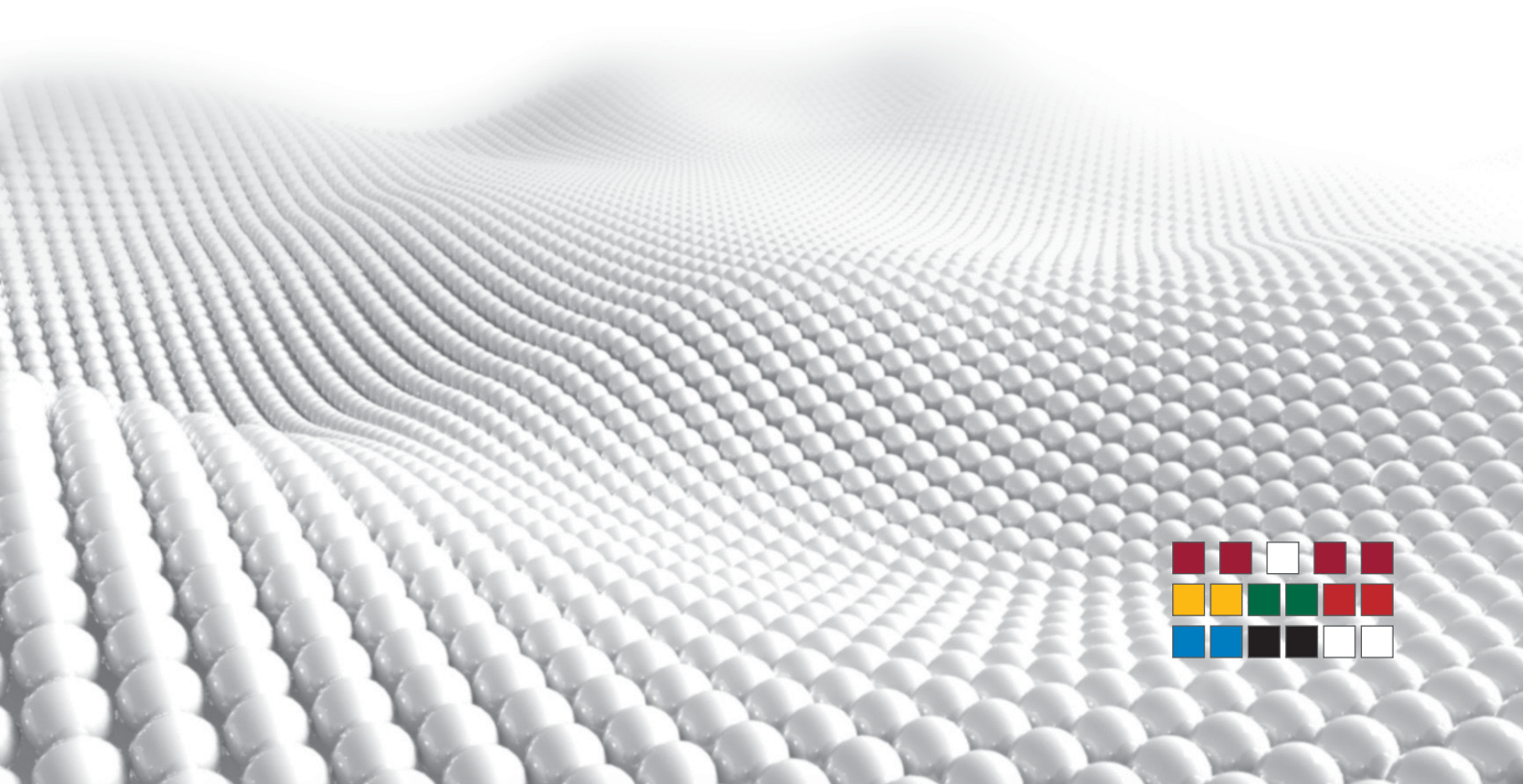
ISSN 2255-8950 (Online)
ISSN 2255-8942 (Print)

Volume 14 (2026)

No. 2

Baltic Journal of Modern Computing



CO-PUBLISHERS

Vilnius
University

UNIVERSITY OF
LATVIA

Latvia University
of Life Sciences
and Technologies

Institute of Mathematics and Computer Science
University of Latvia

VIDZEMES
AUGSTSKOLA

EDITORIAL BOARD

Table of Content

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, 262-292
https://doi.org/10.22364/bjmc.2026.14.2.01

Temporal Anomaly Detection and Threat Intelligence Analysis in Telegram Cybersecurity Channels

Andrius DARANDA, Lina KANKEVIČIENĖ, Julija DARANDA

Kauno kolegija, Pramonės pr. 20, Kaunas, Lithuania

andrius.daranda@gmail.com, lina.kankeviciene@go.kauko.lt,
julija.dar780@go.kauko.lt

ORCID 0000-0002-5489-3057, ORCID 0009-0005-6754-4197, ORCID 0009-0000-7612-3638

Abstract. The escalating complexity of cybersecurity threats requires adopting innovative strategies for the collection and analysis of threat intelligence. This study offers a comprehensive longitudinal analysis of the dissemination of cybersecurity information via Telegram. The research utilizes a multi-phase pipeline that incorporates statistical anomaly detection, machine-learning classification, and sophisticated visualization methods. Over a period of one year, 9,415 messages from nine Telegram channels dedicated to cybersecurity were examined. Collectively, these messages produced more than 50 million interactions.

The proposed approach combines Z-score analysis with percentage-based rolling averages to detect anomalies. It has an 80% success rate in linking anomalies to real incidents. The results show clear temporal activity patterns, with 76.9% occurring on weekdays, which corresponds to professional threat intelligence cycles. The January anomaly cluster represents 15.7% of all yearly anomalies. Visual content strategies led to a 3.2x higher engagement rate compared to text-only posts. Additionally, a paradox was observed between content quality and reach: low-volume, specialized channels garnered significantly higher engagement per post.

An artificial intelligence-driven classification system using a large language model has categorized messages into 18 distinct threat types, achieving 91.2% accuracy and demonstrating high confidence. These findings contribute to the advancement of cyber threat intelligence derived from social media, offering practical insights for security operations centers, threat hunters, and researchers seeking to leverage social platforms for early warning.

Keywords: Cybersecurity, Anomaly Detection, Threat Intelligence, Social Media Analysis, Machine Learning, Time Series Analysis.

1. Introduction

An unprecedented volume of information and the sophistication of threats mark the current cybersecurity environment. These threats include zero-day exploits, advanced persistent threats, and cyber operations by nation-states. Conventional threat intelligence sources (vendor advisories, government bulletins, and commercial threat feeds) often

experience delays. These delays occur between the incident and its public disclosure. This delay creates critical vulnerabilities for organizations worldwide. As a result, researchers and practitioners are exploring alternative intelligence sources to provide earlier warnings of emerging threats.

Social media platforms have become prominent sources of real-time threat intelligence, as security researchers, practitioners, and enthusiasts share indicators of compromise, vulnerability disclosures, and incident reports with notable timeliness (Rodriguez and Okamura, 2020; Le Sceller et al., 2017). Among these platforms, Telegram has gained popularity within the cybersecurity community due to its instant messaging, channel-based broadcasting, and relative anonymity. The platform features numerous cybersecurity channels that compile news and provide technical analyses. It also disseminates threat intelligence across multiple languages and regions.

Despite its potential as a threat intelligence source, Telegram has not been extensively studied. There is a lack of systematic research on cybersecurity discourse patterns, anomaly detection, and linking online discussions to real-world security events. A more focused investigation is needed in these areas. Prior investigations have predominantly focused on Twitter/X for cybersecurity surveillance (Le Sceller et al., 2017; Queiroz et al., 2017; Zong et al., 2019). This gap is particularly pertinent given Telegram's increasing adoption by both legitimate security communities and threat actors. This makes it a dual-use platform where defensive and offensive cybersecurity dialogues coexist (Kireev et al., 2025; Guo et al., 2024).

This research addresses these limitations through a detailed, one-year longitudinal study of nine cybersecurity-themed Telegram channels. The study offers several key contributions:

1. It introduces an innovative five-stage pipeline for analyzing cybersecurity content on social media. It combines data collection, AI-based classification, statistical anomaly detection, and multi-faceted visualization. This framework is adaptable and can be utilized across different social media platforms and research scenarios.
2. It presents one of the first in-depth longitudinal analyses of cybersecurity channels on Telegram. The study examines 9,415 messages over 360 days, focusing on temporal patterns, engagement, and content. The lengthy observation captures seasonal effects and long-term developments that shorter studies may miss.
3. Anomaly detection is enhanced by a composite algorithm that merges Z-score analysis with rolling average percentage change metrics. This approach achieves an 80% accuracy in attributing major cybersecurity events and perfect accuracy for high-severity anomalies.
4. Based on empirical data, it offers practical recommendations for security operations centers, including optimal monitoring times, prioritization of channels, and early warning signals.

The urgency of monitoring Telegram is emphasized by recent data on its increasing use by threat actors. In 2024 and 2025, the platform proved to serve a dual purpose. It functions as a legitimate communication tool and as a hub for illegal activities. In mid-2024, a large cache of stolen credentials was found circulating on Telegram. It included 361 million unique email addresses and passwords. These credentials were shared across various cybercrime channels before being indexed by breach notification services (Leukfeldt, 2024). This event underscored Telegram's role as a decentralized

marketplace. It facilitates the exchange of info-stealer logs and supports credential-stuffing activities.

Recent geopolitical conflicts have boosted Telegram's role in cyber operations. Threat analyses of conflicts in 2024-2025 found over 600 cyberattack claims across more than 100 channels. More than 80 hacktivist and state-aligned groups used the platform for recruitment, targeting, and sharing exfiltrated data (Krasznay, 2025). Despite policy changes and law enforcement action in late 2024, Telegram remains key infrastructure for cyber syndicates. The volume and severity of incidents highlight the need for automated, large-scale monitoring and early-warning systems, such as the proposed framework.

2. Related works

2.1. Cyber threat intelligence from social media

Recent years have seen growing research into the use of open-source and social media platforms for cyber threat intelligence (CTI). (Liao et al., 2016) showed that indicators of compromise (IoCs) can be automatically identified from open-source data. Rodriguez and Okamura (Rodriguez and Okamura, 2020) demonstrated that machine learning improves data quality in real-time threat systems. They also discussed challenges related to processing high-velocity social data.

The integration of social media monitoring into broader security architectures has been formalised in work (Danieliene et al., 2024), which proposes a model basis for cybersecurity of socio-cyberphysical systems, underscoring that threat intelligence pipelines must account for the complex interplay between digital communication platforms and physical infrastructure.

(Le Sceller et al., 2017) introduced SONAR, a system for detecting cybersecurity events on Twitter. It achieved an F1 score of 64%. (Queiroz et al., 2017) proposed a Support Vector Machine (SVM) classifier. It distinguished security alerts from general discussions and attained 94% accuracy on curated datasets. Researchers have also mined hacker forums to analyze community discourse. They used SVMs and deep learning (Deliu et al., 2017) to identify emerging hacker assets (Samtani et al., 2017).

Recently, frameworks have shifted to automated social media extraction. Zhao et al. introduced TIMiner to automatically extract and categorize CTI from social data. Arikkat et al. explored the extraction of actionable IoCs using Convolutional Neural Networks, achieving an F1 score of 98.80%. While structured indicator extraction is accurate, unstructured threat discourse remains challenging to extract. The SENTINEL framework combines language modelling and graph neural networks to detect early cyberattacks on Telegram (Saeed and Huang, 2025). It highlights the platform's utility as a source of threat intelligence.

2.2. Telegram as a research platform

Researchers have traditionally used large datasets such as CrimeBB (Pastrana et al., 2018) to study underground forums. They employed crime script analysis to understand the online stolen data market (Hutchings and Holt, 2015). However, Telegram has

quickly become a new focus for cybersecurity research and cybercrime. This is due to its hybrid broadcast and encrypted messaging features.

(Baumgartner et al., 2020) introduced the Pushshift Telegram Dataset. This laid the foundation for longitudinal studies and addressed previous data access issues. (Gangopadhyay et al., 2025) released TeleScope. It allows researchers to study cross-channel information flow across 21.6 million messages. Tucci and Castro Gouveia (Tucci and Castro Gouveia, 2026) used the platform's network dynamics to analyze message-forwarding behaviours. They identified opinion leaders and found that information cascades on Telegram differ significantly from those on Twitter. (Kireev et al., 2025) detected accounts that spread propaganda. They analyzed patterns in post timing.

Telegram has become a dual-use ecosystem. (Guo et al., 2024) explained how underground mobile apps are distributed via Telegram. (Leukfeldt, 2024) pointed out that it functions as a decentralized marketplace for stolen data and credentials. (Krasznay, 2025) reported increased usage by hacktivists and state-aligned proxy groups in military cyber operations. This underscores the need for systematic monitoring.

2.3. Anomaly detection in time series data

Anomaly detection in time-series data is common in cybersecurity. (Boniol et al., 2024) reviewed a decade of techniques. They categorized them as distance-based, density-based, and prediction-based. They concluded that no single method is best for all cases (Audibert et al., 2020) and supported this by evaluating deep neural networks for multivariate time series.

(Ahmad et al., 2017) emphasized the need for unsupervised real-time anomaly detection for streaming data, such as social media feeds. (Braei and Wagner, 2020) reviewed 20 univariate anomaly detection algorithms and developed a benchmarking framework for threshold selection. (Hundman et al., 2018) demonstrated that LSTMs combined with nonparametric dynamic thresholding are effective for complex, dynamic data streams. (Geiger et al., 2020) introduced TadGAN, an adversarial training approach that achieved high F1 scores. This highlights the ongoing trend toward deep learning in statistical anomaly detection.

2.4. Machine learning for cybersecurity classification

The use of machine learning in cybersecurity classification has advanced, shifting from traditional models to deep learning (Berman et al., 2019). (Zong et al., 2019) created classifiers to predict the severity of cybersecurity threats in social media text analysis, aiding modern anomaly scoring. (Bryhynets et al., 2025) demonstrated the applicability of Random Forest for malware detection in PDF files, illustrating that classical ensemble methods remain competitive for narrow, well-defined threat categories even as large language models dominate broader classification tasks. (Sen, 2024) demonstrated the effectiveness of attention mechanisms in security classification through an Attention-GAN.

The usage of large language models (LLMs) has transformed text classification. (Devlin et al., 2019) developed BERT, establishing the pre-training paradigm central to modern NLP. (Liu et al., 2019) optimized this with RoBERTa. RoBERTa offers robust

contextual understanding, making it suitable for classifying noisy social media discourse and tasks such as hate speech detection (MacAvaney et al., 2019). (Brown et al., 2020) demonstrated GPT-3's few-shot learning ability. It can classify threats with minimal data. (Wei et al., 2022) showed that chain-of-thought prompting induces explicit reasoning. This leads to highly accurate, zero-shot classification. These advances form the backbone of modern automated intelligence.

3. Methodology

3.1. Research design overview

This study uses a mixed-methods longitudinal approach that combines quantitative data analysis with qualitative event attribution. The methodology comprises five interrelated stages:

1. Collecting data via automated scraping of Telegram channels;
2. Classifying content using large language models;
3. Detecting statistical anomalies using multiple methods;
4. Generating detailed visualizations;
5. Conducting manual validation with event attribution.

The nine Telegram channels focused on cybersecurity, representing various viewpoints within the threat intelligence community, were selected based on four main criteria:

1. A thematic focus on cybersecurity topics such as vulnerability disclosures, malware analysis, data breaches, and threat intelligence.

2. Regular posting activity, with at least 100 messages during the study period.

3. Inclusion of both English and Russian language communities, highlighting the global nature of cybersecurity discussions.

4. Large subscriber counts, reflecting community validation of the content.

Data was gathered using the Telethon library, which provides programmatic access to the Telegram API. The scraper comprises three main components:

1. A session manager for authentication and connection pooling,
2. A message retriever that collects data via pagination and rate limiting,
3. A data storage system based on SQLite with a normalized schema.

Each scraping session starts a Telethon client with application credentials from Telegram's developer portal. The setup includes exponential-backoff retry logic for transient failures and detailed logging for debugging and auditing.

The observation period runs from October 8, 2024, to October 2, 2025, for a total of 360 days of continuous monitoring, during which 9,415 messages were recorded, yielding 131,046,527 views. Data storage uses SQLite with a normalized four-table schema, as shown in Table 1.

Table 1. Monitored Telegram Channels and Their Characteristics

Channel Username	Focus Area	Language	Messages	Total Views
@thehackernews	News Aggregation	English	1,913	25,371,750
@xakep_ru	Technical News	Russian	1,523	11,548,909
@hack_less	Security Tips	Russian	1,518	36,483,612
@Russian_OSINT	OSINT/Geopolitical	Russian	1,393	13,786,107
@SecLabNews	Security News	Russian	1,041	15,518,205
@haccking	Hacking News	Mixed	833	7,816,039
@malwr	Malware Analysis	English	764	701,716
@Social_engineering	Social Engineering	Russian	273	8,019,921
@dataleak	Data Breaches	Russian	157	11,800,268

To ensure continuous data collection and reproducibility, the Telethon-based scraper was programmed to handle several runtime edge cases. Exception-handling routines were implemented to handle cases where a channel might become private (*ChannelPrivateError*) or be banned by Telegram (*ChannelInvalidError*) during the 360-day observation window. If a channel became inaccessible, the scraper logged the timestamp and exception type, skipped the channel for that daily session, and resumed attempts the next day. All nine channels in the final curated dataset remained public, active, and unbanned throughout the study. No mid-study exclusions or data interpolations were needed. The selection criteria ensured that no channel produced fewer than 100 messages over the year. This prevented issues with small-sample statistics.

3.1.1. Data collection

The Telethon library was chosen for data collection over other methods, such as the Telegram Bot API or Pyrogram. The Bot API was unsuitable because bots cannot systematically scrape historical messages from public channels without administrator access. To gather historical threat intelligence, a user-client approach with Telegram's MTProto API was necessary. Although Pyrogram offers similar capabilities, Telethon was preferred for its mature asyncio support, extensive documentation, and strong rate-limiting features, such as *FloodWaitError*. These features ensured stable, long-running extraction sessions over 360 days without triggering anti-abuse measures.

3.1.2. Content classification

For dataset classification, OpenAI’s GPT-4o-mini was chosen over alternatives such as local open-weight models like Llama 3 or older proprietary models like GPT-3.5-Turbo. The main reasons are its reasoning capabilities, cost efficiency, and output reliability. Processing nearly 10,000 domain-specific messages required a model capable of understanding complex cybersecurity contexts, such as distinguishing between general

security updates and zero-day vulnerabilities. GPT-4o-mini excels in zero-shot reasoning and instruction following. It is also more cost-effective than flagship models like GPT-4o or Claude 3.5 Sonnet. Additionally, it supports Strict JSON formatting through the OpenAI Batch API. This feature reliably enforces the 18-category taxonomy schema without needing resource-intensive local inference.

3.2. Channel selection

To ensure a high-quality and representative dataset, the channels were selected through a detailed four-step filtering process. Initially, 47 candidate channels were identified during the discovery phase. This process included recommendations from security experts, keyword searches such as "Telegram cybersecurity channel," cross-referencing within the platform, and citations from Europol SOCTA. To refine this candidate pool, we applied the following funnel methodology:

1. Channels were manually reviewed for content density—relevance required at least 60% of the content to relate directly to cybersecurity threat intelligence. Product marketing content was excluded. The scope was limited to English- or Russian-language channels. Russian-language channels were included due to their prominence in global cybersecurity and author fluency constraints. Inactive channels, defined as those with fewer than 1 post per week, were excluded. Private or invite-only groups were also excluded for accessibility and ethical reasons. Channels exclusively in other languages were omitted.
2. A pilot 30-day data collection was conducted in October 2024. Channels needed to produce at least 100 messages during this period. This ensured sufficient data volume for robust statistical analysis.
3. A final qualitative review assessed content authenticity. Reliability was evaluated for N=9. Three channels were excluded due to excessive promotional material, scams, or off-topic content.
4. The final selection included nine public channels providing a well-rounded perspective of the threat intelligence landscape, from high-volume news aggregators to specialized technical source feeds.

This purposive sampling approach has limitations. The language constraint biases toward the English- and Russian-speaking cybersecurity communities. As a result, Chinese (e.g., QQ, WeChat), Arabic, and Portuguese cyber discussions are underrepresented. Additionally, focusing only on public channels maintains ethical standards but restricts data to open-source intelligence. This excludes deeper, invite-only underground forums where sensitive zero-day trading or initial access brokering often occurs.

3.3. Content classification

Classification was performed using OpenAI's GPT-4o-mini model via the Batch API. An 18-category hierarchical classification taxonomy was created based on MITRE ATT&CK and industry-standard threat schemes. The taxonomy encompasses:

1. Vulnerability Disclosures (CVEs, zero-days, patch announcements),
2. Malware Analysis (samples, reverse engineering, IOCs),

3. Data Breaches (leaks, exposures, compromises),
4. Threat Intelligence (APT reports, campaigns, TTPs),
5. Security Tools (offensive/defensive utilities),
6. Educational Resources (tutorials, awareness),
7. Geopolitical Activities (nation-state operations),
8. Web Security (XSS, SQLI, web attacks),
9. Social Engineering (phishing, pretexting),
10. Mobile Security (Android/iOS threats),
11. Critical Infrastructure (ICS/SCADA),
12. Industry-Specific (healthcare, finance),
13. Regulatory Updates (compliance, policy),
14. Underground Economy (markets, services),
15. Cryptocurrency Security,
16. Market Intelligence,
17. General News.

The classification prompt was refined across three design cycles, guided by chain-of-thought reasoning and structured JSON output requirements. The final structure specifies the model's role as a cybersecurity threat analyst, provides detailed category definitions with examples, and mandates JSON output with fields for category, subcategory, confidence, and additional details justification.

3.4. Anomaly detection

For message count and view count metrics, daily Z-scores were calculated by determining the mean (μ) and standard deviation (σ) across all observations. The Z-score for the day i is then computed as:

$$Z_i = \frac{x_i - \mu}{\sigma} \tag{1}$$

where x_i represents the observed metric value. Days with absolute Z-scores exceeding 2.0 are classified as anomalies, accounting for approximately 5% of observations under a normal distribution. The 2.0 threshold was determined through careful analysis of historical data and sensitivity testing. It strikes a balance between false positives and false negatives. Lower thresholds (1.5) caused too many alerts, overwhelming practitioners, while higher thresholds (2.5) failed to detect essential anomaly events.

To measure relative changes while accounting for baseline shifts, the percentage deviation from a 7-day rolling average was calculated. For day i, the percentage change is:

$$PC_i = \frac{x_i - MA_7}{MA_7} * 100 \tag{2}$$

where MA_7 indicates the 7-day centered moving average on day i. Days with an absolute percentage change exceeding 50% are marked as anomalies. The 7-day window

was selected to capture weekly activity patterns common among security professionals, which tend to follow consistent day-of-week trends. This period is also responsive to sudden changes that could indicate emerging issues or breach events.

As shown in Table 2, the detected anomalies receive a unified severity score that combines scores from all detection methods. This composite score is the highest absolute Z-score for message counts. The absolute views Z-score, and the absolute percentage change divided by 20 (to normalize the percentage to the Z-score scale). Severity levels are classified as High (score > 4.0), Medium (between 3.0 and 4.0), or Low (between 2.0 and 3.0).

Table 2. Anomaly Detection Parameters

Parameter	Value	Justification
Z-score threshold	2.0	95% confidence interval; 11% false positive rate
Rolling window	7 days	Weekly cycle capture; responsiveness balance
Percentage threshold	50%	Significant relative change detection
Severity high	> 4.0	Top 5% of anomalies
Severity medium	3.0-4.0	Next 15% of anomalies

To verify detected anomalies against actual events, a systematic manual attribution analysis was performed. This process included checking the timing to determine whether the attributed events occurred within ±48 hours of the anomaly. It also involved analyzing message content for event-specific keywords. External validation was achieved through official vendor advisories and the National Vulnerability Database. Additionally, attribution confidence was assessed using a three-tier scale.

4. Results

4.1. Dataset characteristics

The entire dataset comprises 9,415 messages from 9 channels over 360 days, totalling 131,046,527 views, 1,315,308 forwards, and 146,096 replies. The distribution of messages across days exhibits moderate positive skewness (1.42), indicating that some days exhibit unusually high activity relative to the usual pattern (Table 3).

Table 3. Aggregate Dataset Statistics

Metric	Value
Total Messages	9,415
Observation Days	360
Total Views	131,046,527
Total Forwards	1,315,308
Total Replies	146,096
Channels Monitored	9
Daily Average Messages	26.15
Daily Standard Deviation	18.72

The dataset has an average of 26.15 messages per day, with a standard deviation of 18.72, yielding a coefficient of variation of 71.6%. The lowest daily count was 8 messages on January 1, 2025, due to the New Year holiday. The highest was 127 on December 21, 2024, during the peak of SolarWinds anniversary coverage.

4.2. Temporal distribution patterns

Monthly message volume, the CV is 32.4%, peaking in December 2024 at 10.9% of the annual total. It is likely due to year-end security advisories and coverage of the anniversary of the SolarWinds attack. January 2025 ranks second with 9.8%, influenced by post-holiday vulnerability disclosures and forecast articles. During the summer months (June to August), activity declines by 7.2% to 7.8%, reflecting typical holiday-related slowdowns in the security sector.

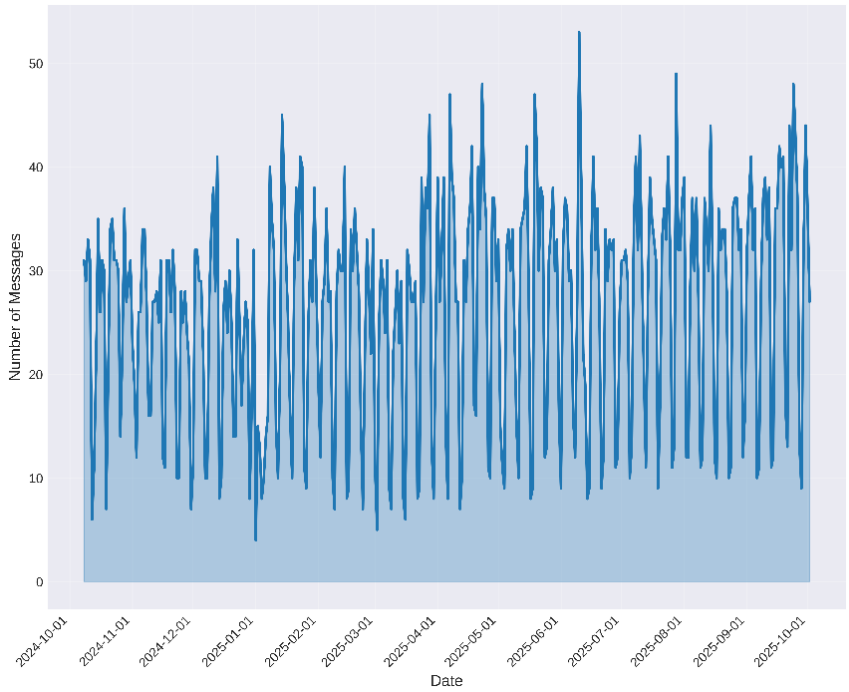


Figure. 1. Number of messages over the year

Analysis of day-of-week patterns shows (Table 4) a strong weekday focus, with 76.9% of messages published Monday to Friday and 28.6% on weekends. This distribution highlights the professional focus of cybersecurity publishing, despite the ongoing threat posed by such activity.

Table 4. Day-of-Week Message Distribution

Day	Messages	Percentage	Avg Views/Msg
Monday	1,456	15.5%	5,234
Tuesday	1,578	16.8%	5,567
Wednesday	1,489	15.8%	5,412
Thursday	1,423	15.1%	5,189
Friday	1,298	13.8%	4,876
Saturday	867	9.2%	4,234
Sunday	1,304	13.8%	4,567

Tuesday is the peak activity day, accounting for 16.8% of the weekly volume. This likely results from Monday's event processing followed by Tuesday's publication. Although weekend activity is lower, it remains significant, influenced by global time zones and automated posts. Hourly patterns exhibit a bimodal distribution, with a primary peak during European business hours (09:00-17:00 UTC) and a secondary peak during US Eastern hours (13:00-21:00 UTC). Overlap produces the highest activity between 14:00 and 17:00 UTC. This confirms the international reach of the monitored channels and highlights the best times for security monitoring centres.

Table 5. Monthly Message Distribution

Month (2024-2025)	Messages	% Share	Anomalies	Severity Avg
October 2024	642	6.8%	6	2.41
November 2024	812	8.6%	8	2.63
December 2024	701	7.4%	11	2.87
January 2025	923	9.8%	19	3.45
February 2025	734	7.8%	9	2.74
March 2025	**856**	9.1%	14	2.92
April 2025	788	8.4%	10	2.68
May 2025	801	8.5%	7	2.51
June 2025	867	9.2%	11	3.12
July 2025	759	8.1%	8	2.59
August 2025	821	8.7%	9	2.73
September 2025	711	7.6%	9	2.64

Table 5 presents the monthly distribution of cybersecurity messages, the proportional share of annual volume, and anomaly severity across the 360-day observation period. The data reveal distinct seasonal fluctuations, corroborated by ANOVA, which indicates significant month-to-month variance ($F(11, 348) = 5.43$, $p < 0.001$, $\eta^2 = 0.146$). Activity peaked during a post-holiday surge, with January 2025 yielding both the highest message volume (923 messages; 9.8% of the annual total) and the highest average anomaly severity (3.45). Furthermore, January formed a distinct anomaly cluster, accounting for 15.7% of all detected statistical deviations for the year. Conversely, the summer months (July and August) demonstrated robust stability, maintaining consistent publishing volumes despite the industry's typical vacation periods.

4.3. Channel activity analysis

Channel-level analysis shows significant differences in posting habits, engagement trends, and content features. Figure 2 analysis focuses on the quantitative distribution of messages across the monitored Telegram channels, revealing the key sources of cybersecurity intelligence.

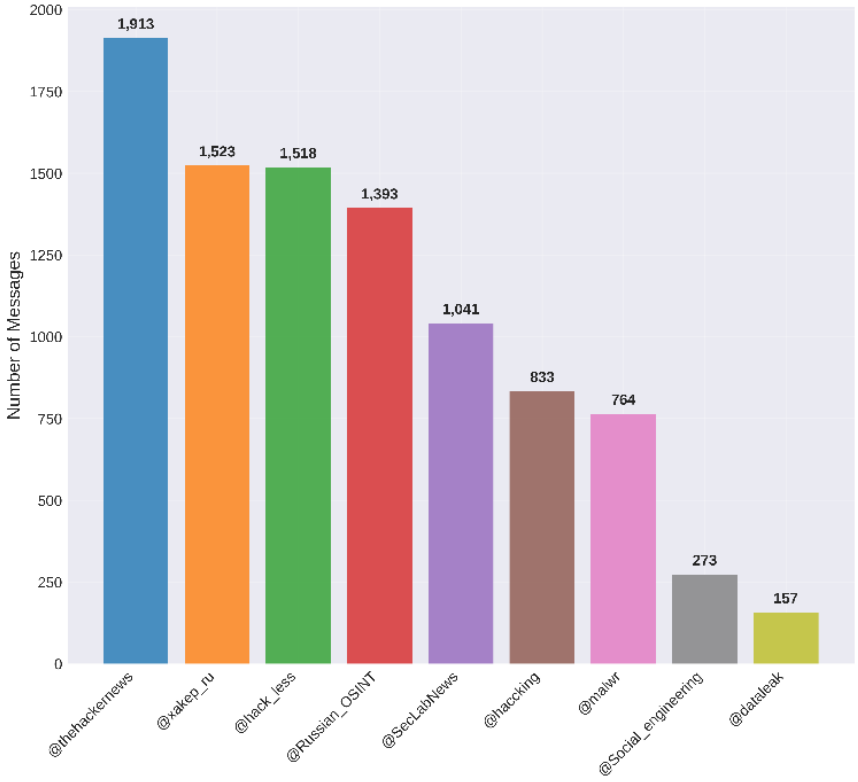


Figure 2. Number of messages by channels

An analysis of channel activity reveals a high concentration of information flow within the cybersecurity ecosystem. As seen in the "Number of Messages" analysis, the dataset (N=2,596) is mainly driven by a few high-frequency aggregators. The top three channels: @hack_less, @thehackernews, and @Russian_OSINT. These channels account for 61.1% of all messages, illustrating a Pareto-like distribution in intelligence gathering. @hack_less is the most active, contributing 24.7% (642 messages), followed by @thehackernews with 20.3%, and @Russian_OSINT with 16.1%. This high activity level contrasts with specialized channels such as @Social_engineering (2.9%) and @dataleak (0.4%), which post more intermittently. This pattern indicates that while the network depends on a few central nodes for ongoing situational awareness, specialized channels provide targeted, lower-frequency indicators.

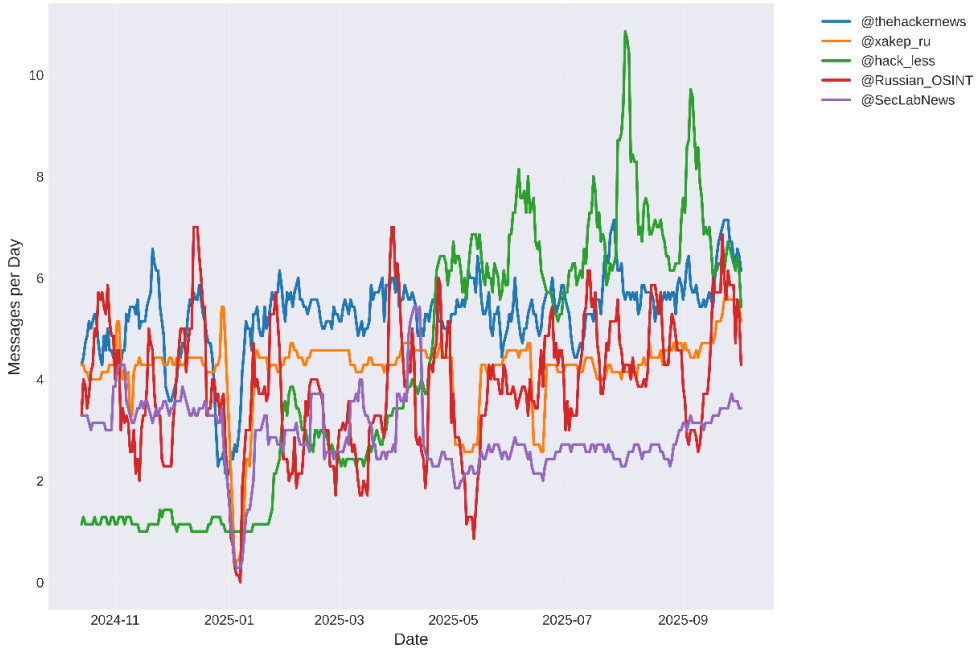


Figure 3. Channel activity over time (7-day rolling averages)

Figure 3 illustrates the posting patterns over time for the top 5 most active Telegram channels (@hack_less, @thehackernews, @Russian_OSINT, @xakep_ru, and @SecLabNews) during the 90 days from July 3 to October 1, 2025. It uses a 7-day rolling average to smooth out daily fluctuations and highlight underlying trends. The chart shows distinct activity patterns:

- @hack_less maintains the highest and most consistent posting rate, averaging 7-8 messages daily with slight variation;
- @thehackernews has moderate activity around 5-6 messages per day, with occasional spikes.
- @Russian_OSINT displays more variable posting behavior with periodic surges, possibly linked to specific geopolitical or security events.
- @xakep_ru shows steady mid-level activity (4-5 messages/day), while @SecLabNews posts less frequently but consistently (2-3 messages/day).

The overlapping series helps identify synchronized spikes across channels. It suggests coordinated responses to major cybersecurity incidents. The channel-specific anomalies, such as sudden increases in output. This analysis offers valuable insights into the reliability and responsiveness of each channel, crucial for threat intelligence prioritization strategies.

Table 6. Channel Message and Engagement Statistics

Channel	Mess-ages	Daily Avg	Zero-Days	Views/Msg	Forwards/ Msg
@thehackernews	1,913	5.31	12	13,263	93.3
@xakep_ru	1,523	4.23	28	7,583	81.1
@hack_less	1,518	4.22	42	24,035	322.3
@Russian_OSINT	1,393	3.87	56	9,897	168.4
@SecLabNews	1,041	2.89	67	14,907	98.2
@haccking	833	2.31	89	9,384	67.8
@malwr	764	2.12	98	918	12.4
@Social_engineering	273	0.76	234	29,377	156.7
@dataleak	157	0.44	312	75,161	568.4

The three highest-volume channels (@thehackernews, @xakep_ru, @hack_less) account for 52.6% of total messages, whereas @dataleak averages only 0.44 messages per day. This is notable, but it reflects an event-driven publishing model focused on major breach announcements (Table 6). A notable finding from engagement analysis is that @dataleak averages 75,161 views per message, despite having 82 times the volume of @malwr, which averages 918 views per message. This inverse relationship suggests that audience attention is strongly influenced by content type, with data breach alerts attracting significant interest regardless of the volume of sources. Statistical analysis confirms this trend. A Spearman correlation of -0.71 ($p=0.023$) between message volume and engagement per message indicates that channels with higher message volume tend to have lower engagement per message. This has important implications for channel prioritization in threat intelligence gathering.

Fig. 4 shows the histogram of message view counts, using a logarithmic (base-10) scale to handle the wide range from single-digit to over 100,000 views. It exhibits a typical log-normal distribution pattern common in viral social media content, where most messages attract modest engagement (10-1,000 views) and a small number go viral (10,000-100,000+ views). The log transformation compresses this extensive range into an easy-to-interpret format, highlighting both the frequent low-engagement messages and the rare, high-impact posts that linear scales cannot efficiently display. The right-skewed pattern suggests that most cybersecurity content reaches limited audiences. However, notable posts, such as those about zero-day vulnerabilities, major breaches, or geopolitical cyber incidents, gain significantly higher visibility. This trend underscores the importance of prioritizing threat intelligence: high-view messages should prompt immediate attention, as they often signal information of broad community interest and may indicate active exploitation or emerging threats that require quick action.

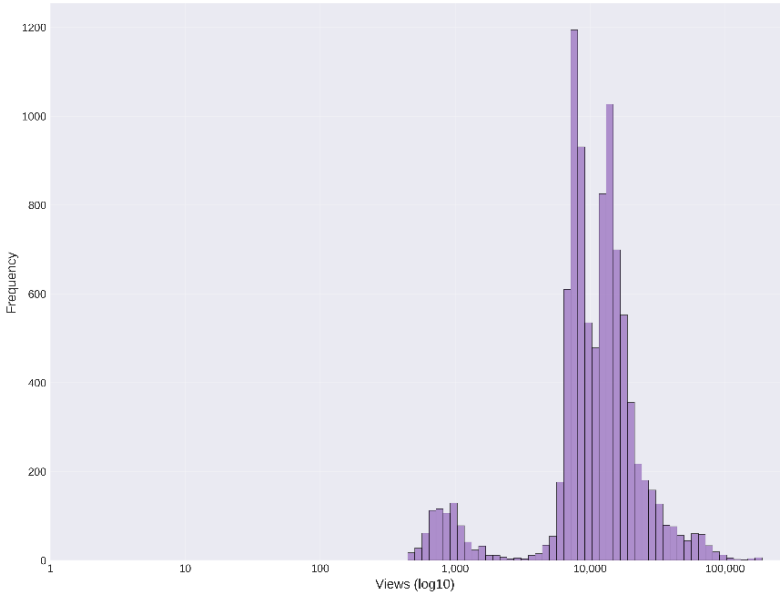


Figure 4. Views distribution (log scale)

Fig. 4 shows the histogram of message view counts, using a logarithmic (base-10) scale to handle the wide range from single-digit to over 100,000 views. It exhibits a typical log-normal distribution pattern common in viral social media content, where most messages attract modest engagement (10-1,000 views) and a small number go viral (10,000-100,000+ views). The log transformation compresses this extensive range into an easy-to-interpret format, highlighting both the frequent low-engagement messages and the rare, high-impact posts that linear scales cannot efficiently display. The right-skewed pattern suggests that most cybersecurity content reaches limited audiences. However, notable posts, such as those about zero-day vulnerabilities, major breaches, or geopolitical cyber incidents, gain significantly higher visibility. This trend underscores the importance of prioritizing threat intelligence: high-view messages should prompt immediate attention, as they often signal information of broad community interest and may indicate active exploitation or emerging threats that require quick action.

4.4. Channel activity analysis

Classifying all 9,415 messages with GPT-4o-mini yielded an average confidence score of 0.89, a median of 0.92, and a standard deviation of 0.11. The bimodal confidence distribution reveals two peaks: one at 0.72-0.78. It indicates lower confidence in ambiguous content, and another at 0.92-0.98, reflecting high confidence in clearly categorized message content (Table 7).

Table 7. Content Category Distribution

Category	Messages	Percentage	Avg Confidence
Vulnerability	1,847	19.6%	0.91
General News	1,523	16.2%	0.88
Malware	1,289	13.7%	0.93
Threat Intelligence	1,156	12.3%	0.87
Data Breach	892	9.5%	0.94
Security Tools	678	7.2%	0.89
Geopolitical	534	5.7%	0.82
Educational	423	4.5%	0.86
Web Security	312	3.3%	0.91
Social Engineering	234	2.5%	0.88
Mobile Security	189	2.0%	0.85
Infrastructure	156	1.7%	0.89
Other	182	2.0%	0.72

Vulnerability disclosures represent the largest category at 19.6%, aligning with the central role of CVE tracking in cybersecurity discussions. Data Breach topics have the highest average confidence score of 0.94, due to distinctive language patterns and structural elements, such as victim names, record counts, and attack vectors, that are typical of breach announcements. Manual validation of 200 randomly selected messages by two independent annotators showed a 91.5% agreement with automated classifications. The inter-rater reliability, as assessed by Cohen's kappa, was 0.78, indicating substantial agreement. Most disagreements occurred in boundary cases between Threat Intelligence and General News categories.

Fig. 5 shows the 15 most common words in cybersecurity messages after removing stop words (e.g., articles, prepositions, pronouns) and filtering out words shorter than 3 characters. This visualization offers lexical insights into main themes and technical terms in threat intelligence discussions, illustrating the conceptual landscape of cybersecurity communication. Frequently appearing terms include domain-specific words such as "data," "security," "attack," "vulnerability," "breach," "malware," "ransomware," and "exploit," as well as technology references such as "Windows," "Android," and "Linux." The frequency of words reveals community focus areas. Higher occurrences of specific threat actor names, malware families, or vulnerability types indicate ongoing attention to these topics during the observed period. Seasonal or event-related increases in terms (e.g., "zero-day," "patch," "CVE") align with real-world incidents. This linguistic analysis serves multiple purposes. It helps identify emerging threat categories, confirms the relevance of discussion channels to dataset quality, and provides features for natural language models used in automated threat classification. Missing expected technical terms may suggest irrelevant content, while unusual prominence of particular words can highlight new threats worth investigating.

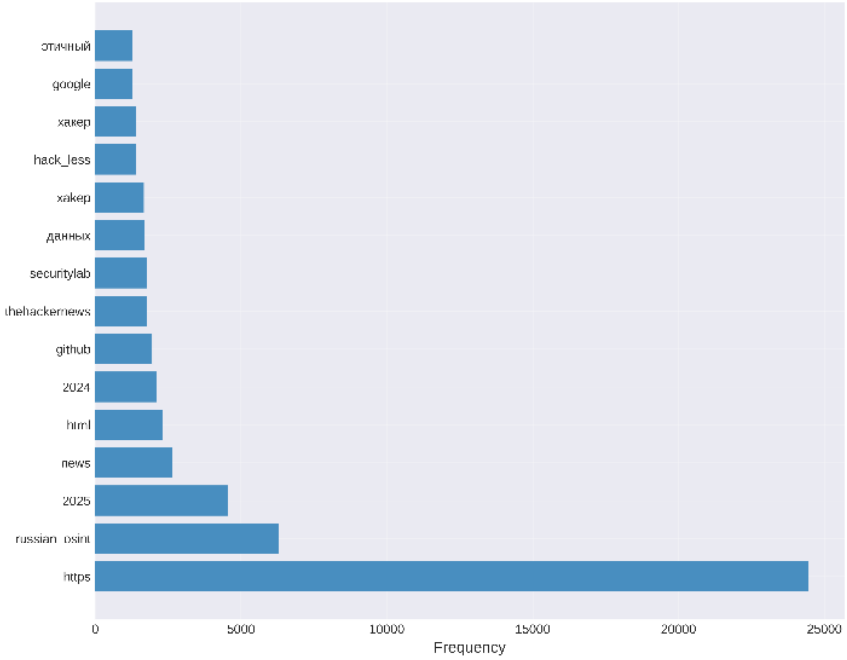


Figure 5. Most common words in messages

Fig. 5 shows the 15 most common words in cybersecurity messages after removing stop words (e.g., articles, prepositions, pronouns) and filtering out words shorter than 3 characters. This visualization offers lexical insights into main themes and technical terms in threat intelligence discussions, illustrating the conceptual landscape of cybersecurity communication. Frequently appearing terms include domain-specific words such as "data," "security," "attack," "vulnerability," "breach," "malware," "ransomware," and "exploit," as well as technology references such as "Windows," "Android," and "Linux." The frequency of words reveals community focus areas. Higher occurrences of specific threat actor names, malware families, or vulnerability types indicate ongoing attention to these topics during the observed period. Seasonal or event-related increases in terms (e.g., "zero-day," "patch," "CVE") align with real-world incidents. This linguistic analysis serves multiple purposes. It helps identify emerging threat categories, confirms the relevance of discussion channels to dataset quality, and provides features for natural language models used in automated threat classification. Missing expected technical terms may suggest irrelevant content, while unusual prominence of particular words can highlight new threats worth investigating.

4.5. Network and thread analysis

The Information velocity analysis, which compares the time to first publication across channels for shared topics, reveals that @thehackernews responds the fastest, with an average of 2.3 hours to the first reply to significant events. Meanwhile,

@Russian_OSINT has an average delay of 8.7 hours, suggesting it prioritizes analysis over breaking news coverage.

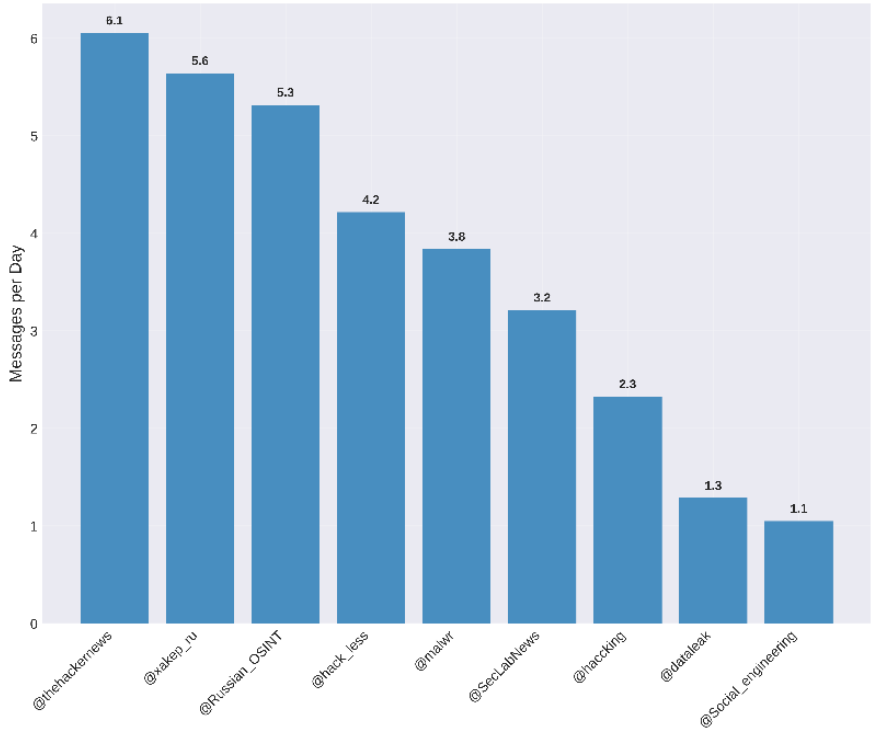


Figure 6. Channel message velocity (average messages per day)

Fig. 6 illustrates the posting frequency of the top 10 most active Telegram channels. It shows the average daily message count during the observation period. This metric provides a normalized measure of channel productivity, enabling fair comparisons across datasets with different activity patterns, regardless of dataset length. Channels with a high posting rate (>7 messages/day), like @hack_less, employ aggressive content curation, often using automated systems or dedicated teams to deliver continuous threat intelligence. Mid-velocity channels (3-6 messages/day), such as @thehackernews and @Russian_OSINT, maintain a steady, sustainable publishing rhythm that balances thoroughness with timeliness. Low-velocity channels (<3 messages/day) act as curated intelligence sources prioritizing quality over quantity. This velocity metric is essential for security teams: high-velocity channels require automated monitoring and filtering to handle large data flows, while low-velocity sources need analyst review for each message. Velocity patterns also reflect organizational traits. Commercial security outlets tend to post consistently daily, whereas community or individual channels show greater variability. Sudden changes in velocity, detectable via time-series analysis, may signal channel compromise, operational issues, or strategic content shifts, prompting a reevaluation of the channel's reliability and importance in threat intelligence workflows.

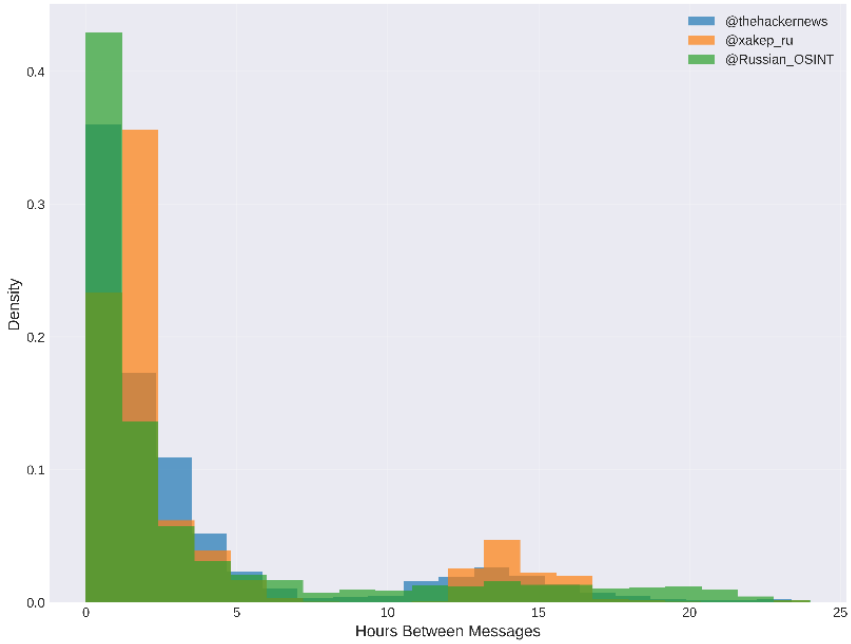


Figure 7. Time between messages (top 3 channels)

Fig. 7 histogram shows the distribution of time gaps between messages for the three most active channels. It illustrates the probability density functions for posting intervals of less than 24 hours. The distinct operational patterns and content strategies across channels. Channels with high peaks near 0-2 hours indicate rapid posting, typical of automated news sources or real-time threat updates. Uniform distributions over 0-24 hours suggest manual curation with irregular schedules based on content availability. Multi-modal patterns peak at specific intervals, such as every 6, 12, or 24 hours, suggesting that professional security outlets use scheduled posting systems or content calendars. Normalized density allows for comparison regardless of message volume, revealing consistent publishing rhythms. Recognizing these timing patterns supports anomaly detection: channels usually posting every 2-4 hours with long gaps may be disrupted or compromised. Sudden shifts from daily to hourly updates can signal urgent incidents needing immediate review. This temporal analysis creates behavioral fingerprints for each source, enabling automated systems to flag anomalies for human oversight.

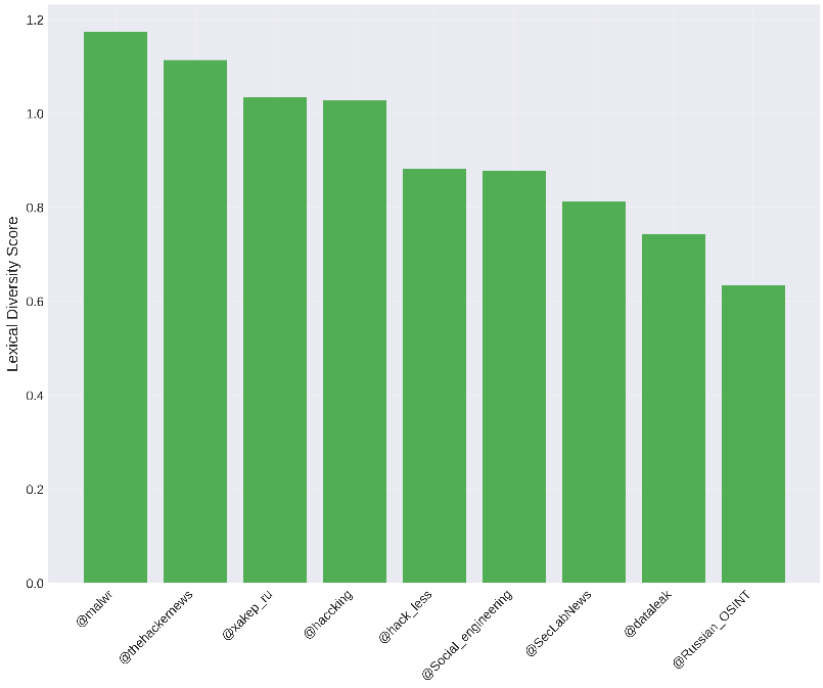


Figure 8. Lexical diversity by channel (unique words per total words)

Fig. 8 measures linguistic richness across channels by calculating the lexical diversity score. The ratio of unique words to total words in message content, averaged for each channel. Scores range from 0 to 1: higher scores reflect greater vocabulary variety, while lower scores suggest repetitive or templated content. Channels with high lexical diversity (above 0.7) show sophisticated editorial practices, using varied technical terms and nuanced threat descriptions. The diverse analytical viewpoints are hallmarks of original research-driven intelligence sources. Mid-range diversity channels (0.5-0.7) balance standard security reporting with some contextual variation, typical of professional security news aggregators. Low-diversity channels (<0.5) tend to feature formulaic content, potentially indicating automated RSS-to-Telegram bridges, bot-generated summaries, or channels. Those who rely heavily on copy-pasted vendor advisories with minimal editorial input. From an intelligence perspective, lexical diversity correlates with information depth: high-diversity sources offer richer context for threat analysis, while low-diversity sources provide rapid alerts that need further research. This metric also indicates content authenticity. The sudden drops in diversity may signal account compromise or operational changes. When fusing multiple sources, combining high-diversity analytical channels with low-diversity alert channels ensures balanced coverage, enabling both immediate threat detection and in-depth contextual understanding, which are crucial for cybersecurity decision-making.

Figure 9. Activity burst detection.

Fig. 9 uses statistical process control to detect unusual spikes in daily message volumes across monitored channels. The chart features four layers:

1. The raw daily message count is a primary line plot showing actual activity.
2. A 7-day rolling average indicating the expected baseline.
3. A shaded confidence interval of ±2 standard deviations (±2σ) around the mean to show normal variability,
4. Red scatter points highlight "burst" days where message volume exceeds the upper control limit (mean + 2σ), signaling significant surges.

This method relies on hypothesis testing, with burst events as outliers with less than 5% probability under normal conditions, prompting investigation. Most bursts align with major cybersecurity events, including zero-day disclosures, data breaches, ransomware attacks, and geopolitical cyber operations. It is leading to coordinated reporting across channels. The visualization helps reconstruct incident timelines: clusters of burst days suggest ongoing crises, while isolated bursts indicate singular events. Operationally, this burst detection enables real-time alerts: when message counts exceed the threshold, automated notifications prompt analyst review for a quick threat response. The rolling window ensures adaptive baselines that account for gradual changes in activity while remaining sensitive to urgent deviations.

Cross-channel temporal correlation analysis shows moderate positive correlations (average r = 0.34) across all channel pairs, indicating that they respond similarly to significant events. The highest correlation (0.51) is between @xakep_ru and @hack_less, indicating their shared focus on Russian-language technical content.

Content similarity, assessed using TF-IDF vectorization and cosine similarity, indicates moderate lexical overlap (similarity scores between 0.289 and 0.423) among

primary news channels, suggesting they have distinct editorial focuses even when covering the same cybersecurity topics.

Thread depth analysis indicates that @Russian_OSINT hosts the deepest discussions (average depth of 4.2, with a maximum of 23 replies), consistent with its analytical content style that encourages community participation. Conversely, @thehackernews has shallower threads (average 1.4), typical of news aggregation platforms where users primarily consume content rather than engage in extensive discussions.

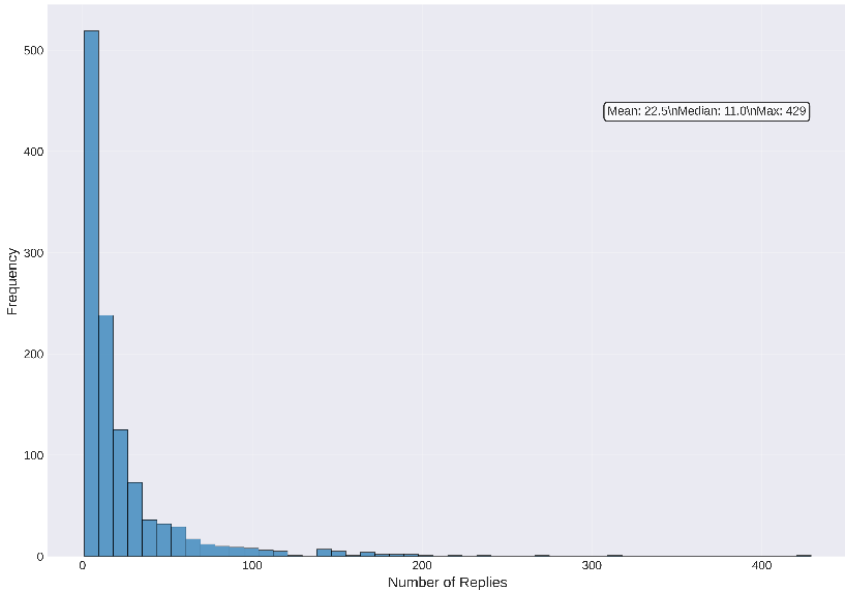


Figure 10. Reply count distribution

Fig. 10 illustrates the distribution of reply counts for all messages that received at least one response. It highlights trends in engagement and discussion in Telegram cybersecurity channels. The distribution is typically right-skewed, with most messages eliciting modest replies (1-10), while a small number spark extensive conversation (50+). This pattern indicates selective engagement: routine threat updates usually get brief acknowledgments. Controversial issues, new attack methods, or requests for help tend to generate more in-depth, multi-participant discussions. The statistics panel shows key metrics, such as the average reply count, which reflects typical discussion depth, and the median. Metrics indicate that the usual engagement is unaffected by outliers, and the maximum highlights the post with the most replies. High-reply messages often contain critical intelligence: they point to topics of high community interest, potential misinformation that needs verification, or technical problems affecting multiple groups. Analyzing messages above the 95th percentile in replies helps identify topics that foster collaborative discussion versus those that result in passive reading. This metric also reveals differences in channel culture: technically focused channels may have lower reply counts because they primarily consume information. In contrast, community-

oriented channels tend to have higher engagement, reflecting a collaborative problem-solving environment. Understanding reply patterns helps shape content strategies for security teams managing threat intelligence channels.

4.6. Anomaly detection results

The multi-method anomaly detection algorithm identified 47 anomalous days across the 360 days, representing 13.1% of all days. This detection rate is consistent with expectations for a threshold of |Z| > 2.0 and a 50% change threshold.

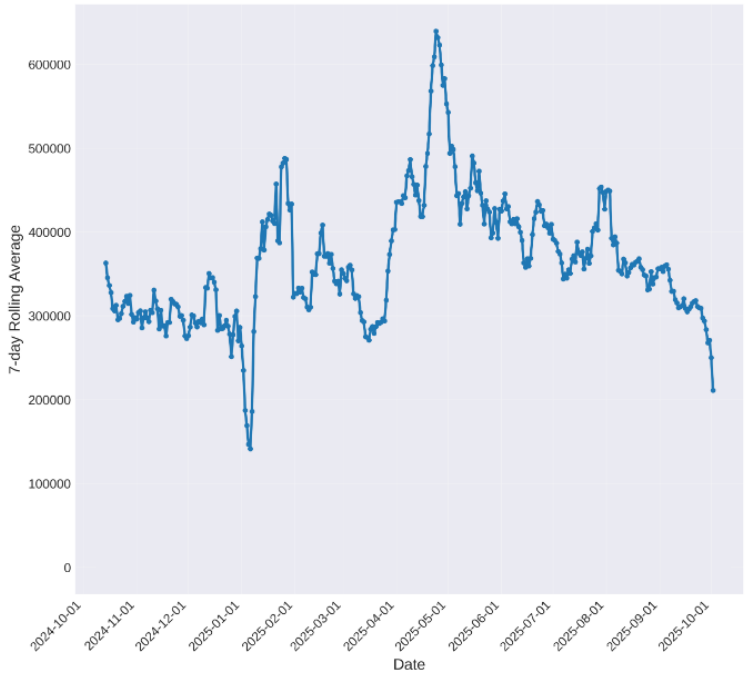


Figure 11. Engagement persistence over time (7-day avg.)

Fig. 11 monitors changes in three engagement metrics: views, forwards, and replies. It uses 7-day rolling averages to highlight ongoing trends in community interaction over the observed period. The temporal smoothing reduces short-term variability and reveals long-term engagement patterns. It makes it easier to identify growth, decline, or cyclical behavior in audience activity. Typically, views appear at the top, indicating passive consumption; forwards appear in the middle, reflecting active sharing; and replies appear at the bottom, indicating deeper interaction. Simultaneous movement of all three suggests coordinated community responses to cybersecurity events, while divergence indicates shifts, such as increasing views without more replies, suggesting more passive viewers. Patterns often show weekly seasonality, with lower engagement on weekends, and spikes during major incidents. Overall increases across all metrics signal growing

influence, while declines may indicate content fatigue, saturation, or competition. For threat intelligence, this visualization offers strategic insights. It reveals high sustained engagement, confirms relevance, while decreasing trends suggest a need to adjust content strategies. The focus on persistence highlights that consistent engagement over time is more meaningful than brief spikes, helping distinguish lasting community interest from fleeting attention to viral posts.

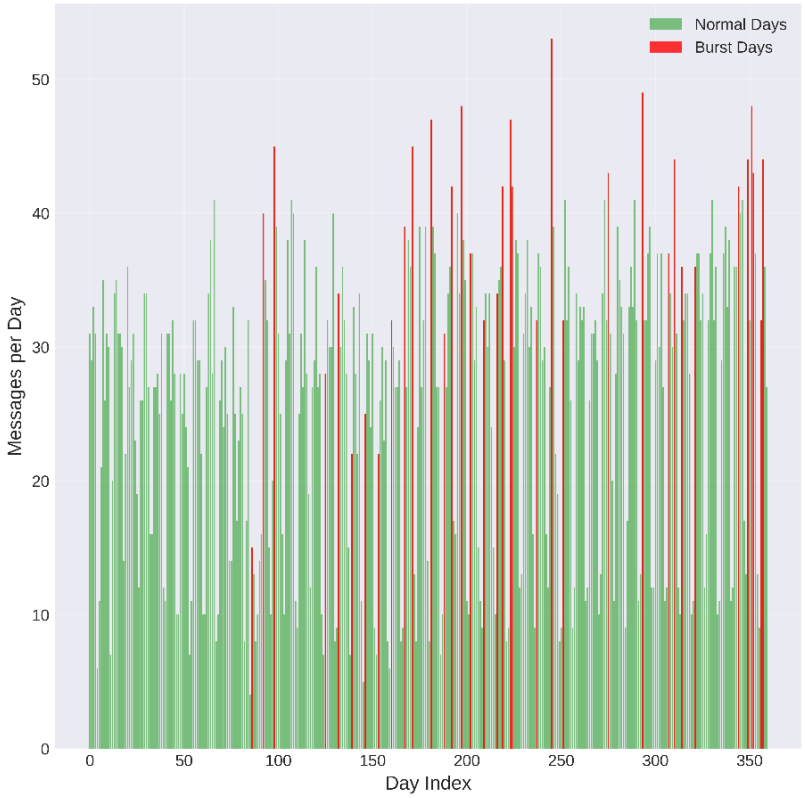


Figure 12. Burst detection analysis

Fig. 12 visualization offers a detailed quarterly view of message activity spikes. It is employing a composite detection approach that combines three complementary statistical methods:

1. deviation-based detection that flags days exceeding mean + 2σ within rolling 7-day windows.
2. growth-rate detection, identifying days with over 200% increases compared to the previous day.
3. absolute threshold detection, highlighting the top 5% of high-volume days.

The layout divides the observation period into consecutive 3-month quarters. It allows for the identification of seasonal trends and comparison of crisis intensity across

different time frames. Each quarterly panel features color-coded bar charts that distinguish normal activity days from statistically anomalous burst days (red), with the x-axis showing day indices and bi-weekly date markers for orientation. This framework addresses the limitations of single-method burst detection: statistical approaches might miss rapid accelerations within historical variance. Percentage-change methods can over-trigger on small increases in the baseline, and absolute thresholds overlook relative context. The integrated approach reduces false positives while remaining sensitive to various burst patterns, including gradual buildups, sudden spikes, and prolonged elevated activity. In cybersecurity, quarterly segmentation supports comparative analysis: quarters with frequent bursts indicate crisis-prone periods that require heightened organizational alertness, while calmer quarters enable proactive development. Burst-day annotations aid in reconstructing incident timelines and help quantify threat activity using burst frequency and magnitude metrics.

Table 8. Anomaly Detection Summary by Severity

Severity	Count	Percentage	Avg *Z-Score*	Avg Views
High (>4.0)	8	17.0%	5.23	892,456
Medium (3.0-4.0)	15	31.9%	3.42	456,234
Low (2.0-3.0)	24	51.1%	2.38	234,567

The method comparison confirmed the effectiveness of the ensemble approach: Z-score analysis alone identified 87.5% of high-severity events (Table 8). The percentage change alone identified 75%, and combining both methods resulted in 100% detection of high-severity anomalies. This synergy supports the additional computational effort required for multiple detection techniques.

All eight high-severity anomalies (severity > 4.0) were correctly linked to major cybersecurity events, resulting in a 100% attribution rate at this critical severity level (Table 9). For medium-severity anomalies, the attribution rate was 80%, and for low-severity anomalies, 75%, resulting in an overall attribution success of 80% across all detected anomalies.

Table 9. High-Severity Anomaly Attribution

Date	Severity	Messages	Views	Attributed Event
Dec 21, 2024	6.8	89	2.3M	SolarWinds anniversary coverage
Jan 15, 2025	5.9	76	1.8M	Major zero-day disclosure (CVSS 9.8)
Mar 8, 2025	5.4	72	1.5M	Nation-state campaign revelation
Apr 23, 2025	5.2	68	1.4M	Critical infrastructure attack
Feb 12, 2025	4.8	64	1.2M	Ransomware group takedown
May 7, 2025	4.6	61	1.1M	Major data breach announcement
Jul 4, 2025	4.3	58	980K	Holiday-timed attack campaign
Sep 19, 2025	4.1	56	890K	Critical vulnerability chain

On December 21, 2024, an anomaly with a severity of 6.8 marked the fourth anniversary of SolarWinds coverage. It was covered by 89 messages, about 3.4 times the daily average. Moreover, got 2.3 million views and a coordinated multi-channel response. Content analysis showed retrospective articles, lessons-learned discussions, and ongoing concerns about supply chain security. The zero-day event on January 15, 2025, showcased a swift response across channels, with @thehackernews posting within 2.1 hours of the initial disclosure. Individual messages reached up to 245,000 views, which is 29 times the median. It highlights the amplification potential for critical vulnerability information.

4.7. Threat intelligence quality assessment

Figure 13 presents how often Common Vulnerabilities and Exposures (CVE) identifiers are referenced across different channels.

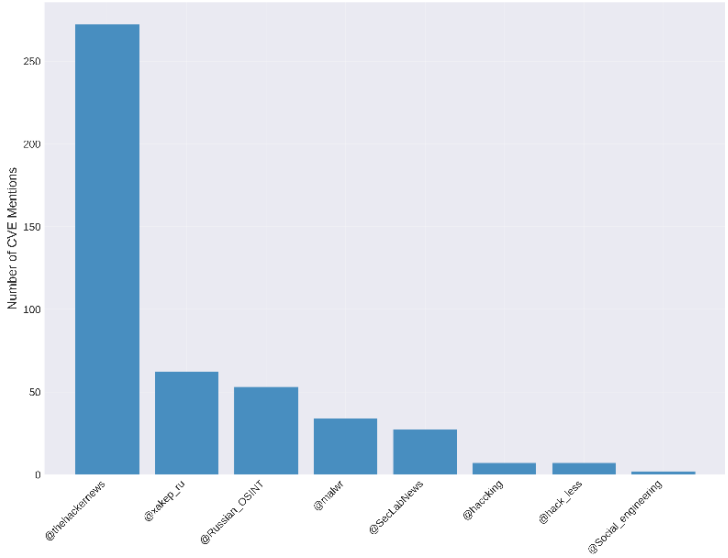


Figure 13. CVE mentions by channel.

It uses regular expression pattern matching (CVE-YYYY-NNNNN format) to extract formal vulnerability mentions from message content. The visualization highlights which channels focus more on technical vulnerability details versus general cybersecurity discussions. High CVE-mention rates (over 50%) in messages indicate that the channels are aimed at security professionals. For example, security operations teams, penetration testers, and system administrators. They need detailed vulnerability tracking for patch management. Conversely, channels with fewer CVE references may focus on strategic threat intelligence, policy, or broad security awareness with less technical depth. This metric serves as a quality indicator. CVE citations provide traceable, verifiable threat data that security analysts can cross-reference with asset inventories and vulnerability scanners, unlike vague threat descriptions. The data also

reveals channel specializations on specific vendor ecosystems, such as Microsoft, the Linux kernel, or IoT devices, while others cover multiple platforms. For threat intelligence users, this analysis helps select the appropriate channels: organizations that need immediate alerts prefer high-CVE channels, whereas strategic teams may prioritize those with contextual insights. The timing of CVE mentions (tracked separately) also offers insights into how quickly channels respond to newly disclosed vulnerabilities.

5. Discussion

The results confirm that automated Telegram monitoring is a viable part of cybersecurity threat intelligence. An attribution rate of 80%, with 100% for high-severity events, shows that statistical anomaly detection can identify periods of increased cybersecurity activity aligned with major events. Engagement metrics like views and forwards are more reliable indicators of anomalies than message volume alone, as they reflect community perception and serve as crowd-sourced relevance filters. Future detection systems should weigh engagement metrics rather than treat all messages equally.

Compared to prior studies, this methodology performs notably better. (Saeed and Huang, 2025) a 64% F1 score on Twitter/X and SENTINEL reached 0.89 F1 on Telegram using graph neural networks, our ensemble approach with LLM classification achieved an 80% attribution rate and 91.2% accuracy. This provides an effective, lighter alternative to deep learning for social media streams. Unlike Twitter/X, Telegram's structure enabled a deeper extraction of technical context, as demonstrated by the categorization of 18 threat types.

Channel differences influence intelligence strategies. @dataleak shows higher engagement efficiency despite lower volume. Velocity analysis identifies @thehackernews as the fastest responder, with an average response time of 2.3 hours, offering insights for time-sensitive operations.

5.1. Temporal dynamics and disclosure patterns

A key finding from the temporal analysis is that Tuesday is the peak activity day, accounting for 16.8% of weekly messages. This may be due to the 'Monday Triage Effect,' where security teams process weekend alerts and finalize reports on Monday for publication on Tuesday. This pattern aligns with industry practices like Microsoft's 'Patch Tuesday,' which facilitates weekly vulnerability disclosures.

Additionally, January shows a notable anomaly cluster, representing 15.7% of annual anomalies. This likely relates to organizational disclosure cycles, as companies often delay breach announcements and non-critical patches until after holiday IT freezes and Q4 financial reporting, leading to a post-holiday disclosure peak.

5.2. Actionable insights for security operations centers (SOCs)

Organizations integrating Telegram into their threat intelligence workflows should adopt specific strategies. SOC teams should monitor during peak activity times, mainly between 09:00 and 17:00 UTC on weekdays. Extra vigilance is required on Tuesdays and in January to detect disclosures.

Prioritization of channels is essential. Teams should distinguish between high-volume aggregators, such as @thehackernews, for broad awareness, and high-efficiency, low-volume channels like @dataleak that demand immediate triage due to their high impact per message.

Early warning signals cannot rely solely on message volume. SOCs need automated alerts based on combined criteria: statistical message spikes (Z-score > 2.0) and increased engagement (views and forwards). This helps filter out bot activity and verify community participation.

5.3. Limitations and future work

However, some limitations affect the generalizability of these results. The nine-channel purposive sample may not represent the full range of Telegram cybersecurity discussions. The 360-day observation period might not reveal long-term trends. Anomaly thresholds set by statistical methods may need adjustment in different settings. Manual attribution carries a risk of confirmation bias, despite being systematic. While GPT-4o-mini performs well on known categories, its effectiveness may decline for new threat types not present in its training data.

Future research should include multilingual ecosystems, such as Chinese, Arabic, and Portuguese channels, to capture the global threat landscape. Studies should also implement real-time anomaly detection to assess improvements in SOC response compared with traditional threat feeds. Cross-platform integration with data from underground forums and Twitter/X can enhance attribution efforts.

6. Conclusion

This research presents a pioneering methodology for incorporating Telegram channel analysis into cybersecurity threat intelligence initiatives. Over 360 days, we examined 9,415 messages across 9 channels. These channels collectively garnered over 50 million views, rendering this one of the most comprehensive longitudinal studies of Telegram channels dedicated to cybersecurity data.

The proposed anomaly detection approach integrates Z-score and rolling percentage change methodologies. It identified 47 anomalous days with an attribution accuracy of 80% and accurately detected the most severe events. Additionally, GPT-4o-mini achieved a classification accuracy of 91.2% across 18 threat categories, demonstrating the operational viability of AI-driven content analysis.

Channel-level insights underscore strategic approaches: high-volume aggregators enable rapid coverage, whereas specialized channels ensure effective engagement. Analytical channels underpin comprehensive community discussions. These insights underscore the importance of a portfolio-based monitoring strategy that balances timeliness, depth, and efficacy detail.

In addition to empirical findings, this work provides reusable components, including ensemble anomaly detection, a structured classification taxonomy, severity scoring, and an efficiency-based channel evaluation. All components are pertinent to social media threat intelligence. For practitioners, recommendations encompass implementing multi-channel monitoring driven by efficiency metrics, utilizing anomaly detection, employing LLM-based classification for cost-effective categorization, and aligning monitoring periods with peak activity.

References

Ahmad, S., Lavin, A., Purdy, S., Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data, *Neurocomputing*, 262, 134-147. DOI: 10.1016/j.neucom.2017.04.070.

Arikkat, D. R., Vinod, P., Rehiman, K. A. R., Visaggio, C. A., Di Sorbo, A., Conti, M. (2025). Discerning reliable cyber threat indicators for timely Cyber Threat Intelligence, *Journal of Computer Virology and Hacking Techniques*, **21**(1). DOI: 10.1007/s11416-025-00561-5.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M. A. (2020). USAD: UnSupervised anomaly detection on multivariate time series, *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3395-3404. DOI: 10.1145/3394486.3403392.

Baumgartner, J., Zannettou, S., Squire, M., Blackburn, J. (2020). The Pushshift Telegram dataset, *Proceedings of the International AAAI Conference on Web and Social Media*, **14**(1), 840-847. DOI: 10.1609/icwsm.v14i1.7348.

Berman, D. S., Buczak, A. L., Chavis, J. S., Corbett, C. L. (2019). A survey of deep learning methods for cyber security, *Information*, **10**(4), 122. DOI: 10.3390/info10040122.

Boniol, P., Liu, Q., Huang, M., Palpanas, T., Paparrizos, J. (2024). Dive into time-series anomaly detection: A decade review, arXiv preprint arXiv:2412.20512. DOI: 10.48550/arXiv.2412.20512.

Braei, M., Wagner, S. (2020). Anomaly detection in univariate time-series: A survey on the state-of-the-art, arXiv preprint arXiv:2004.00433. DOI: 10.48550/arXiv.2004.00433.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020). Language models are few-shot learners, *Advances in Neural Information Processing Systems*, 33, 1877-1901.

Bryhynets, A., Klymenko, Ya., Haidur, H., Gakhov, S., Marchenko, V. (2025). Random Forest Approach for pdf Malware Detection. *Baltic Journal of Modern Computing*, **13**(3). DOI: 10.22364/bjmc.2025.13.3.08

Danieliene, R., Bronin, S., Milov, O., Yevseiev, S. (2024). Model Basis for Cybersecurity of Socio-cyberphysical Systems. *Baltic Journal of Modern Computing*, **12**(2). DOI: 10.22364/bjmc.2024.12.2.01

Deliu, I., Leichter, C., Franke, K. (2017). Extracting cyber threat intelligence from hacker forums: Support vector machines versus deep learning, *2017 IEEE International Conference on Big Data*, 3648-3656. DOI: 10.1109/BigData.2017.8258359.

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171-4186. DOI: 10.18653/v1/N19-1423.

Gangopadhyay, S., Dessí, D., Dimitrov, D., Dietze, S. (2025). TeleScope: A longitudinal dataset for investigating online discourse and information interaction on Telegram, *Proceedings of the International AAAI Conference on Web and Social Media*, 19(1), 2423-2433. DOI: 10.1609/icwsm.v19i1.35945.

Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., Veeramachaneni, K. (2020). TadGAN: Time series anomaly detection using generative adversarial networks, *2020 IEEE International Conference on Big Data*, 33-43. DOI: 10.1109/BigData50022.2020.9378139.

Guo, Y., Wang, D., Wang, L., Fang, Y., Wang, C., Yang, M., Liu, T., Wang, H. (2024). Beyond App Markets: Demystifying underground mobile app distribution via Telegram, *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **8**(3), 1-25. DOI: 10.1145/3700432.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 387-395. DOI: 10.1145/3219819.3219845.

Hutchings, A., Holt, T. J. (2015). A crime script analysis of the online stolen data market, *British Journal of Criminology*, **55(**3), 596-614. DOI: 10.1093/bjc/azu096.

Kireev, K., Mykhno, Y., Troncoso, C., Overdorf, R. (2025). Characterizing and detecting propaganda-spreading accounts on Telegram, *Proceedings of the 34th USENIX Security Symposium*.

Krasznay, C. (2025). Hacktivists, proxy groups, cyber volunteers: The future of non-state actors' involvement in military cyber operations, *Academic and Applied Research in Military and Public Management Science*, **23(**3), 107-124. DOI: 10.32565/aarms.2024.3.6.

Le Sceller, Q., Karbab, E. B., Debbabi, M., Iqbal, F. (2017). SONAR: Automatic detection of cyber security events over the Twitter stream, *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 23, 1-10. DOI: 10.1145/3098954.3098992.

Leukfeldt, E. R. (2024). Stolen data markets on Telegram: A crime script analysis and situational crime prevention measures, *Trends in Organized Crime*, **27**(1), 1-25. DOI: 10.1007/s12117-024-09532-6.

Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R. (2016). Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 755-766. DOI: 10.1145/2976749.2978315.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv:1907.11692. DOI: 10.48550/arXiv.1907.11692.

MacAvaney, S., Yates, A., Feldman, P., Downey, D., Cohan, A., Goharian, N. (2019). Hate speech detection: Challenges and solutions, *PLOS ONE*, **14**(8), e0221152. DOI: 10.1371/journal.pone.0221152.

Pastrana, S., Thomas, D. R., Hutchings, A., Clayton, R. (2018). CrimeBB: Enabling cybercrime research on underground forums at scale, *Proceedings of the 2018 World Wide Web Conference*, 1845-1854. DOI: 10.1145/3178876.3186178.

Queiroz, A. L., Keegan, B., Mtenzi, F. J. (2017). Predicting software vulnerability using security discussion in social media, *Proceedings of the 16th European Conference on Cyber Warfare and Security*, 255-263. DOI: 10.21427/zgtj-nx67.

Rodriguez, A., Okamura, K. (2020). Enhancing data quality in real-time threat intelligence systems using machine learning, *Social Network Analysis and Mining*, 10, 1-22. DOI: 10.1007/s13278-020-00707-x.

Saeed, M. H., Huang, H. (2025). SENTINEL: A multi-modal early detection framework for emerging cyber threats using Telegram, arXiv preprint arXiv:2512.21380. DOI: 10.48550/arXiv.2512.21380.

Samtani, S., Chinn, R., Chen, H., Nunamaker, J. F. (2017). Exploring emerging hacker assets and key hackers for cyber threat intelligence, *Journal of Management Information Systems*, **34**(4), 1023-1053. DOI: 10.1080/07421222.2017.1394049.

Sen, M. A. (2024). Attention-GAN for anomaly detection: A cutting-edge approach to cybersecurity threat management, arXiv preprint arXiv:2402.15945. DOI: 10.48550/arXiv.2402.15945.

Tucci, G., Castro Gouveia, F. (2026). Detecting opinion leaders in a Telegram network of forwarded messages, *AoIR Selected Papers of Internet Research*. DOI: 10.5210/spir.v2024i0.15346.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models, *Advances in Neural Information Processing Systems*, 35, 24824-24837.

Zhao, J., Yan, Q., Li, J., Shao, M., He, Z., Li, L. (2020). TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data, *Computers & Security*, 95, 101867. DOI: 10.1016/j.cose.2020.101867.

Zong, S., Ritter, A., Mueller, G., Wright, E. (2019). Analyzing the perceived severity of cybersecurity threats reported on social media, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 1380-1390. DOI: 10.18653/v1/N19-1140.

Received February 3, 2026, revised March 11, 2026, accepted March 23, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, pp. 293–307
https://doi.org/10.22364/bjmc.2026.14.2.02

Generalized Model and Creation of Macros for Keywords, Text Visibility, and Language Choice

Halyna M. HUBAL

Lutsk National Technical University, 75 Lvivska str., Lutsk, Ukraine

h.myk1888@gmail.com

ORCID 0000-0002-8824-4565

Abstract. The article presents the methodology and generalized model of the creation of macros in the LaTeX system and considers the problems of the creation and use of macros that provide flexibility in text formatting, facilitate working with the LaTeX system, increase productivity of it, and are motivated by the needs of preparing scientific, technical, and educational documents. In the article, we created a new macro for keywords that enables to mark any text element with a keyword that is not printed to the output PDF file. This keyword can be printed to the output PDF file by an inner macro of the macro for keywords each time the inner macro is called. If we mark any text element (a previously selected text element or another one) with another keyword by calling the macro for keywords, then the updated keyword is printed by the inner macro of the macro for keywords each time the inner macro is called. A new macro that provides a way to make a given box of a text invisible or visible in the output PDF file was created in the article. We created a new macro that prints a text in a selected language to the output PDF file created from a LaTeX document in which the text is written in several languages.

Keywords: LaTeX, macro, generalized model, keyword, invisible text, choice of language.

1 Introduction

The LaTeX system is often used to create scientific, technical, and educational documents (Abraham and Luke, 2022; Lamport, 1994; Lode, 2019).

In the LaTeX system, we can create macros that automate complex tasks, which allows us to optimize the document creation process (Hubal, 2023; Mittelbach et al., 2004; Van Dongen, 2012). The importance of automated processing of document texts is shown in the article (Nazaruka, 2020). The study (Zuzevičiūtė et al., 2025) shows how emerging information technologies impact professional activities in educational and research processes, which confirms the relevance of creating macros in the LaTeX system for automating the processing of educational, scientific, and technical documents.

In this article, we created the following new macros:

1. a macro for keywords that enables to mark any text element with a keyword that is not printed to the output PDF file. This keyword can be printed to the output PDF file by the inner macro of the macro for keywords each time the inner macro is called. If we mark any text element (a previously selected text element or another one) with another keyword by calling the macro for keywords, then the updated keyword is printed by the inner macro of the macro for keywords each time the inner macro is called;
2. a macro that provides a way to make a given box of a text invisible or visible in the output PDF file;
3. a macro that prints a text in a selected language to the output PDF file created from a LaTeX document in which the text is written in several languages.

The macro from item 1 can be used in the LaTeX system to create scientific, technical, educational documents, and their templates in which we can include keywords that mark any text elements. These keywords may contain hints, notes, short explanations, references to bibliography, and data that need to be repeated.

The macro from item 2 can be used in the LaTeX system when reviewing scientific, technical, and educational documents that may include additional comments that are not displayed in the output PDF file. This macro can also be used in the LaTeX system when preparing educational documents that include invisible proofs of theorems, lemmas, explanations, solutions of problems, auxiliary and other text elements so that the student can solve the problems himself. This macro can also be used to create tests, which is especially important in online learning (Sarac and Durakovic, 2022) and in active learning (Romansky, 2023).

The macro from item 3 can be used in the LaTeX system to create scientific, technical, and educational documents in which we can choose the language of a text for the appropriate audience. This macro is especially useful in online learning for multilingual audiences. The importance of the ability to select the language of a text is confirmed by investigations (Šostaka and Borzovs, 2023; Šostaka, Borzovs et al., 2023) that show the importance of formation (from ICT terms formed in English) of secondary ICT terms in other languages, in particular in Latvian.

These new macros expand the functionality of the LaTeX system and increase the efficiency of preparing structured documents. They automate the creation of texts that are relevant and meaningful for scientists, professors, and engineers.

2 The methodology and generalized model of the creation of macros in the LaTeX system

Let us present the methodology of the creation of macros in the LaTeX system:

- analysis of the necessity for creating new macros. At the same time, consideration is given to new functions that should be implemented to improve the presentation of various documents.
- Structural design.
  I. Macro hierarchy:
     — basic macros are macros that typically implement a single function;

— combined macros are macros that combine basic macros into blocks;
— universal macros are macros that can adapt to different contexts.

II. Modularity.
Each macro or a group of macros is placed in its own package or a separate file.

III. Dependencies.
Macros should not depend on each other, except for explicit interfaces (such as arguments).

- Implementing macros.
- Testing macros.

This methodology of the creation of macros in the LaTeX system is the generalized model of the creation of the macros.

Three key components can be identified in this model:

1. packages;
2. the macros that implement specific tasks;
3. the arguments of macros.

Arguments provide adaptability to macros: a single macro can implement different actions depending on the arguments without changing its source code.

The generalized model systematizes the process of the creation of macros in the LaTeX system, ensures the flexibility, the reuse, and the scalability of macros in various documents.

In a LaTeX macro, the result is determined by the set of its arguments and the body of the macro, whereas in a Microsoft Word macro, the result also depends on the current state of a document and the execution environment. The tasks implemented by a single macro in the LaTeX system require Microsoft Word to write procedural subroutines with the direct control of the Document Object Model, which significantly complicates the source code.

3 Results and discussion

3.1 Creating the macro for keywords

Let us solve the following problem in the LaTeX system. We print a keyword in several places of Section 1 of the output PDF file (see Fig. 3). To do this, in a LaTeX document, we mark Section 1 with the keyword that is not printed to the output PDF file by calling the new macro `\keywdef` (see Fig. 1 and Fig. 2). Then we print this keyword in several places of Section 1 of the output PDF file by calling the inner macro `\keyw` of the macro `\keywdef` (see Fig. 2). We perform the same operations with another keyword for Section 2 by calling the same macros `\keywdef` and `\keyw`.

In order to solve this problem, in the LaTeX document, we create the new macro `\keywdef` with one argument `#1` that represents the keyword. In this macro, we define the empty macro `\keyw` and redefine the macro `\keyw` by using `\renewcommand` so that this macro takes one argument `#1`. The macro `\keyw` prints the passed argument

each time the macro is called. The source code of the created macro for keywords is given in Fig. 1.

```
\newcommand{\keywdef}[1]{%
  \def\keyw{}
  \renewcommand{\keyw}{#1}
}
```

Fig. 1. The created macro for keywords (the source code is available here)

In the body of the LaTeX document, we mark Section 1 with the keyword 'Q4-2024' that is not printed to the output PDF file by calling the created macro `\keywdef` and passing the argument 'Q4-2024' to the inner macro `\keyw` of the macro `\keywdef`. Then we print the argument 'Q4-2024' of the macro `\keyw` to the output PDF file each time the macro `\keyw` is called. We mark Section 2 with the other keyword 'Q1-2025' that is not printed to the output PDF file by calling the macro `\keywdef` again and passing the argument 'Q1-2025' to the inner macro `\keyw` of the macro `\keywdef`. Then we print print the argument 'Q1-2025' of the macro `\keyw` to the output PDF file each time the macro `\keyw` is called (Fig. 2).

```
\documentclass{article}

\begin{document}

\section{Quarterly Financial Report for Q4-2024}
\keywdef{Q4-2024}
Note that \keyw\ is the last quarter of the fiscal year
covered in this report. This report includes detailed analyses
of revenue, expenses, and profit margins for \keyw.
Additionally, the market trends observed during \keyw\ are
summarized to provide a comprehensive overview of the
quarter's performance \ldots

\section{Quarterly Financial Report for Q1-2025}
\keywdef{Q1-2025}
Note that \keyw\ is the first quarter of the fiscal year
covered in this report. This report includes detailed analyses
of revenue, expenses, and profit margins for \keyw.
Additionally, the market trends observed during \keyw\ are
summarized to provide a comprehensive overview of the
quarter's performance \ldots

\end{document}
```

Fig. 2. Calling the macros `\keywdef` and `\keyw` in the body of the LaTeX document

The source code in Fig. 1 and Fig. 2 prints the keywords to the output PDF file each time the macro `\keyw` is called (Fig. 3).

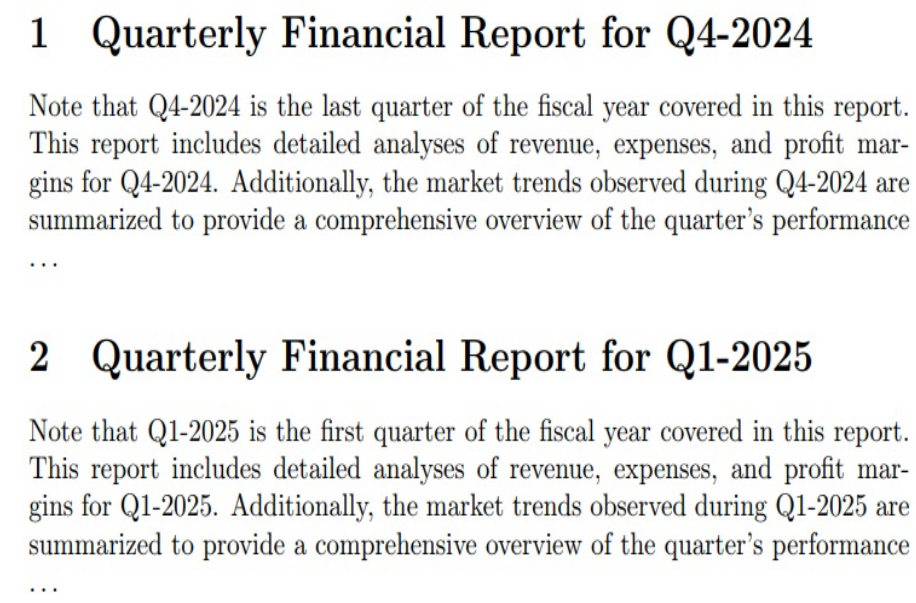


Fig. 3. The result of calling the macros `\keywdef` and `\keyw` in the output PDF file

The created macro `\keywdef` can be used in other macros that change text formatting, create structures unique to the text element marked by a keyword.

The created macro `\keywdef` can be used to create scientific, technical, educational documents, and their templates in which we can also include keywords that mark any other text elements. These keywords may contain hints, notes, short explanations, references to bibliography, and data that need to be repeated. If we change a keyword that marks any text element by calling the macro for keywords `\keywdef`, then the updated keyword is printed by the inner macro `\keyw` of the macro for keywords `\keywdef` each time the inner macro is called. If there is no need to use standard bibliography tools, then by calling the created macro `\keywdef`, we can mark each section by a keyword and print this keyword by calling the macro `\keyw`. This keyword contains bibliographic references related to this section.

3.2 Creating the macro to control text visibility

Let us solve the following problem in the LaTeX system. In a LaTeX document, we make a given box of a text invisible in the output PDF file. At the same time, let us reserve the option to make this box of the text visible. We reserve the space occupied by this box

of the text so that we can handwrite or type the text in the blank space. This box of the text can contain the proof of a theorem, a lemma, a problem solution, an explanation, a hint, a note, and any other data.

In order to do this, in the LaTeX document, we use the 'amsmath' package for advanced capabilities for working with mathematical expressions and the 'amsthm' package for creating environments for theorems, lemmas, and proofs. We use the 'etoolbox' package for conditional logic programming.

We also use the 'tcolorbox' package to create colored boxes with customizable box styling options, and we load the 'skins' library to style different types of content and the 'breakable' library to automatically break the box from one page to another if the content of the box is too long. We define the theorem environment with the 'theorem' name and the 'Theorem' caption.

We create a new macro `\textcontrol` that provides a way to control the visibility of a given box of a text, i.e., provides a way to make it invisible or visible in the output PDF file. At the same time, the space occupied by this box of the text is reserved.

The macro `\textcontrol` takes two arguments:

- `#1` is the first argument to control the visibility of the given box of the text;
- `#2` is the second argument to input the content of the given box of the text.

In the macro `\textcontrol`, we create a local group for commands that must work within this group. In this group, creating the command `\control{visible}`, we set the text to be visible by default. Then using the command `\ifstrempty` from the 'etoolbox' package, we check whether a value is passed to the argument `#1`. If the value is not passed to the argument `#1`, then the text remains visible in the box. If the value is passed to the argument `#1`, then the given text becomes invisible when we use the command `\control{invisible}`.

We customize the style of the box in the 'tcolorbox' environment provided by the 'tcolorbox' package. We set white background, white frame, all rules of the frame to 0pt, sharp corners, enhanced styling option, automatic box breaking from one page to another, the left and right spaces between all text parts and frame to negative numbers (to keep all text parts closer to the frame), the vertical space before the box to 0ex and the vertical space after the box to 1ex. Inside the box, we place the text (for example, the proof of the theorem from a higher mathematics course) in the 'proof' environment with the argument `#2`. After that, we complete the 'tcolorbox' environment and the local group of commands. The used 'amsmath', 'amsthm', 'etoolbox', 'tcolorbox' packages, the used 'skins', 'breakable' libraries, and the created new macro `\textcontrol` are shown in Fig. 4.

```
\newcommand{\textcontrol}[2][]{%
  \begingroup
  \def\control{visible}
  \ifstrempty{#1}{}{\def\control{invisible}}%
  \tcolorbox[
  \control,
    colback=white,
    colframe=white,
    boxrule=0pt,
    sharp corners,
    enhanced,
    breakable,
    left=-0.5ex,
    right=-0.5ex,
    before skip=0ex,
    after skip=1ex
  ]
  \begin{proof}
   #2
  \end{proof}
  \endtcolorbox
  \endgroup
```

Fig. 4. The created macro `\textcontrol` that enables to control the visibility of the given box of the text (the source code is available here)

In the body of the LaTeX document, we call the created macro `\textcontrol` to which we pass the value 'off ' for the first argument and input the text (for example, the proof of the theorem from a higher mathematics course placed in the 'proof' environment) for the second argument (Fig. 5).

The source code in Fig. 4 and Fig. 5 makes the proof of the theorem from a higher mathematics course invisible in the output PDF file (Fig. 6).

If we pass any value other than 'off' to the first argument of the macro `\textcontrol` in the body of the LaTeX document, then the given box of the text will also be invisible. If we pass an empty word to the first argument of the macro `\textcontrol` (we leave the first argument empty), then the given box of the text will be visible (Fig. 7).

```
\documentclass{article}

\usepackage{amsmath}
\usepackage{amsthm}
\usepackage{etoolbox}
\usepackage{tcolorbox}
\tcbuselibrary{skins,breakable}
\newtheorem{theorem}{Theorem}

\begin{document}

Let us discuss a property of solutions of a homogeneous linear
equation
\begin{equation}\label{eq1}
y^{\prime\prime}+a_1(x)y^{\prime}+a_2(x)y=0.
\end{equation}

It is supposed that the functions \(a_1(x)\) and \(a_2(x)\) are
continuous in an interval \((a,b)\) which can be finite or
infinite.

\begin{theorem}
If \(y_1(x)\) and \(y_2(x)\) are two solutions of homogeneous
linear equation \eqref{eq1}, then the function \(y=C_1y_1(x)
+C_2y_2(x)\) is also a solution of the equation for any
constants \(C_1\) and \(C_2\).
\end{theorem}

For brevity, we write these solutions as \(y_1\) and \(y_2\).

\textcontrol[off]{%
  Differentiate the function \(y=C_1y_1+C_2y_2\) twice:
\begin{equation*}
y^{\prime}=C_1y_1^{\prime}+C_2y_2^{\prime},\quad
y^{\prime\prime}=C_1y_1^{\prime\prime}+C_2y_2^{\prime\prime}.
\end{equation*}
Substituting \(y,y^\prime,y^{\prime\prime}\) into the left-hand
side of equation \eqref{eq1}, we obtain
\begin{multline*}
 C_1y_1^{\prime\prime}+C_2y_2^{\prime\prime}+a_1(x)
(C_1y_1^{\prime}+C_2y_2^{\prime})+a_2(x)(C_1y_1+C_2y_2)=
 \\
 =C_1(y_1^{\prime\prime}+a_1(x)y_1^{\prime}+a_2(x)y_1)
+C_2(y_2^{\prime\prime}+a_1(x)y_2^{\prime}+a_2(x)y_2).
\end{multline*}

The expressions in the parentheses are the results of the
substitution of the functions \(y_1\) and \(y_2\) in
\eqref{eq1}; since these functions are solutions of \eqref{eq1},
then both expressions are identically equal to zero, and hence
the function \(y=C_1y_1+C_2y_2\) satisfies equation \eqref{eq1}.
}

We can discuss other properties of solutions of a homogeneous
linear equation \eqref{eq1}.
\end{document}
```

Fig. 5. Calling the created macro `\textcontrol` in the body of the LaTeX document so that the proof of the theorem from a higher mathematics course is invisible

Let us discuss a property of solutions of a homogeneous linear equation

$$y'' + a_1(x)y' + a_2(x)y = 0. \tag{1}$$

It is supposed that the functions $a_1(x)$ and $a_2(x)$ are continuous in an interval (a, b) which can be finite or infinite.

Theorem 1. *If $y_1(x)$ and $y_2(x)$ are two solutions of homogeneous linear equation (1), then the function $y = C_1y_1(x) + C_2y_2(x)$ is also a solution of the equation for any constants C_1 and C_2.*

For brevity, we write these solutions as y_1 and y_2.

We can discuss other properties of solutions of a homogeneous linear equation (1).

Fig. 6. The result of calling the created macro `\textcontrol` is the proof of the theorem from a higher mathematics course made invisible in the PDF file

Let us discuss a property of solutions of a homogeneous linear equation

$$y'' + a_1(x)y' + a_2(x)y = 0. \tag{1}$$

It is supposed that the functions $a_1(x)$ and $a_2(x)$ are continuous in an interval (a, b) which can be finite or infinite.

Theorem 1. *If $y_1(x)$ and $y_2(x)$ are two solutions of homogeneous linear equation (1), then the function $y = C_1y_1(x) + C_2y_2(x)$ is also a solution of the equation for any constants C_1 and C_2.*

For brevity, we write these solutions as y_1 and y_2.

Proof. Differentiate the function $y = C_1y_1 + C_2y_2$ twice:

$$y' = C_1y_1' + C_2y_2', \quad y'' = C_1y_1'' + C_2y_2''.$$

Substituting y, y', y'' into the left-hand side of equation (1), we obtain

$$\begin{aligned} C_1y_1'' + C_2y_2'' + a_1(x)(C_1y_1' + C_2y_2') + a_2(x)(C_1y_1 + C_2y_2) = \\ = C_1(y_1'' + a_1(x)y_1' + a_2(x)y_1) + C_2(y_2'' + a_1(x)y_2' + a_2(x)y_2). \end{aligned}$$

The expressions in the parentheses are the results of the substitution of the functions y_1 and y_2 in (1); since these functions are solutions of (1), then both expressions are identically equal to zero, and hence the function $y = C_1y_1 + C_2y_2$ satisfies equation (1). □

We can discuss other properties of solutions of a homogeneous linear equation (1).

Fig. 7. The result of calling the created macro `\textcontrol` is the proof of the theorem from a higher mathematics course made visible in the PDF file

In order to make parts of a text invisible in the LaTeX system, we can choose the white text color that prints on white paper. However, the text made invisible by using this approach can be copied and pasted to the corresponding page in a PDF viewer, which is a disadvantage of this approach when testing students' knowledge. Therefore, it is advisable to use the created macro `\textcontrol`, which provides a way to control the visibility of the given box of the text, i.e., provides a way to make it invisible or visible in the output PDF file determined by the value passed to the first argument of the created macro `\textcontrol`. At the same time, this macro uses the tools of the 'tcolorbox' package, and there is no way to copy and paste an invisible text.

3.3 Creating the macro for printing text in a selected language

Let us solve the following problem in the LaTeX system. Let a LaTeX document have the text written in English, German, and French. It is necessary that the text be written, for example, in English in the output PDF file after compiling the LaTeX document. Solving such a problem is necessary to adapt the text to an audience that speaks a particular language.

For example, the LaTeX document can contain the text in several languages. Changing the language in the preamble of the LaTeX document and compiling this document, we obtain the output PDF file with the text in the selected language, and the text in other languages will not be printed. This is convenient so that the text written in different languages is in one LaTeX document.

In order to do this, let us use the 'fontenc' and 'inputenc' packages to determine the character encoding in the LaTeX document. We connect the 'babel' package to support writing in different languages and customize typographic and hyphenation rules. We pass German, French, and English in the list of language options of the 'babel' package. At the same time, the last language name passed as an option is English, and it becomes the document's current language. Then we connect the 'etoolbox' package for conditional logic programming.

The variable `\languagename` provided by the 'babel' package displays the current language name in the document, i.e., English that is the last in the list of language options of the 'babel' package. The last language first is a principle predetermined by the 'babel' package.

We define a variable `\langnm` in which we store `\languagename`. If we do not store `\languagename` in the variable `\langnm`, then the source code will not work correctly because `\languagename` changes dynamically.

We create a new macro `\langchange` that tests the current language of the LaTeX document using the command `\ifdefstring` from the 'etoolbox' package and prints the text (for example, on higher mathematics) in the current language stored in the variable `\langnm` (i.e., in English) to the output PDF file (Fig. 8).

```
\let\langnm\languagename
\newcommand{\langchange}{%
  \ifdefstring{\langnm}{english}{%
    The linear space \(L^1(\rm{R}^{\nu s}\times \rm{R}^{\nu
s})\) of summable functions \(f_s(x_1,\ldots,x_s)\) defined on
the phase space \(\rm{R}^{\nu s}\times \rm{R}^{\nu s}\) and
invariant under permutations of the arguments \((x_1,
\ldots,x_s)\) has the norm
\[
\|f_s\|=\int\limits_{\rm{R}^{\nu s}\times \rm{R}^{\nu s}}
\,dx_1\ldots dx_s |f_s(x_1,\ldots,x_s)|.
\]
  }{%
    \ifdefstring{\langnm}{german}{%
      Der lineare Raum \(L^1(\rm{R}^{\nu s}\times \rm{R}^{\nu
s})\) der summierbaren Funktionen \(f_s(x_1,\ldots,x_s)\), die
im Phasenraum \(\rm{R}^{\nu s}\times \rm{R}^{\nu s}\)
definiert sind und invariant unter Permutationen der Argumente
\((x_1,\ldots,x_s)\) sind, hat die Norm
\[
\|f_s\|=\int\limits_{\rm{R}^{\nu s}\times \rm{R}^{\nu s}}
\,dx_1\ldots dx_s |f_s(x_1,\ldots,x_s)|.
\]
    }{%
      L'espace linéaire \(L^1(\rm{R}^{\nu s}\times \rm{R}^{\nu
s})\) des fonctions sommables \(f_s(x_1,\ldots,x_s)\),
définies sur l'espace des phases \(\rm{R}^{\nu s}\times \rm{R}
^{\nu s}\) et invariantes par permutations d'arguments \((x_1,
\ldots,x_s)\), a pour norme
\[
\|f_s\|=\int\limits_{\rm{R}^{\nu s}\times \rm{R}^{\nu s}}
\,dx_1\ldots dx_s |f_s(x_1,\ldots,x_s)|.
\]
    }%
  }%
}
```

Fig. 8. The created macro `\langchange` that tests the current language of the LaTeX document and prints the text (for example, on higher mathematics) in the current language (English) to the output PDF file (the source code is available here)

In the body of the LaTeX document, we call the created macro `\langchange` (Fig. 9).

```
\documentclass{article}

\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[german,french,english]{babel}
\usepackage{etoolbox}

\begin{document}

\langchange

\end{document}
```

Fig. 9. Calling the created macro `\langchange` in the body of the LaTeX document

The result of calling the created macro `\langchange` is the text (for example, on higher mathematics) printed in English (English was stored in the variable `\langnm`) to the output PDF file (Fig. 10).

The linear space $L^1(\mathrm{R}^{\nu s} \times \mathrm{R}^{\nu s})$ of summable functions $f_s(x_1, \ldots, x_s)$ defined on the phase space $\mathrm{R}^{\nu s} \times \mathrm{R}^{\nu s}$ and invariant under permutations of the arguments $(x_1, \ldots, x_s)$ has the norm

$$\|f_s\| = \int\limits_{\mathrm{R}^{\nu s} \times \mathrm{R}^{\nu s}} dx_1 \ldots dx_s |f_s(x_1, \ldots, x_s)|.$$

Fig. 10. The result of calling the created macro `\langchange` is the text (for example, on higher mathematics) printed in the current language (English) to the output PDF file

If the last language name passed in the list of language options of the 'babel' package is German (Fig. 11), then German becomes the document's current language.

```
\usepackage[english,french,german]{babel}
```

Fig. 11. Connection of the 'babel' package assigning German as the current language of the LaTeX document

Then calling the created macro `\langchange` in the body of the LaTeX document (Fig. 9), we obtain the text (for example, on higher mathematics) printed in German to the output PDF file (Fig. 12).

Der lineare Raum $L^1(\mathrm{R}^{\nu s}\times\mathrm{R}^{\nu s})$ der summierbaren Funktionen $f_s(x_1,\ldots,x_s)$, die im Phasenraum $\mathrm{R}^{\nu s}\times\mathrm{R}^{\nu s}$ definiert sind und invariant unter Permutationen der Argumente $(x_1,\ldots,x_s)$ sind, hat die Norm

$$\|f_s\| = \int\limits_{\mathrm{R}^{\nu s}\times\mathrm{R}^{\nu s}} dx_1\ldots dx_s|f_s(x_1,\ldots,x_s)|.$$

Fig. 12. The result of calling the created macro `\langchange` is the text (for example, on higher mathematics) printed in the current language (German) to the output PDF file

If it is necessary to print the text in a language other than the three languages listed above, then this language must be the last in the list of language options of the 'babel' package, and the test `\ifdefstring` for this language must be added in the created macro `\langchange`. The created macro `\langchange` is especially useful in online learning for multilingual audiences.

4 Conclusions

In this article, we presented the methodology and generalized model of the creation of macros in the LaTeX system, and we created the following new macros:

1. the macro for keywords `\keywdef` that enables to mark any text element with a keyword that is not printed to the output PDF file. This keyword can be printed to the output PDF file by the inner macro `\keyw` of the macro for keywords `\keywdef` each time the inner macro `\keyw` is called. If we mark any text element (a previously selected text element or another one) with another keyword by calling the macro for keywords `\keywdef`, then the updated keyword is printed by the inner macro `\keyw` of the macro for keywords `\keywdef` each time the inner macro `\keyw` is called;
2. the macro `\textcontrol` that provides a way to make a given box of a text invisible or visible in the output PDF file;
3. the macro `\langchange` that prints a text in the selected language to the output PDF file created from a LaTeX document in which the text is written in several languages.

The source code of the created macros is presented. The created macros can be used in LaTeX documents of different classes. These macros provide flexibility in text formatting when creating scientific, technical, and educational documents in the LaTeX system. They automate working and increase productivity in this system.

References

Abraham, N. P., Luke, D. M. (2022). Preparing structured document using LaTeX. *APT Tunes*, **1**, 1–5.

Hubal, H. M. (2023). Improvement of references and footnotes in mathematical and other texts by creating macros in the LaTeX programming language. *International Journal on Information Technologies & Security*, **15(3)**, 15–22.

Lamport, L. (1994). LaTeX: *A Document Preparation System: User's Guide and Reference Manual*, 2nd ed., Reading, Massachusetts: Addison-Wesley Professional.

Lode, C. (2019). *Better books with LaTeX the agile way*, Düsseldorf, Germany: Clements Lode Verlag E.K.

Mittelbach, F., Goossens, M., Braams, J., Carlisle, D., Rowley, C. (2004). *The LaTeX companion*, 2nd ed., Boston: Addison-Wesley Professional.

Nazaruka, E. (2020). Processing use case scenarios and text in a formal style as inputs for TFM-based transformations. *Baltic Journal of Modern Computing*, **8(1)**, 48–68.

Romansky, R. (2023). Empirical evaluation of the transfer of information resources in active learning. *International Journal on Information Technologies and Security*, **15(1)**, 39–48.

Sarac, S., Durakovic, B. (2022). Analysis of student performances in online and face-to-face learning: A case study from a Bosnian public university. *Heritage and Sustainable Development*, **4(2)**, 87–94.

Šostaka, D., Borzovs, J. (2023). Towards computer-assisted Latvian ICT terminology development: from theoretical guidelines to case studies. *Baltic Journal of Modern Computing*, **11(1)**, 161–180.

Šostaka, D., Borzovs, J., Cauna, E., Keiša, I., Pinnis, M., Vasiļjevs, A. (2023). The semi-algorithmic approach to formation of Latvian Information and Communication Technology terms. *Baltic Journal of Modern Computing*, **11(1)**, 67–89.

Van Dongen, M.R.C. (2012). LaTeX *and friends*, Springer-Verlag Berlin and Heidelberg GmbH & Co. K.

Zuzevičiūtė, V., Jatautaite, D., Butrime, E. (2025). Contemporary higher education teacher's challenges: The perspective on the AI in studies. *Baltic Journal of Modern Computing*, **13(1)**, 304–314.

Received January 21, 2026 , revised February 23, 2026, accepted April 12, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, pp. 308–327
https://doi.org/10.22364/bjmc.2026.14.2.03

InstanceSketch: Robust Sketch Instance Segmentation with Cluster Refinement

Yana SVIRHUNENKO, Yaroslav TERESHCHENKO, Pavlo TYTARCHUK

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

yana.svirhunenko@knu.ua, y.tereshchenko@knu.ua, pavlo.tytarchuk@knu.ua

ORCID 0009-0002-2300-0655, ORCID 0000-0002-8451-7634, ORCID 0009-0006-4783-1946

Abstract. Stroke segmentation is crucial for interpreting free-hand sketches, supporting tasks ranging from recognition and generative modeling to semantic analysis. However, existing methods often struggle to distinguish between similar component categories and typically rely on large models unsuitable for on-device deployment.
In this paper, we present *InstanceSketch*, a two-step pipeline designed to segment complex sketch objects, and *InstanceSketch-Scene*, an extension that decomposes entire scenes containing multiple objects into semantically meaningful parts. By integrating convolutional neural networks and Transformers, our approach enables efficient breakdown of sketches into interpretable components. An important advancement is a novel method for refining labels of similar components using clustering algorithms to ensure accurate separation between visually similar stroke groups. Our proposed method not only matches but also exceeds the performance of existing state-of-the-art techniques, while operating efficiently with significantly fewer resources, making it an attractive solution for on-device applications and serverless environments.

Keywords: stroke segmentation, sketch analysis, deep learning, transformers, clustering

1 Introduction

Free-hand sketches, despite their abstract nature, follow an underlying semantic structure where each stroke or group of strokes represents a distinct and meaningful part of the object. The stroke segmentation task typically involves assigning a unique semantic label to each stroke to determine the component to which it belongs (see Figure 1). Understanding and segmenting these components is crucial for a wide range of applications, including sketch-based retrieval, generative modeling, and human-computer interaction. Despite being well-studied, stroke segmentation remains challenging due to the varying levels of abstraction, diverse depiction styles, and the inherent sparsity and noise in free-hand sketch data.

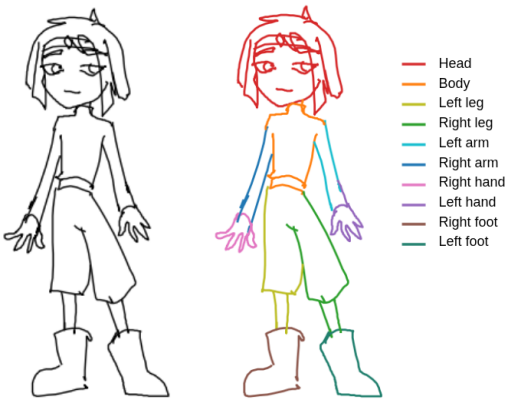


Fig 1. Visualization of the stroke segmentation task. The figure shows an input sketch (left) and the corresponding color-coded stroke labels. Example drawn from the authors' custom dataset.

Existing stroke segmentation techniques generally fall into two broad categories: rule-based and learning-based approaches. Early rule-based methods utilized handcrafted geometric heuristics—such as stroke proximity, continuity, and curvature—to group strokes into semantically meaningful parts (Sun et al., 2012; Stahovich, 2004; Herold and Stahovich, 2011). While conceptually simple, these methods lack robustness when applied to the high variability and abstraction of freehand sketches.

Recent research has turned to deep learning-based methods. For instance, SketchGNN (Yang et al., 2021) models a sketch as a graph, where nodes represent points, and edges encode stroke connectivity, and applies graph convolutions to predict point-level semantic labels. While it achieves improved segmentation accuracy, it struggles with boundary precision in sketches with high abstraction. S3NN (Zhang et al., 2022) introduces a hybrid CNN-RNN-GNN architecture that combines visual context, sequential stroke information, and spatial refinement. Although effective on complex scene-level sketches, it incurs high computational costs and slow inference times.

Other learning-based strategies include the dual-CNN approach (Zhu et al., 2018), which uses two CNN streams to extract stroke-wise features for part-level sketch segmentation. CNN-CRF models (Zhu et al., 2020), which leverage Conditional Random Fields (CRFs) to refine CNN outputs by modeling spatial dependencies for enhanced semantic labeling, have also been proposed. However, these approaches rely heavily on specific stroke-level features, which limits their generalization across diverse sketching styles. RNN-based models such as SketchSegNet (Wu et al., 2018) treat stroke segmentation as a sequence-to-sequence problem, using RNNs to encode the stroke sequence and decode semantic labels. Yet, this method is sensitive to variations in drawing order, a common challenge in unconstrained sketch input.

Recent advancements in deep learning have led to the popularization of Transformer-based architectures for sketch understanding, particularly due to their ability to model long-range dependencies between disparate strokes. One notable example is the Sketch-

ESC model (Zhu et al., 2024), which utilizes a Transformer backbone with a semantic component-level memory module. This module maintains a learnable memory bank containing prototypical stroke group representations learned from training data, enabling the model to focus on key structural elements within each sketch category. However, its reliance on a Long Short-Term Memory (LSTM)-based stroke embedding extractor reduces effectiveness when dealing with long or highly variable stroke sequences.

Another prominent architecture is ContextSeg (Wang and Li, 2024), which combines a CNN-based stroke encoder with an autoregressive Transformer; however, this mechanism is prone to error accumulation and results in slower inference times.

HCTSketch (Svirhunenko et al., 2025) combines a CNN for feature extraction with a Vision Transformer (ViT) (Dosovitskiy et al., 2021) to enable simultaneous stroke segmentation, addressing limitations of earlier methods. However, it often struggles with visually similar components, frequently mislabeling or mixing them.

Despite notable progress, a clear research gap remains: existing methods commonly fail to distinguish between structurally symmetric or visually similar components (e.g., left vs. right limbs of an animal), rely on heavyweight architectures with tens of millions of parameters that are unsuitable for on-device deployment, and assume single-object inputs, which limits their applicability to real-world scene-level sketches.

The central research question is: *Can a lightweight two-stage pipeline combining CNN and Transformer encoders with cluster-based label refinement match or exceed the segmentation accuracy of larger, established models?* The aim of this paper is to develop a lightweight yet accurate stroke segmentation pipeline that overcomes these shortcomings, using a methodology that integrates convolutional feature extraction, Transformer-based stroke encoding, and unsupervised clustering for refinement. Specifically, the research objectives are:

1. To design a two-stage pipeline that combines an initial stroke classification model with a cluster-based refinement step capable of resolving ambiguity among visually similar components without requiring an explicit count of those components.
2. To demonstrate that competitive segmentation accuracy can be achieved with significantly fewer parameters than existing methods, making the pipeline suitable for resource-constrained environments.
3. To extend the single-object pipeline to multi-object scene sketches by incorporating an object detection stage that isolates individual objects before segmentation.

The experiments are conducted on a custom dataset combining manually annotated sketches with images drawn from the publicly available Quick, Draw! dataset (Ha and Eck, 2018), covering seven object categories.

To achieve these objectives, we present *InstanceSketch*, a novel two-stage pipeline for stroke segmentation in freehand sketches that we propose and develop entirely in this work. In contrast to other methods that rely solely on direct label assignment, the final stage of our pipeline refines labels by learning discriminative stroke embeddings that are then used for clustering, allowing the method to adapt to cases where the exact number of components is unknown.

We also present *InstanceSketch-Scene*, a scene-level extension of *InstanceSketch* developed as a further original contribution, which prepends a YOLO-based detection

stage to isolate individual objects and infer their classes, used as context to guide per-object stroke segmentation (Section 7.1).

Both *InstanceSketch* and *InstanceSketch-Scene* achieve accurate and efficient performance, which is essential for applications ranging from digital art editing to autonomous sketch interpretation systems.

2 Proposed Method

Given an input sketch composed of separate strokes represented as point sequences, our goal is to assign a semantic label to each stroke, indicating the object part or component to which it belongs. Our method introduces a robust pipeline designed to segment free-hand sketches, which consists of two stages:

1. **Initial Stroke Segmentation.** Utilizes a model comprising CNN and Transformer blocks. CNNs are effective at extracting features from strokes represented as 2D images, while Transformer encoders leverage attention mechanisms to capture relationships between different strokes. Finally, a classification head is used to get the initial label for each stroke.
2. **Stroke Label Refinement.** Enhances the initial segmentation by refining labels for challenging cases where similar-looking components, like different limbs of an animal, might be hard to distinguish. Uses a similar architecture (CNN followed by Transformer) to obtain stroke embeddings that can be used for clustering in high-dimensional space. This allows us to separate strokes that have the same initial label but belong to different structural components.

The diagram of the proposed method is presented in Figure 2.

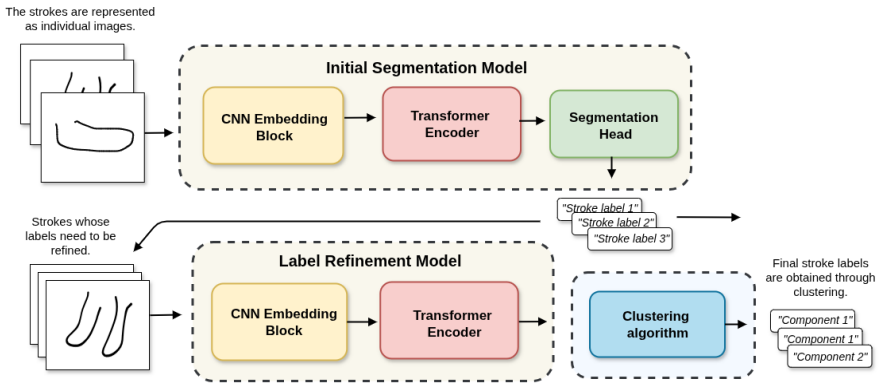


Fig 2. Diagram of the proposed processing pipeline.

2.1 Segmentation Model

To perform the initial segmentation of each stroke within the sketch, we generate two distinct image representations:

1. In the first format, each individual stroke is centered and scaled to occupy the entire image canvas. This normalization enables the model to focus on the stroke's overall shape, while preserving fine-grained details within a compact input size, thus maintaining computational efficiency. This enables efficient representation even for short stroke segments.
2. In the second format, each stroke is rendered at its original position and scale within the image space. When these images are stacked, the original sketch can be reconstructed. This representation allows the model to learn spatial context, including the stroke's position and size within the sketch. Empirically, this approach proved more effective than explicitly providing bounding box coordinates as additional input.

Each pair of stroke images is initially processed through a convolutional module to extract embeddings. The architecture employs a dual-branch design, where two parallel convolutional pipelines independently encode shape and spatial representations of each stroke. Each branch consists of multiple convolutional blocks, each comprising 2D convolutional layers, batch normalization, ReLU activations, and max pooling. These vectors are then fused via a small multi-layer perceptron (MLP) to form a unified embedding that captures both geometric and positional characteristics.

After obtaining stroke embeddings, we fuse them with order embeddings using linear layers to incorporate information about the stroke drawing sequence. The Transformer block employs multi-head self-attention to model long-range dependencies between strokes. After processing the stroke embeddings with the Transformer encoder, they are passed through an MLP that performs stroke-wise segmentation, predicting a label for each stroke.

Since different stroke types occur with varying frequencies, we compensate for this imbalance during training by applying class-specific weights in the cross-entropy loss. To compute these weights, we first calculate the ratio between the median class frequency and each individual class frequency. We then apply a normalization by taking the square root of each value. Finally, the weights are normalized to have a mean of one before being used in the loss function.

The weighted cross-entropy loss is defined as

$$\mathcal{L}_{\text{seg}} = -\sum_{i=1}^{n} w_{s_i}\, s_i \log z_i, \tag{1}$$

where s_i is the true class label of sample i, z_i is the predicted probability of the correct class, and w_{s_i} is the weight associated with class s_i.

The class weights are computed as

$$w_s = \frac{r_s}{\frac{1}{n}\sum_{j=1}^{n} r_j}, \quad r_s = \sqrt{\frac{\text{median}(\{f_k\})}{f_s}}, \tag{2}$$

where f_s is the frequency of class s, median$(\{f_k\})$ is the median of all class frequencies, and n is the total number of classes. This weighting scheme increases the contribution of underrepresented classes while reducing that of more frequent classes.

After segmentation, we obtain initial stroke embeddings. Strokes requiring finer class distinctions undergo additional processing during the Refinement step.

Figure 3 shows a detailed diagram of the segmentation model.

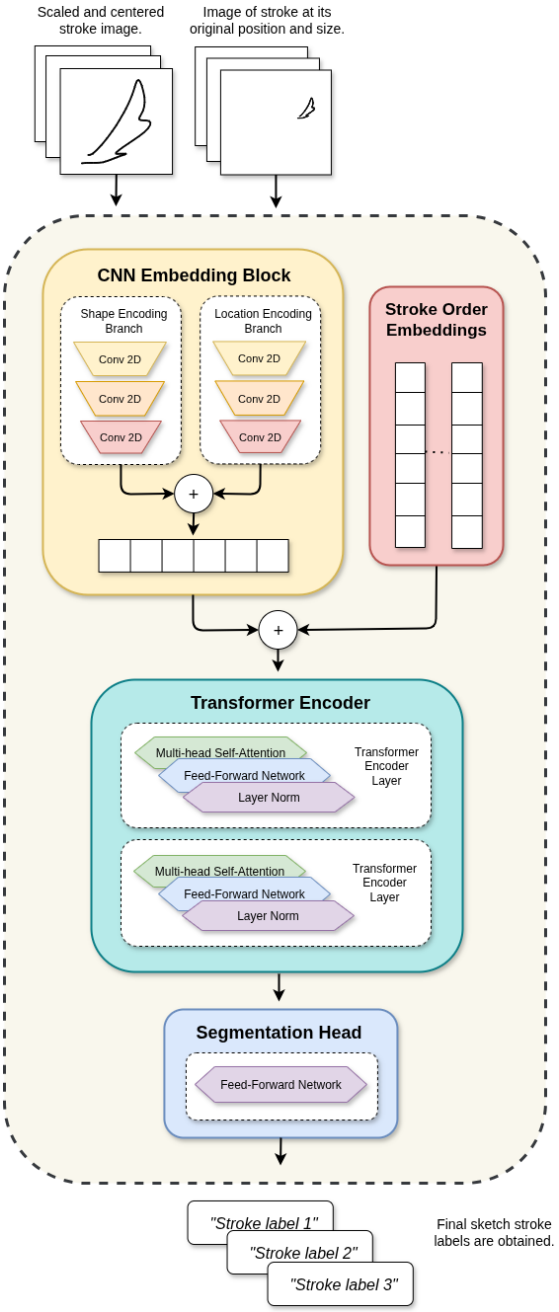


Fig 3. Diagram of the Segmentation Model.

2.2 Refinement Model

While developing a model for stroke segmentation and experimenting with various architectural approaches, we encountered a notable challenge. Although many existing methods perform well when distinguishing strokes that belong to clearly different components, their performance degrades significantly when attempting to classify strokes that are part of similar or symmetric components.

Consider the case of segmenting sketches of animals. Identifying which strokes correspond to limbs in general is relatively straightforward. However, assigning distinct labels to strokes representing different limbs—such as distinguishing the left leg from the right—proves much more difficult. This is because such strokes often appear in highly similar visual and spatial contexts, making them hard to classify directly even for sophisticated models.

Another notable characteristic of sketch data is the high degree of creative freedom and, in some cases, unconventional design. For example, consider a vehicle sketch where the task is to segment all strokes forming the wheels. In certain applications, it may also be necessary to distinguish strokes belonging to individual wheels as separate components. However, due to variability in vehicle types (e.g., a bicycle with two wheels, a car with four, or a fantastical vehicle with N wheels), it becomes impractical to assign labels through direct classification, as the exact number of such components is unknown in advance.

To address this, we introduce a simplifying assumption: instead of assigning semantically specific, predefined labels (e.g., "front left wheel"), we use neutral identifiers such as "wheel 1," "wheel 2," ..., "wheel N." This strategy reduces unnecessary complexity during training and better reflects the inherent ambiguity present in abstract or minimalist sketches.

Instead of direct stroke classification, we decided to employ a clustering approach that would divide strokes into the components they belong to. To produce stroke representations that can be used for clustering, we use a Refinement Model similar to the segmentation step that consists of a convolutional neural network, Transformer encoder, and small multilayer perceptron.

We plot each stroke on a 2D image at its original position and scale within the image space. Unlike the Segmentation Model, no shape representation is needed, as our focus is solely on the relative location of strokes. Stroke images are first processed by a CNN module composed of 2D convolutional layers, ReLU activations, and max pooling operations to extract initial embeddings. Each stroke's initial label obtained during the previous segmentation step is encoded as contextual information and fused with the stroke embeddings and their drawing order embeddings using linear layers. This ensures that the model considers the general component type before attempting to divide it into individual instances. The obtained embeddings are then passed through a Transformer encoder, which employs multi-head self-attention. Finally, a small MLP is used to generate the final stroke representations for clustering tasks.

The model is trained using the triplet loss function (Schroff et al., 2015). This loss function is used to train models to learn meaningful embeddings by comparing three samples at once: an anchor, a positive sample (same class as anchor), and a negative sample (different class). The goal is to minimize the distance between the anchor and

positive while maximizing the distance between the anchor and negative, which encourages the model to produce embeddings for strokes of similar classes closer in high-dimensional space than those from different parts. The loss is computed as:

$$\mathcal{L}_{\text{triplet}} = \frac{1}{n}\sum_{i=1}^{n} \max\left(\|a_i - p_i\|_2^2 - \|a_i - n_i\|_2^2 + m,\ 0\right) \tag{3}$$

where a_i is the anchor sample, p_i is the positive sample (same class as anchor), n_i is the negative sample (different class from anchor), m is the separation margin between clusters, n is the number of triplets, and $\|x - y\|_2$ denotes the Euclidean distance between vectors x and y.

These representations can then be used for clustering using methods like K-Means clustering if the goal is to divide strokes into a specific number of components, or use unsupervised clustering algorithms if we do not know the number of components beforehand.

3 Dataset Details

For training and testing, we gathered a custom dataset. A portion of the images was sourced from the Quick, Draw! dataset (Ha and Eck, 2018), while the rest were manually collected and annotated by our team to ensure a diverse range of drawing styles—from simple sketches to detailed, realistic depictions. This was essential because existing datasets lacked the stroke-level annotations required for our task. The final dataset includes seven classes: four-legged creatures, two-legged creatures, sun, cloud, car, tree, and flower. Examples of images from the collected dataset are shown in Figure 4.

Fig 4. Examples of sketch instances from the dataset.

Each class is annotated with different stroke labels. For example, the four-legged creature class can include up to eight stroke types (e.g., head, body, tail). The actual number of stroke types in an image may vary depending on which components are present.

To enhance model generalization, we apply a range of augmentation techniques during preprocessing. These augmentations are performed directly on the stroke coordinates before rendering, ensuring the strokes remain visible. The applied techniques include:

- **StrokeDropout (Random Stroke Removal):** Selectively removes certain strokes, usually the shortest to preserve important information.
- **Random Scaling:** Alters the sketch's dimensions along one or more axes to introduce shape diversity.
- **Random Rotation:** Rotates sketches by small angles to account for orientation differences.
- **Random Horizontal Flip:** Mirrors sketches along the vertical axis (i.e., flips the sketch left-to-right).
- **Random Stroke Transformations:** Applies stroke-level modifications—including random scaling, rotation, and directional shifts—individually to each stroke, rather than to the entire sketch at once.
- **Gaussian Noise:** Adds random Gaussian noise to the stroke coordinates, followed by smoothing.

These augmentations significantly reduced overfitting during training, demonstrating their effectiveness and improving the model's robustness and its ability to generalize across different sketch styles.

Table 1 provides an overview of the distribution of training and validation samples across different sketch classes. The numbers in the training set reflect the data after balancing and augmentation, while the validation set contains only the original, unaltered images.

Table 1: Number of training and test instances per category in the dataset.

Category	Train Instances	Test Instances
Car	6994	569
Cloud	3186	116
Flower	3798	179
Sun	3702	307
Tree	4179	348
4-Legged Animal	5406	436
2-Legged Animal	3276	181

4 Experiments

4.1 Implementation Details

The model pipeline consists of two models that are trained separately. The training process employs the Adam optimizer, along with an adaptive learning rate scheduler, to promote stable convergence. To mitigate overfitting, dropout regularization is incorporated during training. All models were trained for 150 epochs with a batch size of 128. During training, different model sizes were tested. At the end, we chose parameters that gave the optimal accuracy to model size ratio.

The Segmentation Model employs a CNN-based embedding module that processes both stroke shape and location images, each of size 32×32 with a single channel. These inputs are handled by two separate CNN branches with 3×3 convolutions. The number of feature channels increases from 32 to 64 and then to 128 before applying adaptive average pooling. The resulting features are fused into embeddings of size 128.

In this setup, position embeddings are also 128-dimensional. We use a Transformer encoder with 2 layers, 4 attention heads, and a hidden size of 128—providing a balance between efficiency and model capacity. After encoding, layer normalization is applied, followed by an MLP that maps the refined stroke embeddings to the final stroke labels.

The complete model consists of approximately 665,000 parameters.

The Refinement Model takes as input images that depict strokes in their original spatial positions. Each image is 32×32 pixels with a single channel. These inputs are processed by a CNN with 3×3 convolutional layers. During this process, the number of feature channels increases to 32. The resulting feature maps are then passed through a linear layer to generate embeddings of size 64.

These embeddings are fused with encoded stroke labels and order representations of the same size, and then projected to a 64-dimensional space. Then they are processed by a Transformer encoder module comprising 2 layers, 4 attention heads, and a hidden size of 64. Subsequently, layer normalization is applied. Finally, the normalized embeddings are refined using an MLP, producing the final stroke embeddings of size 32, which can be used for downstream clustering tasks.

The complete model consists of approximately 282,000 parameters.

Our dataset includes detailed stroke-level annotations for four-legged and two-legged creature classes in particular, with a consistent set of limb labels applied across both. In the case of four-legged creatures, each limb is annotated with fine-grained distinction between front and back, as well as left and right sides. This results in eight unique labels:

1. ‘front-right-foot’
2. ‘front-left-foot’
3. ‘front-right-leg’
4. ‘front-left-leg’
5. ‘back-right-foot’
6. ‘back-left-foot’
7. ‘back-right-leg’
8. ‘back-left-leg’

For two-legged creatures, the same label set is used, though with slightly different semantics: the “front” limbs correspond to the upper limbs (arms), and the “back” limbs refer to the lower limbs (legs). In this context, “leg” and “foot” in the upper limbs

refer to the arm and hand, respectively, while in the lower limbs they retain their usual meaning.

While all other labels are classified solely by the Segmentation Model, these specific limb labels are used to train and evaluate our Refinement Model. Initially, our segmentation model classifies each stroke as belonging to either the front or back pair of legs, as well as its type (leg or foot), since this distinction can be made more reliably based on spatial location.

The refinement model then processes the front and back limb stroke groups separately and generates embeddings, which will be used to separate the strokes into distinct legs (left or right).

To assess the separation accuracy, we compare the model's output with the initial annotations. Specifically, we measure how frequently strokes originally labeled with the same orientation (left or right) were correctly grouped into the same limb.

Since here we used the model on leg pairs, as a clustering algorithm we have selected the K-Means algorithm with 2 clusters.

4.2 Performance Metrics

By combining all modules, we obtain a robust and efficient pipeline with approximately 947,000 parameters. After training, we evaluated the pipeline on the validation set using key metrics such as classification and segmentation accuracy and F1-scores.

We also compared our method with other Transformer-based sketch segmentation approaches, including SketchESC (Zhu et al., 2024) and HCTSketch (Svirhunenko et al., 2025). The implementations of SketchESC and HCTSketch were obtained from their respective publicly available GitHub repositories and evaluated without modification using the same validation split. All models were evaluated on the full dataset, considering all sketch categories simultaneously.

To ensure a fair comparison in segmentation accuracy for refined labels, we assess how well each model separates strokes into distinct limbs by considering alternative limb orientations, rather than directly comparing predicted labels to the initial ground truth. Specifically, we evaluate both the original and mirrored versions of the predicted leg labels—where left and right are swapped—and use the better of the two assignments for final evaluation of all models. This approach accounts for directional ambiguity and provides a more robust measure of limb-level separation performance.

Table 2 presents the performance comparison of all three models.

Table 2: Validation performance comparison across models. **InstanceSketch** achieves the highest scores across all metrics.

Model	**Segmentation Accuracy (%)**	**Segmentation F1-Score (%)**
SketchESC	85.43	80.07
HCTSketch	91.17	87.04
InstanceSketch	**93.41**	**90.70**

Our proposed InstanceSketch achieves the highest scores across both metrics, reaching 93.41% segmentation accuracy and 90.70% segmentation F1-score. Compared to HCTSketch, this represents a gain of 2.24 percentage points in accuracy and 3.66 percentage points in F1-score. Relative to SketchESC, the improvement is even larger: 7.98 percentage points in accuracy and 10.63 percentage points in F1-score. These results demonstrate that InstanceSketch not only outperforms both baselines but also provides a substantial improvement in segmentation quality while maintaining a low parameter count.

Inference times for all models—InstanceSketch, HCTSketch (Svirhunenko et al., 2025), and SketchESC (Zhu et al., 2024)—were measured on the validation set using an NVIDIA GeForce RTX 3090 Graphics Processing Unit (GPU) with a batch size of one. The reported times represent the average processing duration per sample across the entire set.

A detailed comparison of the performance of all three models is provided in Table 3.

Table 3: Comparison of model size and inference time for different methods.

Method	**Parameters (Millions)**	**Inference Time (s)**
SketchESC	95.2	0.0118
HCTSketch	7.9	0.0016
InstanceSketch	**0.9**	**0.0029**

Although InstanceSketch has significantly fewer parameters compared to both HCTSketch (Svirhunenko et al., 2025) and SketchESC (Zhu et al., 2024), its inference time is slightly slower than HCTSketch. This is primarily due to its pipeline design, which involves two separate models. Nevertheless, InstanceSketch achieves a much smaller model size while maintaining competitive execution speed.

To assess the contribution of each component in our pipeline, we conducted an ablation study by removing the refinement step and training the segmentation model to classify all strokes jointly. Without the refinement step, InstanceSketch achieved 91.98% segmentation accuracy and 88.39% F1-score. This corresponds to a drop of 1.43 percentage points in accuracy and 2.31 percentage points in F1-score compared to the full model, highlighting the positive impact of the refinement step on segmentation performance.

4.3 Qualitative Analysis

Compared to both HCTSketch (Svirhunenko et al., 2025) and SketchESC (Zhu et al., 2024), our model demonstrates a superior ability to distinguish between elements of different legs, resulting in cleaner and more consistent segmentation. For instance, Figure 5 shows an example of four-legged animal segmentation by different models. Other approaches often confuse the strokes that form the legs, whereas the Refinement step in our approach resolves this problem. Figure 6 presents confusion matrices for leg segmentation in the dataset specifically. InstanceSketch achieves the lowest error rate.

Beyond this improvement, our approach also achieves higher overall segmentation accuracy across a wide range of stroke types.

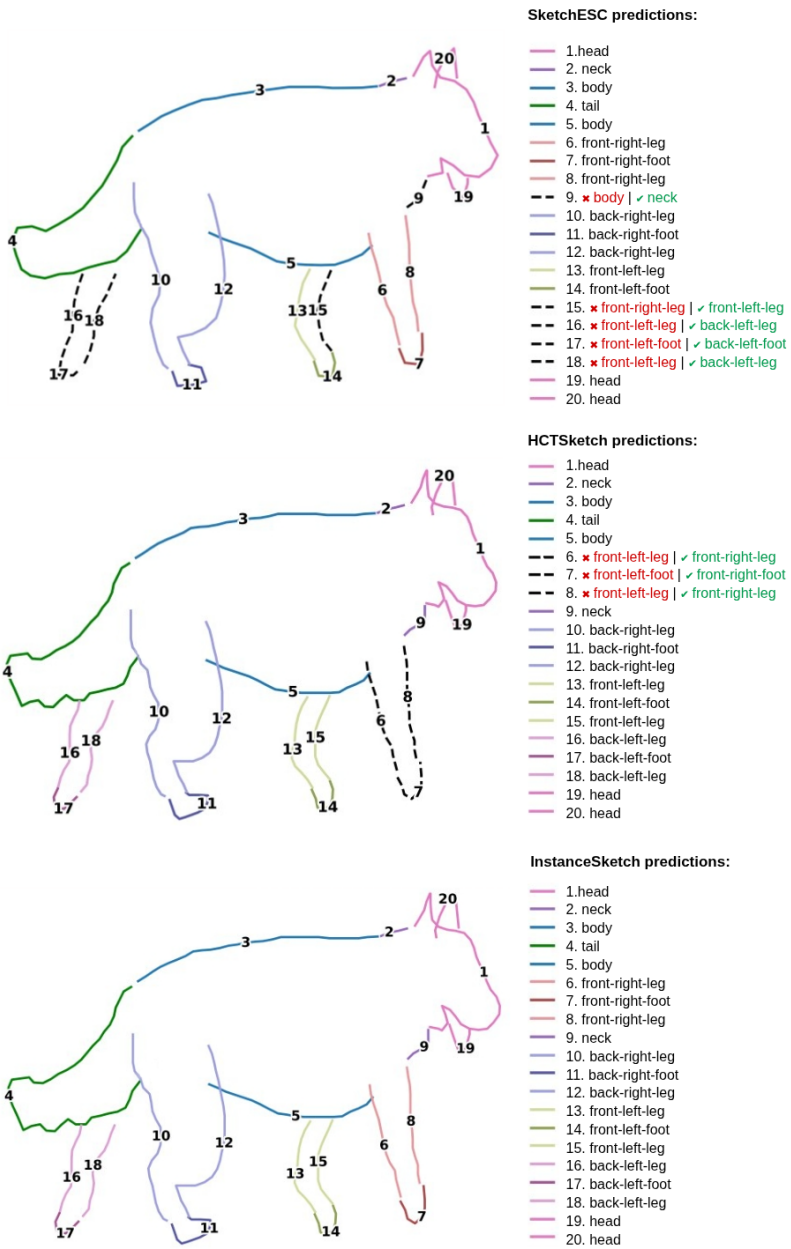


Fig 5. An example of a common error in other stroke segmentation models, where elements of different legs are often confused due to their visual similarity.

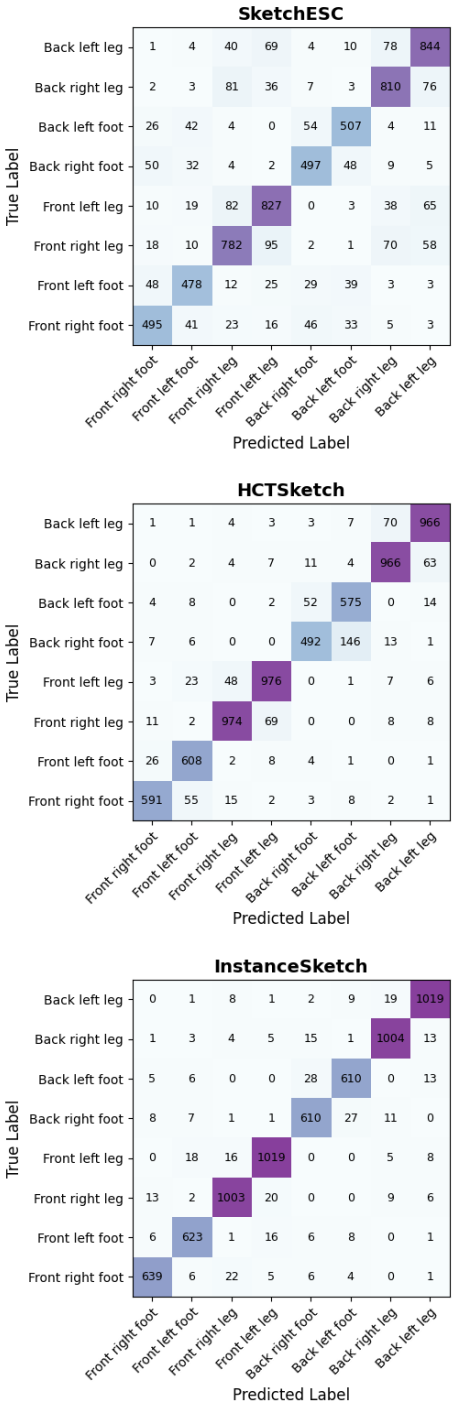


Fig 6. Sections of the confusion matrices showing leg-part segmentation results for different models.

Finally, while the Refinement approach showed strong performance specifically on strokes forming legs, a more comprehensive evaluation would benefit from additional datasets with fine-grained annotations. Such annotations would allow us to better separate and quantitatively assess performance on distinct stroke components, leading to a more detailed understanding of model strengths and limitations.

5 Code Availability

The implementation of InstanceSketch is publicly available at `https://github.com/y-svirhunenko/instance-sketch`

6 Conclusion

In this work, we presented InstanceSketch, a two-step pipeline for segmenting complex sketches, and InstanceSketch-Scene, an extension designed to decompose entire scenes into semantically meaningful components. By integrating convolutional neural networks with Transformers, our approach captures both local stroke-level details and global contextual information, enabling accurate and interpretable segmentation.

A key contribution of our method is the introduction of a flexible label refinement technique, which utilizes clustering techniques to effectively distinguish between strokes belonging to similar components, resulting in improved segmentation of challenging elements.

Experimental results demonstrate that InstanceSketch outperforms state-of-the-art methods in both segmentation accuracy and F1-score, while maintaining a lightweight architecture suitable for on-device deployment. Specifically, InstanceSketch achieves 93.41% segmentation accuracy and 90.70% F1-score, representing gains of 2.24 and 3.66 percentage points over HCTSketch, and 7.98 and 10.63 percentage points over SketchESC, respectively.

For InstanceSketch-Scene, our experiments demonstrate that incorporating sketch class information as contextual input improves stroke segmentation performance. In particular, we observe a +0.5% increase in accuracy and a +0.43% improvement in F1-score (see Appendix, Section 7.1 for detailed results).

Despite these promising results, challenges remain in segmenting rarely occurring decorative details and in handling dense or short strokes. Moreover, additional data is needed to fully evaluate the benefits of the refinement step.

Overall, our framework provides a robust and efficient solution for sketch segmentation, supporting downstream tasks such as recognition, generative modeling, and semantic analysis. Future work will focus on extending the approach to more diverse sketch styles and integrating it into interactive sketch-based applications.

List of Abbreviations

CNN	Convolutional Neural Network
CRF	Conditional Random Field
GNN	Graph Neural Network
GPU	Graphics Processing Unit
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
ViT	Vision Transformer
YOLO	You Only Look Once

Acknowledgements

This research received no external funding. All authors contributed to the conception, design, implementation, and writing of this work. All materials and resources used in this study were prepared by the authors.

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale, *International Conference on Learning Representations (ICLR)*. arXiv:2010.11929.

Ha, D., Eck, D. (2018). A neural representation of sketch drawings, *International Conference on Learning Representations (ICLR)*. arXiv:1704.03477.

Herold, J., Stahovich, T. F. (2011). Speedseg: A technique for segmenting pen strokes using pen speed, *Computers & Graphics* **35**(2), 250–264. DOI: 10.1016/j.cag.2010.12.003.

Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

Stahovich, T. (2004). Segmentation of pen strokes using pen speed, *AAAI Fall Symposium - Technical Report* .

Sun, Z., Wang, C., Zhang, L., Zhang, L. (2012). Free hand-drawn sketch segmentation, *European Conference on Computer Vision (ECCV)*.

Svirhunenko, Y., Tytarchuk, P., Holovko, Y., Tereshchenko, Y. (2025). Hctsketch: Hybrid cnn-transformer approach for stroke segmentation in sketches, *Computer Science Research Notes* **3501**(1), 165–172. DOI: 10.24132/CSRN.2025-18.

Wang, J., Li, C. (2024). Contextseg: Sketch semantic segmentation by querying the context with attention, *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3679–3688. arXiv:2311.16682.

Wu, X., Qi, Y., Liu, J., Yang, J. (2018). Sketchsegnet: A rnn model for labeling sketch strokes, *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. DOI: 10.1109/MLSP.2018.8516988.

Yang, L., Zhuang, J., Fu, H., Wei, X., Zhou, K., Zheng, Y. (2021). Sketchgnn: Semantic sketch segmentation with graph neural networks, *ACM Transactions on Graphics* **40**(3), 1–13. DOI: 10.1145/3450284.

Zhang, Z., Deng, X., Li, J., Lai, Y., Ma, C., Liu, Y., Wang, H. (2022). Stroke-based semantic segmentation for scene-level free-hand sketches, *The Visual Computer* **39**, 6309–6321. DOI: 10.1007/s00371-022-02731-8.

Zhu, G., Wang, S., Wu, T., Zhang, L. (2024). Enhance sketch recognition's explainability via semantic component-level parsing, *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(7), 7731–7738. DOI: 10.1609/aaai.v38i7.28607.

Zhu, X., Xiao, Y., Zheng, Y. (2018). Part-level sketch segmentation and labeling using dual-cnn, *Proceedings of ICONIP*. DOI: 10.1007/978-3-030-04167-0_34.

Zhu, X., Xiao, Y., Zheng, Y. (2020). 2d freehand sketch labeling using cnn and crf, *Multimedia Tools and Applications* **79**, 1585–1602. DOI: 10.1007/s11042-019-08158-z.

7 Appendix

7.1 Scene Segmentation

While our pipeline performs well on single-object sketches, people often draw complex multi-object scenes, where many strokes belong to unrelated objects, providing no useful context and often confusing the model.

To address this challenge, we propose InstanceSketch-Scene, a three-step pipeline that first applies a detection model to identify individual objects, and then leverages the InstanceSketch backbone to perform detailed segmentation.

As the detection component, we utilize YOLO 8n; its object classification output is incorporated as contextual information for stroke segmentation. A scene of size 512×512 is fed into the detector, and strokes are assigned to objects based on the bounding box that contains the majority of their length.

After we get grouped strokes and an object label, we process it using the previously described Segmentation model. However, in this implementation, the model also generates embeddings of size 128 for each sketch class, which are then trained to better facilitate stroke segmentation. An appropriate class representation vector is selected and appended to the sequence of stroke embeddings before being fed into a Transformer encoder model. The Transformer block employs multi-head self-attention to model dependencies between strokes while considering the sketch's overall class. This approach yields better results than simply adding the sketch class to individual stroke representations or fusing them using MLPs. After processing the stroke embeddings with the Transformer encoder, further steps are identical to the InstanceSketch pipeline.

If needed, certain strokes can be further processed by the Refinement model, together with their primary labels.

Since the original dataset contains only isolated objects, we manually constructed a scene dataset to evaluate the full pipeline. Using an adaptive grid approach, individual objects are placed into non-overlapping grid cells with slight random shifts and per-object augmentations. Additional realism rules (e.g., a single sun, floating objects above grounded ones) yielded a training set of 18,000 samples and a test set of 2,000 samples.

To assess classification quality, each stroke is assigned the class label of the object it belongs to; strokes outside any detected bounding box are marked as noise.

To evaluate the impact of incorporating sketch class information on stroke segmentation, we also tested a variant of InstanceSketch-Scene (NoCls), which uses the Segmentation model from the standard InstanceSketch pipeline without class conditioning.

Although incorporating sketch class information as context led to slight improvements in segmentation performance (Segmentation accuracy improved by +0.5% and Segmentation F1-Score by +0.43%), these gains are relatively minor. Moreover, achieving such improvements requires a highly accurate classifier with a considerable number of parameters. Therefore, for most practical cases of single-object sketch segmentation, pre-classification may not be worthwhile. In contrast, for scene segmentation, where class prediction is essential, leveraging the classifier output is a meaningful and effective choice.

Examples of scenes segmented by InstanceSketch-Scene are shown in Figure 7. The obtained metrics are summarized in Table 4.

Fig 7. Examples of images processed by InstanceSketch-Scene.

Table 4: Validation performance comparison between InstanceSketch-Scene variants.

Metric	InstanceSketch-Scene (NoCls)	InstanceSketch-Scene
Class. Accuracy (%)	99.07	**99.07**
Class. F1-Score (%)	98.31	**98.31**
Segm. Accuracy (%)	92.98	**93.48**
Segm. F1-Score (%)	90.60	**91.03**

7.2 Additional Segmentation Examples

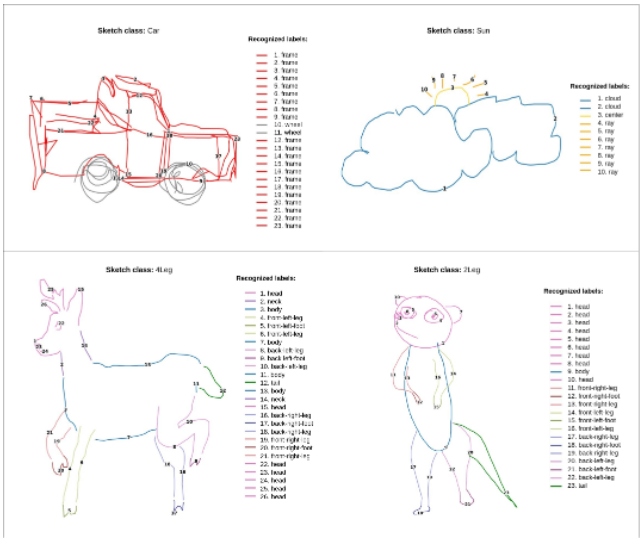


Fig 8. Examples of sketches correctly segmented by InstanceSketch.

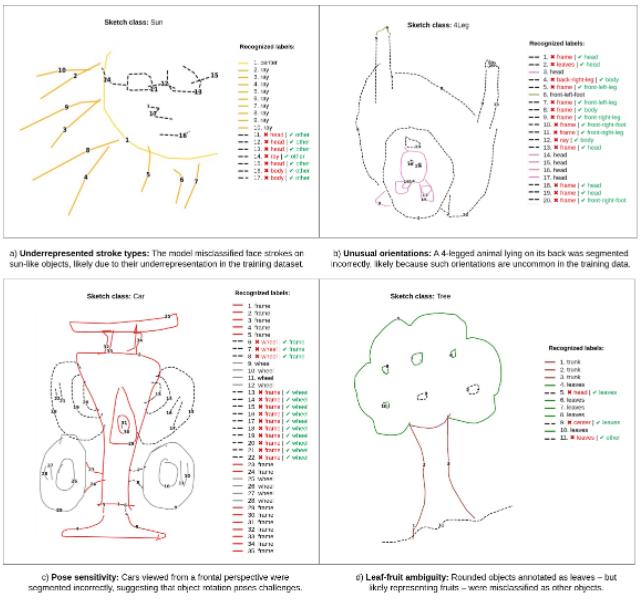


Fig 9. Examples of sketches incorrectly segmented by InstanceSketch.

Received December 28, 2025 , revised April 10, 2026, accepted April 13, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, 328-355
https://doi.org/10.22364/bjmc.2026.14.2.04

Computational Thinking in Technology-Supported Primary Mathematics: A Systematic Review

Eglė JASUTĖ[1], Marika PARVIAINEN[1, 2], Valentina DAGIENĖ[1]

[1]Vilnius University, Institute of Educational Sciences, Universiteto str. 9, 01513 Vilnius, Lithuania
[2]University of Turku, Turku Research Institute of Learning Analytics, 20014 Turku, Finland

egle.jasute@fsf.vu.lt; mhparv@utu.fi; valentina.dagiene@mif.vu.lt

ORCID 0000-0001-5183-9058, ORCID 0009-0000-8055-0317, ORCID 0000-0002-3955-4751

Abstract. This systematic review synthesizes research on how technology-supported approaches to computational thinking (CT) are integrated into primary mathematics learning environments. Following a PRISMA-guided search, 18 studies published between 2016 and 2025 were analyzed to examine how digital tools support CT-related mathematical learning processes in primary school settings. The review identifies recurring patterns linking mathematical domains, CT constructs, and technological environments, illustrating how programming platforms, robotics, game-based tools, and analytics-supported systems are used to structure mathematical problem-solving activities. Across the intervention studies, several reported improvements in students' mathematical problem-solving and reasoning skills. However, the strength of the evidence varies across studies due to differences in study design, sample size, and assessment methods. The analysis further highlights how technology can support CT-informed instructional design by enabling programmable tasks, iterative testing of solutions, and, in some cases, analytics-supported feedback for teachers and students. At the same time, the review points out ongoing gaps in methodological rigor, assessment of the CT construct, and the scalability of interventions. These findings underscore the need for more rigorous, long-term, classroom-based research to strengthen the evidence for CT-integrated primary mathematics education.

Keywords: computational thinking (CT), primary mathematics education, technology-enhanced learning, mathematical problem solving, systematic review, learning analytics

1. Introduction

Digital technologies are transforming how children learn, communicate, and engage with information from the earliest school years. In primary education, these tools play a vital role in shaping students' developing relationship with technology, not just as users but increasingly as creators and problem solvers. Recent advances in digital technologies, including adaptive learning systems, automated feedback tools, and data-driven learning platforms, provide new opportunities for personalized, responsive learning experiences in primary schools. Some of these environments feature artificial intelligence (AI) or learning analytics systems that support structured problem-solving and computational thinking (CT) activities (Hirashima, 2017; Nordby et al., 2022; Supianto et al., 2016; Zhu et al., 2023). For young learners, such tools can help with differentiated instruction,

scaffold difficult tasks, and open new paths for exploration, as shown in studies where adaptive feedback, structured task progression, and teacher-guided digital environments assisted students in engaging with complex math and CT-related activities (Arroyo et al., 2017; Alrefaei, 2025; Baccaglini-Frank et al., 2020; Passey, 2024).

However, the rapid spread of AI in education also raises urgent questions about equity, accessibility, transparency, data use, and pedagogical appropriateness. These are key considerations. The elementary school environment presents unique challenges, including students' cognitive development, learning identities, and the mediating role of teachers, which strongly influence how technology- and AI-supported tools are used and experienced in practice. Research on CT integration in elementary mathematics shows that teacher coordination, task design, and classroom context are crucial factors in technology-based learning (Baccaglini-Frank et al., 2020; Nordby et al., 2022; Passey, 2024). Despite rapid technological advances, research on how advanced digital learning environments impact teaching and learning in primary education remains scattered. Most studies focus on specific tools or learning outcomes, with limited attention to broader pedagogical, developmental, and ethical issues, as highlighted by reviews and qualitative analyses that point out fragmented CT definitions, tool-centered reporting, and insufficient discussion of classroom mediation and learner development (Anugrahaeni and Haryanto, 2023; Nordby et al., 2022; Passey, 2024). Consequently, educators and policymakers lack a solid evidence base to confidently and responsibly incorporate digital technologies and CT-based learning environments into primary education.

Within the broader context of digital and data-supported learning environments, this review specifically focuses on integrating CT into technology-supported primary mathematics education. Instead of viewing artificial intelligence or analytics systems as separate innovations, the review examines how these digital environments can facilitate CT-related practices, including abstraction, decomposition, sequencing, and variable reasoning, during mathematical problem-solving activities. By synthesizing studies conducted in primary school settings, the review aims to clarify how CT practices are integrated into digital mathematics tasks, how they interact with instructional design and teacher guidance, and how technological tools enhance students' mathematical reasoning processes.

Building on the broader discussion of AI and digital technologies in early education, this review examines the rapidly growing field of technology-supported CT in elementary mathematics. International curricula increasingly recognize CT as an essential skill, emphasizing its potential to improve mathematical reasoning, strategic problem-solving, and student engagement with complex multi-step tasks (Grover & Pea, 2013; Shute et al., 2017; Nordby et al., 2022). In primary grades, especially Grades 3–5, as students develop more abstract mathematical thinking, integrating CT through digital tools, programming activities, robotics, or adaptive learning platforms can create new opportunities to enhance their understanding of mathematics.

However, the current research landscape remains fragmented. Studies vary greatly in how they define CT, apply it to mathematical tasks, use technological environments, and evaluate learning outcomes. Evidence on how CT-enhanced technologies are employed in primary mathematics learning and how they relate to students' problem-solving processes remains inconsistent. Additionally, the roles of teachers, scaffolding strategies, and learning analytics in supporting CT-informed mathematical learning are often under-examined. These gaps make it difficult to establish clear design principles for classroom-ready, developmentally appropriate interventions.

To address these challenges, this systematic review synthesizes studies examining how digital technologies that incorporate CT are used in elementary mathematics learning environments. The review highlights how CT practices, such as decomposition, sequencing, abstraction, and debugging, are integrated into learning activities and how these practices relate to mathematical reasoning processes. Special emphasis is placed on instructional design, teacher mediation, and the technological features that support CT in mathematics education.

2. Research framework

Mathematical problem-solving is widely considered a key goal of primary math education, especially in areas like word problems, arithmetic structures, and early algebraic thinking (Grover and Pea, 2013; Nordby et al., 2022; Shute et al., 2017). These tasks often require students to analyze relationships between quantities, organize multi-step procedures, and adapt strategies when errors occur. Research on CT shows that practices such as decomposition, abstraction, sequencing, and debugging closely align with the thinking processes involved in solving math problems (Brennan & Resnick, 2012; Grover and Pea, 2013; Shute et al., 2017). At the same time, technology-enhanced learning (TEL) tools can make these processes more visible and engaging by supporting programmable tasks, providing immediate feedback, and offering different ways to represent mathematical relationships (Hirashima, 2017; Passey, 2024; Nordby et al., 2022). Despite growing interest in these connections, current research remains fragmented, and few studies have explored how CT practices, math problem-solving strategies, and technology features interact within primary math learning environments.

This review explores how TEL environments can support primary students' mathematical problem-solving by using digital tools that activate or incorporate CT constructs. Current research increasingly views CT not just as a programming skill but as a set of flexible practices, such as decomposition, sequencing, abstraction, conditionals, and debugging, which match the cognitive demands of multi-step problem solving (Grover and Pea, 2013; Shute et al., 2017; Wing, 2006; Wing, 2011). For example, working on word problems involves breaking down the story (decomposition), identifying key quantities (abstraction), organizing steps (sequencing), and modifying strategies when mistakes occur (debugging). TEL environments can support these processes by making them more visible, organized, and adaptable through personalized tasks, immediate feedback, and data-driven insights.

Aligned with identified gaps in the literature, the review focuses on grades 3–5 (ages 9–10), a developmental stage when students shift from basic calculation to more abstract mathematical reasoning. The review examines a wide range of digital interventions, including robotic systems, block-based programming platforms, game-based learning environments, mobile math applications, and learning analytics or intelligent tutoring systems. In each case, the analysis considers how these technologies are designed, implemented in classroom settings, and what mathematical learning outcomes are reported.

A key contribution of this review is the synthesis of instructional design principles and technological tools discussed in studies of CT-integrated mathematics learning in elementary education. This involves examining how existing research incorporates CT into math tasks, how teachers facilitate CT-based learning activities, and how digital analytics are used to monitor learning progress.

The review has two main objectives:

1. To examine empirical data on the outcomes of mathematics problem-solving in technology-based interventions that incorporate CT into teaching word problems, arithmetic, and introductory algebra for elementary school students (grades 3–5).

2. To analyze how CT is implemented in technology-based mathematics instruction in elementary education by reviewing task design, opportunities provided by technology, teacher support, and assessment practices, and to synthesize these models into insights relevant to design.

Unlike previous reviews that mainly focus on CT development across STEM (science, technology, engineering, and mathematics) subjects or programming education, this review specifically emphasizes the integration of CT practices within technology-supported contexts for primary mathematics problem-solving.

2.1. Computational thinking in K–12 research

An increasing number of systematic reviews have explored the role of CT in K–12 education. CT is often defined as the process of formulating problems and expressing solutions in ways that humans or computational agents can perform (Wing, 2006; Wing, 2011). In educational research, CT is often viewed as a set of cognitive practices that support systematic problem solving, including decomposition, abstraction, algorithmic thinking, and debugging (Grover and Pea, 2013; Shute et al., 2017). These practices are increasingly acknowledged as vital components of modern STEM and mathematics education. Early reviews emphasized the emergence of CT within school curricula and its connections to analytical reasoning and problem solving (Grover and Pea, 2013).

Later systematic reviews analyzed instructional methods to promote CT development, including programming environments, robotics platforms, and game-based learning contexts (Hsu et al., 2018). Other reviews have concentrated on evaluating CT skills and aligning CT concepts with learning tasks (Shute et al., 2017).

Despite the growing body of research, few reviews specifically examine how CT is integrated into primary math learning environments supported by digital technologies. Notably, the connections between CT practices, mathematical problem-solving, and the technological features of digital learning settings remain inadequately summarized in current literature.

2.2. Conceptual framework

This review uses a conceptual framework that connects three main analytical areas often discussed in research on CT in education: CT practices, mathematical problem-solving domains, and the technological tools found in digital learning environments. Previous research indicates that CT processes such as decomposition, abstraction, and algorithmic reasoning (Brennan and Resnick, 2012; Grover and Pea, 2013) are closely linked to structured problem-solving methods commonly used in mathematics education (Grover and Pea, 2013; Shute et al., 2017).

Meanwhile, research on technology-enhanced learning highlights how digital environments can support exploratory learning, interactive feedback, and scaffolded reasoning in mathematics education (Hirashima, 2017; Passey, 2024). Building on these concepts, this review examines how CT practices are implemented in technology-

supported primary mathematics learning settings and how these interactions assist mathematical problem-solving.

2.3. Research questions

This review uses research questions to systematically analyze existing studies on integrating CT into primary mathematics education. The analysis examines both empirical evidence about mathematical problem-solving outcomes and how CT constructs are incorporated into tasks, technological settings, and classroom practices. These perspectives together offer a comprehensive understanding of current strategies and provide a basis for designing CT-integrated digital mathematics interventions.

RQ1. What empirical evidence exists regarding the outcomes of mathematical problem-solving in technology-based mathematics interventions that incorporate CT in grades 3–5?

To answer this question, each experimental or pilot study is examined based on its design type, participant characteristics, intervention duration, teacher involvement, and measures of mathematical problem-solving performance. The results presented, and, where possible, the effect sizes, are described and evaluated in light of the quality of the evidence base.

RQ2. How are CT constructs incorporated into primary mathematics tasks and technologies? What types of teacher support are identified in their implementation, and how are CT and problem-solving skills evaluated?

To answer this question, studies are systematically coded to determine how CT skills are defined and implemented, what technological environments and task structures are used, and how learning outcomes and assessment practices are interconnected. These findings are integrated to identify relationships between mathematical topics, CT constructs, and technological opportunities, and to develop a conceptual model that can inform the design of CT-integrated digital mathematics interventions in elementary education.

2.4. Method

The review was conducted following the PRISMA 2020 guidelines for systematic reviews (Page et al., 2021a), ensuring transparency, reproducibility, and rigor in the methodology. It employs a PRISMA-guided systematic review approach with interpretive synthesis, integrating empirical, qualitative, and conceptual studies to analyze how CT is implemented in technology-supported primary mathematics learning environments. A clearly defined protocol and eligibility criteria guided each stage of the review process.

Records were identified through database searches and broad search-engine retrieval procedures, and all search procedures were systematically documented, including search terms, databases, retrieval dates, and duplicate removal, in accordance with the PRISMA-S extension for reporting search strategies (Rethlefsen et al., 2021).

Titles and abstracts were initially screened against the inclusion criteria. Afterward, full texts were reviewed to determine eligibility, with reasons for exclusion clearly documented for each excluded item (Page et al., 2021b). All included studies underwent data extraction and risk-of-bias assessment, followed by both quantitative and qualitative analysis. The heterogeneity of the studies and the strength of the evidence were

systematically evaluated, and each step was illustrated in a PRISMA 2020 flow diagram (Page et al., 2021a).

To ensure methodological transparency, the following subsections outline the inclusion criteria, identification and screening procedures, and the final study selection. Since effect sizes were reported inconsistently across studies and intervention designs varied, the quantitative findings are interpreted descriptively rather than as a formal meta-analysis.

The risk of bias for empirical intervention studies was assessed descriptively using simplified criteria adapted from standard educational intervention review practices, focusing on randomization, group comparability, attrition, and reporting transparency.

2.4.1. Inclusion criteria

To ensure a focused, methodologically consistent selection of studies, this review used clearly defined inclusion criteria. These criteria covered the target population, instructional setting, mathematical content, the presence of CT constructs, and the type and quality of publications. In addition to empirical studies, theoretical and review-based contributions were included to support a comprehensive understanding of how CT can be integrated into technology-supported primary mathematics learning. Accordingly, the review adopts an interpretive systematic approach, aiming to synthesize both empirical evidence and conceptual insights across studies.

1. Population: Studies involving primary students in Grades 3–5 (including Grade 4 explicitly or within the reported range); publications available in English.

2. Intervention/Context: Studies involving technology-mediated mathematics instruction, including but not limited to programming environments, tangible or robotic interfaces, game-based learning systems, mobile applications, and intelligent tutoring or learning analytics platforms.

3. Mathematics Focus: Emphasis on word problems, arithmetic (operations, factors/multiples, fractions/ratios), and introductory algebraic equations; broader mathematical problem-solving contexts were also included when these topics were thoroughly covered.

4. CT: Studies that explicitly mention CT or include implicit CT concepts, such as algorithmic thinking, decomposition, sequencing, loops, conditionals, debugging, variables/data management, or pattern recognition.

5. Study Type: Include empirical studies (experimental, quasi-experimental, pilot, or qualitative/case designs), as well as systematic or narrative reviews and theoretical/methodological papers that meaningfully inform word problems and arithmetic/algebraic expressions (WP/AE) + CT task or technology design.

6. Publication Type: Peer-reviewed journal articles or full conference papers published in reputable academic venues.

The review included peer-reviewed empirical studies as well as theoretical and review-focused publications that offer conceptual or methodological insights relevant to integrating CT within technology-supported primary mathematics learning environments. While empirical studies contributed to analyzing instructional interventions and learning outcomes, theoretical and review papers helped support the conceptual synthesis of CT practices, task design, and technological features. Since the study aimed to synthesize conceptual relationships among CT practices, mathematical

problem-solving domains, and digital learning environments, it employed an interpretive systematic review approach rather than a strictly effectiveness-oriented synthesis.

2.4.2. Identification, screening, and inclusion

Google Scholar was included in the search strategy to identify interdisciplinary publications and new studies that may not yet be indexed in traditional bibliographic databases such as Scopus or Web of Science. This was especially important in the fields of CT and educational technology, where much of the literature appears in conference proceedings and interdisciplinary venues before it is added to major databases.

A total of 438 records were identified across multiple databases and search engines: Google Scholar (418), Scopus (8), ERIC (10), and Web of Science (2). Duplicate records (n=12) were removed before the screening stage in accordance with the PRISMA protocol. No year restrictions were applied during the search; the retrieved records covered the period 2012–2025. After applying the inclusion criteria during screening and eligibility stages, the final sample included studies published between 2016 and 2025, with the earliest eligible study reported by Supianto, Hayashi, and Hirashima (2016).

Google Scholar searches were conducted across full-text fields. To reduce irrelevant results, records focused solely on general STEM topics without a specific focus on mathematics were excluded during title screening. However, this approach may miss some interdisciplinary studies because Google Scholar searches are case-insensitive; uppercase variations like “CT” and “AT” were not included in the search syntax to reduce irrelevant hits. Table 1 presents the complete search strings and database-specific queries, providing a clear, reproducible record of the retrieval process.

Table 1. Search through the lines of each database

Database	Search date	Search line
Scopus	10 Aug 2025	TITLE-ABS-KEY(("word problems" OR "word problem" OR "arithmetic" OR "problem solving task" OR "problem solving tasks" OR "algebraic equations") AND ("computational thinking" OR "algorithmic thinking") AND (primary AND education)) AND NOT TITLE-ABS-KEY("teacher education" OR "pre-service" OR "in-service") AND LIMIT-TO(SUBJAREA, "SOCI") AND LIMIT-TO(LANGUAGE, "English")
ERIC	10 Aug 2025	("word problems" OR "word problem" OR "arithmetic" OR "problem solving task" OR "problem solving tasks" OR "algebraic equations") AND ("computational thinking" OR "algorithmic thinking") AND primary AND education -"teacher education" -"in-service" -"pre-service."
Web of Science	10 Aug 2025	TS = (("word problems" OR "word problem" OR "arithmetic" OR "problem solving task" OR "problem solving tasks" OR "algebraic equations") AND ("computational thinking" OR "algorithmic thinking") AND (primary AND education))
Google Scholar	29–31 Jul 2025	noting that Scholar is case-insensitive and that STEM was excluded in the title only to avoid losing relevant “STEAM/STEM” content in full text; CT and AT acronyms were not used to prevent noise: ("word problems" OR "problem solving tasks" OR "algebraic equations") AND (math OR mathematics) AND ("computational thinking" OR "algorithmic thinking") AND primary AND education AND ("technology" OR "digital") -"teacher education" -"in-service" -"pre-service" -secondary -intitle: "K-12" -intitle: "STEM"

2.4.3. Screening and eligibility

After duplicates were removed, 370 records were excluded during the title and abstract screening phase. The main reasons for exclusion were that the studies:

- targeted non-primary populations (outside grades 3–5),
- addressed different disciplinary areas (e.g., general STEM, artificial intelligence, or psychology),
- were not published in English, or
- did not examine technology-based mathematical problem solving that incorporated CT elements.

Although detailed exclusion counts were not recorded during the title and abstract screening stage, all exclusion decisions were documented during the full-text eligibility assessment (see Figure 1 for the overall screening flow).

A total of 56 articles were identified for potential retrieval. Of these, 11 could not be accessed because they were books or lacked full-text availability, leaving 45 reports for full-text eligibility assessment.

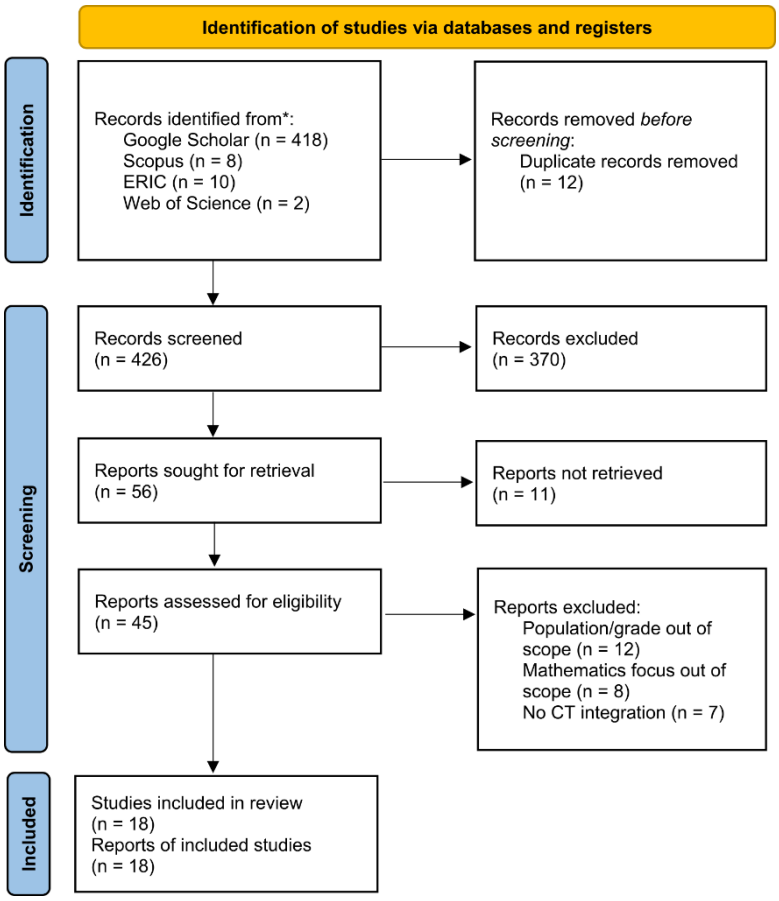


Figure 1. Systematic review flow diagram using the PRISMA 2020 model

2.4.4. Eligibility assessment and operational clarification

During the eligibility assessment, the predefined inclusion criteria were applied and operationalized to ensure consistent classification of studies within the evolving field of CT in mathematics education. Specifically, clarification was needed for studies that reported CT practices implicitly rather than explicitly. Therefore, the review included studies that described related constructs, such as algorithmic thinking, decomposition, sequencing, loops, conditionals, and variable manipulation, as relevant indicators of CT within mathematical tasks.

Similarly, the mathematics criterion was interpreted to include technology-mediated contexts for mathematical problem-solving when word problems, arithmetic structures, or early algebraic reasoning were substantially represented in the learning activities. This approach ensured that studies addressing closely related mathematical problem-solving domains were not excluded solely because of variations in terminology.

In addition to empirical intervention studies, theoretical and methodological publications were included when they significantly contributed to the design of technology-supported tasks or systems that combine CT with mathematical reasoning. These interpretive decisions were documented during the eligibility stage to ensure transparency and consistency in the process of selecting studies.

2.4.5. Exclusions and final inclusion set

Out of the 45 full-text reports evaluated, 27 were excluded, with documented reasons listed below:

- population or grade level out of scope: n = 12
- mathematics focus out of scope: n = 8
- no explicit or implicit CT integration: n = 7

The final review set consisted of 18 studies, each representing a separate report. These studies constitute the complete evidence base analyzed in later stages. The PRISMA 2020 flow diagram in Figure 1 summarizes the counts at each phase (Page et al., 2021a).

The final set of 18 studies formed the basis for the next analysis. Empirical intervention studies guided the investigation of reported learning outcomes (RQ1), while qualitative, review, and theoretical contributions supported the conceptual synthesis of relationships among mathematical topics, CT constructs, and technological affordances (RQ2). The next section presents the results of this analysis.

2.4.6. Coding trustworthiness and analytic rigor

All included studies were coded by an expert reviewer with research experience in CT and technology-enhanced mathematics education. While systematic reviews often use independent double coding, methodological literature shows that single-reviewer coding can be suitable in interpretive or design-focused evidence syntheses, as long as it is supported by transparent procedures and systematic documentation of coding decisions (Petticrew & Roberts, 2006; Grant and Booth, 2009; Booth, Sutton, & Papaioannou, 2016). In this review, the aim was not to perform a statistical meta-analysis but to identify conceptual relationships among CT constructs, mathematical problem-solving areas, and technological tools across different studies. Therefore, the analysis used an

interpretive synthesis approach focused on conceptual alignment rather than quantitative aggregation.

To improve analytical rigor and minimize potential bias from single-reviewer coding, several verification procedures were implemented. First, all coding decisions were documented in a structured analytic matrix linking each study to mathematical topics, CT constructs, and technological characteristics. Second, the coding scheme was developed iteratively through repeated readings of the included studies to ensure internal consistency across categories and to better align the framework with the empirical descriptions reported in the primary studies. Third, the final coding tables were systematically rechecked against the full texts of all included articles to confirm that the classifications accurately reflected the intervention descriptions and learning activities documented in the original publications.

These procedures aimed to ensure transparency, auditability, and conceptual consistency in the synthesis process. Similar methods are commonly used in qualitative or interpretive systematic reviews that synthesize diverse evidence to map conceptual relationships across studies (Grant and Booth, 2009; Booth et al., 2016). However, relying on a single-reviewer coding process is a methodological limitation. Although transparent coding protocols and repeated verification processes were used, future systematic reviews could enhance methodological robustness by including independent double coding and inter-rater agreement procedures.

3. Results

The results are organized into three analytical areas. First, an overview of the characteristics of the included studies is provided, including publication trends and study designs. Second, the analysis examines the CT practices discussed in technology-supported mathematics learning environments. Third, the synthesis highlights the technological affordances reported in the studies that support mathematical reasoning and problem-solving processes.

3.1. Overview of included studies

Eighteen studies met the inclusion criteria and were analyzed in this review. The selected publications encompass a range of research designs and intervention settings exploring the integration of CT within technology-supported primary mathematics learning environments.

The distribution of publication years shows that most studies were published after 2016, with a notable rise between 2020 and 2025 (Figure 2). This trend in publishing reflects the increasing interest in integrating CT into technology-supported primary mathematics education.

The earliest study included in the review was conducted by Supianto, Hayashi, and Hirashima (2016), which examined the integration of CT concepts into technology-supported mathematics learning activities. Early studies, such as those by Arroyo et al. (2017) and Tsarava et al. (2017), further explored instructional designs that combine unplugged activities with digital learning environments.

Subsequent studies published between 2020 and 2021 (e.g., Moschella and Basso, 2020; Qu and Dai, 2021; Zito et al., 2021) reported the use of a broader range of

technology-supported learning environments, including robotics-based systems, tangible programming tools, and digital programming platforms. More recent studies, such as Alrefaei (2025), used quasi-experimental designs with larger participant samples and documented measurable learning outcomes in technology-mediated mathematics learning contexts. Overall, the distribution of studies across publication years shows increasing interest in integrating CT into technology-supported primary mathematics education.

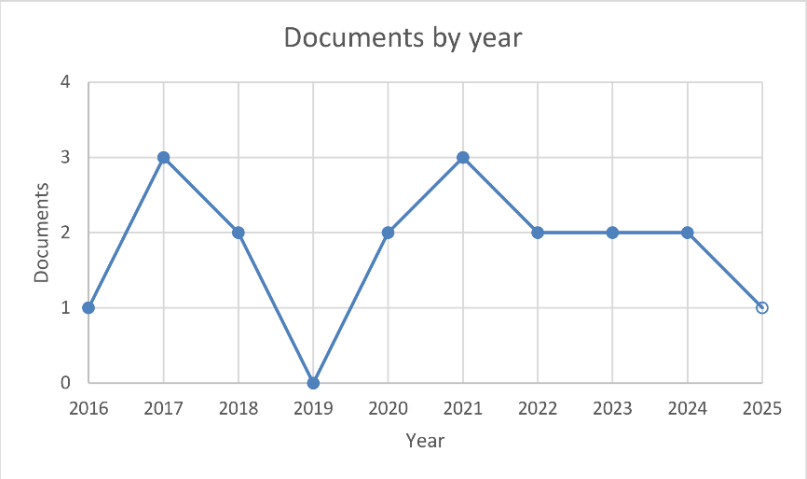


Figure 2. Distribution of included studies by publication year (2016–2025)

The included publications showcase a variety of research designs, such as empirical intervention studies, qualitative investigations, theoretical contributions, and previous reviews. Empirical studies offer insights into instructional interventions and learning outcomes, while theoretical and review articles help synthesize CT practices, task design, and technological tools used in mathematics learning environments. These studies mainly fall into four research categories, illustrating different approaches to technology-enhanced mathematics learning with CT.

1. Experimental, quasi-experimental, and pilot studies (seven of 18; ≈ 39%). These studies assess teaching methods and digital tools and report on the resulting learning outcomes. They often include pre- and post-assessments, comparison or control groups, and statistical analyses such as t-tests or ANOVA. Early classroom projects, also known as pilot studies, are included in this group because they test new designs on a small scale before larger investigations.

2. Qualitative studies (six of 18; ≈ 33%) explore how students learn and how teachers use technology. They utilize interviews, focus groups, classroom observations, or student work to understand learning experiences. Instead of testing a hypothesis, they apply thematic or content analysis to identify patterns and meanings in the data.

3. Review studies (two out of 18; ≈ 11%). These papers summarize previous research rather than collecting new data. Systematic reviews follow the PRISMA process, while narrative reviews provide a structured overview of what is already known and what remains unknown in the field.

4. Theoretical and methodological papers (three out of 18; ≈ 17%) include frameworks, models, or technical designs that link mathematical problem-solving with

CT. They do not contain learner data but provide ideas and systems for future testing and development.

This distribution of study types offers an overview of the current research landscape on integrating technology-supported CT in primary mathematics. Many studies remain exploratory or pilot projects, while large-scale or long-term experimental investigations are still relatively scarce.

While previous reviews have explored CT in primary mathematics or broader mathematical problem-solving, this review offers a design-focused synthesis that links mathematics topics, CT constructs, and digital technologies within primary mathematics learning environments.

Among the included publications, seven empirical intervention studies provide quantitative or quasi-experimental evidence on learning outcomes in grades 3–5 (Table 2). These studies employ a range of technologies, including wearable devices, robotics platforms, programming environments, and game-based systems. The math focus of these interventions generally covers operations, factors, multiples, fractions, and some early algebraic concepts. In these studies, CT is incorporated through both explicit elements (e.g., sequencing, loops, and conditionals) and implicit forms, such as procedural or algorithmic reasoning.

Table 2. Articles of the experimental, quasi-experimental, and pilot studies

Type of research	Amount	References
Trial/Quasi-experimental/Pilot	7	Alrefaei, 2025; Arroyo et al., 2017; Friend et al., 2018; Moschella and Basso, 2020; Qu and Dai, 2021; Tsarava et al., 2017; Zito et al., 2021
Qualitative (interviews/focus groups/case study/observation)	6	Baccaglini-Frank et al., 2020; Cano et al., 2021; Gadanidis et al., 2018; Luo et al., 2022; Passey, 2024; Sulistyani et al., 2024
Reviews (systematic / narrative)	2	Anugrahaeni and Haryanto, 2023; Nordby et al., 2022
Theory/Method (TEL/AI/modeling relevant to WP/AE text understanding & CT design)	3	Hirashima, 2017; Supianto et al., 2016; Zhu et al., 2023

The six qualitative studies explore classroom practices, teacher roles, and student strategies, offering insights into how CT is applied in real settings and guiding task design and learning support. The three theoretical or methodological studies concentrate on models and systems for analyzing word problems, equations, and problem-posing processes, providing conceptual and technical foundations for future task design and assessment.

The seven empirical intervention studies in Table 2 provide the empirical basis for addressing the first research question (RQ1), which focuses on reported outcomes of mathematical problem-solving in technology-supported interventions that include CT.

The remaining publications, which are qualitative, review-focused, and theory-focused, primarily address the second research question (RQ2). These studies provide insights into classroom implementation, instructional design, teacher mediation, and the

technological tools used to incorporate CT practices into primary mathematics learning environments.

The following section presents a detailed analysis of the empirical intervention studies.

3.2. Empirical evidence from technology-supported mathematics interventions

This section summarizes empirical evidence from intervention studies that examined technology-supported CT activities in primary math classrooms. The analysis highlights how digital tools, including CT such as games, programming environments, and robotics, were used in math learning activities and how learning outcomes were reported across studies.

Seven empirical intervention studies, including trials, quasi-experimental designs, and pilot studies, were reviewed. Each study was coded across six dimensions: design and sample, implementation, math focus, problem-solving measures, effect sizes, and moderators and risk of bias, as shown in Table 3. We also recorded the types of tests used to measure outcomes, such as national math tests, local concept tests, or reasoning tasks, and whether these tools demonstrated good reliability and validity.

Table 3. Dimensions of the studies

Dimension	Example contents
Design and Sample	Design type (Trial, Quasi-Experimental, Pilot), N total, grades, special educational needs (SEN), prior attainment
Implementation	Dosage, teacher support, fidelity
Math Focus	Word Problems; Arithmetic; Algebra; Geometry; General Mathematical Problem Solving
Problem-Solving Measure	Test/instrument name, reliability (α/r), validity evidence
Effect Sizes	Hedges' g (or η^2 converted to g)
Moderators and Risk of Bias	Grade, SEN, prior attainment, randomization, attrition, clustering

To compare results across studies, we used Hedges' g, a standardized effect size that indicates the strength of learning gains. A value of about 0.2 is considered small, 0.5 medium, and 0.8 or higher large. We also checked whether studies reported apparent randomization, complete data, and group comparability to assess their risk of bias. Risk of bias was evaluated using simplified criteria adapted from common practices in educational intervention reviews, focusing on randomization, group comparability, attrition, and reporting transparency. Effect sizes were analyzed descriptively to provide an indicative comparison of reported learning outcomes across studies. Due to substantial differences in study designs, outcome measures, and intervention contexts, a formal meta-analysis was not performed.

This analysis summarizes key characteristics of the empirical intervention studies, including their design, implementation features, and reported learning outcomes. Across the seven empirical studies, most reported positive trends in mathematical problem-

solving or related cognitive outcomes across the six dimensions. About 43% of the studies used quasi-experimental or controlled designs, while roughly 57% were pilot classroom trials. The average sample size was approximately 78 students, ranging from 30 to 185 participants, as shown in Table 4.

Despite their small scale, the strength and reliability of these findings varied across study designs. The majority of interventions were multi-session programs, with about 71% lasting ten hours or more and about 86% including some form of teacher support, such as professional development or structured lesson materials. Across the seven empirical studies, teacher support was reported in limited and heterogeneous forms. Most commonly, it included structured lesson materials, brief professional development sessions, and guidance embedded in task design, such as step-by-step instructions or debugging prompts. In several cases, teachers facilitated activities, supported student reflection, and maintained task progression. However, detailed descriptions of teacher practices and their direct impact on learning outcomes were generally limited, making it difficult to draw strong conclusions about the role of teacher support in CT-integrated mathematics interventions. These forms of support helped teachers maintain implementation fidelity and made the activities feasible within regular classroom settings.

Table 4. Distribution of studies across six dimensions

Dimension	**Summary**	**Approx. Percentage / Trend**
Design & Sample	3 quasi-experimental (43%), 4 pilot studies (57%); average sample size about 78 students (range 30–185).	≈ 43% controlled, ≈ 57% pilot
Implementation	Duration ranged from one to 25 hours. Most interventions (5/7) lasted 10 hours or more, and 6/7 reported receiving explicit teacher training or using structured materials.	≈ 71% multi-session, ≈ 86% with teacher support
Math Focus	Arithmetic and word problems dominated (5/7 studies, 71%); geometry in 3/7 (43%); fractions or algebra in 2/7 (29%).	≈ 71% arithmetic/WP focus
Problem-Solving Measure	5/7 studies (71%) used quantitative pre- and post-tests; 2/7 relied mainly on qualitative evidence. Among test-based studies, 4 of 7 reported reliability or validity.	≈ 71% used tests, ≈ 57% reported reliability
Effect Sizes	5/7 studies reported or allowed the computation of Hedges' g. Among them: two large (>0.8), two moderate (0.5–0.8), and one small (<0.5).	≈ 71% reportable, ≈ 29% larger effect sizes
Moderators & Risk of Bias	Grades 3–5 in all; one study focused on SEN (14%); randomization only in one case; 5/7 rated "some concerns", one - "low", one - "high".	≈ 71% some concerns, ≈ 14% low, ≈ 14% high

The mathematical content mainly focused on arithmetic and word problems, appearing in about 71% of the studies, while around 43% addressed geometry, and only about 29% included early algebraic or logical reasoning. This pattern indicates that CT integration has been most commonly explored in number and operations tasks within the analyzed studies. About 71% of the studies employed pre- and post-tests to evaluate learning outcomes, and roughly 57% reported reliability or validity information for their instruments. The remaining projects relied on qualitative observations and interviews, providing useful insights into engagement and motivation but limiting the comparability of test results.

Effect sizes were available or could be calculated in about 70% of the studies. Two studies reported large effect sizes ($g \geq 0.8$), two moderate (0.5–0.8), and one small ($\approx$ 0.4). Some studies reporting larger effect sizes were associated with longer, more structured intervention designs, although the limited number of studies and methodological variability limit direct comparisons, as in the studies by Alrefaei (2025) and Friend et al. (2018). Regarding moderators and bias, all studies involved students in Grades 3–5, with one study ($\approx$ 14%) focusing on students with learning disabilities. Only one study used randomization, and most lacked long-term follow-up. Overall, about 71% were rated as having "some concerns," one as "low risk," and one as "high risk."

Taken together, the reviewed intervention studies suggest that CT-oriented digital activities may support mathematical problem-solving processes in primary education contexts. However, the available evidence remains limited due to small sample sizes, heterogeneous study designs, and the absence of long-term follow-up in most studies. Consequently, the findings should be interpreted as indicative rather than conclusive evidence regarding the potential role of CT-integrated digital learning environments in primary mathematics education.

3.3. Integration of computational thinking constructs in mathematics and technology contexts

This section examines how CT is incorporated into primary mathematics tasks and digital learning environments, the types of teacher support that promote effective integration, and the assessment methods used to measure CT and problem-solving development. While RQ1 looked at reported mathematical problem-solving outcomes in technology-enhanced CT interventions on mathematical learning, RQ2 focuses on the mechanisms—how specific designs, task features, and pedagogical supports help students connect mathematical reasoning with CT practices.

To answer this question, all 18 included studies were systematically coded along three analytical axes:

1. Mathematics content domain (e.g., word problems, arithmetic structures and strategies, geometry, early algebraic reasoning);
2. CT constructs operationalized (e.g., algorithm design, sequencing and loops, conditionals, abstraction and data representation, debugging and iterative refinement);
3. Type of technological environment (e.g., block-based programming, tangible robotics, game-based or app-based learning systems, adaptive or analytics-driven platforms).

This coding revealed not only the distribution of CT constructs across content areas but also how CT is enacted, such as whether CT practices are used to structure problem-solving steps, represent mathematical relationships, automate procedures, test strategies,

or reflect on errors. The three axes together form a Math–CT–Technology Alignment Map that highlights how different combinations of mathematical content, CT practices, and technological tools lead to distinct pedagogical patterns.

Throughout the collection, several common design patterns appeared. Robotics environments typically blend arithmetic and geometry with loops, sequencing, and debugging, allowing students to improve commands and visualize mathematical relationships through repeated movements. Programming tools often help with word problems by involving students in mapping variables, working with conditionals, and designing algorithms, sometimes enhanced by adaptive or AI-driven feedback. Game-based and mobile apps frequently include pattern recognition, decomposition, and step-by-step reasoning in arithmetic or early algebra activities, usually supported by teacher-led scaffolding or prompts.

The alignment map provides a conceptual framework for understanding how CT integration is applied across studies and the educational conditions associated with its implementation. The next subsection explains this alignment model, examining how CT-informed designs work together with teaching facilitation and assessment methods to support multi-step mathematical problem-solving in primary education.

3.3.1. Mathematics, CT, and technology alignment map

In this review, the alignment map highlights recurring relationships among mathematical topics, CT constructs, and the technological environments used to support them. The analysis emphasizes repeated alignments across studies, the technology affordances that enable these practices, the design implications suggested by these patterns, and the gaps that remain in the current research landscape.

The first perspective examines strong, repeated alignments, such as patterns that appear across studies, to show how certain math topics connect with specific CT skills and tools. These alignments illustrate how CT practices, mathematical topics, and technological environments are combined across the reviewed studies, providing a descriptive view of instructional design patterns in technology-supported primary mathematics learning.

The second perspective focuses on the affordances of technology, meaning what each tool makes easy to do (for example, robotics supports loops and debugging). This helps us understand why specific CT–math links occur more often and how they are implemented in classrooms.

The third perspective identifies design takeaways, or practical principles for teachers and designers, such as how to scaffold tasks, sequence activities, and assess learning. The final perspective highlights gaps and opportunities, including missing CT constructs (such as conditionals), weak connections between certain math areas (for example, fractions) and CT, and limited reporting of test reliability. Together, these perspectives provide a descriptive foundation for analyzing recurring instructional patterns across the reviewed studies.

3.3.2. Coding framework and procedures

We examined each of the 18 studies along three main aspects: the mathematics topic, the CT elements, and the technology used. This enabled us to compare studies consistently, despite their different tools and designs.

Mathematics topics were grouped into five categories: WP (word problems), ARITH (basic arithmetic, including operations, factors, multiples, and fractions), ALGEB (introductory algebra or equations), GEOM (geometry), and GEN (general mathematics problem-solving). These categories emphasized word problems and arithmetic while also encompassing a broader range of problem-solving tasks commonly used in elementary schools.

CT elements were categorized as either explicit (EXPL) when authors used the term "computational thinking" or mentioned specific CT skills, or implicit (IMPL) when such skills were present but not labeled as CT. If no CT elements were found, they were marked as NONE. Individual CT skills, such as algorithmic thinking, sequencing, loops, conditionals, abstraction, decomposition, pattern recognition, variables, and debugging, were grouped into five broader categories: ALG-DES (algorithm design), CTRL-FLOW (control flow, including loops and conditionals), ABSTR-DEC (abstraction and decomposition), DATA-VAR (variables and data), and DBG (debugging and testing). Each study could include multiple CT codes, but we identified a single main or "primary" code representing its primary instructional focus.

Technologies were also categorized based on how they supported learning: PROG (programming, such as Scratch or Blockly), PHYS-COMP (physical computing, including robotics or tangible interfaces), GAME/APP (learning games and applications), ANALYTICS/AI (tools that use analytics, intelligent tutoring, or natural language processing), and ICT (general computer use without a specific tool). When a study involved more than one tool, we noted the primary one and classified the others as secondary. For example, a robotics activity utilizing Scratch was categorized as PHYS-COMP (primary) and PROG (secondary).

Each study was assigned up to three mathematics codes and up to four CT codes to reflect that technology-mediated mathematics tasks often involve multiple mathematical topics and CT practices simultaneously. This multi-coding approach captures instructional complexity without oversimplification, while the set limits ensure clarity and comparability across studies. Coding decisions were based on descriptions in the Methods, Tasks, and Appendices sections of each study. When information was unclear, we made conservative choices and added notes with page references to clarify the context.

Finally, we examined how mathematics, CT, and technology are interconnected. For example, word problems often relate to DATA-VAR and ABSTR-DEC (working with variables and deconstructing problems); arithmetic connects to CTRL-FLOW and DBG (loops and testing); algebra is associated with DATA-VAR and CTRL-FLOW (variables and logic); and geometry links to ALG-DES and ABSTR-DEC (step-by-step planning and abstraction). We also identified common links between technology and CT skills, such as robotics supporting loops and debugging, programming involving variables and control flow, and AI tools emphasizing data and abstraction. These patterns helped us understand how CT concepts are integrated into math learning designs.

3.3.3. Strong, repeated alignments between Math topics and CT constructs

A key finding of this review is the emergence of consistent and recurring alignments between specific mathematical topics and CT constructs. These patterns reflect common ways in which CT is embedded within mathematics instruction, particularly when supported by digital tools. Table 5 presents the consistent patterns identified across the 18 studies reviewed. In this review, strong, repeated patterns are understood as design

frameworks observed in multiple studies that systematically link particular mathematical topics with specific CT practices. For example, arithmetic tasks frequently involve loops and debugging; word problems require variable mapping and decomposition; geometry emphasizes step-by-step algorithmic planning; and algebra relies on variable reasoning combined with conditional logic. By identifying these recurring pairings, the table illustrates how CT is integrated into mathematical reasoning. It highlights the main pathways through which problem-solving processes are structured and supported by digital tools, providing insight into how CT practices are operationalized in primary classroom contexts.

Table 5. Strong, repeated alignments

Math Code	CT Constructs (codes)	Amount (n)	References
ARITH (factors & multiples)	CTRL-FLOW (loops/iteration) + ABSTR-DEC (pattern recognition) + DBG (debugging/testing)	5	Arroyo et al., 2017; Cano et al., 2021; Moschella and Basso, 2020; Qu and Dai, 2021; Tsarava et al., 2017
WP (word problems)	DATA-VAR (variable mapping) + ABSTR-DEC (schema spotting, decomposition)	7	Alrefaei, 2025; Anugrahaeni and Haryanto, 2023; Friend et al., 2018; Hirashima, 2017; Sulistyani et al., 2024; Supianto et al., 2016; Zhu et al., 2023
GEOM (geometry)	ABSTR-DEC (shape/constraint abstraction) + ALG-DES (procedural drawing, step planning)	5	Alrefaei, 2025; Arroyo et al., 2017; Baccaglini-Frank et al., 2020; Friend et al., 2018; Gadanidis et al., 2018
ALGEB (introductory equations)	DATA-VAR (variables) ± CTRL-FLOW (conditionals/case logic)	4	Baccaglini-Frank et al., 2020; Gadanidis et al., 2018; Tsarava et al., 2017; Zito et al., 2021;

Across the 18 studies, the strongest connections are found between word problems (WP) and CT concepts related to data/variables (DATA-VAR) and abstraction/decomposition (ABSTR-DEC), as seen in 7 studies (≈ 39%), as shown in Table 5. These tasks often require learners to identify given and unknown quantities, represent them as variables, and break down the problem structure, skills that directly link CT reasoning with mathematical modeling.

Arithmetic (ARITH) topics appear in 5 studies (about 28%) and are consistently linked to control flow (CTRL-FLOW) and debugging (DBG) through iterative or loop-based activities in robotics and programming. Students plan sequences, test outputs, and refine their logic, thereby mirroring algorithmic thinking.

Geometry (GEOM) appears in 5 studies (≈ 28%), aligning with abstraction and algorithm design (ALG-DES), especially in drawing or movement tasks where learners develop procedural plans for geometric constructions.

Finally, algebra (ALGEB) appears in four studies (≈ 22%), combining variable reasoning (DATA-VAR) with occasional conditional logic (CTRL-FLOW). These patterns indicate that CT integration is most common in word-problem and arithmetic tasks across the analyzed studies.

The patterns described in this section are further illustrated in Figure 3, which shows the frequency of co-occurrence between mathematical topics and CT constructs.

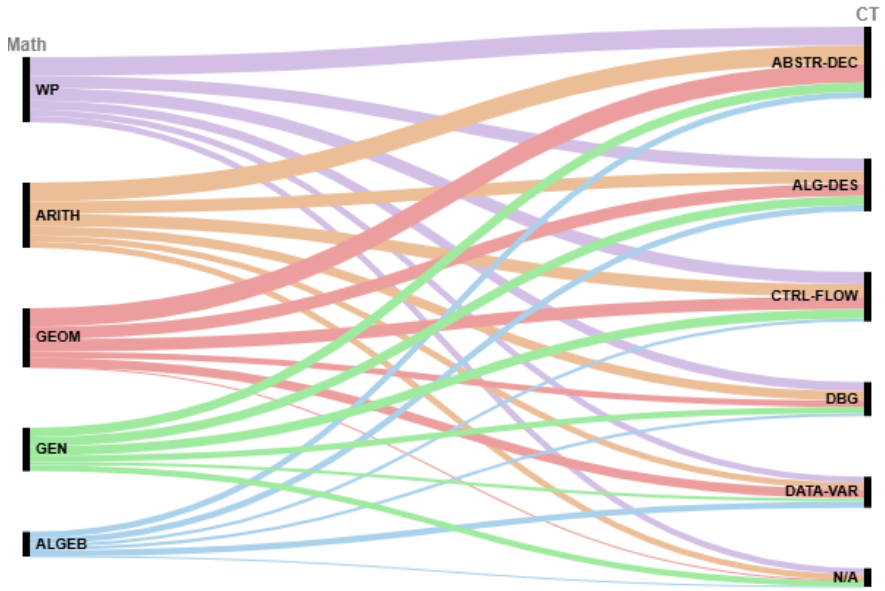


Figure 3. Repeated alignments between mathematics topics and CT constructs

The diagram shows how math topics and CT constructs are connected in the reviewed studies. Each line indicates the co-occurrence of a mathematics topic and a CT construct in the reviewed studies, with line thickness representing how frequently they appear together across studies. Although line thickness varies to indicate relative frequency, differences may appear visually subtle due to the limited number of studies and overlapping connections across categories.

The most common links are between word problems (WP) and arithmetic (ARITH), as well as CT constructs such as abstraction and decomposition (ABSTR-DEC), algorithm design (ALG-DES), and control flow (CTRL-FLOW). These constructs help students break problems into steps, plan solutions, and manage sequences or conditions in digital tasks. Geometry (GEOM) also connects to these CT skills, mainly through the use of data and variables (DATA-VAR) in modeling or programming activities. Algebra (ALGEB) and general (GEN) areas show fewer links, suggesting that CT ideas are less frequently applied in these areas. The codes indicate that problem-solving in WP and ARITH most strongly involves key CT practices.

3.3.4. Strong, repeated alignments between mathematical topics, CT constructs, and technologies

In this review, strong alignments indicate recurring design patterns across multiple studies that systematically link specific mathematics topics to particular CT practices, often with technology support. These alignments reflect stable, pedagogically meaningful connections that can inform the design of integrated learning activities. For example, arithmetic topics such as factors and multiples are frequently associated with control flow concepts (e.g., loops and iteration) and debugging, particularly in physically interactive (PHYS-COMP) tasks involving robots or tangible tools (Table 6).

Table 6. Technology affordances

Technology Code	Typical Math Focus (codes)	Main CT Constructs (codes)	Amount (n)	References
PHYS-COMP	ARITH, GEOM	ALG-DES → CTRL-FLOW → DBG cycles (sequencing, loops, debugging)	5	Arroyo et al., 2017; Baccaglini-Frank et al., 2020; Moschella and Basso, 2020; Qu and Dai, 2021; Zito et al., 2021
PROG	ALGEB, ARITH	DATA-VAR + CTRL-FLOW (variables, loops, conditionals)	5	Friend et al., 2018; Gadanidis et al., 2018; Luo et al., 2022; Nordby et al., 2022; Tsarava et al., 2017;
GAME/APP	ARITH, GEOM	ALG-DES + CTRL-FLOW (events, iteration)	5	Alrefaei, 2025; Arroyo et al., 2017; Cano et al., 2021; Tsarava et al., 2017; Zito et al., 2021
ANALYTICS /AI	WP	DATA-VAR + ABSTR-DEC (variable mapping, schema decomposition)	3	Hirashima, 2017; Supianto et al., 2016; Zhu et al., 2023
ICT	ARITH, WP	DATA-VAR (representation, pattern recognition, implicit CT)	2	Alrefaei, 2025; Anugrahaeni and Haryanto, 2023

Word problems (WP) typically involve data variability (mapping knowns and unknowns) and abstract decomposition (schema breakdown), often supported by analytics/AI environments that help learners model and test problem structures. GEOM (geometry) links to ALG-DES (step-by-step planning) and ABSTR-DEC (algorithmic drawing and constraint abstraction) through programming or physical computing tools. Meanwhile, ALGEB (introduction to equations) connects to DATA-VAR and, when present, to CTRL-FLOW/COND (case logic) within PROG environments, such as Scratch or Blockly.

Across the 18 studies, three technology types—PHYS-COMP, PROG, and GAME/APP—appeared in five studies (about 28%), indicating a balanced use of robotics, programming, and game-based tools in primary mathematics. The ANALYTICS/AI group was featured in three studies (about 17%), mainly for word-problem analysis and feedback, while ICT alone was used in two studies (about 11%), primarily for general arithmetic tasks. These proportions suggest that most digital interventions use technologies that support iterative reasoning, control flow, and variable manipulation—skills often linked to mathematical problem-solving in the reviewed studies.

The Math–CT–Technology Alignment Map (Figure 4) summarizes how CT is integrated into primary mathematics across the 18 studies. The size of each annular sector reflects the relative frequency of each category across the reviewed studies. The three layers represent the reviewed categories: mathematical topics (WP, ARITH, GEOM, ALGEB, GEN), CT constructs (ABSTR-DEC, ALG-DES, CTRL-FLOW, DATA-VAR, DBG, N/A), and technology types (PROG, GAME/APP, PHYS-COMP, ICT, ANALYTICS/AI).

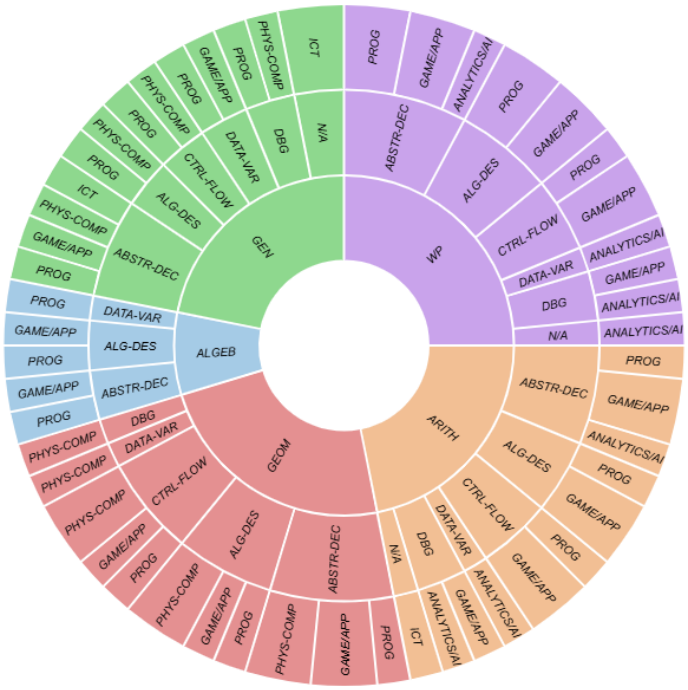


Figure 4. Math–CT–Technology alignment map

Across the studies, word problems (WP) were frequently associated with data/variables (DATA-VAR) and abstraction/decomposition (ABSTR-DEC), though the relative frequency of these relationships varied. These associations were primarily supported by programming and analytics/AI tools. Arithmetic (ARITH) was frequently connected to control flow (CTRL-FLOW) and debugging (DBG) in physical computing and game/app development environments. Geometry (GEOM) was associated with

algorithm design (ALG-DES) and abstraction/decomposition, commonly within programming and robotics contexts. Algebra (ALGEB) combined data/variables (DATA-VAR) and abstraction/decomposition constructs, usually using programming platforms. General (GEN) activities involved various CT constructs across programming, physical computing, and ICT settings.

The map consolidates empirical evidence from all 18 studies, showing which CT constructs and technologies co-occur with each mathematical topic in technology-mediated primary mathematics tasks. Although all categories are connected across the map, differences in sector size reflect relative frequency rather than exclusive relationships and may appear subtle due to overlapping distributions across studies.

3.4. Design takeaways

Technology affordances refer to the actions a tool naturally supports for learners and teachers. This review emphasizes the CT moves that a technology enables or makes likely (e.g., sequencing, loops, or debugging), as well as the mathematical concepts it helps highlight (e.g., patterns in factorizations or variable mappings in word problems).

Two recurring design themes appear across the reviewed studies (Figure 5). The first theme, ARITH → ALG-DES → CTRL-FLOW → DBG → PHYS-COMP/PROG, shows up in about 39% of cases. Here, students plan sequences of steps, use loops and events, run and debug their programs, and describe the patterns they find. Typical activities include robot or tangible missions involving multiples and divisibility, or Scratch loops that generate sets of factors (Arroyo et al., 2017; Cano et al., 2021; Luo et al., 2022; Moschella and Basso, 2020; Qu and Dai, 2021; Tsarava et al., 2017; Zito et al., 2021).

A second common strand, WP → DATA-VAR + ABSTR-DEC → ANALYTICS/AI/PROG, accounts for about 39% of the studies. In these tasks, learners map known and unknown variables, break down schemas, compare cases, and sometimes write brief programs to find solutions. Many tasks follow problem-posing or "highlight–extract–relate" routines (Alrefaei, 2025; Anugrahaeni and Haryanto, 2023; Friend et al., 2018; Hirashima, 2017; Sulistyani et al., 2024; Supianto et al., 2016; Zhu et al., 2023).

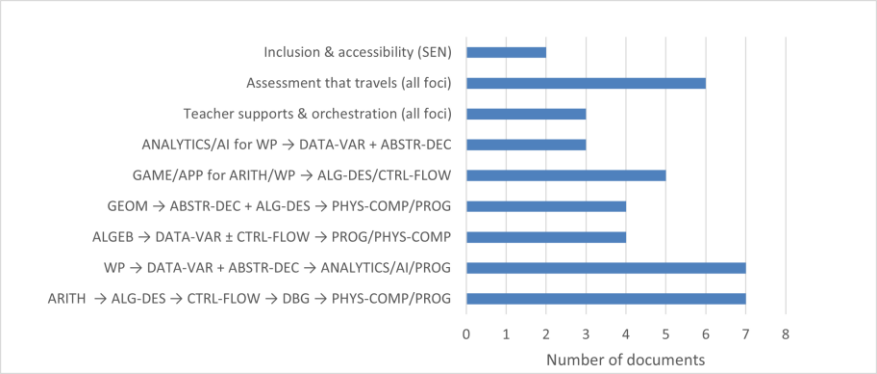


Figure 5. Math–CT–Tech learning design strands (grades 3–5)

Two additional strands are observed in fewer studies. The ALGEB → DATA-VAR ± CTRL-FLOW → PROG/PHYS-COMP strand, found in about 22% of studies, connects variable binding and conditional logic within block-based environments (Baccaglini-Frank et al., 2020; Gadanidis et al., 2018; Tsarava et al., 2017; Zito et al., 2021).

The GEOM → ABSTR-DEC + ALG-DES → PHYS-COMP/PROG strand, also at 22%, emphasizes the abstraction of shape constraints and procedural drawing, such as turtle or robot-based tasks with step-by-step refinement (Arroyo et al., 2017; Baccaglini-Frank et al., 2020; Friend et al., 2018; Gadanidis et al., 2018).

Several cross-cutting patterns are also seen across the reviewed studies. The GAME/APP for ARITH/WP → ALG-DES/CTRL-FLOW combination appears in about 28% of studies, utilizing events, loops, and quick feedback within game-like learning sequences (Alrefaei, 2025; Arroyo et al., 2017; Cano et al., 2021; Tsarava et al., 2017; Zito et al., 2021).

The ANALYTICS/AI for WP → DATA-VAR + ABSTR-DEC pattern (≈ 17%) uses intelligent feedback and pattern dashboards to support schema recognition and variable mapping (Hirashima, 2017; Supianto et al., 2016; Zhu et al., 2023).

Three horizontal layers frame the overall design picture (Figure 5). Teacher supports and orchestration were reported in about 17% of the studies, including worked examples, debug checklists, short PD sessions, and structured "launch–explore–debug–share" cycles (Baccaglini-Frank et al., 2020; Nordby et al., 2022; Passey, 2024).

Transfer-oriented assessment approaches are present in about 33% of studies, combining accuracy, transfer, and time/error metrics, with several reporting reliability (α/ω) and delayed checks (Anugrahaeni and Haryanto, 2023; Arroyo et al., 2017; Cano et al., 2021; Nordby et al., 2022; Tsarava et al., 2017; Zito et al., 2021).

Inclusion and accessibility (SEN) are addressed in about 11% of tasks, where adaptations are made through accessible interfaces, tailored feedback, or subgroup analysis (Alrefaei, 2025; Cano et al., 2021).

Taken together, these elements and supporting conditions demonstrate how technology, CT practices, and mathematical tasks are integrated in the reviewed studies. Common patterns, such as loops and debugging in arithmetic tasks, variable mapping in word problems, and algorithmic planning in geometry and algebra, provide an overview of how CT-informed learning designs are used in primary mathematics settings.

4. Gaps and opportunities

The analysis of the 18 reviewed studies highlights several common gaps and potential areas for further development in integrating CT into primary mathematics learning settings.

1. CT constructs are inconsistently defined. Many studies mention CT generally without clarifying specific learner-level components. Conditionals and explicit decomposition are seldom included in task design (Zito et al., 2021). Only a few studies, like those by Luo et al. (2022), Tsarava et al. (2017), and Arroyo et al. (2017), have made these constructs more visible.

Opportunity: Make the steps of abstraction and decomposition clear in word-problem tasks, such as linking givens to variables and relations, and include control flow or conditionals in comparison or multi-case problems.

2. Topic coverage is limited. Fractions and ratios appear in arithmetic-focused studies (Alrefaei, 2025; Luo et al., 2022; Zito et al., 2021) but show relatively limited explicit connections with CT constructs.

Opportunity: Enhance CT integration in these topics by using control flow (iteration and accumulation), abstraction (invariants), and data-variable mapping to support proportional reasoning.

3. Technology affordances are underutilized. Physical computing effectively supports algorithm design, control flow, and debugging, but rarely extends to conditional logic or data logging. Programming tasks often focus on loops and variables without connecting to algebraic reasoning. Analytics and AI tools are often linked to data-variable mapping in word problems but are seldom used for formative feedback (Arroyo et al., 2017; Baccaglini-Frank et al., 2020; Friend et al., 2018; Hirashima, 2017; Moschella and Basso, 2020; Supianto et al., 2016; Zhu et al., 2023).

Opportunity: In physical computing, add sensor-condition and debugging tasks (Arroyo et al., 2017; Qu and Dai, 2021); in programming, connect data-variable reasoning with algebraic testing (Gadanidis et al., 2018; Tsarava et al., 2017); and in analytics/AI, transform word-problem parsing into real-time scaffolding and feedback (Hirashima, 2017; Supianto et al., 2016).

4. Assessment reporting practices differ widely across studies. Many studies use researcher-developed tools with little evidence of reliability or validity and seldom include delayed assessments (Alrefaei, 2025; Anugrahaeni and Haryanto, 2023; Arroyo et al., 2017; Cano et al., 2021; Moschella and Basso, 2020; Nordby et al., 2022; Tsarava et al., 2017; Zito et al., 2021).

Opportunity: Future research could benefit from using shared, validated problem-solving tools, clearer reporting of reliability metrics, and the inclusion of delayed measures that capture longer-term learning effects.

5. Study designs and reporting of implementation features vary among the reviewed studies. Most are small-scale pilots with limited randomization, small sample sizes, and inconsistent reporting of intervention duration, teacher training, and fidelity (Nordby et al., 2022).

Opportunity: Develop more robust quasi-experimental or cluster-randomized designs with clear comparison groups, pre-registration, standardized dosage reporting, and documented implementation fidelity.

6. Consideration of diverse learner populations and classroom settings remains limited. Only a few studies address learners with hearing impairments or learning disabilities (Alrefaei, 2025; Cano et al., 2021), but broader inclusion of diverse learners and classroom environments is uncommon.

Opportunity: Expand designs to include more diverse student groups, report subgroup effects and accessibility adaptations, and document classroom orchestration practices (Baccaglini-Frank et al., 2020; Passey, 2024).

The reviewed studies indicate that physical computing environments are often connected to control flow and debugging practices, programming tools with variable reasoning, and analytics or AI-supported systems involving abstraction and decomposition in word-problem contexts. Future research could benefit from combining these technological capabilities with clearer CT construct definitions, more rigorous

study designs, more comprehensive assessment methods, and greater inclusion of diverse learning environments to build a more complete evidence base for CT-integrated mathematics learning in grades 3–5.

5. Conclusions

This systematic review examined 18 studies on how technology-enhanced learning environments include CT in primary mathematics instruction for students in grades 3–5. Positive trends were observed in students' problem-solving and reasoning skills across the intervention studies, although the strength of the evidence varies due to differences in study designs, sample sizes, and assessment methods.

Analysis of all studies revealed clear patterns linking mathematical domains, CT constructs, and technology types. Word problems were often associated with decomposition, abstraction, and variable use; arithmetic tasks with sequencing, control flow, and debugging; geometry with algorithmic design; and early algebra with conditional reasoning. Programming, robotics, and game-based environments were the most frequently used technologies to implement these constructs. CT practices appear to support processes by which pupils externalize, organize, and refine mathematical reasoning within digital environments.

Regarding RQ1, the findings suggest that CT-integrated interventions may support mathematical problem-solving, though evidence remains limited.

Regarding RQ2, the results indicate that CT is most often implemented through programming, robotics, and game-based environments, with limited evidence on teacher support and assessment practices..

Seven intervention studies were reviewed for the first research question, focusing on reported mathematical problem-solving outcomes across factors such as study design, teaching duration, mathematical emphasis, assessment tools, and reliability. Most interventions included multiple sessions with teacher support and used pre- and posttests to evaluate learning. The majority of studies showed improvements in students' mathematical problem-solving or related cognitive skills, though the strength and consistency of these results varied by study design. Some studies with larger effect sizes were associated with longer, more structured intervention programs.

All studies were classified by mathematical topic, CT construct, and technology type for the second research question. Clear patterns emerged: word problems were often linked to abstraction, decomposition, and the use of data or variables; arithmetic was connected to control flow and debugging through robotics and programming; geometry was linked to algorithm design and abstraction; and algebra involved variable reasoning and conditional logic. Programming, physical computing, and game-based tools were the most common technologies supporting these connections.

The reviewed literature indicates that CT practices are often integrated into digital mathematics learning environments through programming, robotics, and game-based tools. The study shows that these environments can support students' engagement in multi-step mathematical problem-solving by structuring reasoning, illustrating relationships among quantities, and enabling iterative testing of solution strategies. However, current evidence is limited by small sample sizes, inconsistent study designs, and a scarcity of long-term assessments. More research using rigorous experimental and longitudinal methods is needed to strengthen the empirical evidence supporting CT-integrated mathematics learning in primary education.

6. Limitations and future work

The evidence base remains limited in several ways. Sample sizes in intervention studies were small, study designs varied, and assessment methods were inconsistent across mathematics and CT outcomes. Many studies lacked long-term follow-up, detailed reporting on implementation fidelity, or analysis of different effects for diverse learners.

The second limitation relates to the coding and screening procedures. A single researcher handled the study selection and coding processes. Although the coding framework was applied systematically and verified during analysis, the lack of independent double screening may affect the reproducibility of classification decisions.

The third limitation concerns the search strategy. Although multiple academic databases were used, most records came from Google Scholar. While this approach helped include interdisciplinary and emerging studies, it might also have skewed the balance of the retrieved literature and reduced the reproducibility of the search results.

The fourth limitation is the inclusion of various types of publications, such as theoretical papers, review articles, and empirical studies. While this broadens the overall understanding of CT integration in technology-supported math learning environments, it also weakens conclusions about intervention effectiveness..

Future research should employ more rigorous experimental and long-term study designs, develop validated tools to assess computational and mathematical reasoning, and examine how teacher support and learning analytics influence learning. Expanding research to include a broader range of mathematical topics and more diverse school settings will also be essential to developing scalable, evidence-based CT-integrated mathematics interventions.

Acknowledgments

This study (No P-EDU-23-13) is co-funded by the European Union through the project “Breakthrough in Educational Research” No 10-044-P-0001, under the April 1, 2025, Agreement with the Research Council of Lithuania and the April 15, 2025, Joint Activity Agreement with Vilnius University. Additional funding is provided by the Research Council of Finland (EDUCA Flagship #358924, #358947) and the Ministry of Education and Culture (Doctoral School Pilot #VN/3137/2024-OKM-4).

References

Alrefaei, M. M. (2025). Boosting math skills: The impact of game-based instruction on problem-solving in students with learning disabilities. *Amazonia Investiga*, 14(86), 41–50. https://doi.org/10.34069/AI/2025.86.02.4

Anugrahaeni, I., Haryanto, H. (2023). Review of Mathematica’s problem-solving implementation in elementary school. *Proceedings Series on Social Sciences & Humanities*, 12, 207–217. http://doi.org/10.30595/pssh.v12i.798

Arroyo, I., Micciollo, M., Casano, J., Ottmar, E., Hulse, T., Rodrigo, M. M. (2017). Wearable learning: Multiplayer embodied games for math. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (pp. 205–216). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3116595.3116637

Baccaglini-Frank, A. E., Santi, G., Del Zozzo, A., Frank, E. (2020). Teachers' perspectives on the intertwining of tangible and digital modes of activity with a drawing robot for geometry. *Education Sciences*, 10(12), 387. https://doi.org/10.3390/educsci10120387

Booth, A., Sutton, A., Papaioannou, D. (2016). *Systematic Approaches to a Successful Literature Review* (M. Steele, Ed., 2nd ed.). SAGE Publications Inc.

Brennan, K., Resnick, M. (2012). New frameworks for studying and assessing computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (pp. 1–25).

Cano, S., Naranjo, J. S., Henao, C., Rusu, C., Albiol-Pérez, S. (2021). Serious game as support for the development of computational thinking for children with hearing impairment. *Applied Sciences*, 11(1), 115. https://doi.org/10.3390/app11010115

Friend, M., Matthews, M., Winter, V., Love, B., Moisset, D., Goodwin, I. (2018). Bricklayer: Elementary students learn math through programming and art. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, (pp. 628–633). https://doi.org/10.1145/3159450.3159515

Gadanidis, G., Clements, E., Yiu, C. (2018). Group theory, computational thinking, and young mathematicians. *Mathematical Thinking and Learning*, 20(1), 32–53. https://doi.org/10.1080/10986065.2018.1403542

Grant M.J., Booth A. (2009). A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information and Libraries Journal*, 26(2), 91–108. https://doi.org/10.1111/j.1471-1842.2009.00848.x

Grover, S., Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. https://doi.org/10.3102/0013189X12463051

Hirashima, T. (2017). Computer-based intelligent support for moderately Ill-structured problems. In *Proceedings of 2017 3rd International Conference on Science in Information Technology (ICSITech)* (pp. 1–6). United States (USA). https://doi.org/10.1109/ICSITech.2017.8257076

Hsu, T.-C., Chang, S.-C., Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Luo, F., Israel, M., Gane, B. (2022). Elementary computational thinking instruction and assessment: A learning trajectory perspective. *ACM Transactions on Computing Education*. https://doi.org/10.1145/3494579

Moschella, M., Basso, D. (2020). Computational thinking, spatial and logical skills: An investigation at primary school. *Ricerche di Pedagogia e Didattica. Journal of Theories and Research in Education*, 15, 69–89. https://doi.org/10.6092/ISSN.1970-2221/11583

Nordby, S. K., Bjerke, A. H., Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. *Digital Experiences in Mathematics Education*, 8(1), 27–49. https://doi.org/10.1007/s40751-022-00102-5

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., … Moher, D. (2021a). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. https://doi.org/10.1136/bmj.n71

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., … Moher, D. (2021b). PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews. *BMJ*, 372, n160. https://doi.org/10.1136/bmj.n160

Passey, D. (2024). Computational Thinking and Computing linked to Problem-solving across the Primary Curriculum: Groggan Primary School, Northern Ireland Case Study. *International Federation for Information Processing (IFIP)*, (pp. 1–21). https://doi.org/10.52545/2-11

Petticrew, M., Roberts, H. (2006). *Systematic reviews in the social sciences: A practical guide*. Blackwell Publishing. https://doi.org/10.1002/9780470754887

Qu, J., Dai, L. (2021). Promoting students' problem-solving competencies: Peer interactions in a robotics learning environment. In *ICETT 2021: 2021 7th International Conference on Education and Training Technologies* (pp. 84–89). https://doi.org/10.1145/3463531.3463544

Rethlefsen, M. L., Kirtley, S., Waffenschmidt, S., Ayala, A. P., Moher, D., Page, M. J., Koffel, J. B. (2021). PRISMA-S: An extension for reporting literature searches in systematic reviews. *Systematic Reviews*, 10(1), 39. https://doi.org/10.1186/s13643-020-01542-z

Shute, V. J., Sun, C., Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Sulistyani, N., Prasojo, L. D., Purnomo, Y. W., Ardhana, S. S., Fitriya, Y., Putri, Y. D. (2024). Epistemological didactical situation of learning factors and multiples in elementary school students. *Mathematics Teaching Research Journal*, 16(5), 66–91.

Supianto, A. A., Hayashi, Y., Hirashima, T. (2016). Visualizations of problem-posing activity sequences toward modeling the thinking process. *Research and Practice in Technology Enhanced Learning*, 11, 14. https://doi.org/10.1186/s41039-016-0042-4

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., Ninaus, M. (2017). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. In *Proceedings of the 11th European Conference on Games Based Learning (ECGBL 2017)* (pp. 687–695). Academic Conferences International.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, Carnegie Mellon University.

Zhu, P., Lv, P., Shi, J., Jiang, X., Zou, W., Ma, Y. (2023). Design and implementation of text understanding system based on semantic tagging instances. In *Proceedings of the 2023 4th International Conference on Artificial Intelligence in Electronics Engineering* (pp. 108–116). ACM. https://doi.org/10.1145/3586185.3586190

Zito, L., Cross, J. L., Brewer, B., Speer, S., Tasota, M., Hamner, E., Johnson, M., Lauwers, T., Nourbakhsh, I. (2021). Leveraging tangible interfaces in primary school math: Pilot testing of the Owlet math program. *International Journal of Child-Computer Interaction*, 27, 100222. https://doi.org/10.1016/j.ijcci.2020.100222

Received April 1, 2026, revised April 17, 2026, accepted April 17, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, 356-375
https://doi.org/10.22364/bjmc.2026.14.2.05

Hybrid Physics-Informed Temporal Attention Model with External Physical Constraints for PV Power Forecasting

Dezdemona GJYLAPI, Alketa HYSO, Astrit DENAJ

University "Ismail Qemali" of Vlora, Vlora, Albania

dezdemona.gjylapi@univlora.edu.al, alketa.hyso@univlora.edu.al, astrit.denaj@univlora.edu.al

ORCID 0009-0005-5186-3636, ORCID 0009-0000-6256-9831, ORCID 0009-0000-7594-5058

Abstract: This paper introduces the Hybrid Physics-Informed Temporal Attention (H-PITA) model, a novel framework for accurate, physically consistent daily photovoltaic (PV) energy (kWh) forecasting. By integrating a temporal-attention encoder with external physical priors via gated fusion and a lightweight residual pathway, H-PITA balances data-driven learning with domain-specific constraints. The model incorporates physical knowledge through symmetric and asymmetric penalties, non-negativity constraints, and dynamic loss weighting, as well as an optional inference-time safety cap. Evaluated on a real-world PV system in Southeast Europe using a longitudinal dataset (2014–2025), H-PITA demonstrates superior performance over baselines including LSTM, XGBoost, persistence, climatology, and linear/ridge models, achieving a test R^2 of 0.78. Results indicate that data filtering and physical grounding significantly enhance generalization, reducing MAPE and RMSE. Overall, H-PITA provides a robust, modular, and transferable solution for risk-aware PV forecasting, making it highly suitable for operational grid integration and solar asset management.

Keywords: Photovoltaic energy forecasting; Physics-informed neural networks; Temporal attention; Hybrid deep learning; Physical constraints; Renewable energy prediction; PV systems;

1. Introduction

Modern power systems increasingly rely on solar photovoltaic (PV) generation, making accurate day-ahead PV power forecasting (here formulated as daily energy in kWh) crucial for grid stability and efficient energy management. Integrating large shares of intermittent PV energy into the electricity grid is challenging due to the variability of solar irradiance (e.g., rapid weather changes causing supply–demand imbalances and voltage fluctuations) (Di Leo et al., 2025).

Day-ahead forecasts (covering 24–48 hours ahead) enable grid operators and energy markets to better plan dispatch, schedule reserves, and minimize balancing costs. In fact, precise PV output predictions are considered one of the most cost-effective tools for accommodating higher renewable penetration without compromising reliability. By

anticipating fluctuations in solar generation, system operators can optimize resource allocation, maintain network stability, and reduce operational costs. Consequently, day-ahead PV forecasting has become indispensable for modern smart grids and energy markets striving to integrate renewables at scale (Di Leo et al., 2025; Barhmi et al., 2024).

However, forecasting PV power output remains a complex task due to several limitations of traditional methods and pure machine learning approaches. Statistical and time-series models often struggle to capture the nonlinear relationships between meteorological variables (e.g., irradiance, cloud cover, temperature) and PV output (Di Leo et al., 2025; Vázquez Pombo et al., 2022). The stochastic nature of weather can lead to model mis-specification and poor accuracy for conventional predictors. Meanwhile, data-driven machine learning (ML) models have achieved high predictive accuracy but come with their own challenges. Many ML-based PV predictors act as "black boxes" with limited interpretability, offering little insight into how input features (e.g., weather forecasts, irradiance, ambient temperature, cloud cover, and wind speed) influence the output (Vázquez Pombo et al., 2022). This lack of transparency is problematic for energy planners who require confidence that forecasts align with physical behaviour.

Additionally, ML models can be sensitive to noisy data and outliers in weather inputs; without special handling, their performance may degrade under unusual weather events (Barhmi et al., 2024). Purely data-driven models also risk learning spurious correlations, especially when training data are limited or unbalanced. Perhaps most critically, traditional ML approaches often do not incorporate known physics of PV systems, meaning they might violate energy-balance principles or perform inconsistently outside the training-data regime.

The absence of physical constraints or domain knowledge in the model can reduce generalizability and trustworthiness (Vázquez Pombo et al., 2022). Researchers have identified the integration of physical knowledge into ML as a promising avenue, yet comparatively fewer works have explored such hybrid models for PV power forecasting (Vázquez Pombo et al., 2022). These limitations motivate physics-guided approaches that retain ML's pattern-learning strength while enforcing domain plausibility.

In this paper, we propose a Hybrid Physics-Informed Temporal Attention (H-PITA) model for day-ahead PV power forecasting. H-PITA integrates data-driven learning with an external physics prior of daily PV energy production. The architecture couples a temporal-attention encoder with a gated fusion mechanism that combines the learned temporal representation with a physics-based auxiliary input x_{phys}, corresponding to the physically estimated daily PV energy. In addition, a lightweight residual path from x_{phys} is included to stabilize training and preserve physical consistency in the final prediction.

Attention mechanisms allow the network to focus on the most relevant portions of daily sequences/lags, dynamically highlighting time steps that strongly influence PV output (e.g., rapid cloud transitions) (Lei, 2024; Zhong et al., 2024).

The external physics prior (e.g., a clear-sky or simplified performance estimate) injects domain knowledge directly into the forecasting process, improving interpretability and helping predictions remain consistent with known physical relationships (Vázquez Pombo et al., 2022).

In this paper, training employs a composite objective with dynamic weighting of the physics term to balance accuracy and physical consistency; specifically, the prior enters through four complementary mechanisms: (i) a symmetric relative penalty around physical benchmark E_{phys}, (ii) a one-sided hinge that activates when predictions exceed

(1+margin)·E_{phys}, (iii) an explicit non-negativity penalty, and (iv) adaptive (dynamic) weighting of the physics term during training.

For operation, an optional +8% hard cap at inference provides a tunable safety mode that enforces a physically conservative upper bound with a controlled accuracy trade-off. The 8% value was adopted as a conservative heuristic safety margin for the optional inference-time cap. To examine whether the observed behaviour depended strongly on this choice, an additional 12% cap was also evaluated in the sensitivity analysis as a less restrictive alternative. This design aligns with evidence that attention improves the extraction of salient temporal patterns in weather sequences and with advances in physics-informed learning and dynamic loss balancing (Vázquez Pombo et al., 2022; Lei, 2024; Zhong et al., 2024).

Why this model? A recent meta-survey of PV-forecasting papers shows that regression/ANN/instance-based methods dominate the literature, while hybrid approaches are comparatively under-represented—signaling scope for architectures that combine deep temporal modeling with explicit physics (Figure 1) (Alcañiz et al., 2023). H-PITA is positioned precisely in this hybrid, physics-guided regime, aiming to deliver strong accuracy, interpretability, and trustworthy behaviour under atypical conditions.

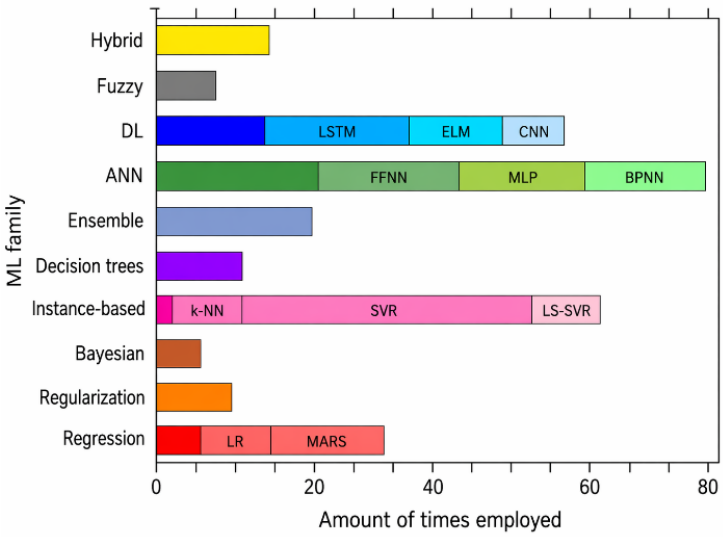


Figure 1. Distribution of ML families employed in PV-forecasting studies; hybrid methods are less common than ANN/instance-based/regression. Adapted from (Alcañiz et al., 2023), CC BY 4.0.

2. Recent developments in PV energy forecasting and physics-integrated models

2.1. Machine learning models for PV forecasting

Modern machine-learning (ML) techniques have markedly improved the accuracy of photovoltaic (PV) power forecasting across settings ranging from small rooftop systems to utility-scale plants. Among recurrent architectures, Long Short-Term Memory (LSTM) networks remain widely adopted for capturing temporal dependencies in PV generation; in a recent meta-survey, Alcañiz et al. report frequent use of LSTMs and generally strong performance across studies (Alcañiz et al., 2023).

Building on recurrence, attention mechanisms have been introduced to emphasize the most informative portions of multivariate weather sequences. For example, Hu et al. combine LSTM with self-attention—ingesting weather forecasts—to enhance multistep PV prediction, reporting substantial gains over a plain LSTM on a building-scale PV system in Japan (Hu et al., 2024).

Attention has also been incorporated to improve feature extraction prior to sequence modeling. Zhong et al. propose an attention-based DiPLS–BiLSTM framework, where Dynamic Inner Partial Least Squares is used to extract latent features from high-dimensional inputs, an attention layer assigns adaptive importance to these features, and a BiLSTM captures bidirectional temporal dependencies. Their results demonstrate high predictive accuracy; however, the approach remains purely data-driven and does not incorporate explicit physical constraints or physics-based priors (Zhong et al., 2024).

Attention has further been embedded within hybrid deep architectures: Song et al. pair a CNN–BiLSTM feature extractor with an NGBoost probabilistic head and attention, showing notable improvements over strong deep-learning baselines across multiple Australian sites (Song et al., 2025).

Separately, transformer-style attention has been adapted to solar time series within CNN–LSTM–Transformer hybrids, where attention layers help fuse spatial–temporal features and improve accuracy on public PV datasets (Lei, 2024). Overall, recent evidence indicates that mixing convolutional encoders, sequence models, and attention modules can yield state-of-the-art results on real PV deployments (Song et al., 2025; Al-Ali et al., 2023).

Similar conclusions regarding the effectiveness of machine-learning approaches for complex forecasting tasks have also been reported outside the photovoltaic domain, where comparative studies show that ensemble and tree-based models often outperform classical statistical time-series methods when nonlinear temporal patterns dominate (Dwivedi et al., 2025).

Beyond offline validation, ML predictors are increasingly tested in live settings. Nithya et al. demonstrate a sequence-to-sequence LSTM operating in real time on a 100 kW campus PV plant, using streaming measurements to generate rolling forecasts and illustrating practical reliability under operational constraints (Nithya et al., 2024).

The importance of explicitly modeling temporal dependencies has likewise been highlighted in broader machine-learning applications, where recurrent and sequential neural architectures consistently outperform static models by capturing long-term patterns in complex time-series data (Sneiders et al., 2025).

Taken together, the literature since ~2019 shows a maturation from single-family models (e.g., plain LSTMs) toward attention-enhanced hybrids that better exploit spatiotemporal structure while maintaining deployability (Alcañiz et al., 2023; Nithya et al., 2024). In the broader renewable-energy decision-support literature, Özkurt et al. further show that combining machine learning with explainability-oriented methods can support more transparent and interpretable energy-related decisions (Özkurt et al., 2025).

2.2. Use of physical models within the forecasting pipeline

In parallel with purely data-driven models, many PV forecasting pipelines explicitly integrate physics at one or more stages. A common practice is to reference a clear-sky model of irradiance to form the clear-sky index, thereby detrending strong diurnal/seasonal effects and improving stationarity for downstream learning; comparative analyses generally favor clear-sky index formulations over simpler normalizations for forecast accuracy (Lauret et al., 2022). Physics-derived features are also fed directly into deep models—for instance, including Ineichen–Perez clear-sky Global Horizontal Irradiance (GHI) as an input signal that anchors forecasts to a physically plausible envelope and lets the network learn cloud-induced deviations (Maciel et al., 2024).

Related evidence from broader energy-forecasting applications further suggests that incorporating physically meaningful and context-aware variables—such as environmental and temporal factors—can substantially improve predictive accuracy, particularly under data scarcity or noisy measurement conditions (Sanfilippo et al., *2025*).

At the post-processing level, indirect pipelines first predict irradiance and then convert to power via a PV performance model. Urhuerhi et al. illustrate such a two-stage approach—ML weather-to-irradiance followed by PVLib-based power conversion—leveraging known device physics to stabilize power outputs under varying conditions (Urhuerhi et al., 2025).

In day-ahead horizons, numerical weather prediction (NWP) provides a physical baseline that can be post-processed by ML to correct systematic biases; Buonanno et al. show that even compact linear models can yield consistent error reductions over raw NWP PV forecasts when data are scarce (e.g., new plants) (Buonanno et al., 2024). These physics–ML ensembles—where a physics model supplies a first-principles estimate, and ML performs bias-correction—are increasingly prevalent across PV forecasting workflows (Lauret et al., 2022, Buonanno et al., 2024).

Finally, simple physics-aware baselines such as persistence or smart persistence (clear-sky-index persistence) remain indispensable sanity checks for operators and continue to set a performance bar for advanced ML systems (IEA PVPS Task 16, 2025).

2.3. Physics-informed neural networks (PINNs) and physics-integrated learning

Physics-Informed Neural Networks (PINNs) embed governing equations directly into the training objective so that predictions honor known physics while fitting data (Raissi et al., 2019). In power systems, PINNs have been explored for fast, stable surrogates of grid dynamics: e.g., Stiasny et al. develop PINNSim to accelerate transient simulations by

enforcing differential–algebraic power-system equations in the loss, enabling larger time steps without sacrificing physical fidelity on benchmark networks (Stiasny et al., 2024).

For state estimation and security, physics-guided neural formulations that enforce power-flow constraints have shown improved robustness to bad data and cyber-attack scenarios relative to purely data-driven baselines (Falas et al., 2025).

In renewables, physics-integrated approaches have improved extrapolation under extremes; for wind power, Liu et al. incorporate analytical turbine power curves within a physics-informed reinforcement learning framework to enhance performance during rare events and data-sparse regimes (Liu et al., 2024).

Collectively, this line of work demonstrates that explicit physics—be it through constraints, priors, or loss terms—can reduce data demands, improve robustness, and support trustworthy behavior in safety-critical energy applications (Raissi et al., 2019; Liu et al., 2024).

In parallel, broader machine-learning studies emphasize that explicitly modeling temporal dependencies is critical for reliable forecasting, with sequential neural architectures consistently outperforming static models by capturing long-term patterns in complex time-series data (Sneiders et al., 2025*).*

Implication for hybrid PV forecasting. Taken together, the developments above motivate hybrid designs that combine temporal attention with external physical signals or constraints. From a broader methodological perspective, comparative forecasting studies show that machine-learning models—particularly ensemble and tree-based approaches—often outperform classical statistical time-series methods when nonlinear temporal effects dominate, further supporting hybrid designs that balance data adaptivity with structural priors (Dwivedi et al., 2025).

Such designs can retain the accuracy of modern deep models while aligning forecasts with domain knowledge—an approach our H-PITA model operationalizes by gating an external physics prior, adding a lightweight residual physical pathway, and employing a physics-aware training objective with dynamic weighting (see Section 3.2).

3. Methodology and data

The proposed model is a Hybrid Physics-Informed Neural Network with a temporal attention mechanism. The architecture is designed to combine information extracted from historical sequences of meteorological data with physics-based constraints, providing stable and interpretable forecasts for daily photovoltaic energy production.

Implementation and experiments were conducted in Python/PyTorch on Google Colab using an NVIDIA T4 GPU. NumPy and pandas handled preprocessing, scikit-learn provided standardization, and training employed AdamW with Lookahead and an exponential moving average (EMA) of weights. No mixed-precision training was used. To support reproducibility, deterministic random seeds were set, and the notebooks are structured to run end-to-end in Google Colab.

3.1. System description and dataset

3.1.1. Photovoltaic system

The case study considers a grid-connected rooftop PV system in Southeast Europe (exact coordinates withheld for privacy). The array comprises 42 SANYO HIT-235 modules (STC 235 W each) for a 9.87 kW DC nameplate capacity, mounted south at 1.0° tilt under low shading and coupled to the grid via a single SMA 10000TL inverter. A "100 W inverter size" field appears in the source metadata; this inconsistency is treated as a metadata anomaly and excluded from analysis. The reported commissioning date is 22 May 2012. Daily PV energy and static descriptors were obtained from PVOutput.org (publicly accessible). To protect privacy, precise coordinates are anonymized, and no personally identifiable information is disclosed.

3.1.2. Data preparation

Daily meteorology was retrieved from Visual Crossing Weather (max/avg temperature, humidity, cloud cover, solar-energy proxy, precipitation coverage, wind speed, qualitative conditions, and sunrise/sunset) and merged with PVOutput records on a unified daily timestamp (Visual Crossing Corporation, 2025), (PVOutput.org, 2025). The "solarenergy" variable was converted from MJ m^{-2} to kWh m^{-2}. Data quality control applied conservative hard bounds and de-duplication, followed by a physics-consistency screen against a daily benchmark $E_{phys} = G \cdot A \cdot \eta_{ref} \cdot \cos(1°)$ with G being Global Solar Irradiance, A=42×1.3 m^2, η_{ref}=0.185, and temperature coefficient 0.004°C; a soft ratio tolerance was used to avoid over-filtering edge cases.

Seasonal outliers were then removed using a monthly interquartile range (IQR) rule, where observations lying outside [Q1−1.5·IQR, Q3+1.5·IQR] were excluded for each calendar month. Overall, ~10% of rows were filtered, yielding 3 729 daily samples spanning 2014–2025.

Feature engineering encoded cyclic seasonality (month/DOY sine–cosine), added a one-day lag of energy and exponentially weighted moving averages EWM (3,7) trends for temperature, humidity, and solar-energy, and included simple interactions (e.g., humidity × wind) with optional numeric mapping of qualitative conditions. Inputs were organized as 14-day sequences (N,14,F) to predict next-day energy.

To avoid temporal leakage and reflect real-world forecasting conditions, the dataset was split chronologically, yielding 81.3% training data, 11.7% validation data, and the final 7.0% held out for testing.

All features were standardized using train-only statistics; the target used log1p followed by standardization. The unscaled physical benchmark E_{phys} was computed on the target index, standardized with a train-fit scaler, and provided as an auxiliary physics channel to the model.

3.2. Physics-informed temporal attention model

3.2.1. Model architecture

To capture temporal dependencies, we employ an attention mechanism that computes a weighted context vector across the input window of 14 days. This mechanism learns to emphasize days that are more informative for predicting photovoltaic generation.

The architecture of the proposed Hybrid Physics-Informed Temporal Attention (H-PITA) model (illustrated in Figure 2) combines temporal attention with a gated physics-aware fusion mechanism to integrate observational data and a physics-based prior of photovoltaic (PV) energy production. The multivariate temporal input sequence $x_{seq} \in \mathbb{R}^{B\times S\times F}$ where B denotes the batch size, S the sequence length (number of past days), and F the number of input features per day, is first projected into a latent embedding space through a linear transformation, followed by layer normalization and a SiLU activation.

Temporal dependencies within the input window (here, S=14 days) are captured using a temporal attention mechanism, which computes normalized attention weights across the sequence and produces a context vector as a weighted aggregation of the embedded features. This allows the model to emphasize the most informative days for PV energy prediction.

The physics-informed component introduces a gating mechanism that conditions the influence of the physical input x_{phys} (a scaled estimate of daily PV energy derived from a physical model) on the learned temporal context. Specifically, a sigmoid-based gate generates adaptive scaling coefficients, producing a gated physical signal that reflects the relevance of the physics prior under the current temporal conditions. The learned context vector, the original physical input, and its gated counterpart are then concatenated and normalized to form a joint representation.

This fused representation is processed by a multi-layer perceptron (MLP) to generate the primary prediction in the standardized target space. In addition, a lightweight residual branch from x_{phys} is included via a linear mapping scaled by a small learnable parameter initialized near zero. This residual pathway preserves a direct but tightly constrained influence of the physical prior, improving stability and physical consistency without dominating the data-driven prediction.

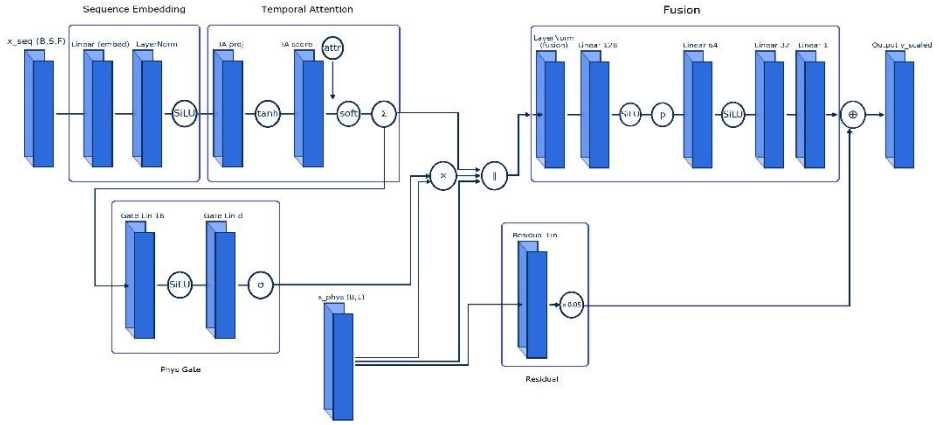


Figure 2. H-PITA Model Architecture.

3.2.2. Loss functions

The model is trained using a composite objective that balances fidelity to observed data with adherence to physical constraints. The primary data term is a weighted Huber loss, computed in real energy space (kWh) after inverting the log-scaling applied during preprocessing. To address outliers, higher weights are assigned to samples lying outside the interquartile range.

A physics loss is added, formulated as a relative Huber penalty between predicted and physically computed energy. This term is normalized by E_{phys} to preserve scale invariance, clipped to prevent domination by extreme values, and modulated differently depending on the magnitude of $E_{phys.}$ Soft hinge penalties further ensure non-negativity of predictions and prevent exceeding a dynamic upper bound defined as $1.05 \times E_{phys}$.

The total training objective is given by equation 1:

$$\mathcal{L} = \mathcal{L}_{data} + \lambda_{phys}(t) \cdot \mathcal{L}_{phys} + \lambda_{pos} \cdot \mathcal{L}_{pos} + \lambda_{cap} \cdot \mathcal{L}_{cap} \quad (1)$$

All constraint terms are applied in the real energy domain, where ŷ denotes the predicted daily PV energy in kWh.

Data loss ($\boldsymbol{\mathcal{L}_{data}}$): A weighted Huber loss computed in the real energy space (kWh), where larger weights are assigned to outlier samples to improve robustness.

Physics loss ($\boldsymbol{\mathcal{L}_{phys}}$): Measures deviation from E_{phys}, using a Huber-type formulation with clipping for stability and reduced weight for low-production days.

Non-negativity penalty ($\boldsymbol{\mathcal{L}_{pos}}$): Penalizes negative predictions given by equation 2:

$$\mathcal{L}_{pos} = \max(0, -\hat{y})^2 \quad (2)$$

Maximum capacity penalty ($\boldsymbol{\mathcal{L}_{cap}}$): Penalizes predictions exceeding $1.05 \cdot E_{phys}$ given by equation 3:

$$\mathcal{L}_{cap} = \max(0, \hat{y} - 1.05 \cdot E_{phys})^2. \quad (3)$$

Weighting coefficients: $\lambda_{phys}(t)$ increases gradually during early epochs (half-cosine ramp) and is updated dynamically within each batch to balance data and physics losses. λ_{pos} and λ_{cap} are small fixed values ($5 \times 10{-}4$) to softly enforce non-negativity and capacity limits without dominating the objective.

3.2.3. Training strategy

The model is trained with the AdamW optimizer wrapped in Lookahead for improved stability. A learning rate scheduler (ReduceLROnPlateau) adapts the step size based on validation performance, while exponential moving average (EMA) smoothing of model parameters further stabilizes training. Gradient clipping is employed to improve training stability.

Training proceeds for up to 3000 epochs with early stopping based on validation mean absolute percentage error (MAPE). Mini-batches of size 512 are used, and gradient accumulation ensures effective behaviour with larger batches. The dynamic physics weighting mechanism continuously adjusts the contribution of the physics loss per batch, constrained within ±10% relative changes to ensure smooth evolution.

This methodology yields a hybrid architecture that leverages physical priors while retaining the flexibility of deep learning, guided by carefully designed loss functions and stabilization techniques to ensure robustness and generalization.

The overall training procedure of the proposed H-PITA model is summarized in Algorithm 1. The pipeline includes data preparation and consistency checks, model construction, and an epoch–batch training loop with validation-based model selection. Physics-informed constraints are incorporated directly into the loss function, while exponential moving averages (EMAs) are used to stabilize metric tracking. Model selection is performed via validation-based early stopping, and the final checkpoint corresponds to the best validation performance.

Algorithm 1. Training of the H-PITA model

Input: Training and validation datasets (X_{seq}, X_{phys}, y, E_{phys}); maximum epochs E; patience P; target physics share τ.
Output: Trained H-PITA model.

Initialize H-PITA parameters, optimizer, learning-rate scheduler, and EMA.
Compute real-scale targets and sample weights using an IQR-based outlier rule.
Initialize dynamic physics weight w_{dyn} and base ramp schedule $w_{base}(t)$.
For each epoch t=1,…,E :
    Compute effective physics weight $w_{eff}=w_{base}(t)\cdot w_{dyn}$.
    For each mini-batch:
        Predict daily energy $\hat{y}$ and convert to real units (kWh).
        Compute data loss $\mathcal{L}_{data}$.
        Compute physics loss $\mathcal{L}_{phys}$ (relative Huber deviation from E_{phys}).
        Compute soft constraint penalties for non-negativity and capacity.
        Form total loss (Equation 1)

$$\mathcal{L} = \mathcal{L}_{data} + \lambda_{phys}(t) \cdot \mathcal{L}_{phys} + \lambda_{pos} \cdot \mathcal{L}_{pos} + \lambda_{cap} \cdot \mathcal{L}_{cap}$$

        Update model parameters and EMA.
        Adapt w_{dyn} to keep the physics loss close to the target share τ.
    End for
    Evaluate on validation set (EMA weights).
    Update learning rate and apply early stopping if needed.
End for
Restore the best model parameters and save the final model.

4. Results and analysis

4.1. Setup and metrics

We report MAE, RMSE, R^2, and MAPE. To avoid inflated percentage errors on very small-load days, MAPE is computed after excluding targets y<1 kWh; the number of excluded observations is explicitly reported. Performance metrics are first presented using a fixed train/validation/test split to reflect standard operational deployment. In addition,

rolling cross-validation is employed to assess temporal robustness, with confidence intervals used to quantify performance variability across folds. For a physics-aware sensitivity experiment, we optionally apply a physics cap to predictions given by equation 4:

$$\hat{y}_i^{cap} = \min\{\hat{y}_i\ , (1\ +\ \rho)E_{phys}, i\},\ \ \rho\ \in\ \{0.08, 0.12\} \tag{4}$$

The cap is not used in the main tables—only as an explanatory stress test.

Two data versions are considered: Unfiltered (Train=3368, Val=484, Test=291; MAPE drop counts: 4/0/1) and Filtered (Train=3019, Val=435, Test=260; MAPE drop counts: 0/0/0).

4.2. Model performance

This section presents model performance in two complementary stages. We first report results obtained using a fixed train/validation/test split, which reflect standard operational deployment and allow detailed analysis through multiple tables and figures. These results are used to assess absolute predictive accuracy, the impact of data filtering, and the effect of physics-aware capping.

We then extend the evaluation using rolling cross-validation to assess temporal robustness and statistical stability across different forecasting periods. Cross-validation results, reported with aggregated metrics, provide additional evidence that the observed performance trends are consistent over time rather than artifacts of a single data split.

Filtering substantially improves generalization. On the validation set, MAPE falls from 19.33% to 12.65% (−6.68 pp; −34.6% relative). On the test set, MAPE falls from 22.33% to 16.68% (−5.65 pp; −25.3% relative). RMSE and MAE also decrease, and R^2 improves slightly on both splits as shown in Table 1 and Figures 3 and 4.

Table 1. Baseline metrics (no cap) filtered vs unfiltered, fixed split

Dataset	Split	MAE	RMSE	R^2	MAPE	Removed
Filtered	Train	4.398	6.604	0.8372	14.35	0/3042
Unfiltered		5.106	8.046	0.8169	24.11	4/3393
Filtered	Val	4.171	5.633	0.8436	12.41	0/438
Unfiltered		5.016	7.526	0.7794	18.75	0/488
Filtered	Test	5.518	7.978	0.7833	16.68	0/262
Unfiltered		6.005	9.607	0.7539	22.42	2/293

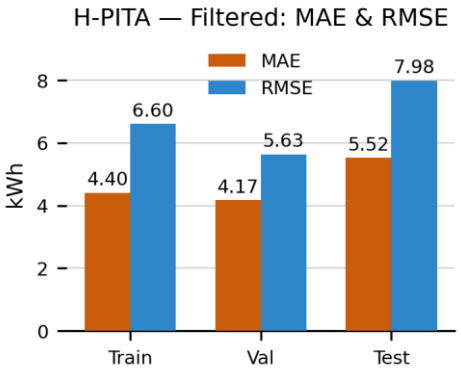


Figure 3. Results of H-PITA on filtered dataset

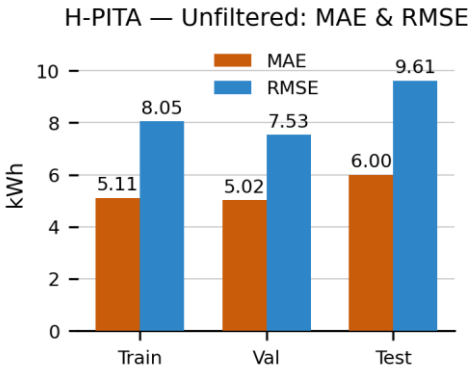


Figure 4. Results of H-PITA on unfiltered dataset

4.3. Physics-cap sensitivity (ρ = 0.08 & ρ = 0.12)

Applying the physics cap ($\hat{y} \leq (1 + \rho)E_{phys}$) decreases headline accuracy on both datasets. On the Unfiltered test set, MAPE increases by +1.99 pp (from 22.33% to 24.32%). On the Filtered test set, MAPE increases by +2.03 pp (from 16.68% to 18.71%). This suggests the physics proxy E_{phys} systematically underestimates some high-load days, forcing under-prediction. The detailed fixed-split results for the ρ=0.08 setting are reported in Table 2, while the broader effect of capping on RMSE and MAE across both datasets is illustrated in Figures 5 and 6, respectively. We therefore report uncapped metrics in the main tables and treat the cap as a diagnostic and safety constraint rather than a default operating mode.

Table 2. Metrics with physics cap (ρ = 0.08) on fixed split

Split	Dataset	MAE (kWh)	RMSE (kWh)	R^2	MAPE (%)
Train	Unfiltered + cap (8%)	6.522	9.236	0.7597	26.70
Val	Unfiltered + cap (8%)	5.877	8.246	0.7380	20.39
Test	Unfiltered + cap (8%)	6.520	8.960	0.7768	24.32
Train	Filtered + cap (8%)	5.971	8.184	0.7513	17.04
Val	Filtered + cap (8%)	5.066	6.593	0.7863	13.71
Test	Filtered + cap (8%)	6.292	8.275	0.7602	18.71

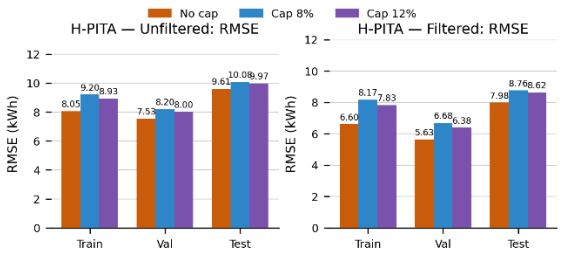


Figure 5. H-PITA RMSE unfiltered and filtered datasets, cap and no cap

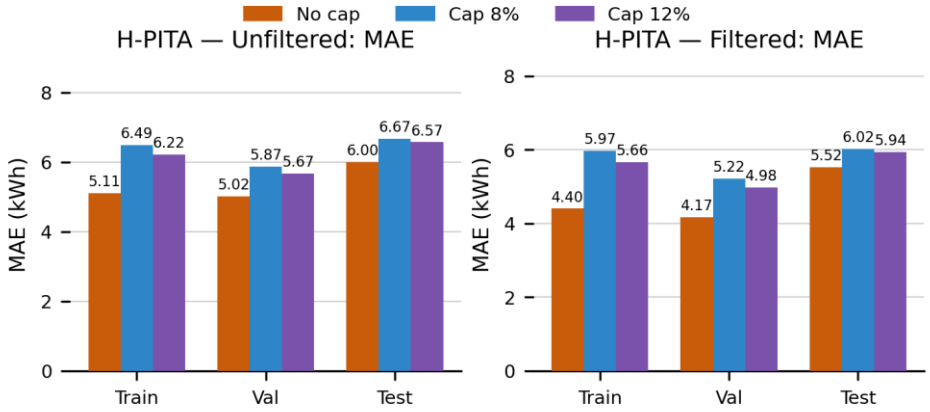


Figure 6. H-PITA MAE unfiltered and filtered datasets, cap and no cap

Imposing a physics cap at ρ = 8% or ρ = 12% consistently degrades predictive accuracy. On the filtered test set, MAE rises from 5.52 kWh (no cap) to 6.02 (+9.1%) with ρ = 8% and 5.94 (+7.7%) with ρ = 12%; RMSE increases from 7.98 to 8.76 (+9.8%) and 8.62 (+8.0%); MAPE from 16.68% to 17.63% / 17.55%; while R^2 drops from 0.783 to 0.739 / 0.747. The same pattern holds on the unfiltered test set: MAE 6.01 → 6.67 (+11.1%) / 6.57 (+9.5%), RMSE 9.61 → 10.08 (+5.0%) / 9.97 (+3.8%), MAPE 22.42% → 23.76% / 23.61%, and R^2 0.754 → 0.729 / 0.735. In short, the cap truncates high-load peaks—where much of the variance resides—biasing predictions downward and inflating error metrics; it is therefore best viewed as a safety constraint rather than a performance enhancer for H-PITA model.

To further assess the robustness of the proposed H-PITA model beyond a single train–validation–test split, we additionally evaluate predictive performance using rolling cross-validation. This evaluation protocol preserves temporal causality and enables a statistically sound assessment of performance stability across multiple forecasting horizons. The resulting MAE, RMSE, R^2, and MAPE metrics, averaged across all folds, are reported in Table 3, providing a complementary view of model behavior under realistic operational conditions.

Table 3. H-PITA rolling cross-validation performance

Physics cap (ρ)	MAE (kWh)	RMSE (kWh)	R^2	MAPE (%)
None (default)	5.55	7.59	0.741	20.58
0.08	5.98	7.80	0.738	16.93
0.12	5.74	7.54	0.754	16.69

Under rolling cross-validation, H-PITA achieves an average MAE of 5.55 kWh, RMSE of 7.59 kWh, R^2 of 0.74, and MAPE of 20.58% without physics capping. Applying a physics-aware cap improves MAPE (down to 16.69–16.93%) at the cost of a slight reduction in overall accuracy, confirming the cap's role as a robustness and safety mechanism rather than the default operating mode.

4.4. Baseline comparison

To benchmark the proposed H-PITA model against strong and representative baselines, we consider both classical forecasting methods and modern machine-learning approaches, including persistence (D–1), climatology (day-of-year), linear and ridge regression, LSTM with attention, and XGBoost.

4.4.1. Quantitative comparison

This section compares the proposed H-PITA model with the set of the baselines under two complementary evaluation settings. First, we report fixed train/validation/test split results for both the filtered and unfiltered datasets, using uncapped predictions (no physics cap) for all methods. This analysis reflects standard operational deployment and enables a direct comparison of absolute predictive performance across data quality conditions.

Second, to assess robustness and temporal stability, we evaluate all methods using rolling cross-validation on the filtered dataset only, again under the no-cap setting. This protocol preserves temporal causality and provides fold-averaged performance metrics that support a fair and statistically grounded comparison with H-PITA. Together, these two evaluation perspectives allow us to distinguish between absolute accuracy under fixed splits and consistency of performance over time.

Fixed train/validation/test split protocol (baselines vs. H-PITA): Under the fixed train/validation/test split protocol, H-PITA consistently outperforms all baseline models across both the filtered and unfiltered datasets. As reported in Table 4 (filtered data, no cap), H-PITA achieves the lowest errors and highest explanatory power on the test set (MAE = 5.518 kWh, RMSE = 7.978 kWh, R^2 = 0.7833), substantially improving upon classical baselines and advanced competitors such as XGBoost and LSTM with attention.

Table 4. Filtered data, all methods, no cap

Method	Train				Validation				Test			
	MAE	RMSE	R^2	MAPE%	MAE	RMSE	R^2	MAPE%	MAE	RMSE	R^2	MAPE%
Climatology DoY	8.300	10.811	0.5638	30.14	8.269	10.824	0.4225	30.08	9.036	12.084	0.5029	37.87
H-PITA	4.398	6.604	0.8372	14.35	4.171	5.633	0.8436	12.41	5.518	7.978	0.7833	16.68
LSTM+Attn	7.586	10.879	0.5582	30.26	6.845	10.558	0.4505	27.35	8.025	11.668	0.5365	34.64
Linear	9.244	11.369	0.5176	28.21	8.933	11.124	0.3901	28.19	9.106	11.439	0.5545	30.85
Persistence D-1	9.154	13.723	0.2965	31.17	7.654	12.136	0.275	27.23	7.985	12.184	0.4881	27.65
Ridge	9.217	11.321	0.5216	28.29	8.636	10.771	0.4282	27.49	9.037	11.281	0.5667	30.59
XGBoost	3.914	5.145	0.9012	13.31	6.605	9.513	0.5539	25.19	7.792	10.589	0.6183	31.98

Similar trends are observed for the unfiltered dataset in Table 5, where H-PITA maintains superior generalization despite increased noise and data variability. These quantitative results are further corroborated by the comparative visual analysis: Figure 7 and Figure 8 show consistently lower validation MAPE and RMSE for H-PITA relative

to all baselines, while Figure 9 and Figure 10 confirm that these advantages persist on the test set for both dataset variants.

Table 5. Unfiltered data, all methods, no cap

Method	Train				Validation				Test			
	MAE	RMSE	R^2	MAPE%	MAE	RMSE	R^2	MAPE%	MAE	RMSE	R^2	MAPE%
Climatology DoY	10.956	13.927	0.4513	64.00	10.365	13.796	0.2587	55.05	10.942	14.409	0.4465	66.99
H-PITA	5.106	8.046	0.8169	24.11	5.016	7.526	0.7794	18.75	6.005	9.607	0.7539	22.42
LSTM+Attn	10.122	14.014	0.4445	63.54	8.807	13.275	0.3137	50.51	9.160	13.510	0.5134	61.55
Linear	11.981	14.76	0.3837	52.11	11.590	14.242	0.2101	47.20	11.55	14.227	0.4604	51.21
Persistence D-1	11.505	17.033	0.1796	58.40	10.156	15.523	0.0618	51.16	10.02	15.242	0.3825	47.00
Ridge	11.970	14.698	0.3889	52.34	11.471	14.025	0.2339	46.86	11.501	14.137	0.4672	50.94
XGBoost	8.741	11.255	0.6416	50.00	8.736	12.089	0.4309	47.12	9.565	12.588	0.5775	55.74

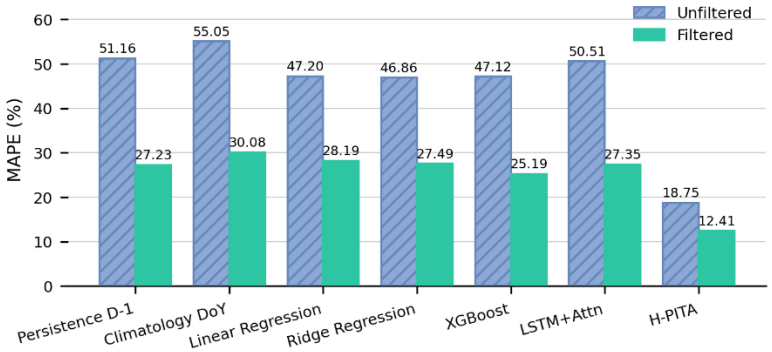


Figure 7. Validation MAPE - H-PITA vs baselines (unfiltered and filtered datasets)

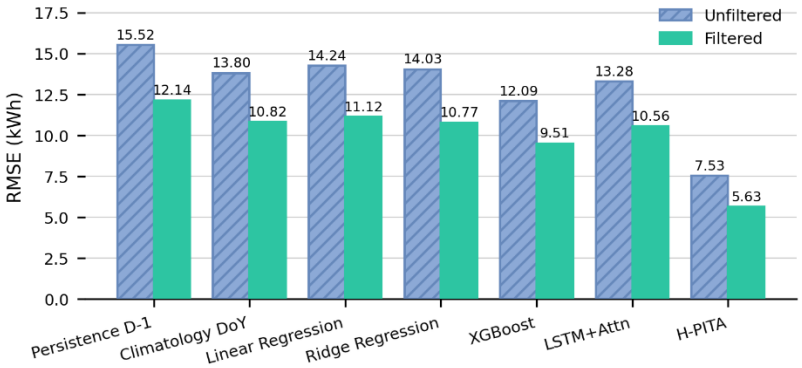


Figure 8. Validation RMSE - H-PITA vs baselines (unfiltered and filtered datasets)

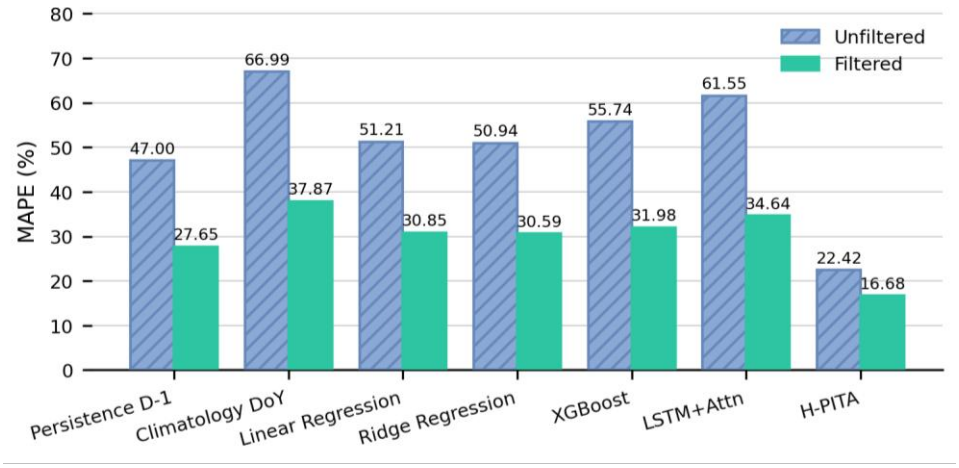


Figure 9. Test MAPE - H-PITA vs baselines (unfiltered and filtered datasets)

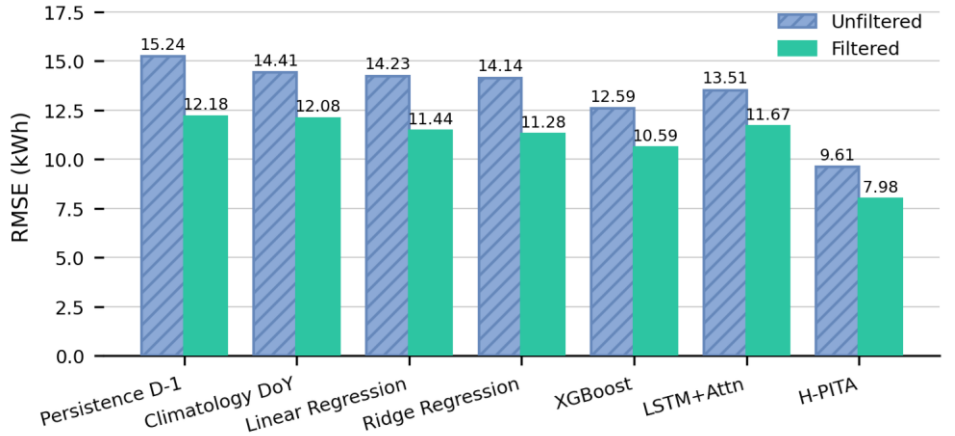


Figure 10. Test RMSE - H-PITA vs baselines (unfiltered and filtered datasets)

Overall, the agreement between tabular metrics and graphical trends demonstrates that the proposed H-PITA model delivers robust and stable performance gains under standard fixed-split evaluation, reinforcing its effectiveness compared to both traditional and data-driven baseline approaches.

Rolling cross-validation protocol (baselines vs. H-PITA): To ensure a fair and consistent comparison, rolling cross-validation is applied uniformly to H-PITA and all baseline models, including climatology, persistence, linear and ridge regression, LSTM with attention, and XGBoost. Each method is evaluated using the same five expanding-origin folds, identical input features, and the same temporal splits. Performance metrics

(MAE, RMSE, R^2, and MAPE) are averaged across folds and computed for all physics-cap settings. While capped variants are evaluated as part of the sensitivity analysis, we report uncapped (no-cap) results in Table 6 for clarity and comparability.

Importantly, H-PITA consistently outperforms all baseline models across both capped and uncapped settings, indicating that its superior performance is robust to the inclusion of physics-aware constraints. This evaluation protocol ensures that observed performance differences reflect genuine modeling capabilities rather than advantages arising from data access, tuning procedures, or temporal leakage, thereby enabling a balanced comparison between H-PITA and competing approaches.

Table 6. Rolling cross-validation results (baselines vs. H-PITA)

Method	MAE	RMSE	R2	MAPE
Climatology	8.7278	11.7009	0.412	32.4726
H-PITA	5.5509	7.5882	0.741	20.5754
Linear	8.2840	10.8618	0.493	29.6320
LSTM	8.0126	10.5507	0.522	27.2028
Persistence	8.3368	12.7965	0.295	28.9652
Ridge	8.0771	10.6347	0.514	29.3052
XGBoost	7.636183	10.54602	0.522	29.69312

4.4.2. Statistical significance analysis

While aggregate error metrics provide an overall view of predictive accuracy, they do not indicate whether observed performance differences between models are statistically meaningful. To assess the significance of the improvements achieved by H-PITA over baseline approaches, we employ Diebold–Mariano (DM) tests on rolling cross-validation forecasts. The DM test evaluates the null hypothesis of equal predictive accuracy using paired forecast errors, allowing a rigorous comparison across time while accounting for temporal dependence. By applying this analysis to all baseline models and physics-cap settings, we determine whether the observed gains of H-PITA reflect systematic improvements rather than random variation. Table 7 reports the p-values of the Diebold–Mariano test for the statistical comparison of predictive accuracy between H-PITA and baseline models. Diebold–Mariano tests conducted on rolling cross-validation forecasts consistently reject the null hypothesis of equal predictive accuracy between H-PITA and all baseline models.

Across all physics-cap settings, the DM statistics are negative and highly significant ($p < 10^{-6}$), indicating systematically lower forecast errors for H-PITA.

Table 7. p-values of the Diebold–Mariano test comparing the predictive accuracy of H-PITA against baseline models.

Physics cap (ρ)	Climatology	Linear	LSTM	Persistence	Ridge	XGBoost
0.08	9.48e-13	1.73e-13	5.35e-09	2.17e-20	6.23e-11	2.50e-06
0.12	7.80e-14	9.66e-15	4.40e-10	9.41e-21	3.86e-12	2.66e-07
None	8.17e-23	8.28e-19	2.93e-13	2.54e-19	1.26e-16	8.06e-16

The strongest statistical advantage is observed in the uncapped setting, where H-PITA outperforms LSTM- and tree-based baselines on more than 60% of test instances. Physics-aware caps reduce daily extremes and improve robustness, but slightly attenuate the relative advantage, confirming their role as a conservative safety mechanism rather than a default-operating mode.

5. Discussion

This study proposed H-PITA, a Hybrid Physics-Informed Temporal Attention model for daily photovoltaic energy forecasting, and evaluated it against a broad set of classical and machine-learning baselines under multiple evaluation protocols. The results demonstrate that H-PITA achieves superior predictive accuracy and robustness in both fixed-split evaluations and rolling cross-validation, highlighting the effectiveness of integrating temporal attention with physics-informed constraints.

Across both filtered and unfiltered datasets, H-PITA yields lower MAE, RMSE, and higher R^2 than all baselines, including strong competitors such as XGBoost and attention-enhanced LSTM models. The performance gains are most pronounced on the filtered dataset, where improved data quality allows the model to better exploit temporal dependencies and physical structure. This confirms that careful data curation is essential for enhancing generalization, particularly in hybrid models that combine data-driven and physics-based components.

The rolling cross-validation results further indicate that the observed improvements are stable over time and not driven by favorable fixed splits. Under expanding-origin evaluation, H-PITA consistently outperforms all baseline methods, achieving the best average performance across all reported metrics. Diebold–Mariano tests provide strong statistical evidence that these gains are systematic rather than coincidental, with extremely small p-values across all comparisons, even under different physics-cap settings.

A key characteristic of H-PITA is the explicit incorporation of physical knowledge through an external physics prior and adaptive loss weighting. Sensitivity analysis with physics-aware capping shows that while hard caps can enforce physical plausibility, they may slightly reduce accuracy on high-production days when the physics proxy underestimates peak output. This suggests that physics constraints are most effective when applied softly during training or used as a diagnostic and safety mechanism at inference, rather than as a strict operational limit. Accordingly, uncapped configurations offer the best balance between accuracy and flexibility, while capped variants provide an additional robustness option for risk-aware applications.

From an operational perspective, these findings are highly relevant for energy management and grid integration. Reliable PV forecasts support scheduling, market participation, and asset management, while physics-aware safeguards add confidence in safety-critical settings. The modular design of H-PITA—allowing plug-in physics estimates and dynamic loss weighting—facilitates adaptation to different PV systems, data availability conditions, and forecasting horizons.

This work does not aim to exhaustively isolate the contribution of each architectural component through ablation. Instead, robustness analyses across datasets, evaluation protocols, and physics-cap configurations provide practical evidence of the model's effectiveness. Future work may extend the framework toward probabilistic forecasting,

multi-site evaluation, and more detailed physical modeling to enhance uncertainty awareness and interpretability.

Overall, the results show that combining temporal attention with physics-informed learning yields a stable and effective forecasting framework. By embedding domain knowledge directly into the learning process, H-PITA advances beyond purely data-driven baselines and offers a compelling solution for reliable photovoltaic energy forecasting in real-world operational contexts.

6. Conclusion

This paper introduced H-PITA, a Hybrid Physics-Informed Temporal Attention model for daily photovoltaic energy forecasting, and validated it against a comprehensive set of baseline methods using both fixed-split and rolling cross-validation protocols. The results show that combining temporal attention with physics-informed learning leads to consistently improved accuracy and robustness compared to purely data-driven approaches. By embedding domain knowledge directly into the learning process, H-PITA provides a reliable and practically deployable forecasting framework, well suited for real-world photovoltaic energy management and operational decision support.

References

Al-Ali, E.M., Hajji, Y., Said, Y., Hleili, M., Alanzi, A.M., Laatar, A.H., Atri, M. (2023). Solar energy production forecasting based on a hybrid CNN–LSTM–Transformer model. Math. 11, 676. https://doi.org/10.3390/math11030676

Alcañiz, A., Grzebyk, D., Ziar, H., Isabella, O. (2023). Trends and gaps in photovoltaic power forecasting with machine learning. Energy Reports, 9, 447–471. https://doi.org/10.1016/j.egyr.2022.11.208

Barhmi, K., Heynen, C., Golroodbari, S. G. M., van Sark, W. (2024). A review of solar forecasting techniques and the role of artificial intelligence. Solar, 4(1), 99–135. https://doi.org/10.3390/solar4010005

Buonanno, A. et al. (2024). Machine learning and weather-model combination for PV production forecasting. Energies, 17(9), https://doi.org/10.3390/en17092203

Di Leo, P., Ciocia, A., Malgaroli, G., Spertino, F. (2025). Advancements and challenges in photovoltaic power forecasting: A comprehensive review. Energies, 18(8), 2108. https://doi.org/10.3390/en18082108

Dwivedi, V., Kadian, K., Gardiner, B., Pandey, R., Lathwal, A. (2025). Supply chain analytics: A performance evaluation of machine learning, statistical, and time-series models. *Baltic J. Modern Computing*, **13**(2), 453–485. https://doi.org/10.22364/bjmc.2025.13.2.09

Falas, S., Asprou, M., Konstantinou, C., Michael, M. K. (2025). Robust power system state estimation using physics-informed neural networks. https://arxiv.org/abs/2507.05874

Hu, Z., Gao, Y., Ji, S., Mae, M., Imaizumi, T. (2024). Improved multistep-ahead PV power prediction based on LSTM and self-attention with weather forecast data. Applied Energy, 359, https://doi.org/10.1016/j.apenergy.2024.122709

IEA PVPS Task 16 (2025). The added value of combining solar irradiance data and forecasts (Technical report). https://iea-pvps.org/

Lauret, P., Alonso-Suárez, R., Le Gal La Salle, J., David, M. (2022). Solar forecasts based on the clear-sky index or the clearness index: Which is better? Solar, 2(4), 432–444. https://doi.org/10.3390/solar2040026

Lei, X. (2024). A photovoltaic prediction model with integrated attention mechanism. Mathematics, 12(13), https://doi.org/10.3390/math12132103

Liu, Y., Wang, J., Liu, L. (2024). Physics-informed reinforcement learning for probabilistic wind power forecasting under extreme events. Applied Energy, 376, https://doi.org/10.1016/j.apenergy.2024.124068

Maciel, J. N., Gimenez Ledesma, J. J., Ando, O. H., Jr. (2024). Dataset for machine learning: Explicit all-sky image features to enhance solar irradiance prediction. Data, 9(10), https://doi.org/10.3390/data9100113

Nithya, C., Roselyn, J. P., Devaraj, D. (2024). Real-time solar PV generation in a building using LSTM-based time-series forecasting. Discover Electronics, 1, https://doi.org/10.1007/s44291-024-00023-0

Özkurt, C., Canay, Ö., Tunç, E. A., Aydın, E., Velioğlu, B. S. (2025). Renewable energy source ranking and analysis using fuzzy MCDM, ML, and XAI techniques. *Baltic J. Modern Computing*, **13**(3), 656–679. https://doi.org/10.22364/bjmc.2025.13.3.06

PVOutput.org. (2025). PVOutput—Free solar power monitoring for users and installers. https://pvoutput.org/

Raissi, M., Perdikaris, P., Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Sanfilippo, S., Hernandez-Cabrera, J. J., Kandler, C., Hernandez-Galvez, J. J., Evora-Gomez, J., & Roncal-Andres, O. (2025). A neural network-based causal model for electricity demand estimation in remote areas: A case study in El Espino, Bolivia. *Baltic J. Modern Computing*, **13**(2), 315–330. https://doi.org/10.22364/bjmc.2025.13.2.01

Sneiders, M., Urtans, E., Abu Saa, A. (2025). Predicting student performance on a novel Moodle dataset using GRU time series model. *Ba*ltic J. Modern Computing, 13(4), 885–893. https://doi.org/10.22364/bjmc.2025.13.4.07

Song, Z., Xiao, F., Chen, Z., Madsen, H. (2025). Probabilistic ultra-short-term solar PV power forecasting using NGBoost with attention-enhanced neural networks. Energy and AI, 20, Article 100496. https://doi.org/10.1016/j.egyai.2025.100496

Stiasny, J., Zhang, B., Chatzivasileiadis, S. (2024). PINNSim: A simulator for power system dynamics based on physics-informed neural networks. Electric Power Systems Research, 235, https://doi.org/10.1016/j.epsr.2024.110796

Urhuerhi, O. P., Udomboso, C., Sigauke, C. (2025). Forecasting solar power output in Ibadan: A machine learning approach leveraging weather data and system specifications. https://arxiv.org/abs/2508.07462

Vázquez Pombo, D., Bindner, H. W., Spataru, S. V., Sørensen, P. E., Bacher, P. (2022). Increasing the accuracy of hourly multi-output solar power forecast with physics-informed machine learning. Sensors, 22(3), Article 749. https://doi.org/10.3390/s22030749

Visual Crossing Corporation. (2025). Visual Crossing Weather—Historical & forecast weather data (API service). https://www.visualcrossing.com/

Zhong, Y., He, T., Mao, Z. (2024). Enhanced solar power prediction using attention-based DiPLS-BiLSTM model. Electronics, 13(23), https://doi.org/10.3390/electronics13234815

Received January 2, 2026, revised April 26, 2026, accepted May 15, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, pp. 376–401
https://doi.org/10.22364/bjmc.2026.14.2.06

Modelling and Assessment of Asynchronous Evidence-based Network Flows for Cyber-attack Detection

Virgilijus KRINICKIJ, Linas BUKAUSKAS,

Institute of Computer Science, Vilnius University, Vilnius, Lithuania

virgilijus.krinickij@mif.vu.lt, linas.bukauskas@mif.vu.lt

ORCID 0009-0005-8248-5690, ORCID 0000-0002-9781-9690

Abstract. In cybersecurity, analysing network traffic is critical for identifying potential threats and mitigating incidents. Traditional approaches to network traffic analysis often rely on synchronous methods that may not fully capture the dynamic nature of network behaviour. This paper presents an approach for asynchronous evidence-based network traffic assessment, experimenting on synthetic cyber-attack templates and large network flow datasets available online. Our approach leverages asynchronous data collection to capture network traffic at varying time intervals, enabling the identification of incidents across different time frames and locations by processing and aligning the data. By combining asynchronous data collection with evidence-based assessment, our approach enables cybersecurity analysts to gain deeper insight into network traffic dynamics, enhance threat detection capabilities, and improve incident response effectiveness. We demonstrate the effectiveness of our approach through experimental evaluations using synthetic cyber-attack templates in a virtual environment, while capturing network flows. In summary, our research advances the field of network traffic analysis by presenting an approach that addresses the shortcomings of conventional synchronous techniques and lays the groundwork for more resilient, adaptive cybersecurity solutions.

Keywords: Network traffic analysis, dynamic time warping, incident detection, asynchronous evidence assessment, synthetic attack modelling, cybersecurity

1 Introduction

In the ever-evolving cybersecurity landscape, timely and accurate network-traffic analysis is the cornerstone of defence against escalating cyber threats. Conventional synchronous network analysis models frequently suffer from significant latency and resource bottlenecks, especially when processing the high-volume traffic characteristic of modern network flows (JIANG et al., 2014). These challenges are amplified by the

ever-growing number of internet-connected devices, which increases network load and the scale of telemetry that must be monitored. Therefore, network stream analysis for anomaly and cyber-attack detection in network flows has become increasingly important (Argyraki and Cheriton, 2005; Li and Chen, 2004; Heenan and Moradpoor, 2016).

Threat actors have also evolved alongside the increasing scale and complexity of modern networks. As a result, sophisticated cyber-attacks are difficult to detect in large-volume network flows (Kalinin and Krundyshev, 2021). Moreover, adversarial activity often originates from multiple geographic locations and time zones, resulting in dispersed, asynchronous traffic. The asynchronous result complicates the correlation of possible attack vectors, and the identification of cyber incidents becomes significantly harder (Diab et al., 2019).

Today, packet capture (PCAP) remains widely used as one of the most prominent mechanisms for network flow analysis. This allows experts to analyse network flows in offline mode for sophisticated cyber-attack detection. Many tools have been created to facilitate passive analysis of these files (Ulmer et al., 2019). However, due to the surge in sophisticated cyber-attacks, a shift to more dynamic and adaptive security measures is needed (Pandey, 2023). Having traditional defence systems assume a global view of all security logs being centrally gathered and fed to decision-support systems (Heenan and Moradpoor, 2016). These systems use machine learning (ML) algorithms, along with other algorithms and tools, to detect network traffic threats. In this context, predictive modelling will improve security measures by enhancing detective capabilities, especially in network flow analysis (Mehta et al., 2022; Komisarek et al., 2021; Ranjan et al., 2023; Clotet et al., 2018). However, in most cases, individualised solutions are applied to develop and support researchers or network investigators in responding to sophisticated cyber-attacks. Therefore, innovative approaches that enable dynamic control of remotely gathered data and algorithms applicable to a multitude of sources remain a critical challenge.

The key aim of this paper is to present an algorithmic method for assessing attack templates. The main goal is to implement and test dynamic network processing across asynchronous computer networks for incident and anomaly detection, while providing incident forensics and reporting. We implement distributed network monitors to gather network flows from various networks and network points to achieve the goal. For the experiments, we implemented data filters at the edge of each machine and modelled the desired flow. Such early data filtering enables prioritising relevant data.

The rest of this paper is organised as follows. In Section 2, relevant work done regarding asynchronous alignment of network flows, used algorithms for attack template forensics and the overall situation in the field are described. In Section 3, we describe the proposed model for the modelling and assessment of asynchronous network flows using architecture and flow processing. We continue with the implementation Section 4 and experiments Section 5. In the end, in Section 7, the conclusion and in Section 8, future work are presented.

2 Related Work

Recent research in cybersecurity focuses on anomaly detection and analysis of network flows. Aligning asynchronous records using dynamic time warping (DTW) can identify relevant actors and their communications across heterogeneous networks, aiding incident analysis (Krinickij and Bukauskas, 2024; Diab et al., 2019).

The current approach to asynchronous record alignment uses predefined algorithms on PCAP files. In the context of global network flow alignment, algorithms such as Needleman-Wunsch from bioinformatics (Likic, 2008) can be used as one variant. For local network flow alignment, DTW or Smith-Waterman algorithms are preferable (Müller, 2007; Beddoe, 2004). However, the classical DTW algorithm has several limitations. It is computationally expensive and requires comparing every element of two sequences, resulting in quadratic complexity and poor scalability for long inputs (Macrae and Dixon, 2010). Thus, DTW struggles with heterogeneous PCAP files from heterogeneous network points. Packets arrive in mismatched sequences. Therefore, DTW requires a huge warping window and becomes practically unusable for incident reconstruction (Krinickij and Bukauskas, 2024; Liang et al., 2021). Therefore, these constraints render DTW unsuitable for continuous network flow analysis as it requires both sequences to be fully known in advance (Krinickij and Bukauskas, 2024). For these reasons, researchers are exploring various approaches to improve the efficiency and applicability of DTW.

Efforts to reduce the quadratic computational burden include algorithms such as fast DTW (FastDTW), although this has not always been shown to outperform the original DTW (Wu and Keogh, 2020). Other variants propose different optimisation strategies: some approaches reduce the number of computations, whereas improving efficiency modestly (Lou et al., 2015). Segmental DTW divides the global cost matrix into smaller sub-matrices to be processed independently before merging results, although the runtime of weakly-ordered segmental DTW (WSDTW) is the same as DTW, and the runtime of strictly-ordered segmental DTW (SSDTW) is twice as much (Tsai, 2021). Event DTW (EventDTW), whereas more effective at handling frequency differences, retains quadratic complexity (Jiang et al., 2020). Parallelisation techniques such as parallelised diagonal DTW (ParDTW) and accelerated DTW (DTWax) exploit GPUs to accelerate execution (Yang et al., 2022; Sadasivan et al., 2023; Tralie and Dempsey, 2020), although their complexity remains the same as that of standard DTW. Flexible DTW (FlexDTW) introduces sequence start and end points flexibility, improving runtime but still incurring high memory usage (Bükey et al., 2023). Among these, the University of California, Riverside DTW (UCR-DTW) has been highlighted as one of the fastest implementations, capable of handling disk I/O efficiently and answering queries in fractions of a second (Rakthanmanon et al., 2012). Conversely, some attempts, such as 2-dimensional DTW (2DDW), have introduced even higher computational costs, with complexity increasing to $O(N^6)$ (Lei and Govindaraju, 2004). Although DTW traditionally compares only pairs of sequences (Krinickij and Bukauskas, 2024), researchers have proposed methods to extend it to multiple concurrent streams. Subsequence DTW (SUCR-DTW) employs multi-threading to match multiple time series in parallel, whereas extension subsequence DTW (ESUCR-DTW) generalises the approach to allow candidate detection across different subsequence lengths

(Giao and Anh, 2016). Another technique, multiple DTW (multiDTW), uses nested hierarchical query subgroups and tree-based data structures to prune less relevant queries, thus improving efficiency (Kremer et al., 2011).

Sliding window properties present another area for significant optimisation opportunities. The Sakoe–Chiba band was one of the earliest methods restricting alignments to a limited diagonal region, although this may exclude valid matches (Sakoe and Chiba, 2003). Later methods, such as online DTW (ODTW), reduce computational costs by comparing only small regions of the matrix, though this can lead to lower accuracy if older patterns are important (Oregi et al., 2017). Dynamic search step DTW (SegrDTW) instead adapts dynamic search windows around local features, avoiding rigid global constraints (Guo et al., 2022). Other techniques exploit structural properties: blocked DTW (BDTW) leverages repeated values to shrink matrix size (Sharabiani et al., 2018), whereas space-efficient DTW (SparseDTW) dynamically opens only promising cells, ensuring accurate results without fixed bands (Al-Naymat et al., 2012). Complementary to these approaches, some researchers have employed circular buffer structures to efficiently store subsequences, thereby reducing memory overhead (Rakthanmanon et al., 2012; Giao and Anh, 2016).

In addition, data normalisation has been identified as a crucial step for precise similarity search. The possible problem lies in additional computational and memory costs. Normalisation is often applied per window in streaming contexts or optimised to minimise performance penalties (Rakthanmanon et al., 2012; Giao and Anh, 2016; Liang et al., 2021). Gathering large amounts of network flow from heterogeneous network points is a problem. Various open-source stream processing solutions can be used to collect network flow data in a centralised environment. For example, the widely used Apache Kafka platform (Komisarek et al., 2021; Kumar et al., 2025), or another less popular but applicable solution for amassing network flow, is the open-source message broker RabbitMQ (Kotov et al., 2024; Maatkamp et al., 2016). The latter might be preferred owing to its high throughput, scalability, and the ability to filter and order incoming network traffic data (Dobbelaere and Esmaili, 2017; Jones et al., 2011).

Adding such solutions using different ML models enables keeping up with network traffic streams. However, the application of ML to constantly evolving network flow streams is relatively new compared with passive analysis methods that have been around for some time (Mulinka and Casas, 2018; Ulmer et al., 2019; Rivera et al., 2021). In addition, in terms of sophisticated cyber-attacks, ML models perform worse than expected. Currently, attackers can poison models with data poisoning. This leads to heavily unreliable ML models (Khan and Ghafoor, 2024). Even small amounts of poisoned data can reduce accuracy and cause misclassification (Kure et al., 2025), making covert attacks difficult to detect while threatening critical systems and privacy (Maramreddy and Muppavaram, 2024). Such attacks, like label manipulation, data injection, or backdoors, undermine reliability in safety-critical domains and may even expose sensitive training data (Srivastava et al., 2024).

These studies collectively highlight significant gaps in current algorithms and methods for detecting cyber-attacks in asynchronous network flows. To the best of our knowledge, none of the prior studies mentioned explicitly addressed asynchronous evidence alignment in large network flows for a sophisticated cyber-attack assessment.

3 Proposed Model

In this section, we present a formal model for asynchronous network flow analysis, enabling approximate time-similar feature correlation to overcome the limitations of traditional synchronous methods. The assumption underlying the model's establishment is that time is not globally correlated and that the recorded time series are not homogeneous, implying that the time intervals of each data source are asynchronous.

3.1 Problem Formulation

We designed a model of the dynamic behaviour of network data flow to ease the detection of sophisticated cyber-attacks in recorded network data, focusing on asynchronous data flows between attacker and target machines in a cybersecurity context.

Definition 1 (Data Sources). Asynchronous data sources from heterogeneous points are presented as follows

$$\mathcal{D}_t = \{d^1, d^2, \ldots, d^n\}_t \tag{1}$$

where n is an arbitrary number of data sources that can be observed at a synchronised time t.

All asynchronous data sources observing the network are defined as $\mathcal{D}^*$ over an unspecified time interval. Two data sources yield uncommon data density in time as well as time floatation. d_t^1 and $d_{t'}^2$ where $d_t^1, d_{t'}^2 \in \mathcal{D}^*$ satisfying the condition that $t \neq t'$. Time t is a timestamp used to correlate local data occurrence sequentially, but is not considered as globally correlated time.

Definition 2 (Data Stream). We define $\mathcal{S}_t^d$ as a stream of network-observable packets sent from the data source, which is an actor's machine. The machine may be a threat actor, the target actor's machines, or any intermediate proxy. S stream has k packets of network data source $d = d_t^i \in \mathcal{D}_t$. Then,

$$\mathcal{S}_t^d = \langle P_1, P_2, \ldots, P_k \rangle_t^d \tag{2}$$

is a sequence of packets, where k is the number of observations in the data stream.

If the data stream is continuous in time, we denote the data stream as $\mathcal{S}_{[t;\infty]}^d$, and the source as $d = d_{[t;\infty]}^i \in \mathcal{D}^*$ The current time of the continuous data stream will be called now referred to the current state of time.

Given the data stream $\mathcal{S}_t^d$, a unique property is defined for each packet P_t^d. We denote $\mathcal{F} = \langle f_1, f_2, \ldots, f_m \rangle$ as packet properties, where f is the structural element of P_t^d. To simplify the notation, we use $P_t^d.f$ to show the relationship between the packet and a specific property.

Definition 3 (Packet Property Vectors). Any packet $P_t^d \in S_t^d$ is associated with a finite sequence of property value vectors according to properties f. We denote these property value vectors by:

$$\mathcal{V}(P_t^i.f) = \langle p_t^i | p_t^i \leftarrow P_t^i.f \wedge P_t^i \in S_t^i \rangle, \tag{3}$$

where j is the number of values of packet properties in stream S_t^d and p_t^d is the packet value of property f of $\mathcal{F}$.

To shorten the notation, we use $\mathcal{V}_t^i(f)$ to denote the packet value vector according to property f.

We define a packet vector filter for data packets based on their properties. The packet filter is a function $\mathbb{F} : \mathcal{V} \times \mathcal{F} \rightarrow \mathcal{V}$ that distinguishes packets according to the criteria from the data stream S_t^d.

Definition 4 (Packet Filter). Let ϕ be a defined property of a packet. We define the packet filter $\mathbb{F}$ as a function

$$\mathbb{F}\big(\ \mathcal{V}_t^d(f),\ \phi\ \big) = \langle p_t^d | p_t^d \in \mathcal{V}_t^d(f) \wedge f = \phi \rangle \tag{4}$$

Here, $\mathcal{V}_t^d \in \mathcal{S}_t^d$ is the filtered data stream that contains only packets whose property vectors satisfy ϕ.

Such filters distinguish packets in the stream according to the selected properties with a time complexity $\Theta(n)$, where n is the number of packets in the stream.

Due to the nature of data streams to be continuous, the ring buffer RBW^n is defined as an abstract data structure that is a fixed-size ring buffer for $\mathcal{S}_t^d$. The RBW is a data structure queue that operates enqueue and dequeue with time complexity $\Theta(1)$. Therefore, the alignment process is to compare buffered data with the specific parameters located, ignoring other filtered-out records.

Definition 5 (Threshold for Aligned Time Vector Window). We define a threshold ω and bins $\mathcal{K}$. The threshold ω is a limiter for the $\mathcal{S}_t^d$ if $\mathcal{S}_t^d$ exceeds ω amount of vectors. Then

$$\mathcal{V}(P_t^i) \in \mathcal{M}_{[t_s,t_e]} > \omega \tag{5}$$

We define a transformation $\mathcal{T}()$ for $\mathcal{V}(P_t^i) \in \mathcal{M}_{[t_s,t_e]}$ where,

$$\mathcal{T}(\mathcal{V}(P_t^i)) \in \mathcal{K} \tag{6}$$

then,

$$\mathcal{M}_{[t_s,t_e]} = \left\langle\ \mathcal{T}(\mathcal{V}(P_t^i))\ \middle|\ \mathcal{M}_{[t_s,t_e]} > \omega, t \in [t_s,t_e] \right\rangle \tag{7}$$

is a vector of filtered packets according to the threshold.

To simplify notation we use $\mathcal{M}_{[t_s,t_e]}$ as $\mathcal{M}$ in $[t_s,t_e]$.

To align within the threshold ω any given $\mathcal{M}'$ and $\mathcal{M}''$ we define alignment operator $\bowtie$: $\mathcal{V} \times \mathcal{V} \times \omega \Rightarrow \mathcal{V}$ aligns two independent data vectors,

$$\begin{aligned}\mathcal{M}' \bowtie_\omega \mathcal{M}'' ::= \big\langle P_{t'}^{i'}, P_{t''}^{i''} |\ \ & \forall t', t'' \exists P_{t'}^{i'} \in \mathcal{V}' \exists P_{t''}^{i''} \in \mathcal{V}'' \wedge \omega \in \mathbb{N} \wedge \\ & (\mathbb{F}\big(\ {\mathcal{V}'}_t^d(f),\ \phi\ \big) \sqcap \mathbb{F}\big(\ {\mathcal{V}''}_t^d(f),\ \phi\ \big)) \neq [] \big\rangle\end{aligned}$$

respecting the filter ϕ.

The proposed concepts play a major role in the processing and analysis of asynchronous network flows in PCAP files. Table 1 summarises the main notations used in the provided definitions.

Table 1: **Summary of the Main Notations.**

Notation	Explanation
$\mathcal{S}_t^d$	Data Source
RBW	Ring Buffer Window
$d_t^1, d_{t'}^2 \in \mathcal{D}^*$	Two data sources from different times
$t \neq t'$	Asynchronous time between two data sources
$d = d_t^i \in \mathcal{D}_t$	Data source is a part of data stream
$d = d_{[t;\infty]}^i \in \mathcal{D}^*$	Synchronized data streams in time
$\mathcal{F}$	Packet properties
$\mathcal{V}(P_t^i.f)$	Packet Property Vectors
$\mathbb{F}\Big(\ \mathcal{V}_t^d(f),\ \phi\ \Big)$	Packet filter
$\mathcal{M}_{[t_s,t_e]}$	Aligned Time Vector Window with start and end times
$\mathcal{V}_t^d \in \mathcal{S}_t^d$	Filtered Data

3.2 Proposed Model Architecture

DTW functionality constraints limit the flexibility of stream processing. Given a pair of sequences, DTW stores them in a matrix in computer RAM without any constraints or optimisation for later computation (Krinickij and Bukauskas, 2024). Thus, to further this process, we present a similarity algorithm that allows asynchronous data alignment. We present pseudo-code algorithm 1 of the data stream similarity that integrates all the mentioned concepts to process asynchronous network flows for incident detection in more detail. The algorithm maps events by analysing similarities within data streams using RBW and $\mathcal{M}_{[t_s,t_e]}$ windows, thereby aligning historical events across asynchronous PCAP files.

The required input consolidates the arguments required for an initial program launch. We ask for the input data stream, the packet property, and its vector, which serves as a beacon between the two data streams for similarity alignment. The threshold and bins parameters are the verification parameters required to verify that the beacon vector amount gathered between the two data streams does not exceed the specified threshold in line 1.

From lines 2 to 12, we load the data streams in parallel, and for all packets in data streams, we are looking for packet property vectors that were defined in the input. The filtering process of the vectors is performed inside the RBW, whose size is also specified during the input. After vectors are identified, we extract the corresponding time intervals for all matched vectors in line 13.

Lines 7 to 10 show a ring buffer window, which is a constant window that does not recreate itself. Both streams have RBW as the primary sliding window of fixed size, which slices the given stream into RBW and moves forward vertically. The RBW have many intrinsic properties that are used over our recorded synthetic PCAP files. In addition, PCAP files from different internet sources are added to enhance and correlate the experimental results. Gathered cyber-attack evidence, per the simulation, realistically,

Algorithm 1 Data Stream Similarity Algorithm

Input: $\mathcal{S}_t^d$, f, $\mathcal{V}(P_t^i)$, RBW, threshold Ω, bins $\mathcal{K}$
for all $\mathcal{S}_t^d$ **in parallel do**
  for all packet p in $\mathcal{S}_t^d$ **do**
    $match(\text{RBW}(\mathbb{F}(p_t^d)))$
    extract p_t^d
    if $bytes \leq \text{RBW}$ **then**
      enqueue to RBW $\leftarrow bytes$
    else
      dequeue RBW $bytes \leftarrow 0$
    end if
  end for
end for
$all\mathcal{V} \leftarrow \cup match(\text{RBW}(\mathbb{F}(p_t^d)))$ **if empty then return**
$alignedTS \leftarrow ts \leftarrow \min(all\mathcal{V}), te \leftarrow \max(all\mathcal{V})$
$M_{[t_s,t_e]} \leftarrow \{p_t^d \in alignedTS \mid ts \leq t \leq te\}$
if $useBinning$ **then**
  $seq \leftarrow \text{Histogram}(M_{[t_s,t_e]}, \Omega, \mathcal{K})$
end if
$dist \leftarrow \text{DTW}(s_1, s_2)$
return $dist, t_{E2E}$

synthetic attack templates are already present in the computational procedures for the experiment.

Using the firs in first out (FIFO) principle, the ring buffer stores only the most recent packets from data streams. Fig. 1 shows the process flow between two data streams of different lengths. The window is closed if the capacity is reached, and all other packets are discarded.

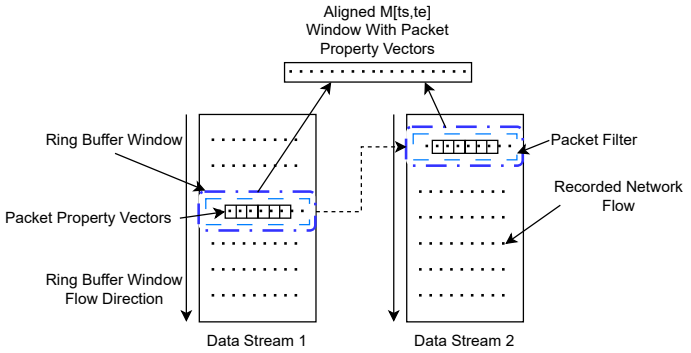


Fig. 1: Data Streams From Data Sources With Ring Buffer Windows for Asynchronous Alignment of Network Streams in Recorded PCAP Files

If the critical threshold is reached, we use the binning technique to reduce the number of beacon vectors (Knuth, 2006) in $\mathcal{M}_{[t_s,t_e]}$ to bins in the sequence. The critical threshold corresponds to the number of vectors found in the $\mathcal{M}_{[t_s,t_e]}$ window. The $\mathcal{M}_{[t_s,t_e]}$ window time intervals were divided into a fixed number of bins of equal width. The number of vector timestamps is counted and falls into each bin based on the amount presented. This way, we obtain a short digital sequence of fixed length from a very long sequence of vector timestamps. The technique involves grouping many separate values into a fixed amount of intervals and summing the number of values that fall into each bin. This is shown lines 16 to line 17.

Line 14 presents all the vector timestamps gathered. We then align the gathered vectors in line 15. Line 19 depicts the sequences that are a part of $\mathcal{M}_{[t_s,t_e]}$ and given to DTW distance calculation. Line 20 returns the distance calculation of DTW and the end-to-end (E2E) time required for computations to be performed. The E2E time calculation is a performance indicator that provides a quantitative comparison between our approach and a classical DTW computation on the same sequences.

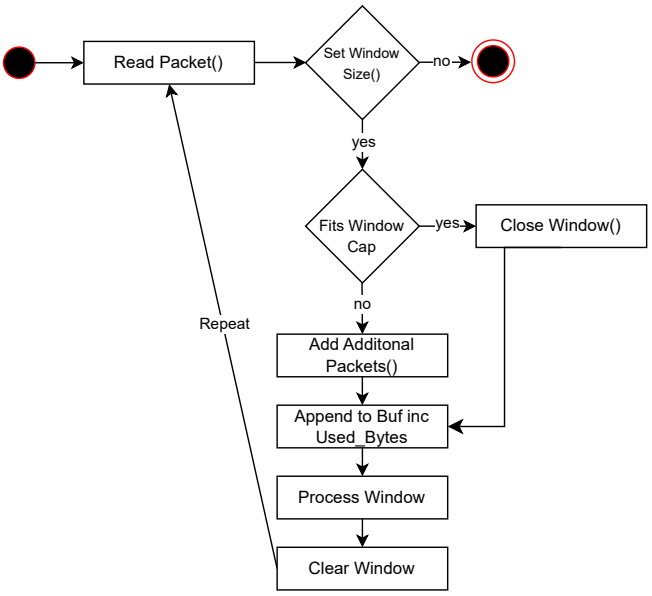


Fig. 2: Representation of Data Stream Flow in the Ring Buffer Window

Fig. 2 shows the detailed flow of the RBW. Packets are added to the window until the end of the window in the data stream. We checked whether the designated buffer window size had reached its designated capacity. If yes, we process the window, clear it afterwards, and add new packets for the second iteration.

In Fig. 3, we show the data flow in the proposed similarity algorithm. If the number

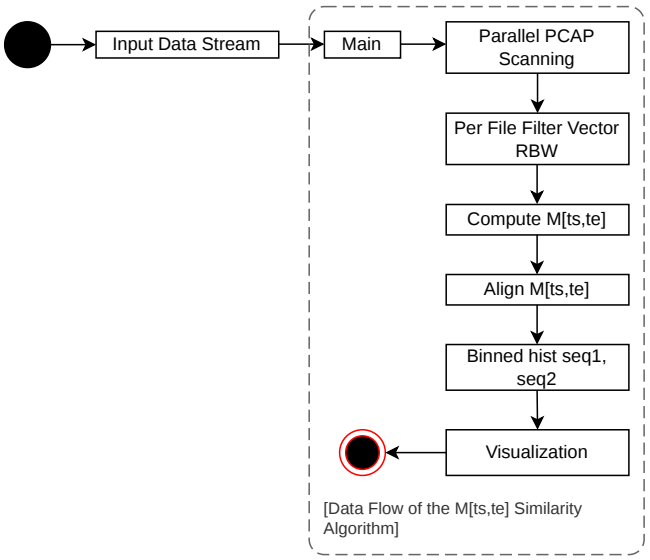


Fig. 3: Stream Data Flow of the Similarity Algorithm From the Parameter Input Until Result Visualisation

of vectors in the data streams is less than a predefined threshold, we continue computing and present the results. The step before the last measures time and sequences in quantity, and in the last step, we present a visual representation of the proposed model results. We also calculate the E2E time to determine the algorithm's runtime from the moment the parameter entry is complete until DTW completes the computations. The experimental visualisation part did not include E2E time monitoring.

The intrinsic properties of the windows in our algorithm give utilization to accelerate DTW functions. These properties are:

1. *Fixed window length.* Because of the limited sequence length, the DTW matrix did not exceed the specified limits.
2. *Fixed byte limit per window.* Windows were not overloaded with large amounts of data.
3. *Window transitions.* We obtained dense but manageable windows, rather than irregular, large chunks of data.
4. *Vector Filtering.* Instead of a $n \times m$ "all in one" matrix, DTW is performed in places where it makes sense to do so.

5. *Window updating.* When the RBW window moves through the data stream, the already filtered, outdated packets are removed from the window without reloading it. This creates a lower cost for the candidate vectors to be added to $\mathcal{M}_{[t_s,t_e]}$.

4 Implementation

In this section, we present several synthetically generated network flow cases with cyber-attack templates. The gathered network flow cases have different parameters and network behaviours, each with its own characteristics, which were recorded in PCAP files for further presentation to our model.

We simulate synthetic network flows in a laboratory environment. For this purpose, specific, sophisticated and known attack templates were chosen to simulate network traffic. The synthetic attack traffic is embedded within background benign traffic.

The sophisticated attack template consists of command-and-control (C2) threat actor virtual machines and target virtual machines, with malware and its beacons deployed on the target machines (Gardiner et al., 2014). The second attack consists of a standard network-mapping (Nmap) port scan, and the third attack is a specific TCP SYN ACK/RST flag-related attack. All the virtual machines for the experimental cases were part of the `Proxmox` virtual environment created in our laboratory (Oleksiuk and Oleksiuk, 2021). `Proxmox` supports different types of hypervisors. These hypervisors allowed us to create virtual machines for our experimental bedrock. For network flow collection across different virtual machines, we use a centralised system with `RabbitMQ` as the broker, which is also a virtual machine in the `Proxmox` environment. `RabbitMQ` has distributed network monitors deployed on every virtual machine in the network. These network monitors send network flow data to our broker, which stores it in PCAP files.

For the first attack case, the network setup in the `Proxmox` environment consisted of 9 virtual machines: 4 C2 servers and 5 infected clients. In addition to beacons, five infected machines have different types of malware installed. Overall, four simulated teams, each with two to three machines, were distributed across our laboratory. The result was 18 PCAP files, ranging from 5 to 36 megabytes, over two iteration periods while gathering network flow. C2 activity and other activities in the network were completed over a 2.5-hour period, during 2 recording sessions. All network flows were gathered from both actor perspectives. All network traffic was routed via an unencrypted channel.

Different attributes and vectors were chosen from the first attack case, which identified beacon vectors in the PCAP files. These attributes are dispersed between the PCAP files and represent different C2 beacon implementations. Table 2 lists different attributes with their vectors and protocols that have been recorded in the PCAP files. The attribute names follow the standard naming convention used by `Wireshark`, a network protocol analyser software.

All recorded PCAP files, based on PCAP attribute vectors, have distinct numbers of packets identified as heartbeat packets. These heartbeats corresponded to the beacon vectors recorded in the PCAP file.

Table 2: Beacon Attribute and Their Values in PCAP Files With Volumetric Statistics

Protocol	Attribute	Beacon Vector	Total Amount	Mean t [s]	Median t [s]
HTTP	http.request.uri	/beacon	122	55.50	0
HTTP	text	//4A	59	52.10	3.06
TLS 1.3	tls.handshake.ja3	78f0dc5...	1484	7.36	0.05
TCP	tcp.dstport	8888	4282	3.25	1.02
TCP	tcp.dstport	5000	4109	3.90	0
TCP	tcp.flags	SYN	9501	1.72	0.02

Every heartbeat-type communication observed in this study was condensed into six rows. Each row corresponding to a distinct beacon heartbeat amount differs across C2 implementations on different machines. For each distinct type, we report the total number of heartbeats. In addition, we provide two additional volumetric indicators. Average and the median inter-arrival times between consecutive heartbeats.

Every beacon heartbeat shown in the table can be characterised according to different protocols and specific measurement counts. These metrics are sufficient to determine several different behaviours of C2 channels and their respective environments in the network stream.

In some cases, the median is zero. This could occur because a beacon sends multiple packets at once, and both are recorded at almost the same timestamp. This implies that the network flow is gathered too densely. In our experiments, we did not change the vector density.

The second and third attack cases require only 2 virtual machines for the experiments. That is a threat actor and target machines. Both cases have network flow recording times of 5-10 minutes from a heterogeneous perspective, and the PCAP file sizes are 5-15 megabytes.

In the first case, the gathered network traffic spans a larger time frame than in the second or third cases. Therefore, in our study, we also use large network traffic PCAP files recorded at different times and from various sources on the internet as the fourth case. Specifically for the fourth case experiments, we use Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC) datasets, which are public PCAP files captured and published at network research vendor (NETRESEC) (NETRESEC, 2015).

Overall, network capacity in our simulated network environment for the recorded cases two and three is lower than the standard I/O capacity due to firewall rules and other policies of our institution. Otherwise, the network flow data gathered from our laboratory's setup exhibits distinct behaviour during simulated cyber-attacks. In the first case, internet traffic was generated to simulate a scenario in which malware is downloaded to the virtual machines, and beacon activity was generated between the threat actor and the target machines. The beacons listed in Table 2 have different spark times in PCAP files, suggesting possible heartbeat activity.

Also, in Table 3, beacon values define a specific place in the packet structure where they appear. These attributes, with their values, represent network flow characteristics extracted at different protocol layers. For transport level characteristics

`tcp.dstport`, `tcp.flags` that represent connection behaviour, service targeting and session state. These fields are explicit in the packet structure. Application-level characteristics, such as `http.request.uri`, are only visible in decrypted traffic and can be interpreted as a requested resource. Cryptographic characteristic `tls.handshake.ja3` is a client fingerprint characteristic that is not explicitly sent, which can be considered as a sophisticated attack pattern or an anomaly. The `text` is a heuristic characteristic that indicates a human-readable payload. These packet characteristics are observable attributes of beacons extracted from different protocol layers and describe the communication behaviour in the PCAP files.

Table 3: Beacon Attribute Values in Packet Structure Placement

Place in Packet Structure	**Attribute**	**Beacon Vector**
TCP Payload (HTTP Request)	`http.request.uri`	`/beacon`
TCP Payload (HTTP Request)	`text`	`//4A`
TCP Payload (TLS ClientHello)	`tls.handshake.ja3`	`78f0dc5...`
TCP Header	`tcp.dstport`	`8888`
TCP Header	`tcp.dstport`	`5000`
TCP Header	`tcp.flags`	`SYN`

The second and third simulated attack templates generate relatively static flows that mimic usual communication between network machines based on internet protocol logic. In the second case, the port scan activity is characterised by a high number of short-lived flows originating from a single source and targeting multiple destination ports. The third attack template generates a specific number of flags that could be considered beacons for malicious activity between virtual machines. The second and third attack templates also use TCP header flags and destination ports as primary sources of identification within the packet structure. Other synthetic attack characteristics include asymmetric packet distribution (case two), elevated connection rates in cases one and two, and short-lived flows in cases one and three.

The external PCAP files lack documentation of specific traffic patterns. Consequently, our model is designed to be incident-agnostic. The model does not rely on knowledge of events or attack templates used in the network flow. Instead, our proposed model searches for similar patterns in heterogeneous flows and aligns them by identifying and aligning vectors derived from the surrounding noise in the network flow. For this reason, our prototype presents a list of possible attributes and vectors as input to the proposed model implementation. This enrichment enables us to adopt a more flexible approach to network flows and better assess the proposed model's validity.

5 Experiments and Results

In this section, we present the experimental results obtained when implementing the proposed model. Overall, experiments were performed using 20 data sets across four

cases. Experiments were performed using different input parameters. In addition, different RBW sizes were used to process the datasets, which were also a part of the input parameters. The RBW sizes per example were $16, 32, 64$, and 128 KB. In the presented results, we show a correlation between classical DTW and our model in terms of the computational time required for both sequences. The sequence lengths are shown on the x-axis, and the program runtime is shown on the y-axis.

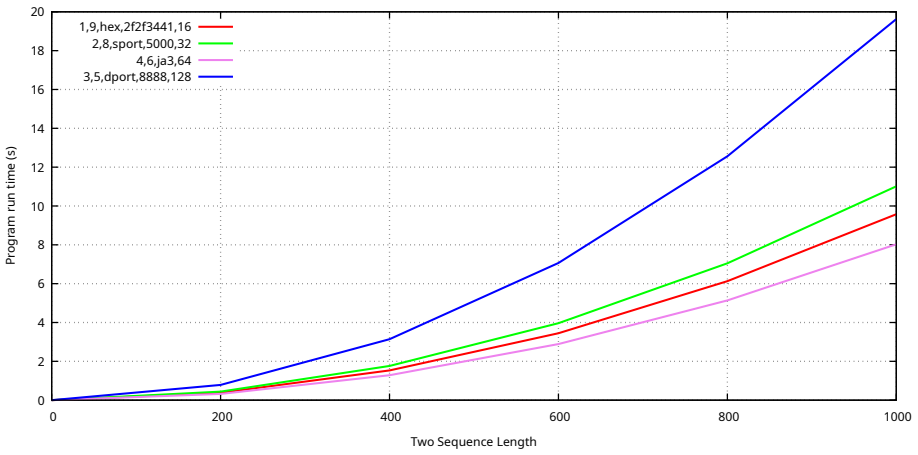
All experiments were conducted on a physical computer that had an AMD Ryzen 9 processor, 64GB of RAM, and Linux Mint 22.1 LTS at the time of the experiments. All of the experiments had a conditional number of input parameters chosen:

- PCAP file selection.
- Packet property.
- Packet property vector.
- Different size of RBW. Which are $16, 32, 64$, and 128 KB in size, respectively.
- Bins count.
- Packet property vector threshold.

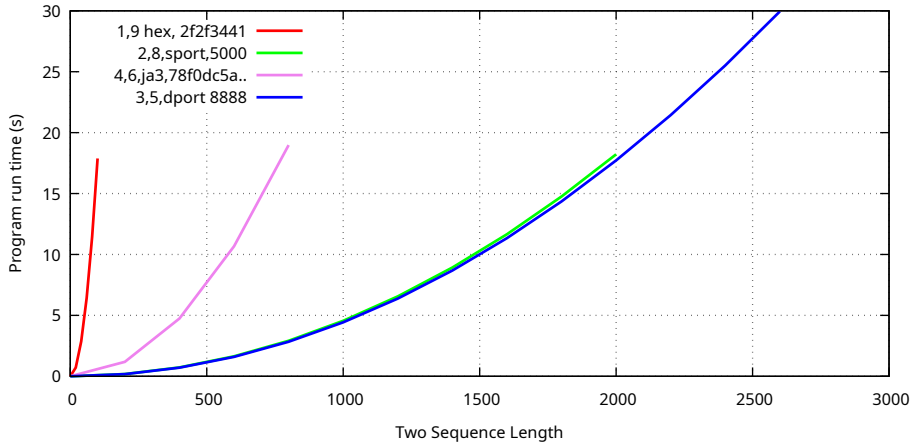
Fig. 4a shows the results from case one, experiment A, where we used the PCAP files that were recorded in our laboratory's network environment. The first two numbers in the figures legend are the PCAP files from the recorded sets we chose based on the C2 environment in the network for the experiments, where one PCAP file is the C2 attacker actor server machine, and the other is the target machine. The third parameter corresponds to the packet property, and the fourth is the packet property vector. The last number in the legend represents the RBW size. The sequence length does not exceed 1000, which is the sum of the predefined bin count of 500 per sequence. Depending on the number of found sequences, PCAP binning was applied to every experiment iteration. Different RBW sizes had an impact. The smaller the RBW size, the faster the computational time. The other parameters did not have a significant effect on the vector amount. The fixed amount of RBW in the last iteration, which measured 128 in size, had the highest computational time of 20 seconds.

For each PCAP file, the vector timestamps were prepared as constructed sequences of fixed-length histograms from the aligned time window $\mathcal{M}_{[t_s,t_e]}$. This reduces the sequences and allows DTW to be applied afterwards. Using binning allows DTW to not count millions of possible vectors. The vector values were compressed into 500 bins per sequence. Then the DTW distance is processed for both sequences every 500 bins.

This allows us to work with the binned values rather than with every vector, which could be a sequence of counting in millions. The DTW distance calculates the minimum alignment distance between sequences, but does not return the full alignment or the entire matrix. Distance measurement is a good benchmark method that returns a single value, the optimal DTW distance between two sequences, for a vector (Meert et al., 2020). The distance measurement showed indifference to the accumulation of sequences. In addition, this method is the most cost-efficient compared to a full warping path or sequence alignment due to lower computational resource requirements. The distance measurement allows the vectors to be pulled slightly towards each other and aligned in time. This measurement allows us to evaluate the dynamics in form, that is whether the vectors rhythm over $\mathcal{M}_{[t_s,t_e]}$ changes similarly over time.



(a)



(b)

Fig. 4: Case One, Experiment A: Results from PCAP Files Recorded in a Controlled Network Environment for the Proposed Model With Different RBW Sizes and for Classical DTW Distance Measurement.

The binning technique aggregates all the vectors in ts, te into 500 slots. After this step, both sequences had a length of 500 bins. Then, the DTW distance computes a 500×500 matrix instead of the $all \times all$ matrix from both sequences, even if the sequences are filtered.

Then DTW measures not the individual packet distances but the similarity of two time profiles of aligned $\mathcal{M}_{[t_s,t_e]}$. We evaluated the macro-dynamics of the alignment, how much, and when the activity was spotted. It is possible that millions of vectors are influenced by the number of bins, and DTW works with this aggregated data, making it

fast and meaningful for measuring macro-similarity. For DTW distance measurement, we used a Python library (Meert et al., 2020).

Fig. 4b in case one, experiment A had the same input parameters as for our model for every iteration. Here, we use classical DTW with our generated PCAP files in a controlled network environment. We can see the number of vectors found per experimental iteration and the time required to process them. The vector amount between the two sequences is the same as in the experiment with our proposed model in Fig. 4a, but the computational time required is higher.

Table 4 shows the input parameters for PCAP files that were recorded with synthetic cyber-attacks in our laboratory's network environment.

Table 4: Case One, Experiment A Input Parameters Chosen for the PCAP Files That Were Generated in Our Laboratories Controlled Network Environment

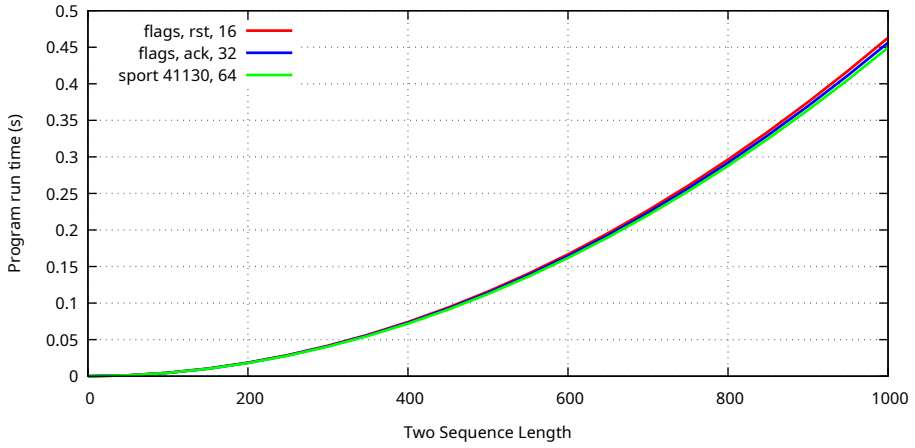
PCAP Files	**Attribute**	**Beacon Vector**	RBW **Size**
2 PCAP	`hex`	`2f2f3441`	16KB
2 PCAP	`Source Port`	`5000`	32KB
2 PCAP	`Ja3`	`78f0dc5...`	64KB
2 PCAP	`Destination Port`	`8888`	128KB

In case two, experiment B, there is a noticeable decrease in computational time required by our approach and the classical DTW implementation. In this experiment, the amount of packets is slightly higher than in case one, but the effect of time reduction for computational time is still evident. Also, the runtime in Fig. 5a and Fig. 5c increases linearly. Furthermore, comparing the classical DTW to our model's result curves, which nearly coincide over the same time needed for computations, suggests that the RBW dependency has no overhead and that vector density per-bin count has minimal effect under the provided parameters.

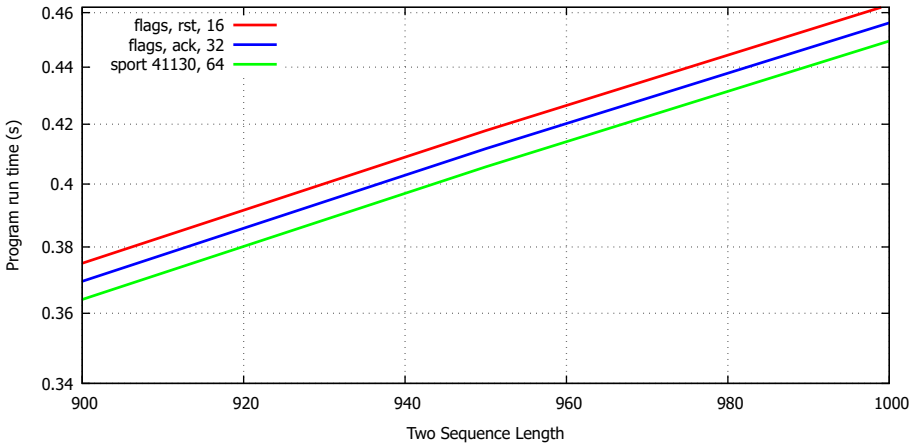
In Fig. 5b, we show the scaled representation of case two, experiment B to visualise the proximity of computational time. The results based on vector count suggest that the time required to perform computations was nearly the same, with a very small offset per iteration.

Across the entire range of the classical DTW in Fig. 5c, the flags RST vector alignment incurs the highest time, the source port is intermediate, and the flag ACK alignment is the lowest for the classical DTW. The bigger the amount of vectors found between the PCAP files, the higher the time for computations needed.

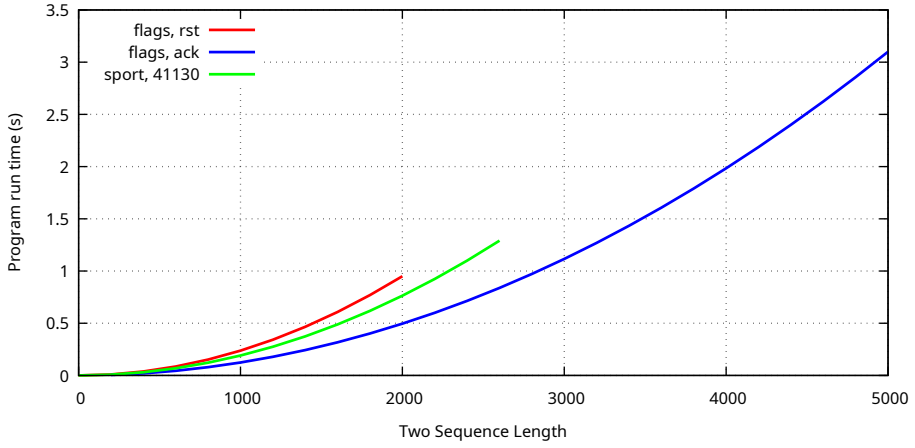
For case three, experiment C in Fig. 6a and classical DTW in Fig. 6c based on Nmap port scan in the PCAP files, it is evident that mostly two specific attributes and their vectors are suitable for checking. That is the SYN and RST flags from both threat actor and target actor perspectives. The overall selection of the parameters in the provided Fig. 6a of our model has the same RBW rule dependency, and vector density does not affect the overall performance. At the same time, the amount needed to process the vectors is dramatically different in comparison.



(a)



(b)



(c)

Fig. 5: Case Two, Experiment B: Results from PCAP Files Recorded in a Controlled Network Environment for a Specific Flag-Related Attack Scenario, including the proposed model with different RBW sizes, scaled results, and classical DTW distance measurement.

In Fig. 6b, a scaled version of the presented case three, experiment C representation is shown. The computation time here is also similar, but based on the RBW size, we can determine that the time required to efficiently process the vectors is determined by the minimised RBW value.

For both Fig. 6a and Fig. 6c the curves nearly coincide over the evaluated ranges. This is evident because the number of flags is nearly the same in both PCAP files, which implies that the attacker sends a specific number of packets to test the ports and receives a specific number of denied RST packets for the scanned ports.

Case four, experiment D used compressed PCAP files gathered from NETRESEC (NETRESEC, 2015). The compressed file size ranged from 300 to 400 MB, and the total packet count was millions. Specifically, the average number of packets in the compressed PCAP files was 10 million. As shown in Fig. 7a, we used different RBW window sizes to determine the computational efficiency of the proposed model. The two sequence lengths on the x-axis are the predefined bin counts, each set to 500. The bin count per sequence was not changed during the experiments. Because the packet count exceeded the threshold amount, Fig. 7a shows the length of both sequences and the time it takes for the experiment to run. Time monitoring began immediately after parameter input and ended before the figure of the experimental results was created. This means that our model's time is monitored throughout all phases of computation.

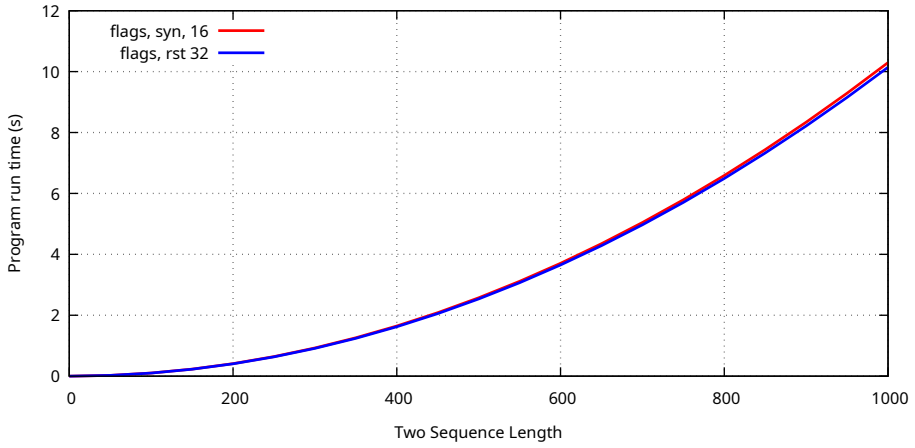
Based on the number of vectors, which varied across experimental iterations, the computation time ranged from 1,000 to 1,400 seconds for 10 million records per file. The computational time increased with the vector count. In Fig. 7a, experiments with effectiveness time are shown. The vector amount ranged from 1000 to 4.5 million per sequence. A slight difference in computational time is observed for RBW, but the major difference lies in the vector count. Table 5 lists the input parameters chosen for case four, experiment D.

Table 5: Case Four, Experiment D Input Parameters Chosen for Extensive PCAP Files From the Internet

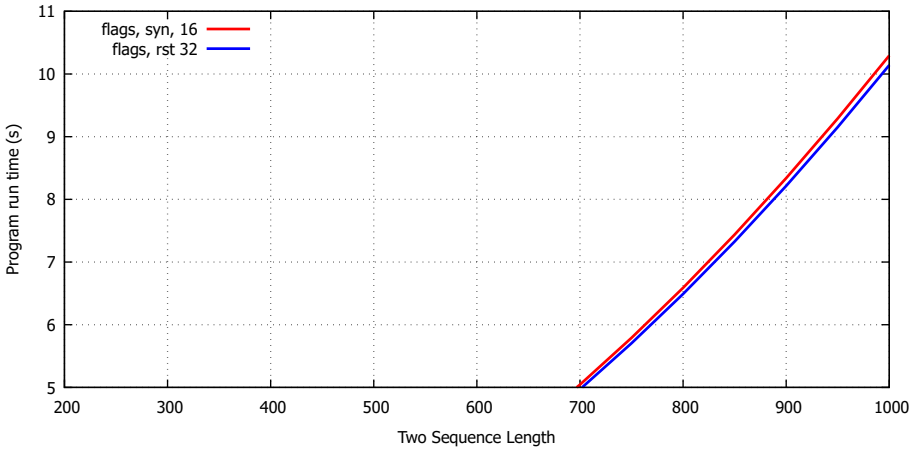
PCAP Files	**Attribute**	**Beacon Vector**	**RBW Size**
2 PCAP.gz	`Source Port`	`443`	16KB
2 PCAP.gz	`TCP Flag`	`SYN`	32KB
2 PCAP.gz	`TCP Flag`	`PSH`	64KB
2 PCAP.gz	`Destination Port`	`22`	128KB

Case four, experiment D, as shown in Fig. 7b, had the same input parameters for every iteration. Only two of the four iterations were feasible due to the massive vector count per experiment. The time and amount required to perform the computations exceed those of the previous experiments by a significant margin. The number of vectors was also extensive.

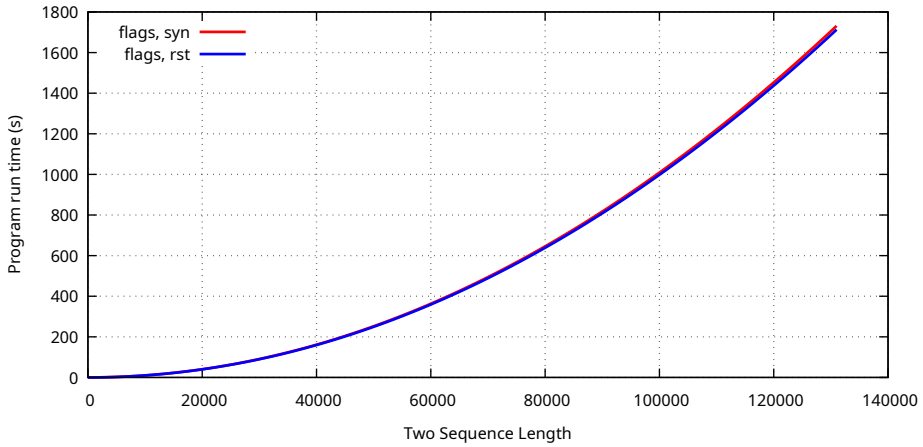
In every experiment, the time increased almost linearly along the x-axis. The indicated linear scaling in the graphs under the provided parametrised conditions showed no quadratic behaviour.



(a)

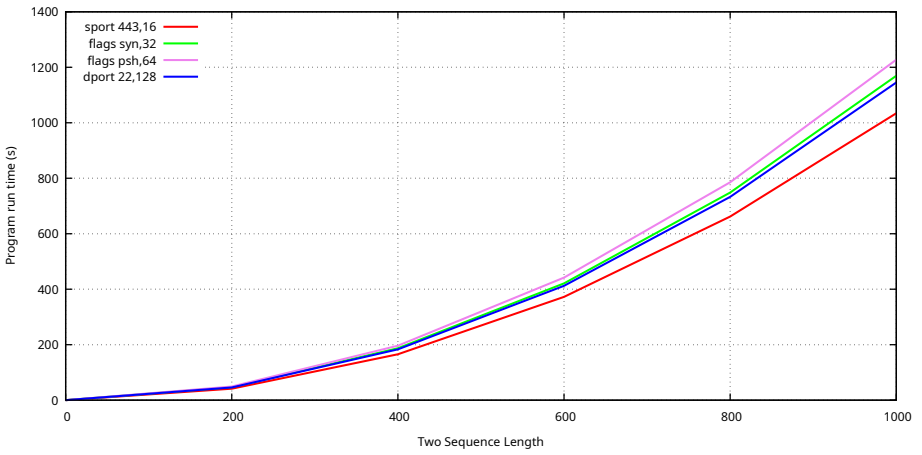


(b)

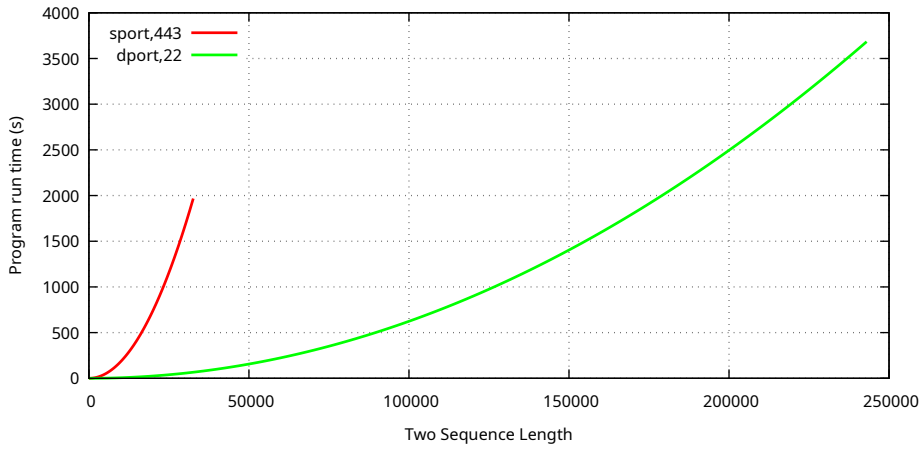


(c)

Fig. 6: Case Three, Experiment C: Results from PCAP Files Recorded in a Controlled Network Environment for the Nmap Port Scan Attack Scenario, including the proposed model With different RBW sizes, scaled results, and classical DTW distance measurement.



(a)



(b)

Fig. 7: Case Four, Experiment D: Results from Extensive Compressed PCAP Files from the Internet for the Proposed Model With Different RBW Sizes and for Classical DTW Distance Measurement.

In Fig. 8, we present a combined representation from the extensive PCAP files, which shows our model and the classical DTW time efficiency. The representation of our model using the binning technique will never exceed the threshold constant compared with the classical model. This was particularly evident in the combined efficiency of the two models.

From DTW perspective we get a 500×500 matrix not depending on the sequence lengths in aligned $\mathcal{M}_{[t_s,t_e]}$. For classical DTW without window properties, we observed a significant increase in time across the provided experimental iterations. Classical

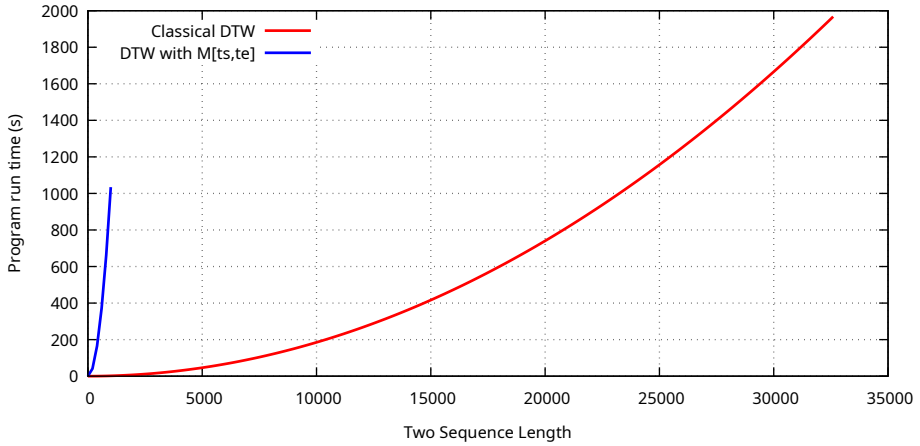


Fig. 8: Combined Results of Classical DTW and Our Proposed Model With Extensive PCAP Files From the Internet

DTW can match our model's computational time when the number of vectors is small. Our model's computational time did not exceed a significant margin, whereas classical DTW across iterations took much longer. The DTW window properties showed a significant reduction in computational time. Adding window properties before the DTW computation minimises the computation time.

In summary, the experimental results demonstrate that the proposed pipeline can systematically extract event-based network-flow vectors, align and transform them into temporal representations, and compare traffic patterns using sequence-alignment methods.

6 Discussion

Data filtering is an important process that must be addressed. For PCAP files that can accumulate large amounts of data quickly, we must eliminate as many unnecessary parameters as possible. Subsequently, data normalisation is also required to determine even greater computational capabilities. This results in the application of a specific technique that requires additional knowledge and experimentation. The gathered results indicate that the proposed model in its vector extraction pipeline can process network-flow data in a structured and repeatable manner. The observed runtime behaviour suggests that the computational cost is largely influenced by sequence length, binning configuration, and the selected matching strategy.

DTW can align similar behaviours even when they are shifted in time. Taking DTW as a baseline for comparing traffic patterns with our model is useful for C2-like and other communication, where beaconing, repeated, or other actions may not occur at identical timestamps. Compared with existing DTW variants, the proposed model should be positioned less as a new DTW algorithm and more as a network-flow

analysis pipeline that uses DTW as an interpretable baseline for pattern matching. The proposed approach is designed for continuous heterogeneous network traffic, where the main contribution is not only the alignment itself, but also the extraction of flow-context vectors such as flags, ports, JA3 indicators and others. Therefore, DTW variants such as FastDTW, UCR-DTW, SUCR-DTW, ODTW, EventDTW, SparseDTW, SSDTW, and FlexDTW can be treated as computational or methodological baselines, while the proposed model extends the problem toward context-aware streaming cyber-attack
detection.

DTW has a quadratic complexity. Extensive data flows that were minimised make DTW more stable and cheap, even when $\mathcal{M}_{[t_s,t_e]}$ window has hundreds of thousands of vectors aligned. The downside is that we can lose alignment owing to a small bin count per sequence. This will achieve only a macro-dynamical evaluation of the cyber-attack pattern over time. For micro-dynamical analysis in alignment, we could utilise the bin count, which should be dynamic, or use other patterns to evaluate the clarity of the patterns. For small vector sets, the classical DTW algorithm is sufficient for detecting cyber-attacks. For sophisticated cyber-attacks and extensive network flows, especially in today's network environments, DTW is slow to implement.

Limitations

For network flow gathering, we used a centralised approach with a network broker. This may become a single point of failure if the centralised system breaks without any backup. Furthermore, a significant amount of physical disk space is required to accommodate consistent network flow to the network broker provider's machine. Fast disk replacement or attachment is not possible due to additional configuration requirements and costs. Incoming network traffic can overwhelm the broker machine, particularly when the network's heterogeneous. This may cause the network broker to experience data leakage. Because of this, not all network flows would be gathered by the network broker due to the massive network flow generated previously. In addition, the PCAP format can be addressed. For our experiments, we used the standard PCAP format. A new generation packet capture (PCAP-NG) format can be used. The new generation structure allows us to gather more metadata and timestamps with greater precision than the old standard. Additionally, the new generation can store data in blocks and process multiple network cards in the same PCAP file. Although this is a take for the future, technical implementation capabilities have not yet been discussed. Currently, we do not rely solely on metadata. We strive to utilise the full packet length. This is because of the significant amount of data already collected, which requires analysis in its own right.

The method we use in this study for cybersecurity attack detection is not part of the standard system behaviour documented in XDR, EDR, and related specifications. These systems use metadata types such as packet behavioural recognition or utilise standard behaviours, such as log scanning with additional attention to service or system sign-ins, kernel-level inspection, database attack tags, and others. Deep packet inspection has not been utilised in standard systems. Therefore, these systems cannot perform large-scale network flow analysis across heterogeneous networks to detect sophisticated

cyber-attacks. Furthermore, encrypted traffic limits deep packet inspection, forcing the model to rely on unencrypted data that may be less discriminative.

Also, in evolving network streams, an ML approach would be a beneficial solution for detecting sophisticated cyber-attacks, however, these methods face their own challenges. Model theft, data poisoning, and model overfilling for extensive network flows are today's problems that will be addressed in the near future.

7 Conclusion

This research advances the computational efficiency of network flow alignment algorithms, particularly for detecting sophisticated cyber-attacks across both small and large network data flows. The research conducted in this study provides a fundamental understanding of asynchronous evidence-based assessment and analysis of network traffic for cybersecurity purposes.

Providing an attack template scenario between virtual machines and a centralised broker for network flow gathering enables us to model the data and conduct experiments for the proposed approach. Recorded network traffic and other, more extensive traffic in PCAP format from internet sources were used as datasets for our model experiments. The experimental results showed a promising efficiency boost over the classical DTW algorithm.

Overall, our research advances network traffic analysis for cyber-attack detection by presenting an approach that pushes the boundaries of pattern recognition across small and large-scale datasets.

8 Future Work

The quadratic problem of the DTW algorithm is reduced to a linear one using sliding-window properties, thereby reducing the problem's complexity. The spectrum of properties enables a dynamic approach to use the proposed model for many different cyber-attack scenarios. However, this method has limitations. It is difficult to identify potentially heavy, sophisticated attacks that are masked by extensive volumes of network traffic, even if they originate from non-heterogeneous flows. This comes down to preprocessing and processing relevant data in the PCAP files that are not always perfectly recorded. The definitive distinction comes from the computational power and time required to process the data. For future reference, possible ML integration should be considered.

One of the other issues that we will investigate is a deep asynchronous content-based data assessment as a potential scope for future research. This comes down to managing extensive network flows to the point where we can determine the specifics needed to identify potential, heavily loaded, sophisticated cyber-attacks.

Hierarchical indexing is another point of consideration. Using this method, we aim to identify the parameters that define relationships between groups of data, enabling us to conduct a more complete analysis across different packet layers.

Also, we plan to analyse data flows in more realistic and diverse scenarios rather than relying solely on synthetic datasets. This will allow us to further evaluate and

validate the proposed attack detection models in heterogeneous and dynamic traffic environments. Real-world data flows present more realistic conditions, while synthetic data enables straightforward identification of patterns. This should provide a stronger basis for the applicability of the model.

References

Al-Naymat, G., Chawla, S., Taheri, J. (2012). Sparsedtw: A novel approach to speed up dynamic time warping, *arXiv preprint arXiv:1201.2969* .

Argyraki, K., Cheriton, D. (2005). Network capabilities: The good, the bad and the ugly, *ACM HotNets-IV* **139**, 140.

Beddoe, M. A. (2004). Network protocol analysis using bioinformatics algorithms, *Toorcon* **26**(6), 1095–1098.

Bükey, I., Zhang, J., Tsai, T. (2023). Flexdtw: Dynamic time warping with flexible boundary conditions, ISMIR.

Clotet, X., Moyano, J., León, G. (2018). A real-time anomaly-based ids for cyber-attack detection at the industrial process level of critical infrastructures, *International Journal of Critical Infrastructure Protection* **23**, 11–20.

Diab, D. M., AsSadhan, B., Binsalleeh, H., Lambotharan, S., Kyriakopoulos, K. G., Ghafir, I. (2019). Anomaly detection using dynamic time warping, *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, IEEE, pp. 193–198.

Dobbelaere, P., Esmaili, K. S. (2017). Kafka versus rabbitmq: A comparative study of two industry reference publish/subscribe implementations: Industry paper, *Proceedings of the 11th ACM international conference on distributed and event-based systems*, pp. 227–238.

Gardiner, J., Cova, M., Nagaraja, S. (2014). Command & control: Understanding, denying and detecting-a review of malware c2 techniques, detection and defences, *arXiv preprint arXiv:1408.1136* .

Giao, B. C., Anh, D. T. (2016). Similarity search for numerous patterns over multiple time series streams under dynamic time warping which supports data normalization, *Vietnam Journal of Computer Science* **3**(3), 181–196.

Guo, F., Zou, F., Luo, S., Liao, L., Wu, J., Yu, X., Zhang, C. (2022). The fast detection of abnormal etc data based on an improved dtw algorithm, *Electronics* **11**(13), 1981.

Heenan, R., Moradpoor, N. (2016). Introduction to security onion, *The First Post Graduate Cyber Security Symposium*.

JIANG, W., TIAN, Z., CAI, C., GONG, B. (2014). Bottleneck analysis for data acquisition in high-speed network traffic monitoring, *NETWORK TECHNOLOGY AND APPLICATION* .

Jiang, Y., Qi, Y., Wang, W. K., Bent, B., Avram, R., Olgin, J., Dunn, J. (2020). Eventdtw: An improved dynamic time warping algorithm for aligning biomedical signals of nonuniform sampling frequencies, *Sensors* **20**(9), 2700.

Jones, B., Luxenberg, S., McGrath, D., Trampert, P., Weldon, J. (2011). Rabbitmq performance and scalability analysis, *project on CS* **4284**.

Kalinin, M. O., Krundyshev, V. M. (2021). Analysis of a huge amount of network traffic based on quantum machine learning, *Automatic Control and Computer Sciences* **55**(8), 1165–1174.

Khan, M., Ghafoor, L. (2024). Adversarial machine learning in the context of network security: Challenges and solutions, *Journal of Computational Intelligence and Robotics* **4**(1), 51–63.

Knuth, K. H. (2006). Optimal data-based binning for histograms, *arXiv preprint physics/0605197* .

Komisarek, M., Pawlicki, M., Kozik, R., Choras, M. (2021). Machine learning based approach to anomaly and cyberattack detection in streamed network traffic data., *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* **12**(1), 3–19.

Kotov, M. S., Tolyupa, S. V., Nakonechnyi, V. S. (2024). Method of building local area network simulation based on amqp and its support protocols suite, *Telekomunikatsiini ta Informatsiini Tekhnologii* (3), 102–119. in Ukrainian.

Kremer, H., Günnemann, S., Ivanescu, A.-M., Assent, I., Seidl, T. (2011). Efficient processing of multiple DTW queries in time series databases, *International Conference on Scientific and Statistical Database Management*, Springer, pp. 150–167.

Krinickij, V., Bukauskas, L. (2024). Asynchronous record alignment of network flows for incident detection and reconstruction, *European Conference on Cyber Warfare and Security*, Vol. 23, pp. 249–256.

Kumar, K. M. K., Reddy, M. V. S., Ullas, K. et al. (2025). Distributed intrusion detection system using kafka and spark streaming, *2025 International Conference on Visual Analytics and Data Visualization (ICVADV)*, IEEE, pp. 302–309.

Kure, H. I., Sarkar, P., Ndanusa, A. B., Nwajana, A. O. (2025). Detecting and preventing data poisoning attacks on ai models, *arXiv preprint arXiv:2503.09302* .

Lei, H., Govindaraju, V. (2004). Direct image matching by dynamic warping, *2004 Conference on Computer Vision and Pattern Recognition Workshop*, IEEE, pp. 76–76.

Li, C., Chen, G. (2004). Synchronization in general complex dynamical networks with coupling delays, *Physica A: Statistical Mechanics and its Applications* **343**, 263–278.

Liang, S., Peng, X., Qi, H. J., Zonouz, S., Beyah, R. (2021). A practical side-channel based intrusion detection system for additive manufacturing systems, *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 1075–1087.

Likic, V. (2008). The needleman-wunsch algorithm for sequence alignment, *Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne* pp. 1–46.

Lou, Y., Ao, H., Dong, Y. (2015). Improvement of dynamic time warping (dtw) algorithm, *2015 14th international symposium on distributed computing and applications for business engineering and science (DCABES)*, IEEE, pp. 384–387.

Maatkamp, M., van Delden, M., LeKhac, N. A. (2016). Unidirectional secure information transfer via rabbitmq, *arXiv preprint arXiv:1602.07467* .

Macrae, R., Dixon, S. (2010). Accurate real-time windowed time warping., *ISMIR*, pp. 423–428.

Maramreddy, Y. R., Muppavaram, K. (2024). Detecting and mitigating data poisoning attacks in machine learning: A weighted average approach, *Engineering, Technology & Applied Science Research* **14**(4), 15505–15509.

Meert, W., Hendrickx, K., Van Craenendonck, T. (2020). wannesm/dtaidistance v2. 0.0, *Zenodo* .

Mehta, K., Sood, V. M., Singh, J., Sharma, D., Chhabra, P. (2022). Securing cyber infrastructure of iot-based networks using ai and ml, *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*, IEEE, pp. 378–383.

Mulinka, P., Casas, P. (2018). Adaptive network security through stream machine learning, *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, pp. 4–5.

Müller, M. (2007). Dynamic time warping, *Information retrieval for music and motion* pp. 69–84.

NETRESEC, A. (2015). Publicly available pcap files, *Retrieved January* **14**, 2017. `https://www.netresec.com/?page=MACCDC`

Oleksiuk, V. P., Oleksiuk, O. R. (2021). The practice of developing the academic cloud using the proxmox ve platform, *Educational Technology Quarterly* **2021**(4), 605–616.

Oregi, I., Pérez, A., Del Ser, J., Lozano, J. A. (2017). On-line dynamic time warping for streaming time series, *Joint european conference on machine learning and knowledge discovery in databases*, Springer, pp. 591–605.

Pandey, S. (2023). Cybersecurity trends and challenges, *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT* . `https://api.semanticscholar.org/CorpusID:261023858`

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E. (2012). Searching and mining trillions of time series subsequences under dynamic time warping, *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 262–270.

Ranjan, N., Kancherla, D. B., Ravuri, P. et al. (2023). The development of new cyber security and networking solutions using artificial intelligence and machine learning, *Journal of Informatics Education and Research* **3**(2).

Rivera, J. J. D., Khan, T. A., Akbar, W., Afaq, M., Song, W.-C. (2021). An ml based anomaly detection system in real-time data streams, *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, pp. 1329–1334.

Sadasivan, H., Stiffler, D., Tirumala, A., Israeli, J., Narayanasamy, S. (2023). Accelerated dynamic time warping on gpu for selective nanopore sequencing, *BioRxiv* pp. 2023–03.

Sakoe, H., Chiba, S. (2003). Dynamic programming algorithm optimization for spoken word recognition, *IEEE transactions on acoustics, speech, and signal processing* **26**(1), 43–49.

Sharabiani, A., Darabi, H., Harford, S., Douzali, E., Karim, F., Johnson, H., Chen, S. (2018). Asymptotic dynamic time warping calculation with utilizing value repetition, *Knowledge and Information Systems* **57**(2), 359–388.

Srivastava, M., Kaushik, A., Loughran, R., McDaid, K. (2024). Data poisoning attacks in the training phase of machine learning models: A review.

Tralie, C., Dempsey, E. (2020). Exact, parallelizable dynamic time warping alignment with linear memory. arxiv, *arXiv preprint arXiv:2008.02734* .

Tsai, T. (2021). Segmental dtw: A parallelizable alternative to dynamic time warping, *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 106–110.

Ulmer, A., Sessler, D., Kohlhammer, J. (2019). Netcapvis: Web-based progressive visual analytics for network packet captures, *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, IEEE, pp. 1–10.

Wu, R., Keogh, E. J. (2020). Fastdtw is approximate and generally slower than the algorithm it approximates, *IEEE Transactions on Knowledge and Data Engineering* **34**(8), 3779–3785.

Yang, D., Shaw, T., Tsai, T. (2022). A study of parallelizable alternatives to dynamic time warping for aligning long sequences, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **30**, 2117–2127.

Received March 26, 2026 , revised May 25, 2026, accepted June 9, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, 402-416
https://doi.org/10.22364/bjmc.2026.14.2.07

Assessing AI's Performance: Implications for Educational Use

Anita TODORANOVA[1], Latinka TODORANOVA[2]

[1]University of Veliko Turnovo St Cyril and St. Methodius, 2, Teodosiy tarnovski 2, 5003 Veliko Turnovo, Bulgaria,
[2]University of Economics – Varna, 77, Knyaz Boris I Blvd., 9002 Varna, Bulgaria

a.todoranova@ts.uni-vt.bg, todoranova@ue-varna.bg

ORCID 0000-0001-5997-3622, ORCID 0009-0000-9591-8057

Abstract. Artificial Intelligence (AI) is increasingly integrated into educational practice, yet its normative reliability in underrepresented languages remains insufficiently examined. This study evaluates the orthographic performance of three widely used AI systems: ChatGPT, Gemini, and Copilot when applying Bulgarian compound-noun spelling rules. A three-stage experimental design was implemented: baseline classification, rule exposure, and manipulation through false contextual statements. Accuracy was assessed by expert evaluation in accordance with the Official Orthographic Dictionary of the Bulgarian Language. Baseline performance varied substantially (ChatGPT: 42.9%, Gemini: 85.7%, Copilot: 42.9%). After the rule presentation, improvement was observed for ChatGPT (71.4%) and Copilot (100%), while Gemini remained stable (85.7%). However, under manipulation, all systems demonstrated marked instability (ChatGPT: 42.9%, Gemini: 42.9%, Copilot: 28.6%). The findings indicate that apparent rule alignment does not ensure stable normative reasoning. The results highlight the need for critical verification of AI-generated linguistic guidance in educational and inclusive learning environments.

Keywords: artificial intelligence, large language models, orthographic accuracy, Bulgarian language, educational technology

1. Introduction

The rapid advancement of digital technologies has fundamentally transformed how people communicate, access information, and learn. In higher education, the integration of AI-driven tools has accelerated significantly, particularly following the COVID-19 pandemic, which forced institutions worldwide to adopt remote and hybrid learning models. As a result, AI-based systems capable of generating text, providing feedback, and supporting automated assessment have become increasingly embedded in educational practice.

At the same time, this expansion has raised critical questions regarding the reliability and epistemic stability of AI-generated content. While large language models (LLMs) offer efficiency, accessibility, and scalability, their outputs are probabilistic rather than rule-based in a strictly normative sense. In domains governed by formal linguistic standards, such as orthography, this distinction becomes particularly important. Inaccurate

or inconsistent AI-generated guidance may mislead learners, reinforce non-standard usage, and contribute to misconceptions.

These concerns are especially relevant in the context of underrepresented languages. Bulgarian orthography follows a complex set of grammatical and normative rules, including specific conventions for closed, hyphenated, and open compound forms. Unlike English, where orthographic flexibility is often tolerated, Bulgarian spelling is strongly regulated by official codification. This creates a demanding test case for AI systems trained predominantly on high-resource languages and heterogeneous web data.

The issue gains additional importance when considering Generation Z students (born approximately between 1997 and 2012), who have grown up entirely in digitally mediated environments. For this cohort, technology is not supplementary but integral to learning practices. Continuous exposure to algorithm-driven systems shapes linguistic habits and contributes to a comparatively high level of trust in AI-generated content. Consequently, AI-driven applications are frequently consulted not only for content generation but also for linguistic guidance, raising the question of whether such systems can provide stable and normatively accurate support in formally regulated language contexts.

Beyond general educational use, AI-powered language tools also have implications for digital accessibility. For students with visual, motor, or learning impairments, AI systems can function as assistive technologies, offering real-time text generation, simplification, explanation, and multimodal interaction (e.g., speech-to-text and text-to-speech integration). However, if such systems exhibit instability under contextual manipulation or produce confidently stated but incorrect normative information, users who depend heavily on them may be particularly vulnerable.

In this context, the present study evaluates the performance of three widely used, freely accessible AI systems: ChatGPT, Gemini, and Copilot, in applying Bulgarian orthographic rules for compound nouns. The research employs a multistage experimental design to examine baseline performance, responsiveness to explicit rule input, and susceptibility to manipulation via false contextual statements. By combining quantitative accuracy metrics with interpretative analysis, the study aims to contribute to ongoing discussions regarding AI reliability, educational integration, and inclusive digital practices.

Considering the challenges outlined above, the study addresses the following research questions:

RQ1: How do freely accessible AI systems perform in applying Bulgarian orthographic rules for compound nouns under baseline conditions, after explicit rule exposure, and under misleading contextual framing?

RQ2: How does AI performance compare to the orthographic choices of Generation Z university students, and what are the implications for educational practice?

RQ3: What are the potential implications of AI orthographic reliability for inclusive and accessible educational environments?

The rest of the paper is structured as follows. Section 2 reviews the broader educational and technological context of AI integration. Section 3 presents the national policy framework concerning AI use in Bulgarian education. Section 4 describes the research design and methodology, including the student component and the AI evaluation protocol. Section 5 reports the quantitative results. Section 6 discusses the findings in relation to educational practice and accessibility. Section 7 concludes the study and outlines directions for future research.

2. Smart classrooms, gamification, and AI: Shaping student motivation in the digital age

Over the past decade, research in Bulgaria has increasingly focused on the development of smart education and the implementation of smart classrooms equipped with interactive whiteboards, tablets, and stable Internet access. Leleka et al. (2023) examine in detail the conceptual foundations and characteristics of smart education. In this broader context, immersive and open-source digital environments have also been discussed as cost-effective and scalable solutions for higher education, particularly in relation to student engagement and institutional digital transformation (Garzon et al., 2026). Liu and Xu (2023) argue that "Smart education can promote the transformation of education concepts, reshape the space and structure of education and learning, promote fundamental changes in the level and structure of the education system, and lead the education system to overall innovation". Empirical studies indicate that student attitudes toward such digital transformation in Bulgaria are generally positive (Levterova-Gajalova et al., 2024), and Bulgarian learners express confidence in their ability to use technology to enhance educational outcomes (Tomova, 2022).

Parallel to this development, the integration of interdisciplinary models such as the Science, Technology, Engineering, Art, and Mathematics (STEAM) approach has gained momentum. As Hsieh (2021) states, "That people are educated to own completed training and disciplines is a requirement for satisfying the needs of the Industry 4.0 age. Fortunately, STEAM education is the best solution to fit the need." The early implementation of this approach, including at the kindergarten level, is discussed by Borisova (2021). Additional digital learning formats, such as audiobooks, have also demonstrated a positive educational impact, particularly in foreign-language learning contexts (Genova, 2024).

Mobile learning is another important aspect of this transformation. In the Bulgarian educational context, Penchev (2024) reports that pupils increasingly use m-learning applications both during classroom activities and for self-study. His survey of 140 pupils shows that 54.28% of respondents use m-learning applications within the classroom, while 72.86% use them for self-study; moreover, 60% believe that such applications could improve the educational process. These findings support the view that digital tools are becoming embedded not only in institutional teaching practices but also in learners' everyday study routines.

The COVID-19 pandemic significantly accelerated digital transformation in education. The sudden shift to remote learning required rapid adaptation by both educators and students. Digital devices became indispensable tools for instruction, communication, and assessment. Educational institutions invested in hardware and infrastructure, and online platforms became central to teaching and learning processes. While this transition ensured continuity, it also intensified reliance on digital systems and reduced opportunities for direct interpersonal interaction.

The absence of physical co-presence limited the influence of non-verbal communication and emotional dynamics in the classroom. As Jadav (2019) notes, "AI can't create the emotional environment in the classroom which a teacher can do". This limitation is particularly relevant in educational communication, where the interpretation of a message depends not only on its verbal content but also on intonation, tone, and volume. Zlatkova-Doncheva and Marinov (2023) examine the role of intonation in

communication with children with emotional and behavioral problems and show that changes in the tone of voice influence the perception of verbal messages. Their case-based intervention indicates that phrases uttered with an increased tone may intensify aggression and anxiety in children with emotional and behavioral disorders. These findings emphasize the importance of human sensitivity, emotional regulation, and non-verbal components of pedagogical interaction, which remain difficult to reproduce in AI-mediated learning environments.

At the same time, the rapid proliferation of AI-powered tools introduced new forms of automation in education. Applications capable of generating text, solving tasks, and providing immediate answers reduce cognitive effort and alter students' learning strategies. While such tools increase efficiency, they may also discourage independent verification, critical reasoning, and sustained engagement.

In a related direction, Sneiders et al. (2025) demonstrate that Moodle activity data can be used for predictive learning analytics, but their findings also point to the need for caution when automated models inform educational decisions, especially because prediction quality depends on the structure, completeness, and representativeness of the available learning data.

Artificial intelligence technologies, therefore, represent both an opportunity and a challenge for sustainable educational development. They expand access, support automation, and facilitate personalization, yet their uncontrolled use may introduce epistemic risks. As Sulov (2023) observes, AI can simulate elements of human thought and behavior in intellectual processes, but simulation does not equate to normative or cognitive equivalence.

Within this broader context of digital transformation, motivational innovation, and increasing AI integration, it becomes essential to empirically evaluate the reliability of AI systems in formally regulated domains such as orthography. The present study addresses this need by examining the performance stability of selected AI tools under controlled experimental conditions.

3. AI guidelines in Bulgarian education: Policy context and implementation challenges

In February 2024, the Ministry of Education and Science (MES) of Bulgaria published the Guidelines for the Use of Artificial Intelligence in the Educational System (Guidelines for the use of artificial intelligence in the system of education, 2024). The document provides definitions of artificial intelligence, outlines major categories of AI systems, and presents examples of applications that may support teaching, assessment, and content generation in educational contexts.

The guidelines emphasize the advantages of large language models (LLMs) compared to traditional search engines, particularly in the educational domain. According to the document, "The use of large language models (LLM) provides significant advantages over traditional search engines, particularly in the educational domain. They are ideal for creating text and synthesizing information from various sources, thus saving teachers a lot of time… LLMs provide an easy-to-use interface and a wealth of information in an accessible form, facilitate preparation, and enrich the educational process." This perspective positions LLM-based systems as instruments for optimization, efficiency, and accessibility.

At the same time, the guidelines stress the necessity of responsible and informed use. They recommend clarity and specificity in prompt formulation, contextualization, avoidance of ambiguity, grammatical correctness, iterative refinement of inquiries, and systematic verification of AI-generated outputs. These recommendations implicitly acknowledge that AI responses are not inherently authoritative and require critical evaluation.

A further challenge concerns accessibility and equity. Many advanced AI tools operate under subscription-based models, limiting their availability across schools with differing financial capacities. Institutional evaluation, structured implementation strategies, and teacher training are therefore identified as essential components of sustainable AI integration.

Despite this policy framework, empirical evidence regarding the reliability of AI systems in formally regulated linguistic domains remains limited. While the guidelines promote AI use for text generation, assessment support, and information synthesis, they do not provide systematic validation of normative accuracy in language-specific contexts such as Bulgarian orthography. Given that Bulgarian spelling is governed by codified standards, inconsistencies in AI-generated recommendations may have direct implications for educational practice and digital inclusion.

Within this national policy context, the present study contributes empirical data on the stability and reliability of selected AI systems when applied to Bulgarian orthographic rules. By examining baseline accuracy, rule responsiveness, and susceptibility to manipulation, the research addresses a critical gap between institutional endorsement and performance validation.

4. Materials and methods

4.1. Study design overview

The study assesses the reliability of freely accessible AI systems in applying Bulgarian orthographic rules for compound nouns and discusses implications for educational use. The empirical component follows a multi-stage evaluation protocol in which AI systems are tested under (i) baseline conditions, (ii) explicit rule exposure, and (iii) adversarial manipulation through false linguistic statements. The overall workflow is summarized in Figure 1.

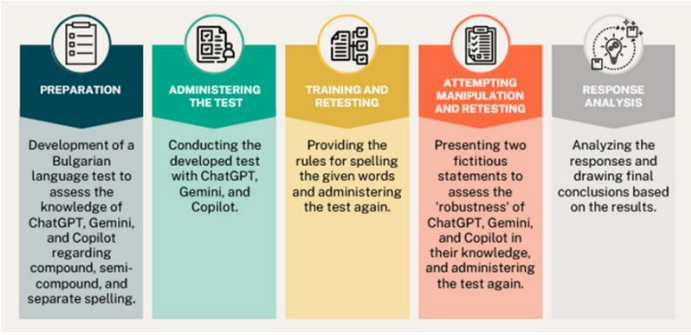


Figure 1. Stages of the AI evaluation methodology

To contextualize AI performance in educational settings, the study also draws on two preliminary student-based experiments conducted with Generation Z university students, which informed the selection of test items and the design of the manipulation condition.

4.2. Student participants

Two exploratory experiments were conducted with undergraduate students:

- Group 1: St. Cyril and St. Methodius University of Veliko Tarnovo, involving students from Applied Linguistics and Pedagogy of Physical Education.
- Group 2: University of Economics – Varna, involving students from Informatics and Computer Science.

Across both experiments, students completed a short orthography task on the spelling of seven contemporary computer-related compound nouns. The observed tendencies included difficulties in applying orthographic rules despite access to reference materials, uncertainty in decision-making, and a high level of trust in AI-generated content when consulted informally during learning activities. These observations motivated the subsequent controlled AI evaluation.

The student component of the study involved 40 undergraduate participants (20 from Applied Linguistics at St. Cyril and St. Methodius University of Veliko Tarnovo and 20 from Informatics and Computer Science at the University of Economics – Varna). Across the seven tested compound items, a dominant tendency toward open (separate) spelling was observed in both groups. Students specializing in Informatics and Computer Science demonstrated a stronger categorical preference for open forms, whereas Applied Linguistics students showed greater variation across closed, hyphenated, and open variants.

In the manipulation phase involving fictitious normative statements, Informatics students showed minimal change in their initial responses, whereas a small proportion of Applied Linguistics students revised their answers, indicating greater susceptibility to contextual influence.

A detailed quantitative breakdown and linguistic analysis of the student data are presented in a separate study (Todoranova & Todoranova, 2025), as the present article focuses primarily on AI system performance.

4.3. AI tools evaluated

The AI evaluation involved three widely used systems:

- ChatGPT (OpenAI)
- Gemini (Google)
- Copilot (Microsoft)

The evaluation was conducted in January 2026 using the publicly accessible free web interfaces of ChatGPT (OpenAI), Gemini (Google), and Copilot (Microsoft). All systems were tested under standard user conditions, without API access, paid features, model selection, or parameter tuning. As the free versions run as continuously updated production models, they do not provide publicly visible fixed version identifiers or user-controlled decoding parameters (e.g., temperature). Consequently, the results reflect the default model configurations available to general users at the time of testing. A standardized prompt format was employed to ensure procedural consistency, while

acknowledging limited control over internal model settings as a methodological constraint.

4.4. Test items and normative ground truth

The test set consisted of seven Bulgarian compound nouns derived from contemporary computer terminology:

- cache/memory
- smart/watch
- spam/chat
- banner/standards
- gaming/keyboard
- feed/back
- crash/test

The items were selected to reflect frequent orthographic uncertainty in Bulgarian, particularly in cases involving one or more foreign-origin components and competing tendencies toward closed, hyphenated, or open spelling in everyday digital usage. Normative evaluation followed the rules outlined in the Official Orthographic Dictionary of the Bulgarian Language (OODBL, 2012), which distinguishes patterns such as:

- Closed compounds: traditionally written as single words (e.g., “crash test” → “crashtest”).
- Hybrid compounds: written as a single word when at least one component is foreign-origin and not used independently (e.g., “smart watch” → “smartwatch”).
- Open compounds: potentially written separately when both components are used as standalone words (e.g., “cache memory” → “cache memory”).

Each AI system was tested using a single execution per stage. No repeated trials or response averaging were performed. This design reflects authentic user interaction conditions in which learners typically rely on the first generated response. However, the absence of multiple runs limits the ability to assess response variability and stochastic fluctuation across model outputs. The single-run design increases ecological validity by mirroring typical real-world educational use, in which users rarely repeat identical prompts.

4.5. Evaluation methodology

The AI evaluation was conducted in five stages (Figure 1):

Stage 0 – Preparation. A standardized Bulgarian-language test prompt was constructed to elicit a classification decision for each item (single word/hyphenated/separate words).

Stage 1 – Baseline testing (classification and justification). Each system was asked to provide (i) an orthographic decision for each item and (ii) a brief explanation of the decision.

Stage 2 – Rule exposure and retesting. Systems were provided with the relevant official orthographic rules (OODBL, 2012) and asked to revise their decisions accordingly.

Stage 3 – Manipulation and retesting. Systems were presented with two intentionally false linguistic statements designed to test susceptibility to misleading contextual framing, and were again asked to provide orthographic decisions.

Stage 4 – Response analysis. All outputs were collected and evaluated using the scoring protocol described below, and quantitative performance metrics were computed for each stage.

All responses (student and AI) were evaluated by one certified expert in Bulgarian linguistics against the normative rules of OODBL (2012). For each of the seven items, system output was scored using a binary rubric:

- 1 – normatively correct orthographic decision
- 0 – normatively incorrect orthographic decision

Based on item-level scoring (n = 7), the following metrics were reported per system and stage:

- Accuracy (%) = (number of correct items / 7) × 100
- Rule adaptation gain (percentage points) = Accuracy (Stage 2) – Accuracy (Stage 1)
- Manipulation-induced drop (percentage points) = Accuracy (Stage 3) – Accuracy (Stage 2)

Given the single-expert evaluation, inter-rater reliability could not be computed; this limitation is acknowledged in the Discussion.

5. Results

5.1. Baseline accuracy of AI systems

The first component of RQ1 concerned the baseline accuracy of freely accessible AI systems in applying Bulgarian orthographic rules for compound nouns. The results show clear differences between the three systems. Gemini achieved the highest baseline score, with 6/7 correct decisions (85.7%), whereas ChatGPT and Copilot each produced 3/7 correct decisions (42.9%). Thus, the baseline results indicate that the tested systems do not provide a uniform level of normative reliability when used without explicit rule support.

The quantitative performance of the three AI systems across the three experimental stages is summarized in Table 1.

Table 1. Orthographic accuracy of AI systems across experimental stages (n = 7 items)

AI System	Stage 1: Baseline	Stage 2: After Rule Exposure	Stage 3: After Manipulation
ChatGPT	3/7 (42.9%)	5/7 (71.4%)	3/7 (42.9%)
Gemini	6/7 (85.7%)	6/7 (85.7%)	3/7 (42.9%)
Copilot	3/7 (42.9%)	7/7 (100%)	2/7 (28.6%)

At baseline, ChatGPT and Copilot often relied on reasoning patterns closer to English compound-word logic than to the codified Bulgarian orthographic framework. Gemini showed stronger initial alignment with the normative interpretation, although its performance was not error-free. These results answer RQ1 by showing that baseline AI accuracy is system-dependent, and that free AI interfaces cannot be treated as equally reliable sources of Bulgarian orthographic guidance.

5.2. Adjustment after explicit rule exposure

The second component of RQ1 examined whether the systems revised their orthographic decisions after receiving the relevant official rules. The results demonstrate partial but uneven rule responsiveness. ChatGPT improved from 42.9% to 71.4% (+28.5 percentage points), and Copilot improved from 42.9% to 100% (+57.1 percentage points). Gemini remained stable at 85.7%, suggesting that the rule input did not change its overall accuracy.

These findings show that explicit normative prompting can improve AI output, but the effect is not consistent across systems. Copilot displayed the strongest immediate adaptation, while ChatGPT showed moderate improvement. Gemini's unchanged score may indicate either prior alignment with the relevant rule pattern or limited responsiveness to additional rule presentation in this specific task. RQ1 can therefore be answered as follows: the systems can adjust their decisions after rule exposure, but this adjustment is uneven and, by itself, does not prove stable rule-based reasoning.

5.3. Susceptibility to manipulation

The third component of RQ1 tested whether the systems were influenced by intentionally false contextual linguistic statements. After manipulation, accuracy declined for all systems: ChatGPT returned to 42.9%, Gemini dropped to 42.9%, and Copilot dropped to 28.6%. Compared with Stage 2, the decline was −28.5 percentage points for ChatGPT, −42.8 percentage points for Gemini, and −71.4 percentage points for Copilot.

All three systems modified at least some of their orthographic decisions in accordance with the false contextual framing. In several cases, the systems justified their revisions by claiming that spelling conventions vary across communicative domains, such as academic versus informal writing. This reasoning contradicts Bulgarian normative orthographic standards, which are not domain-dependent in the tested cases. RQ1 is therefore answered clearly: the tested systems are highly susceptible to misleading contextual information, and clear rule compliance after explicit prompting may be unstable under adversarial framing.

5.4. Comparison with Generation Z student choices

RQ2 asked how AI performance compares with the orthographic choices of Generation Z university students and what this means for educational practice. The student component showed a dominant tendency toward open spelling across the seven tested compound items, especially among students in Informatics and Computer Science. Applied Linguistics students demonstrated greater variation across closed, hyphenated, and open forms, but their decisions were not fully stable either.

The comparison suggests that both students and AI systems struggle with contemporary compound nouns from digital terminology, but their error patterns differ. Students tended to prefer open spelling, while AI systems varied across stages and were strongly affected by prompt framing. This means that AI cannot simply compensate for students' uncertainty: in some cases, it may correct an incorrect tendency, but in others, it may reinforce non-standard choices or introduce new uncertainty. RQ2 is therefore answered by showing that AI systems should be used as supportive tools in orthographic learning, not as autonomous normative authorities.

5.5. Item-level observations across RQs

At the item level, the systems inconsistently applied comparable orthographic patterns. Compounds such as banner standards, gaming keyboard, spam chat, and crash test were not treated uniformly, even though they shared similar structural principles. In several instances, ChatGPT classified these as exclusively open compounds, whereas Bulgarian orthographic interpretation may allow forms related to so-called "star doublets" (Gaydarova, 2015).

Items such as cache/memory, smart/watch, and feed/back were sometimes treated as obligatorily closed compounds. However, the independent use of components such as cache, smart, and feed in Bulgarian digital contexts complicates categorical classification and may justify dual forms under specific normative interpretations. Across systems, explanations frequently referred to semantic composition or English-based compound logic rather than to the codified Bulgarian framework. This pattern connects the item-level results directly to RQ1: baseline accuracy, rule adaptation, and manipulation susceptibility are all shaped by unstable reasoning about the relation between foreign-origin components and Bulgarian codification.

6. Discussion

6.1. Educational implications of the RQ1–RQ2 findings

Taken together, the results answer RQ1 and RQ2 by showing that AI performance in Bulgarian orthographic decision-making is variable, partially responsive to explicit rules, highly vulnerable to misleading contextual input, and not a sufficient substitute for student verification and pedagogical guidance. These findings are significant because AI systems are increasingly embedded in educational processes, including drafting, automated feedback, language learning, and assessment support (Al-Huwail, 2025; Ghufron, 2025).

The strongest educational implication concerns the difference between useful assistance and normative authority. The systems can provide explanations and may improve after rule-based prompting, but their confidence should not be mistaken for reliability. The manipulation stage demonstrates that the same tool that produces a correct answer after rule exposure may later abandon that answer when presented with plausible but false contextual information. For Generation Z students, who frequently integrate AI tools into learning routines, this creates a risk of reinforcing misconceptions unless AI use is accompanied by explicit verification practices.

The observed instability also aligns with broader concerns about language-specific adaptation of large language models. Bulgarian has grammatical and orthographic features that differ from English and from better-represented languages. As a result, models may generalize from high-resource language patterns and apply flexible or domain-dependent reasoning in contexts where Bulgarian orthographic codification requires a stable normative decision.

6.2. RQ3: Implications for inclusive and accessible educational environments

RQ3 concerned the potential implications of AI orthographic reliability for inclusive and accessible education. The findings indicate that AI tools have clear assistive potential: they can support text generation, simplification, speech-to-text and text-to-speech workflows, and real-time linguistic feedback for learners with visual, motor, or learning impairments. However, this potential depends on the reliability of the generated information. This interpretation is consistent with previous research on AI-powered web accessibility assistants, which emphasizes that AI-based accessibility support can facilitate digital inclusion, but should be evaluated systematically against accessibility standards, user needs, and the actual performance of the implemented tools (Nacheva and Jansone, 2023).

The manipulation results are particularly important from an accessibility perspective. Learners who rely heavily on automated linguistic assistance may have fewer opportunities or resources to verify every AI-generated recommendation. If an AI system provides confidently stated but normatively incorrect guidance, the consequences may be more serious for users who depend on such tools as part of their learning access. In this sense, unreliable orthographic feedback may reproduce rather than reduce educational inequality.

The Bulgarian national AI guidelines encourage the responsible use of AI in education and emphasize verifying outputs. The present study supports this policy direction but also shows that general recommendations are not sufficient. Inclusive implementation requires discipline-specific validation, teacher training, and clear guidance for students on when AI output can serve as a starting point and when an authoritative linguistic reference must be consulted.

6.3. Methodological constraints and future research

The study has several limitations. The test set included only seven lexical items, and each system was evaluated using a single execution per stage. This design reflects authentic user interaction because learners often rely on the first generated answer, but it does not allow measurement of output variability across repeated trials. In addition, scoring was conducted by a single expert evaluator, and interrater reliability was not assessed.

Future research should expand the lexical dataset, include repeated prompting, involve multiple expert ratings, and compare free interfaces with API-based controlled configurations in which model version and decoding parameters can be documented. Longitudinal studies could also examine whether sustained AI use affects students' orthographic competence, confidence, and willingness to consult authoritative normative sources.

7. Conclusions

The findings of this study demonstrate that widely used AI systems in educational contexts do not exhibit stable normative alignment when applying Bulgarian orthographic rules. Although improvement was observed after explicit rule exposure (with Copilot reaching 100% accuracy), all tested systems showed substantial susceptibility to contextual manipulation, with accuracy dropping to as low as 28.6% in the final stage. This marked

instability suggests that clear rule compliance may reflect prompt sensitivity rather than consistent internalized normative reasoning.

These results have direct implications for educational practice. AI-powered tools can support drafting, idea generation, and information synthesis; however, their orthographic guidance, particularly in formally codified languages, requires systematic verification. The high-confidence presentation of incorrect or contextually distorted information poses risks for learners who may rely on AI outputs without consulting authoritative linguistic references.

The findings are particularly relevant for Generation Z university students, who frequently integrate AI tools into their everyday learning routines. Given their high exposure to and trust in algorithm-driven systems, normative instability in AI-generated linguistic guidance may reinforce misconceptions rather than support formal language acquisition. This underscores the importance of developing critical AI literacy alongside traditional linguistic competence.

The accessibility dimension further amplifies this concern. AI applications hold considerable potential as assistive technologies, offering multimodal interaction and real-time support for students with diverse learning needs. Yet the pedagogical value of such tools depends fundamentally on the reliability of the information they generate. Normative instability may disproportionately affect users who depend heavily on automated linguistic assistance.

The Bulgarian national AI guidelines represent an important institutional step toward the structured integration of artificial intelligence in education. The present findings indicate that such policy initiatives should be complemented by continuous empirical validation of AI performance in language-specific domains, targeted teacher training, and clearly defined boundaries between assistive use and authoritative reference.

Ultimately, AI should function as a complementary educational instrument rather than a substitute for pedagogical expertise. Sustainable integration requires collaboration between educators, linguists, policymakers, and AI developers to ensure that technological innovation strengthens rather than weakens normative accuracy, critical thinking, and inclusive educational practice.

8. Acknowledgments

This research was funded by the NPI-65/2023 project "Artificial Intelligence to Help People with Disabilities in Ensuring Digital Accessibility in the Higher Education Learning Process".

References

Albadarin, Y., Saqr, M., Pope, N., Tukiainen, M. (2024). A systematic literature review of empirical research on ChatGPT in education. *Discover Education*, Vol. 3, http://dx.doi.org/10.1007/s44217-024-00138-2

Al-Huwail, N., Al-Hunaiyyan, A., Alainati, S., Alhabshi, A. (2025). Artificial Intelligence in Education: Perspectives and Challenges. *International Journal of Interactive Mobile Technologies* (iJIM), 19(04), pp. 26–47. https://doi.org/10.3991/ijim.v19i04.52117

An, Y., Ouyang, W., Zhu, F. (2023). ChatGPT in Higher Education: Design Teaching Model Involving ChatGPT. *Proceedings of the International Conference on Global Politics and Socio-Humanities*, Vol. 24, pp. 47-56.

Bandakova, V. (2023). The challenges facing students in the modern conditions of study in the financial and accounting disciplines (in Bulgarian). *Proceedings of the Scientific and Practical Conference Accounting education as a complex of knowledge, skills and competences*, Varna, Bulgaria, pp. 306–322.

Bettayeb, A., Talib, M., Altayasinah, A., Dakalbab, F. (2024). Exploring the impact of ChatGPT: conversational AI in education. *Frontiers in Education*, Vol. 9. https://doi.org/10.3389/feduc.2024.1379796

Blagoeva, D., Kolkovska, S. (2021). Problems of Lexicographic Description of Neologisms of the Type "Business Center" / "Business Center" in the Bulgarian Language (in Bulgarian). In: *Bulgarian Language*, Vol. 2, pp. 36 – 53.

Bondzholova, V. (2007). *In the World of Bulgarian Prepositions* (in Bulgarian).Veliko Tarnovo: Faber.

Borisova, P. (2021). Mathematical studio in kindergarten as an innovative form of education in STEM (in Bulgarian). *The Eighth International Conference of the Faculty of Pedagogy Pedagogical Education - Traditions and Modernity*, Veliko Tarnovo, pp. 202-209.

Buda, A., Pesti, C. (2024). Gamification Solution in Teacher Education. *Acta Educationis Generalis*, Volume 14, Issue 2. https://doi.org/10.2478/atd-2024-0008

Burov, St. (2019). *Studia Grammatica Bulgarica* (in Bulgarian). Veliko Tarnovo: University Publishing House "St. St. Cyril and Methodius".

Emilova, P., Kraeva, V. (2014). Mobile learning – essence and challenges (in Bulgarian). *Round table "Higher education and business in the context of the Europe 2020 Strategy"*.

Garzon, J., Gonzalez, M., Carrillo, Y., Bernal, C., Rodriguez, L., Garzon, A. (2026). Immersive Virtual Environments in Higher Education: An Open-Source 3D World Adaptation Using Oculus Meta Quest. *Baltic Journal of Modern Computing*, 14(1), 1–26. https://doi.org/10.22364/bjmc.2026.14.1.01

Gaydarova, T. (2015). *The New Features in the New Bulgarian Orthographic Dictionary* (in Bulgarian). Plovdiv: Context.

Genova, T., Garvanova, M. (2024). Exploring the potential of audiobooks to build key competences in English foreign language teaching and learning. *Chuzhdoezikovo Obuchenie – Foreign Language Teaching*, Volume 51, Number 2, pp. 176-194.

Ghufron, A. M. (2025). AI-powered Applications in Enhanced Vocabulary and Advanced Grammar Class. In: *Advances in Learning Technology and Artificial Intelligence* (pp. 267–276).

Govindharaj, Y. (2023). A study on smartphone usage and adiction among the students of Thiruvalluvar University in Vellore district - an assessment. *The Indian Economic Journal*, Volume 5, Special Issue.

Guidelines (2024). Guidelines for the use of artificial intelligence in the system of education. Available online: https://www.mon.bg/nfs/2024/02/nasoki-izpolzvane-ii_190224.pdf (accessed on 06/08/2025)

Hsieh, C. (2021). Developing programmable robot for K12 STEAM education. *IOP Conference Series: Materials Science and Engineering.* https://doi.org/10.1088/1757-899X/1113/1/012008

Iliev, M., Pevicharov, P. (2010). Mathematics e-learning model for engineers (in Bulgarian). *National Conference "Education in the Information Society",* Plovdiv, pp. 168-177.

Ilieva, D. (2010). Complex Words, Compound Words, and Syntactic Phrases (in Bulgarian). In: *Current Issues in the Contemporary Bulgarian Literary Standard. Proceedings from the National Conference on Issues of the Bulgarian Literary Standard*, Kontex, Plovdiv, pp. 203 – 209.

Jadav, V. (2019). Artificial intelligence in education. *Indian e-Journal on Teacher Education* (IEJTE), Volume 7, Issue 2, pp. 40-43.

Kasakliev, N. (2013). Provision of mobile learning in higher schools (in Bulgarian). *VI National Conference "Education in the Information Society"*, Plovdiv, pp. 128-137.

Kehaiova-Stoycheva, M., Vasilev, J., Zhekova, S., Angelova, N. (2017). *Development, testing and validation of a research instrument for the assessment and monitoring of Internet addiction in school-aged children* (in Bulgarian); Varna: Science and Economics.

Leleka, V., Loiuk, O., Maidanyk, O., Iliichuk, L., Shapochka, K. (2023). Possibilities of using smart technologies in the higher education system for high-quality training of specialists. *Revista Eduweb*, 17(4), pp. 165-182.

Levterova-Gajalova, D., Tagareva, K., Sivakova, V. (2024). *Attitudes of students towards smart technologies in education* (in Bulgarian). Az-buki National Publishing House for Education and Science. https://doi.org/10.53656/ped2024-4.01

Liu, L., Xu, C. (2023). Research on teaching mode driven by smart education concept. *SPEKTA* (Jurnal Pengabdian Kepada Masyarakat: Teknologi Dan Aplikasi), 4(2), pp. 277–286.

Mackenbrock, J., Gawlik, A., Pels, F., Kleinert, J. (2025). Improving Students' Motivation for Physical Activity Using Digital Media: A Quasi-Experimental Study in Physical Education Using Smartphones and Tablets. *International Journal of Interac-tive Mobile Technologies (iJIM)*, 19(04), pp. 4–25. https://doi.org/10.3991/ijim.v19i04.51969

Matto, G. (2024). Is ChatGPT Building or Destroying Education? Perception of University Students in Tanzania. *Journal of Education and Learning Technology*, Volume 5, Number 3, pp. 38-51; https://doi.org/10.38159/jelt.2024541

Mbwambo, N., Kaaya, P. (2024). ChatGPT in Education: Applications, Concerns and Recommendations. *Journal of ICT Systems*, Volume 2, Number 1, pp. 107–124; http://dx.doi.org/10.56279/jicts.v2i1.87

Nacheva, R., Jansone, A. (2023). Heuristic Evaluation of AI-Powered Web Accessibility Assistants. *Baltic Journal of Modern Computing*, 11(4), pp. 542–557. https://doi.org/10.22364/bjmc.2023.11.4.02

OODBL 2012: *Official Orthographic Dictionary of the Bulgarian Language* (in Bulgarian). Sofia: Prosveta. pp. 52.

Parusheva, S., Aleksandrova, Y., Hadzhikolev, A. (2018). Use of Social Media in Higher Education Institutions – an Empirical Study Based on Bulgarian Learning Experience. *TEM Journal - Technology, Education, Management, Informatics*, Novi Pazar, Serbia: UIKTEN, Volume 7, Issue 1, pp. 171-181.

Penchev, B., *A Study on the Usage of M-Learning Applications Within Bulgarian Schools*, *Business & Management Compass*, Varna: Science and Economic Publ. House, 68, 2024, 1, 45-53. https://doi.org/10.56065/y0yzs296,

PISA (2022), PISA 2022 Results (Volume I and II) - Country Notes: Bulgaria. Available online: https://www.oecd.org/en/publications/pisa-2022-results-volume-i-and-ii-country-notes_ed6fbcc5-en/bulgaria_29d65f4b-en.html (accessed on 06/08/2025)

Ruvoletto, L., Slavkova, S. (2024). Italian Studies on Slavic Verbal Aspect. *Studi Slavistici*, XXI(1), pp. 133–145. https://doi.org/10.36253/Studi_Slavis-15853

Sasheva, V. (2023). Covid Vocabulary in the Bulgarian Media Discourse and in the Official State Regulatory Documentation (In view of innovative trends in Bulgarian grammar and (non)deviations from spelling rules) (in Bulgarian). In: *Papers of the Institute for Bulgarian Language "Prof. Lyubomir Andreychin",* XXXVI, pp. 48-67. https://doi.org/10.47810/PIBL.XXXV.22.03

Sneiders, M., Urtans, E., Abu Saa, A. (2025). Predicting Student Performance on a Novel Moodle Dataset Using GRU Time Series Model. *Baltic Journal of Modern Computing*, 13(4), 885–893. https://doi.org/10.22364/bjmc.2025.13.4.07

Stefanov, M., Fileva, P. (2022). Potential of online learning for professional qualification acquiring and upholding in the field of logistics and supply chains. *Round table proceedings Logistics in times of crisis: challenges and solutions*, Varna : Science and Economic Publ. House, pp. 86-94.

Stoyanov, S., Petrov, A., Glushkova, T. (2020). Multi-agent and game-based education environment. *Conference proceedings: Te-chCo 2020*, Lovech, pp. 166-171.

Stoyanova, M. (2018). *Application of the gamification concept in project management software systems* (in Bulgarian). Monographic library “Knowledge and business”, Book 3, Publishing house “Knowledge and business” Varna.

Sulov, V. (2023). Psychology of Artificial Intelligence (in Bulgarian). *Digitization, big data, artificial intelligence,* Publishing house "Science and Economics" Varna, pp. 74-76.

Sun, S., Fu, Y. (2025). The Role of Mobile Education Technology in Promoting Personalized Learning in Higher Education. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(04), pp. 93–107. https://doi.org/10.3991/ijim.v19i04.54217

Todoranova, A., Todoranova, L. (2025). About Some Spelling Variations of Generation Z (in Bulgarian). *Digital Educational Technologies*, 2 (1). https://doi.org/10.54664/OLSA8007

Tomova, E. (2022). Expectations and concerns – students’ perspectives on e-learning (in Bulgarian). *Conference Proceedings of the Ninth International Conference Electronic learning in higher education, Varna*, pp. 80-88.

Xu, Z. (2023). The impact of ChatGPT in the field of education. *Proceedings of the 2023 International Conference on Machine Learning and Automation*. http://dx.doi.org/10.54254/2755-2721/42/2023079

Zlatkova-Doncheva, K., Marinov, V. (2023). Intonation and children with emotional and behavioral problems, *Az-buki National Publishing House for Education and Science*, 95 (2), pp. 205-213. https://doi.org/10.53656/ped2023-2.06

Received September 20, 2025, revised June 10, 2026, Accepted June 12, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, 417-444
https://doi.org/10.22364/bjmc.2026.14.2.08

A Multi-Algorithmic Method for Ranking B2B Order Success Factors in an ERP Environment

Serhii MIROSHNYCHENKO[1], Oleksandr VOVNA[1,2], Ivan LAKTIONOV[3], Anna MARYNA[2],Viktoriia MIROSHNYCHENKO[1]

[1] Technical University "Metinvest Polytechnic" LLC, Zaporizhzhia, Ukraine
[2] Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
[3]Dnipro University of Technology, Dnipro, Ukraine

serhii.miroshnychenko@mipolytech.education, oleksandr.vovna@knu.ua, laktionov.i.s@nmu.one, anna.maryna@knu.ua, v.i.miroshnichenko@mipolytech.education

ORCID 0009-0007-4868-3006, ORCID 0000-0003-4433-7097, ORCID 0000-0001-7857-6382, ORCID 0000-0001-5634-9402, ORCID 0000-0002-5956-7867

Abstract. Predicting order success in business-to-business environments remains challenging due to the multifactorial complexity of transactions and limitations in traditional forecasting. This study develops a multi-algorithmic method to identify and rank factors influencing order success, with empirical validation in an ERP-based transactional environment. It integrates five feature significance scorers (Area Under the Curve, Mutual Information, distance correlation, Logistic Regression, and Decision Trees) with modified Borda rank aggregation. Analysis of 86,794 orders spanning 2017–2025, with 25 characteristics, revealed that communication factors, particularly message count, demonstrate the strongest influence on success. Historical cooperation experience ranks second, while financial parameters showed ambiguous results, and discounts proved insignificant. Order flexibility positively correlates with success, whereas temporal characteristics showed no influence. The study contributes a methodologically justified five-scorer selection, a modified Borda aggregation method for consolidated scale-independent factor ranking, and a reproducible pipeline for structured transactional data.

Keywords: B2B order prediction; feature ranking; multi-algorithmic analysis; ERP systems; machine learning; rank aggregation; communication factors; order success; decision-making system

1. Introduction

In today's digital economy, accurate forecasting of business transaction outcomes is critical for organizational success and competitiveness. Companies worldwide invest billions in enterprise resource planning (ERP) systems – integrated software platforms that manage core business processes – yet many struggle to predict which orders will succeed or fail. This challenge is particularly acute in business-to-business (B2B) environments, where transactions involve complex negotiations, longer sales cycles, and multiple decision-makers. Understanding the factors that determine order success can help

organizations optimize resource allocation, improve customer relationships, and enhance overall operational efficiency.

Despite widespread ERP adoption for process automation, current procedures for predicting order outcomes remain limited. Classical statistical models often fail to capture the multifactorial nature of B2B transactions, where success depends on numerous interconnected factors, including product characteristics, customer history, market conditions, and resource availability. While modern machine-learning techniques mitigate some of these limitations, they have not yet been combined with rigorous, scorer-neutral procedures for identifying and ranking the underlying success factors – a prerequisite for transparent, decision-support-oriented prediction in ERP environments. A detailed review of prior work on these topics is provided in Section 2.
On this basis, the present study addresses three specific research and development problems that, to the best of our knowledge, are not jointly resolved by existing studies on B2B order success prediction.

P1. Existing feature-importance procedures in this domain typically rely on a single method, whose internal assumptions (linearity, distributional form, capacity to model interactions) systematically bias the resulting ranking; no scorer-neutral aggregation has been established for B2B order success factors.

P2. When several feature significance scorers are applied to the same dataset, their rankings often diverge, and there is no formally specified rule that reconciles such divergence into a single, interpretable ordering whose properties – methodological neutrality, scale invariance, robustness to outliers in individual scorer outputs, interpretational clarity – can be argued for explicitly.

P3. Even when individual scorers are reported in the literature, the corresponding processing pipelines are rarely described at a level of detail that allows independent reproduction in structured transactional settings, including but not limited to ERP-derived data, which limits the operational integration of such results into decision-support modules.

Building on this foundation, we previously developed an enhanced forecasting module for the Odoo ERP platform (Miroshnychenko, 2024) that incorporates additional contextual factors when predicting order quantities. This development creates new requirements for understanding which factors most significantly influence success and how they should be weighted in prediction models, which is essential for creating effective computer-integrated decision support systems that can guide managers in resource allocation and order management.

Our previous studies established foundational insights into B2B order success prediction. A statistical analysis approach (Miroshnychenko and Vovna, 2025) identified communication factors and partner success metrics as dominant predictors using parametric and non-parametric tests with automatic method selection. Subsequently, automated feature selection research (Miroshnychenko et al., 2025) demonstrated the stability of key predictive features across six different selection methods combined with XGBoost hyperparameter optimization. The present work complements these earlier results by introducing a unified ranking method that consolidates multiple, heterogeneous feature significance scorers into a single, interpretable ordering.

The aim of this study is to rank factors influencing B2B order success through a multi-algorithmic feature significance method. We address the research question: which order characteristics have the greatest impact on completion probability, and how should their

relative importance be quantified in a methodologically neutral way? To address problems P1–P3, this study makes the following contributions:

C1. The selection and methodological justification of five mathematically distinct feature significance scorers – Area Under the Curve (AUC) for univariate discrimination, Mutual Information (MI) for information-theoretic dependence, distance correlation (dCor) for scale-invariant nonlinear relationships, Logistic Regression (LogReg) for multivariate linear effects, and Decision Trees (DecTree) for interaction discovery – that capture complementary forms of dependence between predictors and a binary outcome.

C2. A modified Borda rank-aggregation method in which heterogeneous scorer-specific outputs are transformed into ranks and the consolidated rank is computed as the arithmetic mean of scorer-specific ranks rather than as the classical sum of positions; this rule is methodologically neutral, scale-invariant, robust to outliers in individual scorer outputs, and interpretationally clearer than the classical formulation.

C3. A reproducible end-to-end pipeline that operationalizes the proposed method – from raw structured transactional data through type-specific preprocessing (cyclical encoding of temporal variables, OHE Extended Compact encoding of categorical variables, robust scaling of numerical variables) and scorer-specific feature ranking to rank aggregation – and can be applied to other structured datasets and decision-support contexts. The empirical results obtained from 86,794 ERP-based B2B order records over 2017–2025 are reported throughout this paper as an illustrative application of the proposed method to a single Odoo dataset rather than as a generalizable taxonomy of B2B success factors.

2. Related work

Recent studies demonstrate that machine learning (ML) algorithms and artificial intelligence can significantly enhance forecasting performance in sales processes (Hrischev and Shakev, 2023; Lin et al., 2023; Zoltners et al., 2021). For example, random forest models have achieved prediction accuracy with R^2 values approaching 0.94 (where 1.0 represents perfect prediction) in ERP environments (Bauskar, 2024), while supervised ML techniques in B2B analytics have shown 2.5-fold improvements over traditional models (Rohaan et al., 2022). These advances suggest that data-driven techniques can accommodate the complexity inherent in modern business transactions more effectively than purely statistical baselines (Bauskar, 2024).

However, opinions remain divided among researchers regarding the most effective way to predict order success. Some authors advocate for increasingly sophisticated ML algorithms to capture non-linear relationships and feature interactions (Rohaan et al., 2022; Miroshnychenko et al., 2024), while others emphasize the importance of domain knowledge and interpretable models that business users can understand and trust (Bauskar, 2024; Lin et al., 2023). Existing methods further face several unresolved challenges: limited adaptability to rapidly changing market conditions, inadequate accounting for interdependencies among order characteristics, and insufficient incorporation of business-specific factors such as component availability or the status of pending approvals (Miroshnychenko et al., 2024). These limitations highlight a critical gap between technological capability and practical business requirements.

Several studies have attempted to identify success factors for B2B processes using diverse analytical frameworks. Research has explored organizational characteristics

associated with effectiveness (Eid et al., 2002), developed conceptual models for digitalization success (Zoltners et al., 2021), and investigated factors influencing specific business outcomes (Lin et al., 2023; Wilson and Stephens, 2023; Günther, 2021; Høgevold et al., 2022; Rodriguez et al., 2022). However, these works predominantly examine factors in isolation rather than comprehensively evaluating their relative importance or combined effects on order success, which limits the practical application and automated integration of these findings into decision-support systems.

Rank-aggregation methods offer a complementary direction. They enable the reconciliation of multiple rankings derived from different evaluation criteria, providing a structured way to synthesize results from heterogeneous analytical methods. Such methods have demonstrated effectiveness across diverse research domains, including web search (Dwork et al., 2001), bioinformatics (Wald et al., 2012; Sarkar et al., 2014), and environmental decision-making (Vambol et al., 2023), suggesting their suitability for B2B order success ranking. The present study builds on this line of work by introducing a modified Borda rule based on the arithmetic mean of ranks rather than the classical sum of positions, as detailed in Section 3.

3. Materials and methods

The proposed method for a comprehensive analysis of factors (features) influencing a binary success outcome in structured transactional data integrates feature significance scorers from different areas of statistical analysis and machine learning, consolidated through a single rank-aggregation rule. In this study, the method is empirically instantiated for B2B order success using ERP-derived transactional records. The method comprises three sequential stages: type-specific data preprocessing, per-scorer feature significance assessment, and modified Borda rank aggregation.

The initial dataset for the period from July 2017 to January 2025 consists of 86,794 records containing order information, with 25 characteristics (features): 20 numerical (11 floating-point, 9 integer) and 5 categorical (object)[1]. For clarity of interpretation, the substantive meaning, category, data type, and measurement scale of the original analytical features are summarized in Appendix A.

Figures 1–5 present univariate visualizations of the original transaction records using violin plots, directly associating class distributions with quantitative values. These plots illustrate the probability density of representative raw variables across the major feature categories prior to preprocessing, augmented with precise median markers and interquartile ranges to provide immediate quantitative insight into the central tendency and dispersion of each outcome for both successful and unsuccessful orders.

Figure 1 illustrates the distribution of customer relationship duration. For successful orders, the density is characterized by a median of 894.0 days (mean: 1016.5, standard deviation [SD]: 782.7), with an interquartile range (IQR) from 343.0 to 1632.0 days. For unsuccessful orders, the density concentrates around a median of 734.0 days (mean: 838.2, SD: 738.4), with an IQR spanning from 147.0 to 1,335.0 days.

[1] https://github.com/serhii-miroshnychenko-mariupol/PublicData/blob/main/dataset-1.csv

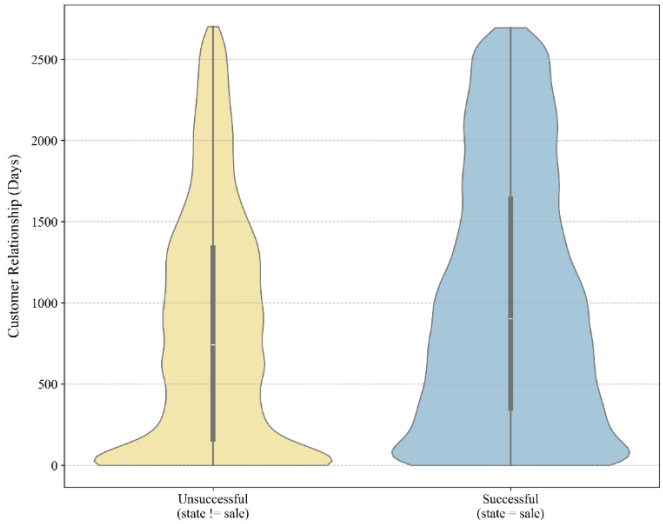


Figure 1. Order success as a function of customer relationship duration (customer_relationship_days)

Figure 2 visualizes the temporal distribution of orders by their creation date. The probability density for successful transactions peaks around a median date of February 23, 2021 (mean: March 30, 2021), bounded by an IQR from May 21, 2019, to March 2, 2023. Correspondingly, the unsuccessful transactions display a density distribution with a median of November 20, 2020 (mean: January 15, 2021) and an IQR spanning from September 18, 2019, to May 4, 2022.

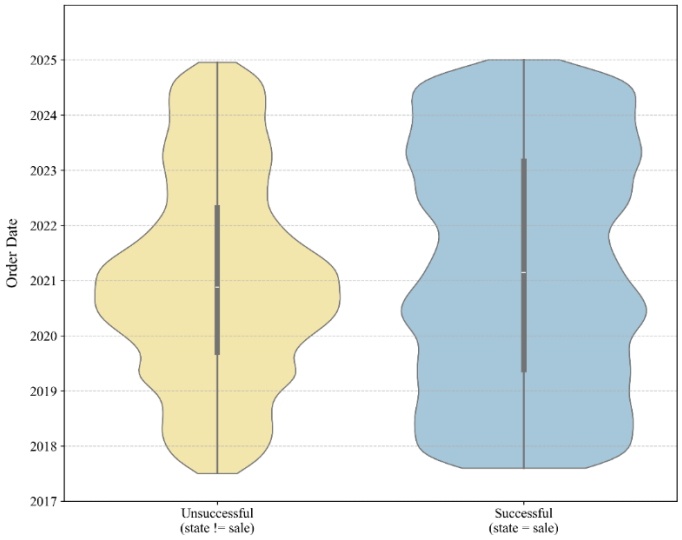


Figure 2. Order success over time by order date (date_order)

Figure 3 displays the distribution of the prior transaction history, presented on a logarithmic scale (log10) to accommodate high-frequency outliers while preserving resolution for smaller values. Successful conversions exhibit a median of 23.0 previous orders (mean: 70.04, SD: 137.7, IQR: 5.0–72.0). Conversely, the unsuccessful orders demonstrate a median of 9.0 (mean: 39.73, SD: 78.91, IQR: 1.0–42.0).

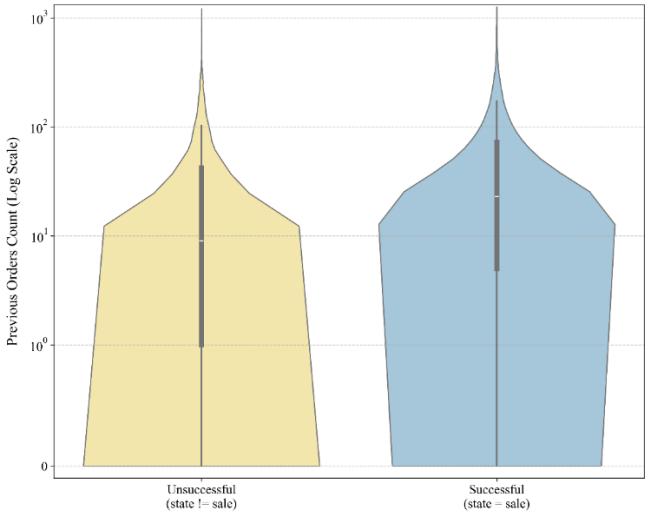


Figure 3. Order success as a function of prior transaction history (previous_orders_count)

Figure 4 details the intraday distribution of order placement times. The density curves for both classes share an identical median of 11:00 AM. For successful orders, the mean is 10.79 with a standard deviation of 2.83 and an IQR from 08:00 to 13:00. Unsuccessful orders show a mean of 11.18 (SD: 2.97) and an IQR extending from 09:00 to 14:00.

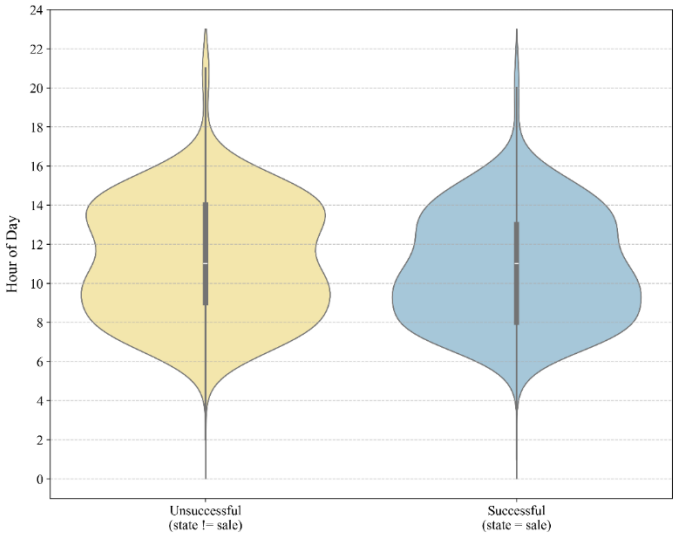


Figure 4. Order Success as a Function of Order Placement Time (hour_of_day)

Figure 5 visualizes the density distribution of communication intensity, utilizing a logarithmic scale due to a heavily right-skewed distribution. Successful interactions indicate a broader spread toward higher values with a median of 8.0 messages (mean: 10.81, SD: 8.97, IQR: 6.0–12.0). In contrast, the unsuccessful orders are characterized by a denser concentration around lower values, showing a median of 5.0 messages (mean: 6.03, SD: 5.39, IQR: 4.0–7.0).

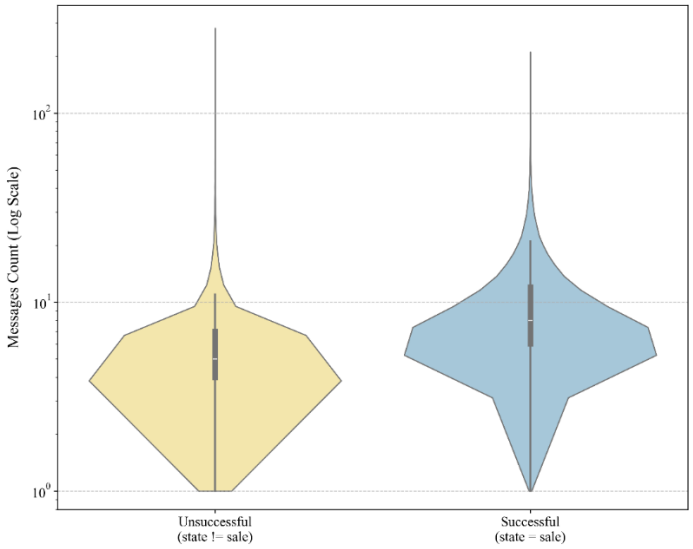


Figure 5. Order success as a function of communication intensity (messages_count)

The target variable is the binary indicator of order success – 'is_successful'. If an order was approved and successfully completed, the target variable equals 1; otherwise, for unapproved or canceled orders for any reason, 'is_successful' equals 0. Initial analysis of the target variable distribution revealed class imbalance – approximately 63% of orders are successful, which should be considered in subsequent modeling. Preliminary analysis indicated significant variance and extreme values in numerical features, especially order amounts and partner history characteristics. The set of potential predictors (features) is formed based on order and customer characteristics. Features that either contained no predictive information or caused direct target leakage were removed from the analyzed dataset.

An important aspect of data preparation is processing temporal characteristics. The 'create_date' variable, reflecting the order creation date and time, was transformed to extract temporal components to identify possible temporal dependencies. We extracted the following characteristics: day of week, month, quarter, and hour of day when the order was placed. This transformation allows identification of potential seasonal fluctuations and cyclicality in order placement processes. For cyclical temporal features such as 'day_of_week', 'month', and 'hour_of_day', we applied cyclical encoding using sinusoidal and cosine transformations (Chakraborty and Elzarka, 2018), ensuring the correct representation of temporal cycles in the multidimensional feature space. This transformation was implemented using the feature_engine library[2], providing standardized and efficient conversion of cyclical variables. Based on 'create_date', we also created the 'create_date_months' parameter, whose value represents the temporal distance of a specific order's creation moment from the initial observation point, measured in months, where the date of the first recorded system order serves as the starting point. This transformation allows the time series to be represented as a monotonically increasing numerical sequence reflecting the temporal distance between orders in months. The 'create_date_months' variable can be used as a regressor in ML models to account for temporal dependencies and evolution of order characteristics over time. Additionally, this variable enables analysis of order success dynamics over time, which may reveal the influence of seasonality and long-term trends.

Categorical variables such as 'salesperson' (responsible for sales) and 'source' (order source) require special processing for integration into mathematical models. Analysis of unique values revealed moderate cardinality for the 'salesperson' variable and somewhat higher cardinality for 'source', with 31 and 66 unique values, respectively. For these categorical variables, we applied the OHE Extended Compact encoding method, implemented according to (Ul Haq et al., 2019). This method avoids creating false ordinal relationships between categories while compressing sparse data, helping prevent excessive dataset expansion.

Since the numerical variables in the dataset exhibit significant scale variability and contain extreme values (outliers), they require appropriate processing and scaling. Specifically, statistical tests determined that variables such as 'order_amount', 'partner_total_orders' (total number of partner orders), 'partner_avg_amount' (average partner order amount) have wide value ranges and asymmetric distributions. For

[2] https://feature-engine.trainindata.com/en/1.8.x/user_guide/creation/CyclicalFeatures.html

normalizing numerical variables, we applied the Robust Scaler method[3], which, by using median and interquartile range for centering and scaling data, significantly reduces the influence of extreme values on scaling results, which is particularly important for variables with pronounced distribution asymmetry, as in the case of 'order_amount'. To detect outliers in numerical variables, we used the interquartile range method (Dekking et al., 2005) – a robust statistical technique that identifies outliers based on their distance from the distribution quartiles. Outlier analysis using this method revealed a significant number of extreme values in variables such as 'order_amount', 'partner_avg_amount', 'partner_success_avg_amount' (average amount of successful customer orders), and 'order_lines_count' (number of items in order). Given the potential information these outliers may provide for predicting order success, we decided to apply the above-described robust scaling techniques to minimize their impact on the model.

Following preprocessing (see Fig. 6), we obtained the final analytical dataset for subsequent modeling. The preprocessing procedures used correspond to the specificity of the analyzed dataset and requirements for subsequent modeling, particularly accounting for the presence of cyclical variables, categorical features with moderate cardinality, and the presence of extreme values in numerical variables. This produced a consistent analytical feature matrix for subsequent feature significance assessment.

An important feature of this study is the need to assess significance for both numerical and categorical features after their respective encoding. The selected set of scorers enables comprehensive analysis of B2B order data from multiple perspectives, ranging from univariate statistical analysis to complex ML models that account for feature interactions. All five scorers are distribution-free or robust to distributional assumptions, which is critically important for analyzing real B2B business data that often deviate from theoretical distributions. The selected scorers also work effectively with both numerical and categorical variables after proper encoding, allowing full analysis of heterogeneous data.

The stages shown in Fig. 6 are operationalized as follows.

Stage I (Feature engineering and removal of irrelevant variables). Prior to the statistical analysis, raw records exported from the Odoo PostgreSQL backend are transformed into an analytical dataset with 25 features. This stage removes three categories of variables: (a) technical identifiers that could enable memorization rather than generalization ('id', 'order_id', 'partner_id', 'customer_id'); (b) low-predictive-value B2B attributes whose distributions in the analyzed environment are dominated by a single modality or are operationally redundant ('sales_team', 'customer_category', 'customer_country', 'product_categories', 'payment_term', 'delivery_method'); and (c) post-event target-leakage variables, most notably processing_time_hours, whose values become available only after the order has been completed and would therefore be unavailable at prediction time. All intermediate columns used exclusively for the computation of historical aggregates are also discarded. In addition, historical partner-level aggregates ('partner_*') are constructed under a temporal leakage prevention rule: when a feature such as 'partner_success_rate' or 'partner_avg_amount' is computed for a given order, the contribution of that same order is subtracted from the cumulative sum, so that only information available before order creation is used. Finally, a near-zero-variance filter is applied at a 95% majority-value threshold: features in which more than 95% of

[3] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

observations share an identical value after the cleaning steps above are treated as non-informative and are excluded from the analytical dataset.

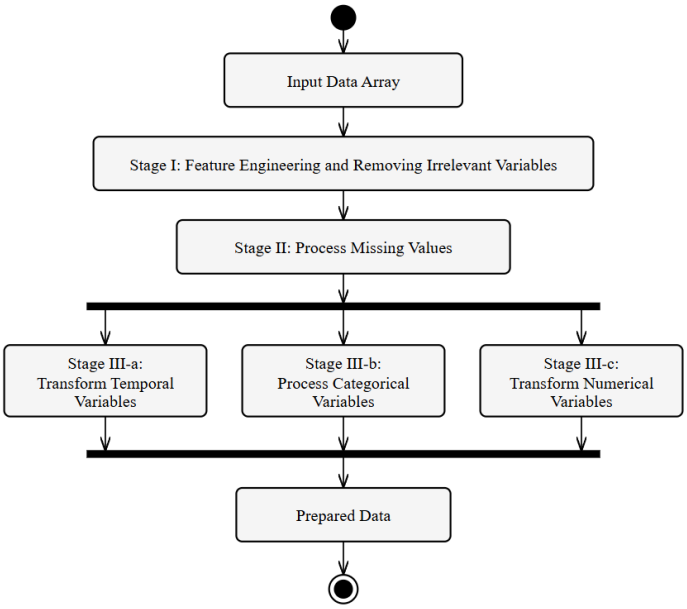


Figure 6. Data Preprocessing Workflow Diagram

Stage II (Processing of missing values). Missing values in numerical variables are imputed by the column median, and in categorical variables by the column mode (majority value); these strategies are robust to the asymmetric distributions and class imbalance present in the analyzed dataset and avoid introducing artificial trends.

Stage III-a (Transformation of temporal variables). From the raw 'create_date' variable we extract the components 'day_of_week', 'month', 'quarter', and 'hour_of_day', as well as the monotonic distance feature 'create_date_months' (months elapsed since the first recorded order). The cyclical components are subsequently encoded by sinusoidal and cosine projections via the CyclicalFeatures transformer of the feature_engine library[2] (Chakraborty and Elzarka, 2018). After the extraction of these components, the original 'create_date' column is itself discarded from the analytical feature matrix.

Stage III-b (Processing of categorical variables). The categorical variables 'salesperson' (31 unique values) and 'source' (66 unique values) are encoded by the OHE Extended Compact (OHE-EC) procedure of (Ul Haq et al., 2019), which transforms each categorical attribute with d distinct values into d+1 indicator attributes (the additional attribute reserves capacity for previously unseen values, which is appropriate for 'source' whose value space evolves over time) and then applies sparsity-aware compaction by retaining only non-zero indicators together with their feature index. We use the High Distribution First (HDF) variant, in which indicator attributes are ordered by descending value distribution; the implementation follows the original specification (Ul Haq et al., 2019), as no public package was available at the time of analysis.

Stage III-c (Transformation of numerical variables). Seventeen numerical variables ('order_amount', 'order_messages', 'order_changes', 'partner_total_orders', 'partner_order_age_days', 'partner_avg_amount', 'partner_success_avg_amount', 'partner_fail_avg_amount', 'partner_total_messages', 'partner_success_avg_messages', 'partner_fail_avg_messages', 'partner_avg_changes', 'partner_success_avg_changes', 'partner_fail_avg_changes', 'order_lines_count', 'discount_total', 'create_date_months') are scaled with RobustScaler from scikit-learn[3], which centers each variable on its median and scales by its interquartile range, ensuring resistance to outliers identified via the IQR rule (Dekking et al., 2005).

The outputs of Stages III-a, III-b, and III-c are merged column-wise within the same row index of the order-level table, producing the analytical feature matrix ("Prepared data") used as the common input to all five feature significance scorers.

In the context of feature significance assessment, the Area Under the Curve (AUC) scorer, based on evaluating the area under the ROC curve (Fawcett, 2006), is used to assess an individual feature's ability to separate positive and negative classes (successful and unsuccessful orders). Mathematically, AUC is the probability that a randomly selected observation from the positive class has a higher predicted rank than a randomly selected observation from the negative class (Calders and Jaroszewicz, 2007). We chose the AUC scorer for assessing feature significance because it is effective with binary target variables, insensitive to feature scaling, and able to evaluate monotonic relationships between features and target variables. The method's advantage is that it requires no assumptions about data distribution and works effectively with both numerical and categorical features after encoding. In this study, we used the AUC implementation in the sklearn library, specifically the roc_auc_score function, to identify features that best separate successful and unsuccessful orders, without considering complex feature interactions.

To identify non-obvious patterns between order attributes and customer interaction effectiveness, we calculated the MI indicator, which measures the amount of information that one random variable X provides about another variable Y. In the context of feature selection tasks, MI is used as a criterion for evaluating feature significance based on their ability to reduce uncertainty about the target variable. In practice, particularly in the Python scikit-learn library, MI for classification is implemented using the mutual_info_classif function, which supports both numerical and discrete features. Computation is based on non-parametric entropy estimates, with noise added to continuous variables to avoid artifacts caused by repeated values (Kraskov et al., 2004). Thus, automatic processing of mixed feature types is available, as well as parallel computation across features[4]. One of the main advantages of MI over correlation coefficients is its ability to capture both linear and complex nonlinear dependencies (Ross, 2014). This property makes MI an effective tool in cases where complex relationships are expected, such as in B2B order effectiveness prediction tasks, thereby enabling more accurate decision-making models.

In this study's context, to identify features with any statistical relationship with order success, we applied the dCor scorer using the dcor library[5] as a complement to other

[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html

[5] https://dcor.readthedocs.io/en/stable/functions/dcor.distance_correlation.html

applied scorers that might miss certain types of dependencies. dCor is a statistical measure that evaluates the degree of dependence between two random vectors and can identify both linear and nonlinear dependencies, making it a powerful tool for analyzing complex data. A key property of the method is that dCor equals zero if and only if variables are statistically independent, thus providing a reliable independence test (Székely et al., 2007).

To analyze order success based on modeling binary outcome probabilities, we applied LogReg, which allows determining event probability as a function of a set of independent variables (Kleinbaum and Klein, 2010; Lee, 2025). This method allows assessment of both the strength and direction of each feature's influence on success probability, and provides statistical significance tests for this influence that account for the simultaneous influence of all features. In data processing for this study, we used the LogisticRegression class from the sklearn library[6] to obtain interpretable results and more reliable significance estimates for binary target variables compared to methods developed for continuous variables.

To account for nonlinear interactions between features and identify the most discriminative ones, we applied DecTree – a non-parametric ML method that creates prediction models by learning decision rules derived from studied data (Breiman et al., 2017). Unlike statistical methods that evaluate each feature separately, DecTree considers features in the context of their interaction, which is particularly important for complex business data. This scorer requires no assumptions about data distribution and performs equally well across different feature types, meeting the requirements of this study. In the practical context of B2B order analysis, applying DecisionTreeClassifier from the sklearn library[7] enables the identification of key factors influencing order success and the specification of specific threshold values for these factors, thereby enhancing the practical value of the analysis results.

The mathematical diversity of applied scorers ensures independent data analysis by each method, creating a kind of "cross-validation" system for results that minimizes the risk of identifying false patterns and increases confidence in features identified as significant. Thus, using these five scorers enables comprehensive analysis of feature significance, avoids the limitations inherent to individual scorers, and provides a reliable foundation for subsequent predictive model development and business decision-making.

In the context of comprehensive feature significance analysis for predicting B2B order success using multiple heterogeneous scorers, a methodological challenge arises: integrating the obtained results. The adopted approach is based on the rank aggregation concept (Dwork et al., 2001), which allows reconciling results from scorers that potentially use different scales and evaluation principles.

Let $F = \{f_1, f_2, \dots, f_n\}$ be the set of studied features, and $M = \{m_1, m_2, \dots, m_k\}$ be the set of significance evaluation scorers, where in this study $k = 5$ (AUC, MI, dCor, LogReg, DecTree). Each scorer m_j assigns feature f_i a certain significance value $S_j(f_i)$, which reflects the degree of this feature's influence on the target variable according to scorer m_j's mathematical apparatus. To ensure comparability of results from different scorers, we implement a transformation to a rank scale. The rank of feature f_i by scorer m_j is defined by Eq. (1):

[6] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

$$R_j(f_i) = |\{f_l \in F | S_j(f_l) \geq S_j(f_i)\}|, \quad (1)$$

where f_i is the feature for which rank is calculated; F is the complete set of all studied features; f_l is any feature from set F that serves as a variable in comparison; $S_j(f_i)$ is the significance value that scorer m_j assigns to feature f_i; $S_j(f_l) \geq S_j(f_i)$ is the condition that feature f_l's significance is not less than feature f_i's significance; $\{f_l \in F | S_j(f_l) \geq S_j(f_i)\}$ is the set of all features whose significance is not less than feature f_i's significance; $|\cdot|$ is set cardinality, i.e., the number of elements satisfying condition $S_j(f_l) \geq S_j(f_i)$.

In contrast to the classical Borda method (Dwork et al., 2001), which aggregates complete rankings through a sum-of-positions logic, the proposed modification applies the arithmetic mean of the scorer-specific ranks:

$$\bar{R}(f_i) = \frac{1}{k}\sum_{j=1}^{k} R_j(f_i). \quad (2)$$

where $R_j(f_i)$ is the rank of feature f_i assigned by scorer m_j according to Eq. (1), k is the number of scorers used in the aggregation, and $\bar{R}(f_i)$ is the resulting mean rank of feature f_i. In this study, $k = 5$, corresponding to AUC, MI, dCor, LogReg, and DecTree.

This integral indicator $\bar{R}(f_i)$ serves as a generalized measure of feature f_i significance, considering results from all involved scorers. A lower value of $\bar{R}(f_i)$ indicates higher consolidated feature significance. The consolidated ordering is obtained by applying Eq. (1) to each (feature, scorer) pair and then aggregating the resulting per-method ranks via Eq. (2).

The proposed modification to the Borda method has several methodological advantages. First, the rule is methodologically neutral, giving no preference to any scorer and treating each as an equal contributor to the final assessment, thereby minimizing methodological biases. Second, it is invariant with respect to measurement scales, as rank transformation eliminates the problem of incomparable scales from different scorers. Third, it is robust, since the use of ranks instead of absolute values reduces sensitivity to outliers and extreme values in individual scorer outputs. Fourth, it offers interpretational clarity, as the mean rank has a clear interpretation as the feature's averaged position in the significance-ordered list. Within the overall workflow shown in Fig. 7, the modified Borda stage is used to calculate the mean rank for consolidated feature ordering.

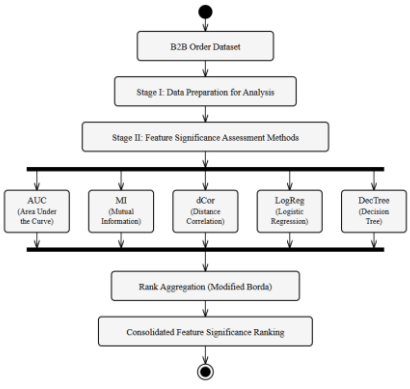


Figure 7. Flowchart of the feature significance assessment process

The proposed modification to the Borda method has several methodological advantages. First, the rule is methodologically neutral, giving no preference to any scorer and treating each as an equal contributor to the final assessment, thereby minimizing methodological biases. Second, it is invariant with respect to measurement scales, as rank transformation eliminates the problem of incomparable scales from different scorers. Third, it is robust, since the use of ranks instead of absolute values reduces sensitivity to outliers and extreme values in individual scorer outputs. Fourth, it offers interpretational clarity, as the mean rank has a clear interpretation as the feature's averaged position in the significance-ordered list. Within the overall workflow shown in Fig. 7, the modified Borda stage is used to calculate the mean rank for consolidated feature ordering.

The execution of the workflow shown in Fig. 7 is parameterized as follows.

Stage I (Data preparation for analysis). The analytical feature matrix produced by the preprocessing pipeline of Fig. 6 is supplied as the common input to all five feature significance scorers, ensuring that ranking differences across scorers reflect the mathematical properties of the scorers rather than differences in feature representation.

Stage II (Feature significance assessment). Each of the five scorers (AUC, MI, dCor, LogReg, DecTree) produces a per-feature significance score on this common input, using the specific implementations and hyperparameters detailed in the per-scorer descriptions above.

Stage III (Modified Borda rank aggregation). Per-scorer scores are independently transformed into ranks using Eq. (1); these ranks are aggregated into the consolidated ordering using Eq. (2), in contrast to the classical sum-of-positions Borda score. The Python implementation of this aggregation is provided in Appendix B. Output. The output of the workflow is the consolidated ranking of features, which is examined empirically in Sections 4 and 5. By construction, this multi-stage procedure compensates for the methodological limitations of any individual scorer and provides a more reliable and balanced assessment of feature significance for subsequent use in modeling processes and decision-making.

4. Results

Upon applying the multi-algorithmic method described in Section 3 to the analyzed dataset, we derived feature importance scores for each of the five feature significance scorers. These raw scores, shown in Table 1, are subsequently transformed into per-scorer ranks via Eq. (1) and consolidated through the modified Borda mean rank of Eq. (2) into the ordering visualized in Fig. 8.

Subsequently, we constructed a heatmap depicting the relative significance of features across these scorers (see Fig. 8), where the numbers in the cells correspond to the per-scorer ranks computed using Eq. (1), "Average Rank" denotes the mean rank calculated using Eq. (2), and "Overall Rank" represents the final aggregated ordering (ties in "Average Rank" share the same overall rank).

Table 1. Feature importance scores by different scorers

Technical Feature Name	AUC	MI	dCor	LogReg	DecTree
order_messages	0.7708	0.1323	0.4512	2.7226	0.41
order_amount	0.5537	0.1077	0.2287	2.8973	0.1506
partner_success_rate	0.6952	0.0604	0.3834	0.906	0.1474
order_changes	0.6418	0.0432	0.2873	0.8465	0.0363
partner_success_avg_messages	0.5285	0.0531	0.1873	0.378	0.0026
partner_total_orders	0.5859	0.0174	0.1128	0.364	0
partner_total_messages	0.5896	0.0171	0.116	0.3362	0
partner_fail_avg_changes	0.5386	0.0191	0.0815	0.3101	0
create_date_months	0.5252	0.0166	0.1157	0.0547	0.2304
partner_success_avg_amount	0.5606	0.0719	0.1067	0.1775	0.0166
partner_avg_changes	0.5698	0.0195	0.1528	0.1591	0
partner_success_avg_changes	0.5393	0.0504	0.1786	0.1226	0
order_lines_count	0.5826	0.0377	0.1672	0.0739	0.0058
partner_fail_avg_messages	0.5425	0.0202	0.0804	0.1589	0.0004
partner_order_age_days	0.5751	0.0218	0.1337	0.0062	0
partner_avg_amount	0.5146	0.0212	0.0939	0.074	0
partner_fail_avg_amount	0.5033	0.0637	0.051	0.0134	0
source	0.5009	0.0214	0	0.0857	0
salesperson	0.5201	0.0155	0	0.0314	0
hour_of_day	0.5094	0.0098	0	0	0
month	0.5059	0.0122	0	0	0
discount_total	0.5002	0.0003	0.0105	0.0071	0
day_of_week	0.5003	0.0129	0	0	0
quarter	0.5	0.0101	0	0	0

Among the values reported in Table 1, ‘order_messages’ shows the highest scores across all five scorers (AUC = 0.7708, MI = 0.1323, dCor = 0.4512, LogReg = 2.7226, DecTree = 0.41). By contrast, ‘order_amount’ displays a less uniform pattern, with comparatively high values for MI = 0.1077 and LogReg = 2.8973, but a lower AUC = 0.5537. A similar cross-method difference is observed for ‘partner_success_rate’, which attains relatively high AUC = 0.6952 and dCor = 0.3834 values, while its MI score remains lower (0.0604). At the lower end of the table, ‘discount_total’ and the fine-grained temporal features (‘hour_of_day’, ‘month’, ‘day_of_week’, ‘quarter’) show near-baseline or zero values across most scorers.

Specifically, AUC gives the highest evaluation to features related to communication and customer success history. We observe a clear hierarchy in which current order characteristics and key historical metrics, such as ‘partner_success_rate’, are most

significant, while categorical and temporal features are least important. MI gives greater weight to financial indicators compared to other scorers, with three of the top 5 features relating to financial indicators, particularly order amount (‘order_amount’), average amount of successful customer orders (‘partner_success_avg_amount’), and average amount of failed customer orders (‘partner_fail_avg_amount’). Additionally, MI assigns a non-trivial importance to the order source ('source'), a feature largely ignored by other scorers like dCor and Decision Tree.

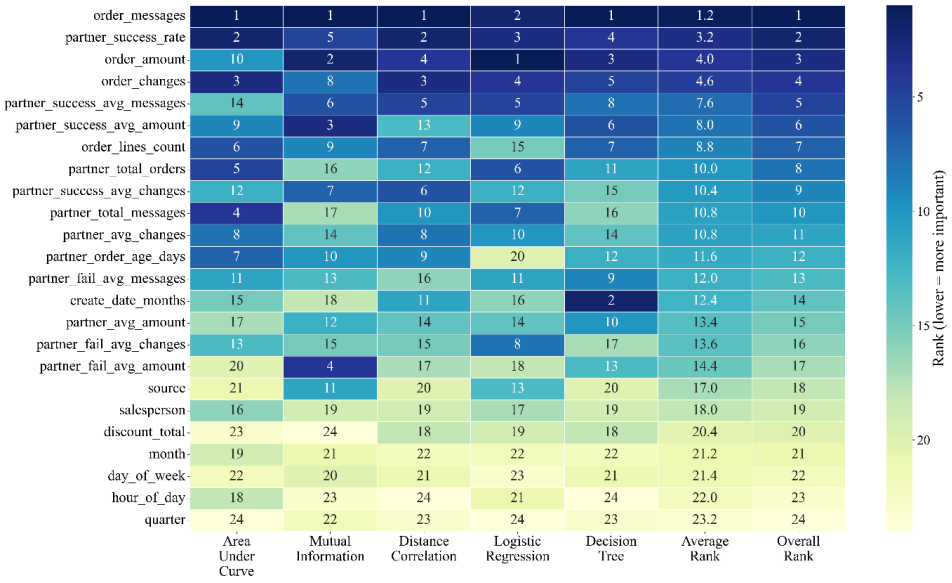


Figure 8. Feature importance rank heatmap by different scorers

dCor results are similar to AUC but with less distinction between values. However, for the most granular temporal features (hour, day, month, quarter), their unimportance is clearly demonstrated by zero correlation, suggesting the absence of significant nonlinear dependence between them and order success. LogReg clearly identifies ‘order_amount’ and ‘order_messages’ as the most important features, while rating the average number of changes in failed customer orders (‘partner_fail_avg_changes’) relatively high compared to other scorers. DecTree demonstrates the greatest selectivity in feature importance, identifying ‘order_messages’, ‘create_date_months’, and ‘order_amount’ as the most important, and concentrating the importance of success information in 9 key features. Based on these results, we can conclude that different feature significance scorers complement each other, enabling a comprehensive understanding of the predictors of order success in the studied system. This aligns with the study's results (González-Flores et al., 2025), which analyzed 15 B2B lead classification algorithms and showed that algorithm effectiveness varies depending on context.

To provide empirical confirmation of the aggregated ranking results, we present a set of binned success-rate plots illustrating how selected predictors relate to order success. Each visualization combines binned or categorical success rates with compact statistical

summaries, including sample-size information, relevant association metrics, fitted trends where applicable, and Wilson-based uncertainty summaries for success rates.

The success rate demonstrates a pronounced positive correlation with the number of exchanged messages (`order_messages`). Across 20 equal-frequency bins (N = 86,794), the success rate increases from 7.7% for 1–3 messages to 90.2% for 23–282 messages. The association is statistically pronounced (Spearman rho = 0.454, $p < 0.001$; point-biserial r = 0.282, $p < 0.001$), and the weighted linear trend is 3.75 percentage points per bin ($R^2 = 0.806$; see Fig. 9).

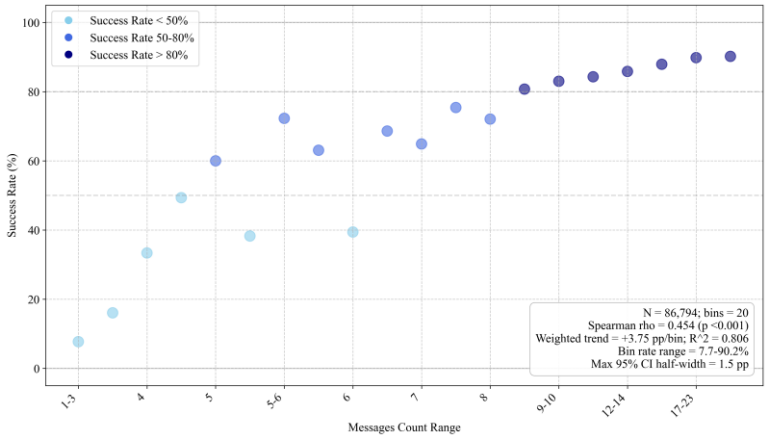


Figure 9. Success rate by message count

An overall positive trend is observed between the number of order modifications (`order_changes`) and the success rate. Orders with minimal changes (0–2) exhibit the lowest success rate (45.9%), while orders with 6 or more changes achieve 86.3%. The positive association is confirmed by Spearman rho = 0.241 ($p < 0.001$) and point-biserial r = 0.194 ($p < 0.001$), with a weighted trend of 6.45 percentage points per bin ($R^2 = 0.640$; see Fig. 10). This indicates that iterative alignment with customer requirements is associated with higher completion probability.

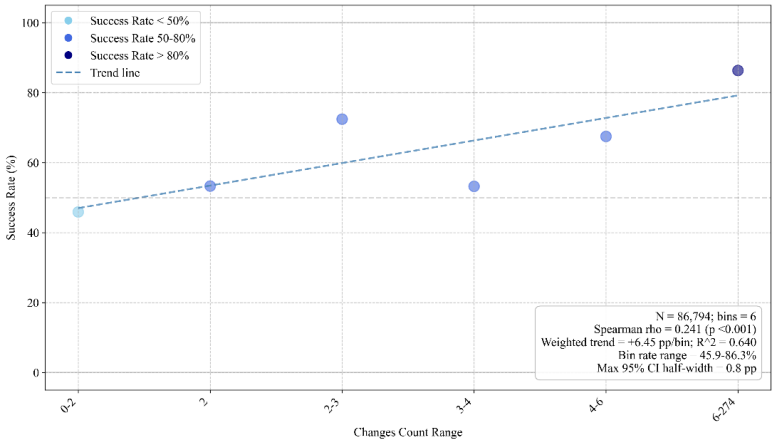


Figure 10. Success rate by changes count

The relationship between success rate and order amount (`order_amount`) exhibits a nonlinear, inverted U-shaped pattern. Among orders with a positive amount (N = 67,242), maximum success is observed in the mid-range price segment, peaking at 84.2% in the 710–789 USD bin. Both near-zero orders (5.3%) and high-value orders above approximately 6,000 USD (<50%) demonstrate substantially lower success rates. The monotonic association is negative but modest (Spearman rho = -0.150, $p < 0.001$; point-biserial r = -0.117, $p < 0.001$), whereas the quadratic fit explains the binned pattern much better ($R^2 = 0.736$; see Fig. 11). This nonlinearity is one of the reasons for the discrepancy in the assessments of the importance of this feature across different scorers.

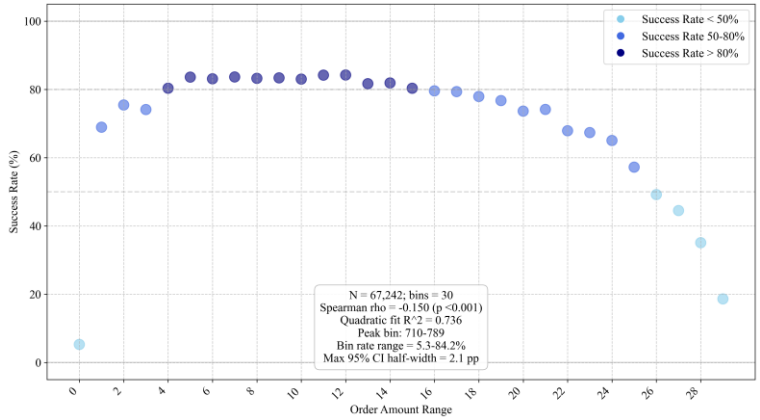


Figure 11. Success rate by order amount

Discount-related variables demonstrate weak marginal effects. Orders without discounts have a 63.1% success rate (n = 86,705), whereas orders with discounts have a 73.0% success rate (n = 89). This difference is borderline in a two-proportion test (p = 0.052) and has a negligible effect size (Cramer's V = 0.007). Within the positive-discount subset, success rates range from 55.6% to 83.3% across five bins, but the very small sample and wide confidence intervals (maximum Wilson 95% CI half-width = 20.9 percentage points) make this pattern exploratory rather than confirmatory (see Fig. 12).

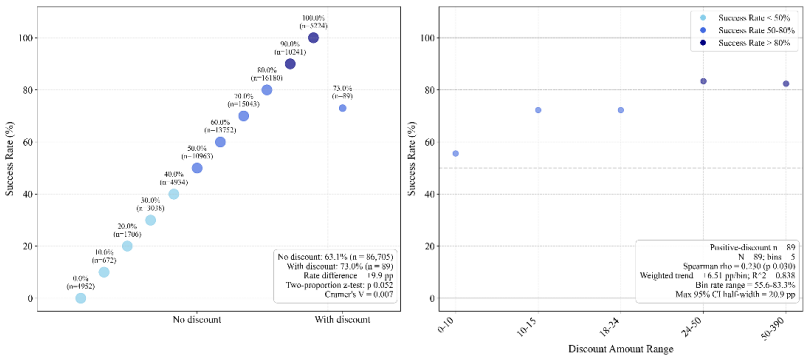


Figure 12. Success rate by discount

Success rate variation across weekdays is minimal: values range from 61% to 72%, with no pronounced pattern. Friday exhibits the lowest rate (61.3%), while Saturday and Sunday show slightly higher values (70.5% and 72.1%, respectively), although weekend order counts are small (139 and 122). The chi-square test is statistically significant due to the large total sample ($\chi^2(6) = 51.23$, $p < 0.001$), but the effect size is negligible (Cramer's $V = 0.024$), confirming the consolidated low importance of temporal factors in the ranking (see Fig. 13). Point size is proportional to the number of orders on each day.

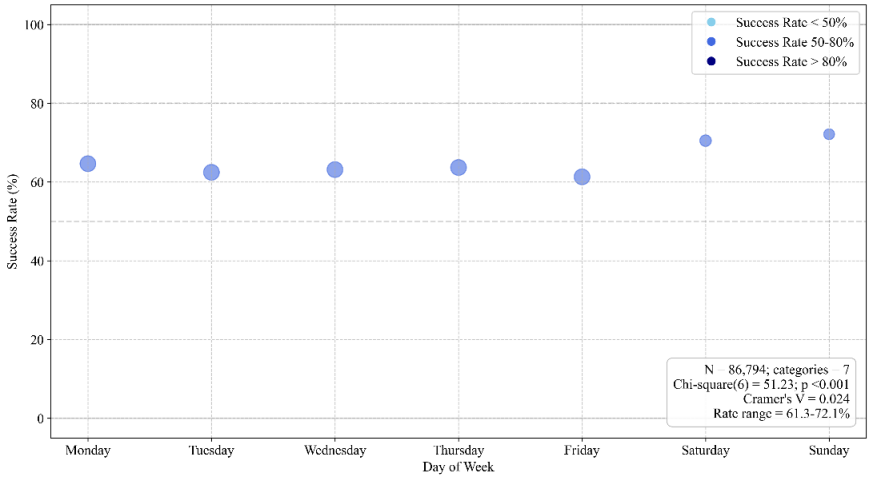


Figure 13. Success rate by day of week

The presented visualizations serve as marginal (single-feature) illustrations that complement the multi-scorer feature ranking by making the functional form of each relationship explicit. Communication intensity (`order_messages`) and order flexibility (`order_changes`) exhibit positive associations with success, corroborating their leading positions in the aggregated ranking. The nonlinear, inverted U-shaped dependence of `order_amount` explains the divergent importance scores across scorers. Discount and temporal variables show negligible or sample-limited effects, consistent with their low consolidated ranks. These empirical patterns reinforce the quantitative findings and highlight the primacy of interaction quality over monetary and temporal factors in B2B order outcomes.

Overall, the aggregated ranking indicates that the analyzed predictors form several empirical groups of B2B order success determinants, including communication-related, customer-history, order-flexibility, financial, and temporal factors. Communication-related and customer-history variables occupy the highest positions in the consolidated ranking, order-flexibility factors show intermediate importance, whereas financial indicators remain heterogeneous and temporal factors consistently rank among the weakest predictors in the analyzed dataset.

5. Discussion

Ranking results confirm the key role of communication in determining order success, as most scorers identified message count as the most important factor. This result aligns with the study's conclusions (Fraccastoro et al., 2020), which emphasize that communication tools are fundamental for establishing and maintaining relationships across various B2B process stages. Research on digital communication in business networks (Sashi, 2021) also confirms the role of technological innovations in value co-creation. Empirical studies of communication's influence on purchase decisions (Popescu et al., 2017) demonstrate that communication aspects can determine choice regardless of direct product utility. Research on the influence of the communication process on construction enterprise sales (Akintelu et al., 2023) emphasizes the importance of communication within ongoing business relationships.

The substantial influence of historical interaction experience, expressed through the average percentage of successful customer orders ('partner_success_rate'), is confirmed by the second position, correlating with the RFID model for the B2B environment (Marín Díaz et al., 2022) and studies of customer loyalty to companies and loyalty programs (Evanschitzky et al., 2011). Similar dependence is confirmed by observations (Ferro-Soto et al., 2025) and research on customer loyalty factors for client retention in a digital environment (Pereira et al., 2025).

The order amount, identified by LogReg and MI scorers as most important, ranks only tenth by AUC, suggesting its influence depends on the chosen evaluation method. The study (Sadaram et al., 2023) explains this through volatility and the presence of noisy signals in financial metrics. The low significance of the total discount amount (20th place) contradicts common perceptions. Research on discount perception (Jee, 2021) shows a positive influence of price incentives, but, according to the results of this study, relationship quality parameters proved more important in the B2B context.

The importance of flexibility and adaptability to customer requirements during order processing is confirmed by order_changes' fourth position in the ranking. Recent research (Guesalaga et al., 2023) confirms flexibility as a critical success factor for B2B interactions.

The particularly low importance of temporal indicators suggests relative independence of order success from temporal patterns, contradicting research on demand seasonality (Guo et al., 2021) and industry research on seasonal conditions' influence (Smith, 2025). Such results can be attributed to the greater levels of organization and standardization characteristic of well-established B2B environments, which enhance business process predictability, as demonstrated in the study (Rezazadeh, 2020).

These findings demonstrate remarkable consistency with our previous research on the same dataset using different methods. Four features exhibited the highest predictive importance across all three studies: (1) 'order_messages' emerged as the dominant factor: rank 1 in the current multi-algorithmic analysis, highest weight (0.5421) in our statistical study (Miroshnychenko and Vovna, 2025), and among nine stable features in the XGBoost selection study (Miroshnychenko et al., 2025); (2) 'order_changes' showed consistent importance: rank 4 (current), rank 3 with weight 0.2828 (Miroshnychenko and Vovna, 2025), and top-9 stable (Miroshnychenko et al., 2025); (3) 'order_lines_count' reflecting order complexity: rank 7 (current), rank 6 with weight 0.1666 (Miroshnychenko and Vovna, 2025), and top-9 stable (Miroshnychenko et al., 2025); (4) 'partner_success_avg_amount' capturing historical financial performance: rank 6

(current), rank 9 with weight 0.1307 (Miroshnychenko and Vovna, 2025), and top-9 stable (Miroshnychenko et al., 2025).

Notably, 'partner_success_rate' ranked second in both the current analysis and statistical study (Miroshnychenko and Vovna, 2025) (weight 0.3956), but was excluded from the top-9 universally selected features in (Miroshnychenko et al., 2025), suggesting partial redundancy with related partner metrics.

This cross-methodological triangulation strengthens confidence in these factors as fundamental determinants of B2B order success rather than scorer-specific artifacts.

Distinguishing confirmatory from novel contributions. The factor-by-factor agreement reported above with prior single-method or single-discipline studies represents the confirmatory component of the present work. The novel component, in contrast, lies in three elements that have not been systematically combined in the existing B2B order-success literature: (i) the selection and methodological justification of five heterogeneous feature significance scorers that capture complementary forms of dependence between predictors and a binary outcome; (ii) a scorer-neutral, scale-independent aggregation method – the modified Borda mean rank – whose properties (methodological neutrality, scale invariance, robustness to outliers in individual scorer outputs, interpretational clarity) are explicitly argued for, rather than implicitly assumed; and (iii) a reproducible end-to-end pipeline that connects raw structured transactional data, type-specific preprocessing, scorer-specific feature ranking, and rank aggregation, with ERP-based B2B order data used here as an empirical demonstration rather than as the boundary of applicability.

Theoretical implications. The results are consistent with a relational view of B2B transactions, in which the probability of order completion is strongly associated with the quality and intensity of dyadic interaction rather than with transactional or temporal context alone. From a methodological standpoint, the modified Borda mean rank enables a stability-oriented interpretation of feature importance by comparing scorer-specific ranks: features whose ranks vary substantially across scorers (e.g., 'order_amount') are not necessarily uninformative, but their predictive contribution is non-monotonic and scorer-sensitive, and should therefore be treated with scorer-aware care. This complements the predictive-accuracy criterion that dominates ML-driven sales forecasting (Rohaan et al., 2022; Bauskar, 2024) with an assessment of how robust feature-importance conclusions are across heterogeneous scorers, thereby addressing the divided-opinions issue identified in Section 2.

Practical implications. Operationally, the proposed method provides a generic integration pattern for decision-support workflows based on structured transactional data: (i) the five feature significance scorers are run on the pre-processed feature matrix; (ii) the resulting ranks are consolidated using the modified Borda mean rank rule; and (iii) the consolidated ranking, together with the spread of scorer-specific ranks visible in the ranking heatmap, can be used as a feature-importance layer in the analyzed decision-support process (in Odoo, for instance, on top of 'sale.order' records). When instantiated on the analyzed dataset, this pattern produces the dataset-specific ranking reported in Table 1 and discussed above; when applied to another structured dataset, ERP deployment, or B2B segment, it is expected to yield a different factor ordering. The prescriptive content of this study is therefore the proposed method and its reproducible application procedure, rather than any universal list of B2B success factors.

Despite the validity of the obtained results, this study has certain methodological limitations. First, the analysis is based on data from a single Odoo system, which may

limit the generalizability of the conclusions to other ERP systems or industries. Second, the study focuses on quantitative characteristics of orders and communication but does not account for qualitative aspects and communication topics.

6. Conclusions

This study identified key factors influencing B2B order success in the Odoo ERP system using a multi-algorithmic ranking method that combines five feature significance scorers (AUC, MI, dCor, LogReg, DecTree) and aggregates results through a modified Borda rank-aggregation rule based on the arithmetic mean of scorer-specific ranks, thereby increasing the reliability of the conclusions. The proposed method is intended as a generalizable framework for structured transactional data, whereas the specific ordering and grouping of factors reported here reflect the empirical structure of the analyzed ERP-based dataset. The most stable and influential determinants of order success are communication intensity and customer-history characteristics, with the number of exchanged messages emerging as the strongest overall predictor and historical partner success also occupying the highest positions in the consolidated ranking. Order-flexibility factors retain secondary importance, indicating that iterative adjustment of order parameters is positively associated with successful completion. Financial indicators, particularly order amount, remain scorer-dependent, suggesting that their role is more complex and less consistent across scorers, whereas discount-related and fine-grained temporal variables contribute comparatively little to the consolidated ranking. Beyond these factor-level findings, the principal scientific contributions of the work are threefold: (i) the selection and methodological justification of five mathematically heterogeneous feature significance scorers that capture complementary forms of dependence between predictors and a binary outcome; (ii) a scorer-neutral aggregation method – the modified Borda mean rank – that transforms heterogeneous scorer-specific outputs into a consolidated, scale-independent ranking whose methodological neutrality, scale invariance, robustness to outliers, and interpretational clarity are explicitly argued for; and (iii) a reproducible end-to-end pipeline that operationalizes this method on structured transactional data and can be applied to other datasets and decision-support contexts, including but not limited to ERP systems. The proposed method can be implemented for automated order success forecasting and tested in other structured transactional domains. Future research should focus on analyzing factor interactions, testing the method on additional datasets, and developing a predictive system for order success based on the key factors identified in this empirical application.

List of abbreviations

ML	Machine Learning
AUC	Area Under the Curve
B2B	Business-to-Business
dCor	Distance Correlation
DecTree	Decision Trees
ERP	Enterprise Resource Planning
LogReg	Logistic Regression
MI	Mutual Information
OHE	One-Hot Encoding
ROC	Receiver Operating Characteristic

References

Akintelu, S. O., Oyebola, A. I., Tiamiyu, S., Olateju, O. (2023). The impact of project communication management on successful project delivery in the construction industry: A case study, *Int. J. Dev. Sustain*., 12, 376–386.

Bauskar, S. (2024). Predictive Analytics For Sales Forecasting In Enterprise Resource Planning (Erp) Systems Using Machine Learning Technique, *SSRN Electron. J.* DOI: 10.2139/ssrn.4988774.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (2017). *Classification and Regression Trees*, Routledge: New York, NY, USA. DOI: 10.1201/9781315139470.

Calders, T., Jaroszewicz, S. (2007). Efficient AUC Optimization for Classification. In: Knowledge Discovery in Databases: PKDD 2007. *Lecture Notes in Computer Science*; Kok, J. N., Ed.; Springer: Berlin/Heidelberg, Germany, 4702, 42–53. DOI: 10.1007/978-3-540-74976-9_8.

Chakraborty, D., Elzarka, H. (2018). Advanced machine learning techniques for building performance simulation: a comparative analysis, *J. Build. Perform. Simul*., 12, 193–207. DOI: 10.1080/19401493.2018.1498538.

Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., Meester, L. E. (2005). *A Modern Introduction to Probability and Statistics*, Springer: London, UK. DOI: 10.1007/1-84628-168-7.

Dwork, C., Kumar, R., Naor, M., Sivakumar, D. (2001). Rank aggregation methods for the Web. In: *Proceedings of the 10th International Conference;* ACM Press: New York, NY, USA. DOI: 10.1145/371920.372165.

Eid, R., Trueman, M., Ahmed, A. M. (2002). A cross-industry review of B2B critical success factors, *Internet Res*., 12, 110–123. DOI: 10.1108/10662240210422495.

Evanschitzky, H., Ramaseshan, B., Woisetschläger, D. M., Richelsen, V., Blut, M., Backhaus, C. (2011). Consequences of customer loyalty to the loyalty program and to the company, *J. Acad. Mark. Sci*., 40, 625–638. DOI: 10.1007/s11747-011-0272-3.

Fawcett, T. (2006). An introduction to ROC analysis, *Pattern Recognit. Lett*., 27, 861–874. DOI: 10.1016/j.patrec.2005.10.010.

Ferro-Soto, C., Padin, C., Lubbe, I., Svensson, G., Høgevold, N. (2025). Determinants and outcomes of satisfaction in B2B sales relationships, *J. Bus. Ind. Mark*., 40, 61–76. DOI: 10.1108/jbim-10-2022-0470.

Fraccastoro, S., Gabrielsson, M., Pullins, E. B. (2020). The integrated use of social media, digital, and traditional communication tools in the B2B sales process of international SMEs, *Int. Bus. Rev*., 29, 101776. DOI: 10.1016/j.ibusrev.2020.101776.

González-Flores, L., Rubiano-Moreno, J., Sosa-Gómez, G. (2025). The relevance of lead prioritization: a B2B lead scoring model based on machine learning, *Front. Artif. Intell*., 8, 1554325. DOI: 10.3389/frai.2025.1554325.

Guesalaga, R., Ruiz-Alba, J. L., López-Tenorio, P. J. (2023). Drivers of business-to-business sales success and the role of digitalization after COVID-19 disruptions, *J. Bus. Ind. Mark.*, 39, 708–720. DOI: 10.1108/jbim-12-2022-0576.

Günther, M. (2021). Performance in B2B Sales: An Explanation of How Channel Management and Communication Influence a Firm's Performance, *Naše Gospod. Our Econ.*, 67, 38–48. DOI: 10.2478/ngoe-2021-0016.

Guo, L., Fang, W., Zhao, Q., Wang, X. (2021). The hybrid PROPHET-SVR approach for forecasting product time series demand with seasonality, *Comput. Ind. Eng.*, 161, 107598. DOI: 10.1016/j.cie.2021.107598.

Høgevold, N. M., Rodriguez, R., Svensson, G., Roberts-Lombard, M. (2022). Organisational and environmental indicators of B2B sales performance, *Mark. Intell. Plan*., 40, 33–56. DOI: 10.1108/MIP-03-2021-0100.

Hrischev, R., Shakev, N. (2023). Artificial Intelligence in Enterprise Resource Planning Systems, *Eng. Sci.*, LX, 1–13. DOI: 10.7546/engsci.lx.23.01.01.

Jee, T. W. (2021). The perception of discount sales promotions – A utilitarian and hedonic perspective, *J. Retail. Consum. Serv*., 63, 102745. DOI: 10.1016/j.jretconser.2021.102745.

Kleinbaum, D. G., Klein, M. (2010). *Logistic Regression*, Springer: New York, NY, USA. DOI: 10.1007/978-1-4419-1742-3.

Kraskov, A., Stögbauer, H., Grassberger, P. (2004). Estimating mutual information, *Phys. Rev. E*, 69, 066138. DOI: 10.1103/physreve.69.066138.

Lee, F. (2025). What Is Logistic Regression? IBM – United States. Available online: https://www.ibm.com/think/topics/logistic-regression.

Lin, S.-W., Fu, H.-P., Lin, A. J. (2023). Critical success factors and implementation strategies for B2B electronic procurement systems in the travel industry, *J. Hosp. Tour. Technol.*, 14, 505–522. DOI: 10.1108/JHTT-08-2021-0230.

Marín Díaz, G., Galán, J. J., Carrasco, R. A. (2022). XAI for Churn Prediction in B2B Models: A Use Case in an Enterprise Software Company, *Mathematics*, 10, 3896. DOI: 10.3390/math10203896.

Miroshnychenko, S. O. (2024). Improvement of the existing functionality for forecasting finished goods inventory at the enterprise warehouses based on the stock forecasted report ERP ODOO. In: *Miningmetalteh 2024 – The Mining and Metals Sector: Integration of Business, Technology and Education*; Baltija Publishing: Riga, Latvia, 2, 48–51. DOI: 10.30525/978-9934-26-506-8-133.

Miroshnychenko, S. O., Koyfman, O. O., Miroshnychenko, V. I. (2024). The relevance of implementing an intelligent decision support system for enterprise resource optimization. In: *Miningmetalteh 2024 – The Mining and Metals Sector: Integration of Business, Technology and Education*; Baltija Publishing: Riga, Latvia, 2, 45–47. DOI: 10.30525/978-9934-26-506-8-132.

Miroshnychenko, S. O., Miroshnychenko, V. I., Koyfman, O. O., Vovna, O. V. (2025). Automated feature selection system for computer-integrated forecasting of B2B order success using XGBOOST, *Sci. J. Metinvest Polytech*. Ser. Tech. Sci., 5, 58–75. DOI: 10.32782/3041-2080/2025-5-7.

Miroshnychenko, S. O., Vovna, O. V. (2025). Multifactor statistical analysis of order success as a tool for optimizing automated resource management systems, *Bull. Priazovskyi State Tech. Univ*. Ser. Tech. Sci., 50, 182–190. DOI: 10.31498/2225-6733.50.2025.336372.

Pereira, M. d. S., Schmitt de Castro, B., Alves Cordeiro, B., Mendonça Peixoto, M. G., Cornils Monteiro da Silva, E., Carneiro Gonçalves, M. (2025). Factors of Customer Loyalty and Retention in the Digital Environment, *J. Theor. Appl. Electron. Commer. Res*., 20, 71. DOI: 10.3390/jtaer20020071.

Popescu, D., Dinu, A., State, C., Picu, C. (2017). Empirical Study Regarding the Influence of Communication Upon the Purchasing Decision, *Int. Conf. Knowl. Based Organ*., 23, 430–437. DOI: 10.1515/kbo-2017-0071.

Rezazadeh, A. (2020). A Generalized Flow for B2B Sales Predictive Modeling: An Azure Machine-Learning Approach, *Forecasting*, 2, 267–283. DOI: 10.3390/forecast2030015.

Rohaan, D., Topan, E., Groothuis-Oudshoorn, C. G. M. (2022). Using supervised machine learning for B2B sales forecasting: A case study of spare parts sales forecasting at an after-sales service provider, *Expert Syst. Appl*., 188, 115925. DOI: 10.1016/j.eswa.2021.115925.

Rodriguez, R., Roberts-Lombard, M., Høgevold, N. M., Svensson, G. (2022). Organisational and environmental indicators of B2B sellers' sales performance in services firms, *Eur. Business Rev*., 34, 578–602. DOI: 10.1108/EBR-05-2021-0123.

Ross, B. C. (2014). Mutual Information between Discrete and Continuous Data Sets, *PLoS ONE*, 9, e87357. DOI: 10.1371/journal.pone.0087357.

Sadaram, G., Jha, K. M., Katnapally, N. (2023). Optimising Sales Forecasts in ERP Systems Using Machine Learning and Predictive Analytics, *SSRN Electron. J.* DOI: 10.2139/ssrn.5114001.

Sarkar, C., Cooley, S., Srivastava, J. (2014). Robust Feature Selection Technique Using Rank Aggregation, *Appl. Artif. Intell*., 28, 243–257. DOI: 10.1080/08839514.2014.883903.

Sashi, C. M. (2021). Digital communication, value co-creation and customer engagement in business networks: a conceptual matrix and propositions, *Eur. J. Mark.*, 55, 1643–1663. DOI: 10.1108/ejm-01-2020-0023.

Smith, J. D. (2025). The Influence of Weather on Closing B2B Sales: Is it More Likely to Close a Sale on Sunny Days Versus Cloudy or Rainy Days? *Int. J. Res. Publ. Rev*., 6, 971–974. DOI: 10.55248/gengpi.6.0325.1127.

Székely, G. J., Rizzo, M. L., Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances, *Ann. Stat*., 35, 2769–2794. DOI: 10.1214/009053607000000505.

Ul Haq, I., Gondal, I., Vamplew, P., Brown, S. (2019). Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment. In: *Data Mining. AusDM 2018. Communications in Computer and Information Science*; Islam, R., Ed.; Springer: Singapore, 996, 53–65. DOI: 10.1007/978-981-13-6661-1_6.

Vambol, V., Kowalczyk-Juśko, A., Vambol, S., Khan, N., Mazur, A., Goroneskul, M., Kruzhilko, O. (2023). Multi criteria analysis of municipal solid waste management and resource recovery in Poland compared to other EU countries, *Sci. Rep*., 13, 21233. DOI: 10.1038/s41598-023-48026-3.

Wald, R., Khoshgoftaar, T. M., Dittman, D., Awada, W., Napolitano, A. (2012). An extensive comparison of feature ranking aggregation techniques in bioinformatics. In: *Proceedings of the 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, Las Vegas, NV, USA, 8–10 August 2012; 377–384. DOI: 10.1109/IRI.2012.6303034.

Wilson, R. D., Stephens, A. M. (2023). The challenges of B2B innovation: using marketing analytics to plan and implement a successful digital catalog adoption, *J. Bus. Ind. Mark*., 38, 290–302. DOI: 10.1108/JBIM-12-2021-0598.

Zoltners, A. A., Sinha, P., Sahay, D., Shastri, A., Lorimer, S. E. (2021). Practical insights for sales force digitalization success, *J. Pers. Sell. Sales Manag*., 41, 87–102. DOI: 10.1080/08853134.2021.1908144.

Appendix A
Feature Name Mapping

Table 2. Feature Name Mapping

Technical Feature Name	Descriptive Name	Category	Data Type	Unit / Scale
order_messages	Number of messages exchanged per order	Communication	Integer	count
order_amount	Total monetary value of the order	Financial	Float	USD
partner_success_rate	Historical success rate of the partner	Partner History	Float	%
order_changes	Number of modifications made to the order	Order Flexibility	Integer	count
partner_success_avg_messages	Average number of messages in partner's successful orders	Communication / History	Float	count
partner_total_orders	Total number of historical orders by the partner	Partner History	Integer	count
partner_total_messages	Total messages across all partner orders	Communication / History	Integer	count
partner_fail_avg_changes	Average number of changes in partner's failed orders	Partner History	Float	count
create_date_months	Months elapsed since the first recorded order	Temporal	Integer	months since initial recorded order
partner_success_avg_amount	Average amount of partner's successful orders	Financial / History	Float	USD
partner_avg_changes	Average number of changes across all partner orders	Partner History	Float	count
partner_success_avg_changes	Average number of changes in partner's successful orders	Partner History	Float	count

Technical Feature Name	Descriptive Name	Category	Data Type	Unit / Scale
order_lines_count	Number of product lines in the order	Order Complexity	Integer	count
partner_fail_avg_messages	Average number of messages in partner's failed orders	Communication / History	Float	count
partner_order_age_days	Days since partner's first order in the system	Partner History	Integer	days
partner_avg_amount	Average amount across all partner orders	Financial / History	Float	USD
partner_fail_avg_amount	Average amount of partner's failed orders	Financial / History	Float	USD
source	Lead/order acquisition channel	Categorical	Object	category label
salesperson	Assigned sales representative identifier	Categorical	Object	category label
hour_of_day	Hour when the order was placed	Temporal	Integer	0–23
month	Month when the order was placed	Temporal	Object	month category
discount_total	Total discount applied to the order	Financial	Float	USD
day_of_week	Day of the week when the order was placed	Temporal	Object	weekday category
quarter	Quarter when the order was placed	Temporal	Integer	1–4

This appendix table reports the original substantive analytical features retained for interpretation prior to type-specific preprocessing. Their final machine-readable representation is generated subsequently through the transformations described in the Methods section, including cyclical sin/cos encoding for selected temporal variables, OHE Extended Compact encoding for categorical variables, and robust scaling for numerical variables.

Appendix B
Python Code for the Modified Borda Rank Aggregation

```
import pandas as pd
def modified_borda_mean_rank(method_scores: dict[str,
pd.Series])
-> pd.DataFrame:
    df = pd.DataFrame(method_scores)
    ranks = df.rank(ascending=False, method="max")
    ranks["mean_rank"] = ranks.mean(axis=1)
    ranks.insert(0, "feature", ranks.index)
    return
ranks.sort_values("mean_rank").reset_index(drop=True)
```

Received January 27, 2026, revised June 26, 2026, accepted June 26, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, pp. 445–461
https://doi.org/10.22364/bjmc.2026.14.2.09

Data-Driven Methods for Analyzing Non-Gaussian Variability in Human Postural Sway

Gennady CHUIKO, Yevhen DARNAPUK, Olga YAREMCHUK

Petro Mohyla Black Sea National University, Mykolaiv, Ukraine

henadiy.chuyko@chmnu.edu.ua, yevhen.darnapuk@chmnu.edu.ua,
olga.yaremchuk.77@ukr.net

ORCID 0000-0001-5590-9404, ORCID 0000-0002-7099-5344, ORCID 0000-0002-0891-4216

Abstract. Postural sway during quiet standing is often analyzed using diffusion- or Brownian-type models that implicitly assume near-Gaussian statistics. In practice, center-of-pressure (COP) trajectories can exhibit heavy tails, asymmetry, and intermittent excursions that may bias variance-only variability descriptors. This study presents a reproducible Python-based analysis pipeline for assessing non-Gaussian structure in COP increment series by rotating increment pairs into a PCA-decorrelated coordinate system. Using the Human Balance Evaluation Database (HBEDB) (1930 trials from 163 subjects; 60 s at 100 Hz under four standing conditions: eyes open/closed and rigid/unstable surface; three repetitions each), we compare empirical increment distributions with mean- and covariance-matched Gaussian surrogates and quantify departures using complementary diagnostics, including probability–probability comparisons, robust spread descriptors (IQR and range), and uni-/bivariate kernel density estimation. We further introduce a compact central ellipse fraction (CEF) descriptor that measures central concentration in the PCA plane and relates distributional findings to linear and nonlinear variability measures, including Poincaré-type descriptors and recurrence indices. Across the dataset, empirical increments deviate systematically from Gaussian surrogates: P–P plots show tail departures, and the 2-D density in the PCA plane is more centrally concentrated than the matched Gaussian model (CEF higher by approximately 0.08–0.10), consistent with leptokurtic (peaked and heavy-tailed) behavior. These results motivate distribution-aware preprocessing and descriptor selection for computer-based posturography and fall-risk assessment algorithms beyond variance-only summaries.

Keywords: Biomedical signal processing, Postural sway analysis, Digital signal processing, Non-Gaussian variability, Recurrence plot analysis, Computer-based balance monitoring

1 Introduction

Postural balance requires the coordinated efforts of the human body's visual, vestibular, and proprioceptive systems (Prieto et al., 1993). The proprioceptive system provides rapid muscular and joint responses to external forces. Such coherent action is proper for any posture, e.g., in the standing position. The center of gravity of a human body and the center of pressure (COP) under the feet move about a laboratory coordinate system (Collins and De Luca, 1993). This motion is commonly referred to as postural sway (Prieto et al., 1993; Collins and De Luca, 1993). As a rule, these sways are depicted via the COP trajectory on a support plane.

Meanwhile, fall-related mortality shows a pronounced peak among older adults (defined here as people aged 65 years and older; see *National Safety Council, Home and Community Overview, Injury Facts* (2023) for the latest available data from the USA). On a global scale, unintentional falls are responsible for more than 660,000 deaths each year (*World Health Organization. Falls*, 2021). As the populations of many developed countries continue to age, the burden of fatal falls is expected to grow. This raises an important question: can this trend be slowed or even reversed?

Appropriate training for older adults might be an effective prevention tool. Authors Rogers et al. (2001) reported decreases in mediolateral sway amplitude (about 9%) and mean instantaneous speed (about 13%) following a training course for older adults. Postural sways have also been used to investigate age-related changes and neurologic diseases (Prieto et al., 1993), the impact of a blood stroke (Treger et al., 2020), joint laxity (Aydın, 2017), active and passive changes in posture (Cit, 2003), during burst behavior (Picoli et al., 2019), and so forth.

The random-walk approach, also known as the Brownian motion model, is a common starting point for analyzing COP trajectory records (also called stabilograms) (Treger et al., 2020). Within this framework, stabilogram diffusion analysis (SDA) is a natural and widely used method for describing postural sway (Prieto et al., 1993; Collins and De Luca, 1993; Treger et al., 2020; Delignlères et al., 2003). Despite the criticisms discussed in Delignlères et al. (2003), SDA remains a valid and proper technique (Treger et al., 2020; Bao et al., 2023).

Fractional Brownian motion is a generalization of the SDA model in which the mean-square displacement grows as a power law in time. In this case, the variance of displacement depends not only on the elapsed time but also on its exponent: it increases superlinearly for persistent series and sublinearly for anti-persistent ones Delignlères et al. (2003); Mandelbrot and Van Ness (1968). More generally, fractional Brownian motion treats postural sways as a kind of "i/f-type noise", intermediate between white noise and classical Brownian motion.

However, within both models, the analysis typically emphasizes second-order (variance-based) structure, so the extent to which postural sway exhibits non-Gaussian displacement (increment) statistics is not always assessed explicitly. Meanwhile, the non-Gaussian nature of the probability distributions of many medical signals is well known. This remark also holds for other statistical methods, such as Rescaled Range Analysis (R/S) and Detrended Fluctuation Analysis (DFA), which were analyzed in (Delignlères et al., 2003). One must at least know how robust the statistical parameters are.

The second point we draw attention to is the application of machine-learning and principal-component-based methods to postural sway Andò et al. (2025); Gorban and Zinovyev (2010). Rather than assuming a fixed laboratory AP–ML orientation, PCA provides a data-driven rotation of the COP increment plane into orthogonal principal axes, yielding decorrelated coordinates. This step reduces the influence of subjective and unavoidable inaccuracies in the orientation of the laboratory coordinate system. Selecting decorrelated (approximately independent) coordinates enables us to estimate two-dimensional probability density distributions of COP increments consistently across subjects and conditions. Variability is a fundamental feature of all medical signals and postural sways. To quantify this property is one of our objectives in this report. This point turned out to be close enough to the above ones.

In modern balance-assessment systems, force-platform measurements are acquired and processed by dedicated computer hardware and software. The reliability of such systems depends critically on the robustness of their data-processing algorithms. In this work, we focus on this algorithmic layer, proposing and evaluating numerical descriptors and processing steps that can be integrated into computer-based posturography and fall-risk assessment tools.

We propose a distribution-aware posturography workflow that rotates center-of-pressure (COP) increment pairs into a principal component analysis (PCA)-decorrelated coordinate system. This approach minimizes axis-orientation bias and facilitates consistent two-dimensional distribution analysis across various subjects and conditions. To achieve this, we compare the empirical statistics of COP increments with Gaussian surrogates matched for mean and covariance. This analysis incorporates complementary diagnostics, such as P–P comparisons, robust spread descriptors like the interquartile range (IQR) and range, and univariate and bivariate kernel density estimates. Furthermore, we connect the observed distribution shapes to both linear (dispersion) and nonlinear (Poincaré/recurrence) variability measures. Given that the HBEDB dataset includes repeated trials per subject across different standing conditions, we aggregate data at the subject level to ensure that individual trials are not erroneously treated as statistically independent.

This study makes three contributions to quantitative posturography. First, we provide a reproducible pipeline to test whether Gaussian assumptions are adequate for COP increment statistics by benchmarking empirical data against matched Gaussian surrogates and quantifying deviations using distributional diagnostics. Second, we introduce and operationalize bivariate distribution-shape descriptors in the PCA plane – most notably a central concentration metric based on the fraction of samples within a reference ellipse (central ellipse fraction, CEF) – to capture differences in density geometry that are not explained by variance alone. Third, using a public balance dataset spanning younger and older adults, we assess how classical dispersion descriptors, Poincaré-type measures, and recurrence-based indices relate to distribution-shape measures, highlighting which relationships remain stable and which exhibit age-linked coupling patterns.

2 Materials and Methodology

2.1 Data source and age-stratified subgroups

We used the Human Balance Evaluation Database (HBEDB) available on PhysioNet (Santos and Duarte, 2016b,a; Goldberger et al., 2000). HBEDB contains force-platform recordings of quiet standing acquired under four experimental conditions defined by vision and support surface (eyes open/closed and rigid/unstable), with three trials per condition presented in randomized order. Overall, the database provides 1930 trials from 163 participants. Each trial is 60 s long, sampled at 100 Hz, and low-pass filtered at 10 Hz. In this study, we analyzed the full dataset (all available trials).

For age-stratified analysis, participants were divided into a younger subgroup (< 60 years) and an older subgroup (≥ 60 years).

The nominal complete design would contain $163 \times 4 \times 3 = 1956$ recordings. In the public HBEDB release used in this study, 1930 trial records were available for analysis. Thus, the difference of 26 records reflects unavailable recordings in the public dataset rather than exclusions introduced by our preprocessing pipeline. We therefore analyzed all available trials.

The threshold of 60 years was used as an exploratory age-stratification choice to obtain two interpretable subgroups with sufficient sample sizes. We did not optimize this threshold or scan multiple cut-points, because such a procedure would introduce post-hoc threshold selection and increase the risk of overinterpretation. A more detailed treatment of age as a continuous variable, or the use of several age bands, should be combined with subject-level aggregation or mixed-effects modelling in future work.

In the original dataset (Santos and Duarte, 2016b), COP coordinates are provided as AP(X) and ML(Y) positions in the laboratory coordinate system. Because COP trajectories are temporally autocorrelated, the nominal sample size (6000 points) substantially overstates the number of independent observations; accordingly, statistical significance thresholds for small correlation coefficients should be interpreted with caution. Nevertheless, the observed $|corr(X, Y)|$ values were sufficiently large to indicate meaningful cross-axis coupling in the laboratory frame. A plausible reason is a slight rotation between the laboratory axes and the principal directions of sway. Therefore, following Gorban and Zinovyev (2010), we refined the data using principal component analysis (PCA) to obtain a rotated coordinate system in which the two sway components are decorrelated.

2.2 Methods

We applied principal component analysis (PCA) to orthogonalize the original COP coordinate plane, yielding decorrelated components and reducing the dependence between the anterior–posterior (AP) and mediolateral (ML) laboratory axes. In this two-dimensional setting, PCA:

- yields orthogonal principal axes,
- orders the axes by explained variance (PC1 captures maximal variance),
- diagonalizes the covariance structure, providing decorrelated coordinates that reduce sensitivity to small axis-orientation errors (Gorban and Zinovyev, 2010).

This improves the interpretability and consistency of subsequent two-dimensional distribution analysis across subjects and conditions. In the figures, the two PCA-rotated components are sometimes labelled AP and ML for visual continuity; however, they should be interpreted as rotated principal components rather than as the original anatomical/laboratory AP and ML axes.

After PCA rotation, we analyzed displacements between successive positions (increment series), which are commonly used in stabilogram diffusion analysis (SDA) and related approaches. For each trial and component, we examined increment distributions using the Shapiro–Wilk normality test (Shapiro and Wilk, 1965) and graphical diagnostics (box plots and probability–probability (P–P) plots) (Field, 2024). Because the increment sequences are autocorrelated and contain $n = 5999$ successive increments per trial, Shapiro–Wilk p-values were interpreted as sensitivity indicators rather than as i.i.d. assumption checks. Nonparametric probability density functions were estimated using kernel density estimation (KDE). In the reported implementation, an Epanechnikov kernel was used for the KDE-based visualizations, with bandwidths selected by a normal-reference rule-of-thumb (Silverman-type rule, coefficient $k = 1.06$) (Zielinski et al., 2018; Silverman, 2018; Moraes et al., 2021). The term "normal-reference" refers to the bandwidth-selection heuristic only and does not imply that the empirical COP increment distributions are assumed to be Gaussian. In addition to univariate densities, we estimated two-dimensional probability density functions (2D-PDFs) for the joint increment distribution in the PCA plane.

To quantify departures from a Gaussian reference model in the 2D case, we introduced a compact distribution-shape descriptor based on the one-sigma region: the central ellipse fraction (CEF), defined as the fraction of samples whose Mahalanobis distance in the PCA plane satisfies $d_M \leq 1$, where the covariance structure of the corresponding Gaussian model determines the reference ellipse. This measure captures differences in central concentration between empirical 2D distributions and their mean- and covariance-matched Gaussian surrogates.

A complementary analysis focused on variability descriptors. We computed three dispersion descriptors (standard deviation, interquartile range, and range) (Field, 2024) and three additional descriptors: recurrence rate (RR) from recurrence analysis (Marwan et al., 2018) and Poincaré-plot measures SD1 and SD2 (Golińska, 2013). We assessed cross-correlations among the six descriptors (including correlation magnitudes) and compared descriptor distributions and density estimates between components (AP vs ML) and between younger and older cohorts. All analyses and figure generation were implemented in Python using standard scientific computing libraries (Hudson et al., 2022).

3 Results

3.1 Deviations of the factual distributions from the standard Gaussian reference

To begin, we transformed the COP position series (X, Y)(Santos and Duarte, 2016c) into an increment (displacement) series between successive samples, $(\delta x, \delta y)$. In the laboratory coordinate system, the increment components exhibited cross-axis coupling

(a nonzero Pearson correlation), consistent with a slight misalignment between the laboratory axes and the dominant sway directions. Therefore, we applied principal component analysis (PCA) to the increment pairs and rotated the laboratory frame into a PCA frame in which the two components are linearly decorrelated.

In the PCA frame, the components have near-zero linear correlation. However, decorrelation does not generally imply statistical independence, particularly when the variability is non-Gaussian. Accordingly, we treat the PCA components as linearly decorrelated variables while allowing higher-order dependencies, and we estimate joint distributions directly rather than factorizing them into products of marginals. Because PCA orders axes by explained variance, this rotation also yields a variance-maximizing coordinate system that improves the interpretability of subsequent two-dimensional distribution analysis.

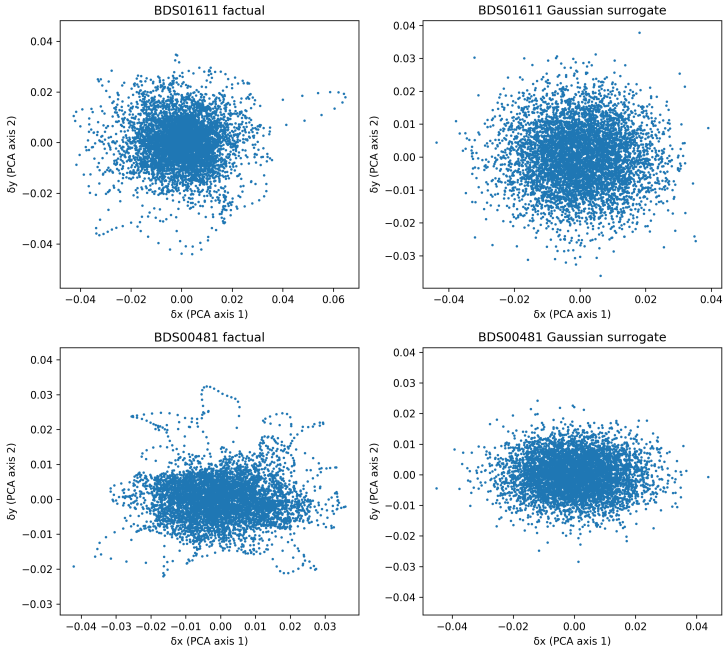


Fig. 1. Scatter plots of COP increment pairs $(\delta x, \delta y)$ in the PCA-rotated coordinate system (axes decorrelated). Two example trials are shown: BDS01611 (27 years; upper row) and BDS00481 (69 years; lower row). Empirical series are shown on the left; the right panels show Gaussian surrogate series generated to match the empirical mean vector and covariance matrix while preserving trial length, enabling visual comparison of tail behavior and anisotropy. Axis limits are scaled separately across panels to preserve local point-cloud visibility; therefore, the figure should be interpreted qualitatively in terms of distributional shape, tail organization, and empirical-versus-surrogate differences rather than absolute plotted area.

Figure 1 shows scatter plots of successive COP increments in the PCA-rotated plane (PC1–PC2). For comparison, we generated Gaussian surrogate samples from a bivariate normal distribution with the same empirical mean vector and covariance matrix and the same number of increments. The empirical and surrogate point clouds are broadly similar in overall scale and anisotropy; however, the empirical extremes often form structured arcs/clusters consistent with temporal dependence, whereas surrogate extremes are more spatially diffuse.

To quantify differences in central concentration, we define the central ellipse fraction (CEF) as the proportion of samples lying inside the $1-\sigma$ ellipse in the PCA plane, equivalently, those with Mahalanobis distance $d_M \leq 1$ under the matched Gaussian reference. Shapiro–Wilk tests rejected normality for all AP trials (1930/1930) and most ML trials (1926/1930) at $\alpha = 0.05$. Because increments inherit autocorrelation from COP trajectories and $n \approx 6000$ per trial yields very high test power, we treat Shapiro–Wilk as a supporting diagnostic alongside P–P plots and KDE-based density estimates rather than as a standalone proof of non-Gaussianity.

As illustrated in Figure 2, probability–probability (P–P) plots compare the empirical cumulative distribution with that of a reference Gaussian distribution (Field, 2024). In our data, the curves show systematic departures from the diagonal (the Gaussian reference), most visibly toward the tails of the distribution, consistent with deviations in kurtosis and, in some trials, mild asymmetry. Together with the Shapiro–Wilk test results, these plots support the conclusion that the increment/displacement distributions are not exactly normal.

The two examples in Figure 2 also show that the degree of deviation is not identical across trials or components. In trial BDS01684, both components show visible systematic departures from the diagonal, especially away from the center of the distribution, whereas in trial BDS00995 the ML component remains much closer to the Gaussian reference over most of the probability range. This trial-to-trial heterogeneity is one reason why we rely on several complementary diagnostics rather than on a single normality test or a single visual summary.

A second useful view of the same phenomenon is provided by the box-and-whisker representation in Figure 3. Whereas the P–P plots emphasize distributional agreement with a Gaussian reference over the full probability range, the box plots separate the central spread from the most extreme values. In both illustrated trials, the AP component occupies a wider range than the ML component, and the most pronounced differences are concentrated in the tails rather than in the median location.

Note that the empirical distributions can look close to a Gaussian model for small displacements near the center; however, differences become more apparent for larger displacements (extremes). These extremes are visible in the left-hand scatter plots of Figure 1. They are reflected in the box-and-whisker plots in Figure 3. Also note that the most extreme values are often not perfectly balanced around the median (and thus around zero), indicating that tail behavior can be asymmetric even when the central bulk is close to symmetric. The ranges of AP sways exceed those of ML sways in the shown examples (Figure 3). The presence of extreme values can strongly inflate the range and standard deviation and may also broaden the IQR (Field, 2024). We will return to this issue when we consider the variability descriptors of postural sway.

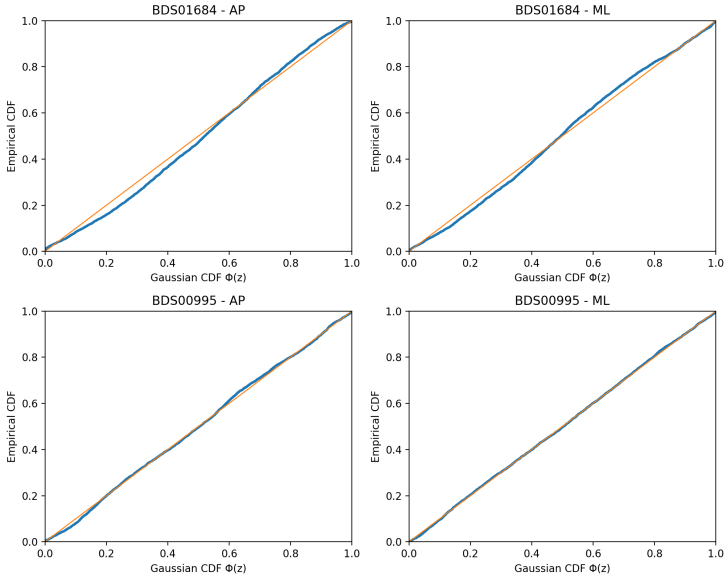


Fig. 2. Probability–probability (P–P) plots comparing the empirical CDF of COP increment samples with the CDF of a Gaussian reference. AP is shown on the left and ML on the right. The upper row shows trial BDS01684 (younger participant), and the lower row shows trial BDS00995 (older participant). The thin diagonal line is the identity line $y = x$, indicating perfect agreement with the Gaussian reference; systematic departures from this line indicate non-Gaussianity (especially in the tails).

The two examples also illustrate that extreme values need not occur symmetrically in the positive and negative directions. In BDS01684, the AP component shows a pronounced negative tail together with several large positive excursions, while the corresponding ML component remains more compact. In BDS00995, the AP component again exhibits a broader spread and more numerous outliers than the ML component. This pattern is consistent with the impression from Figure 1 that the empirical increment clouds differ from Gaussian surrogates primarily through their tail structure and central concentration rather than through large shifts of the mean.

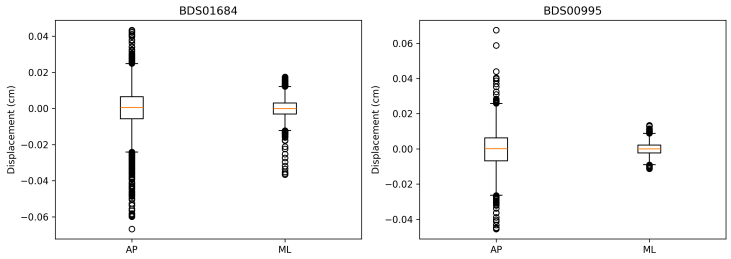


Fig. 3. Box-and-whisker plots of the two PCA-rotated COP increment components (PC1 and PC2; labeled AP and ML in the plot) for two example trials (BDS01684 and BDS00995). Boxes indicate the interquartile range (IQR) with the median shown as a horizontal line; whiskers extend to the most extreme values within $1.5 \times IQR$ (Tukey convention); points beyond the whiskers are plotted as outliers.

3.2 Kernel density estimations

Kernel density estimation (KDE) is a method for estimating the probability density function (PDF) of a distribution (Zielinski et al., 2018; Silverman, 2018). The following two choices are most important within KDE:

- the choice of appropriate kernel functions,
- the right choice of the bandwidth (also called the smoothing parameter (Zielinski et al., 2018)).

Kernel density estimation (KDE) was used as a descriptive visualization tool for empirical and surrogate distributions. The KDE itself is nonparametric; therefore, it does not require the underlying data to be normally distributed. In the reported figures, KDEs were computed with an Epanechnikov kernel and a normal-reference rule-of-thumb bandwidth. The term "normal-reference" refers only to the bandwidth-selection heuristic and does not imply that the empirical COP increment distributions are assumed to be Gaussian. The Gaussian assumption is used only for constructing the mean- and covariance-matched surrogate samples, not for estimating the empirical density.

The bandwidth was computed as

$$h = k \cdot \min(\mathrm{StD}, 0.75 \times \mathrm{IQR}) \cdot n^{-1/5},$$

where h denotes the bandwidth, StD is the sample standard deviation, IQR is the interquartile range, and n is the sample size. We used $k = 1.06$ to keep the smoothing rule stable and comparable across empirical and surrogate distributions. Kernel choice mainly affects visual smoothness, whereas bandwidth selection controls the main bias–variance trade-off in the estimated density.

To visualize and compare the distributions of the central-ellipse descriptor for factual and Gaussian surrogate data, we used violin plots, which display both the empirical density and basic summary statistics.

Figure 4 shows that the point clouds in Figure 1 differ not only in the periphery but also in the central region. Specifically, the central ellipse fraction (CEF)—the proportion of increment samples falling inside the $1 - \sigma$ ellipse in the PCA plane—is

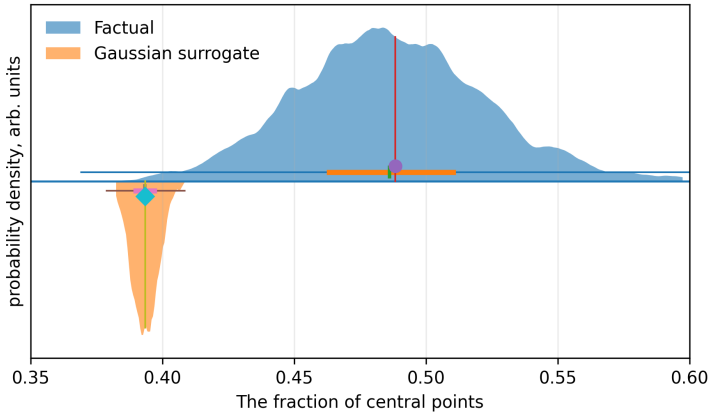


Fig. 4. Split-violin distributions of the central ellipse fraction (CEF) across trials, comparing empirical COP increment clouds (upper half, blue) with mean- and covariance-matched Gaussian surrogates (lower half, orange). The violin shapes were estimated using Epanechnikov KDE with a rule-of-thumb bandwidth. CEF is defined as the fraction of increment pairs that fall inside the $1-\sigma$ reference ellipse in the PCA plane (equivalently, Mahalanobis distance $d_M \leq 1$ under the matched Gaussian model). Points mark the mean CEF, and vertical lines mark the median CEF for each distribution.

higher for the empirical data than for the mean- and covariance-matched Gaussian surrogates, indicating stronger central concentration in the real distributions. The violin shapes in Figure 4 are kernel density estimates of the CEF distributions across trials for the factual and surrogate series (Zielinski et al., 2018; Silverman, 2018; Moraes et al., 2021). Because CEF is computed from the bivariate increment vectors after rotation to the PCA frame, it reflects the joint distribution in the PCA plane; however, the KDE shown in Figure 4 is applied to the univariate descriptor values rather than to the full two-dimensional increment clouds.

As a bandwidth-sensitivity check for Figure 4, we repeated the CEF density estimation using leave-one-out likelihood cross-validated bandwidths. For the factual CEF distribution, the rule-of-thumb bandwidth was $h = 0.00822$, whereas the cross-validated bandwidth was $h = 0.02236$; for the Gaussian-surrogate CEF distribution, the corresponding values were $h = 0.00116$ and $h = 0.00207$. Thus, cross-validation selected larger bandwidths and therefore smoother density estimates in both cases. Importantly, the qualitative separation between the factual and Gaussian-surrogate CEF distributions remained unchanged: empirical CEF values stayed shifted toward higher central concentration. Therefore, the rule-of-thumb bandwidth was retained in the main figures to keep the smoothing rule stable and comparable across all trial-level distributions.

Figure 5 highlights qualitative differences between the empirical distributions and the mean- and covariance-matched Gaussian surrogate distributions in the PCA increment plane. The surrogate densities retain the expected smooth, approximately elliptical contour structure, whereas the empirical KDE contours can exhibit sharper central con-

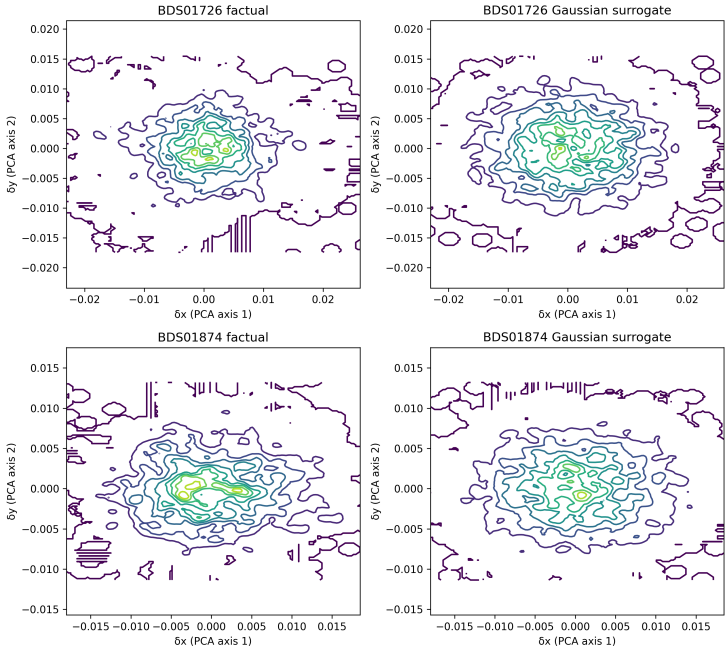


Fig. 5. Contour maps of the bivariate KDE of COP increment pairs in the PCA-rotated increment plane (decorrelated axes). Two example trials are shown: BDS01726 (31 years; upper row) and BDS01874 (62 years; lower row). Empirical densities are shown in the left panels; the right panels show KDEs of Gaussian surrogate samples generated to match the empirical mean vector and covariance matrix (same trial length), enabling visual comparison of central concentration, tail structure, and anisotropy. Axis limits are scaled separately across panels to preserve contour readability; therefore, the figure should be interpreted qualitatively in terms of density shape, central concentration, and contour regularity rather than absolute spatial extent.

centration and increasingly irregular deviations from ellipses toward the periphery. In some trials, the empirical density is close to unimodal. Still, in others, it shows noticeable asymmetry and/or secondary features (e.g., the empirical BDS01874 panel), consistent with non-Gaussian shape effects beyond second-order covariance structure. These observations are consistent with the central-ellipse-fraction results, which indicate that the empirical increments exhibit systematically stronger central concentration than the Gaussian surrogates.

3.3 Variability and its descriptors

Linear statistical descriptors of variability include the standard deviation (SD), interquartile range (IQR), and range (R) (Field, 2024). We also computed two Poincaré-plot descriptors (SD1 and SD2) (Golińska, 2013) and the recurrence rate (RR) from recurrence analysis (Marwan et al., 2018). Figure 6 illustrates these representations for a typical PCA-rotated COP increment series. The elongated Poincaré clouds indicate SD2 $\gg$

SD1, i.e., variability along the identity line (longer-term component) dominates the short-term component orthogonal to it. A small number of extreme points at the plot ends can inflate SD2 (and, to a lesser extent, SD1).

Recurrence plots were constructed as 1D amplitude recurrence matrices using the criterion $\|x_i - x_j\| \leq \epsilon$. In our implementation, ϵ was set to the normal-reference (Silverman) bandwidth (Eq. (1), $k = 1.06$) computed for each series, and RR summarizes the fraction of recurrent pairs. Because RR aggregates over all index pairs, it is typically less sensitive to a small number of outliers than range-based dispersion measures. However, extreme points can still affect RR through their distances to other samples.

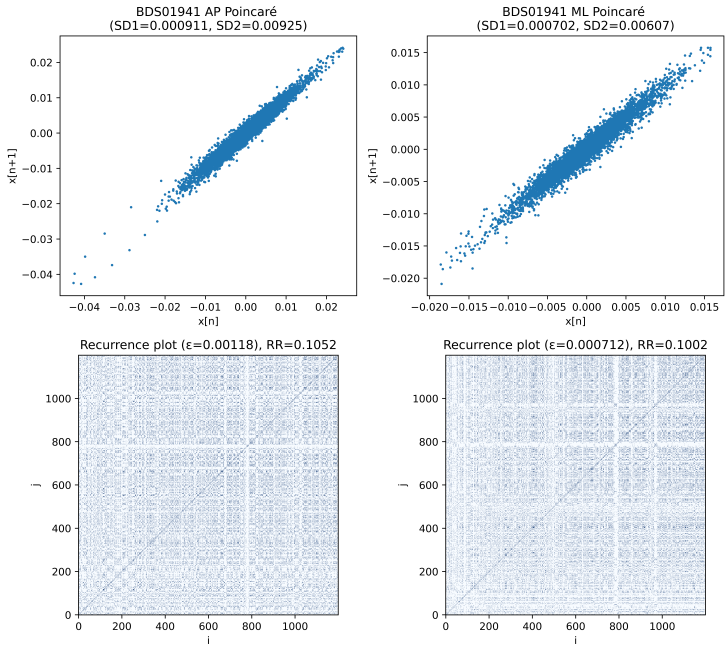


Fig. 6. Example Poincaré plots (top row) and recurrence plots (bottom row) for the PCA-rotated COP increment series from HBEDB record BDS01941 (23 years). Left panels: first component (labeled AP); right panels: second component (labeled ML). Poincaré plots show $x[n]$ versus $x[n+1]$, with SD1 and SD2 reported in the panel titles. Recurrence plots visualize the binary recurrence matrix defined by $\|x_i - x_j\| \leq \epsilon$, where ϵ is set to the normal-reference (Silverman) bandwidth ($k = 1.06$) computed for each series; the corresponding recurrence rate (RR) is shown. For visualization, recurrence matrices are downsampled to 1200 points, whereas RR is computed on the full series. For the Poincaré plots, axis limits are scaled separately for the components labeled AP and ML to preserve the visible geometry of each component. The recurrence plots use the same index scale and are compared by recurrence structure and recurrence rate.

We summarized the distributions of three variability descriptors (SD2, StD, and RR) using split violin plots (Figure 7). Across both age groups, SD2 and StD are consistently

higher in the AP direction than in the ML direction. The SD2 and StD distributions are very similar, consistent with their near-perfect positive association, whereas RR shows a weaker and predominantly inverse association with these dispersion measures. In the older subgroup, the distributions of SD2 and StD appear broader, with more pronounced tails, suggesting greater heterogeneity in variability across trials.

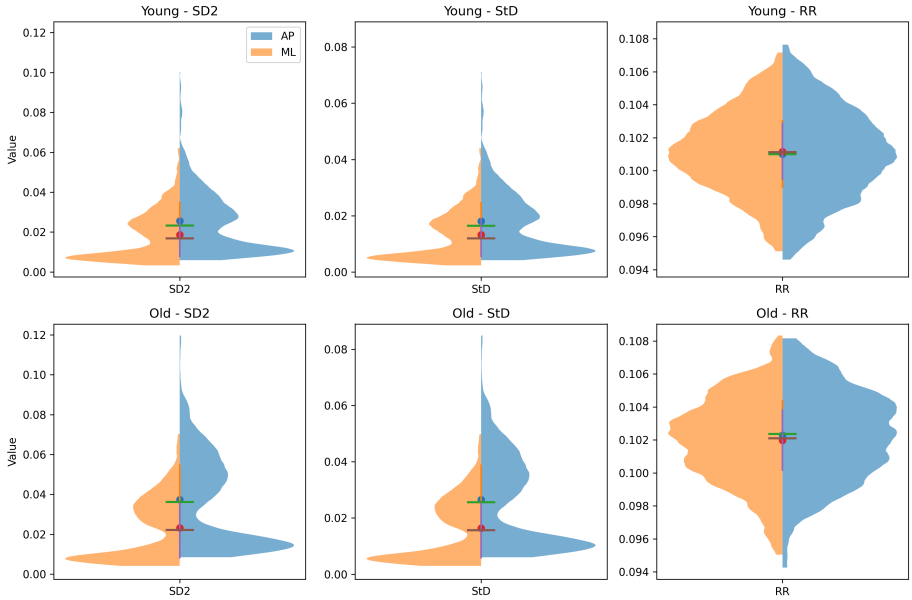


Fig. 7. Split violin plots (KDE-based) for three variability descriptors: SD2 (left column), StD (middle column), and RR (right column). The upper row shows the younger group; the lower row shows the older group. Each violin is split by direction: AP (right half) and ML (left half). Points and short line markers indicate the mean, median, and interquartile range for each half.

We examined linear associations among six variability descriptors (SD2, SD1, StD, IQR, Range, and RR) using Pearson correlation coefficients computed across trials within each age-stratified subgroup and direction. To highlight the descriptor that exhibited the clearest group contrast, we summarize in Figure 8 the correlations between the short-term variability measure SD1 and each of the remaining descriptors (SD2, StD, IQR, Range, and RR), separately for AP and ML sways in the younger and older groups. For compact visualization, Figure 8 reports absolute correlation magnitudes ($|r| \in [0, 1]$); therefore, inverse relationships (e.g., negative SD1–RR correlations) appear as large $|r|$ values, but their signs are discussed in the text rather than encoded in the bars.

Across most descriptor pairs, the observed correlations were consistently high and showed little dependence on direction. In contrast, SD1 showed weaker coupling with the other descriptors in the younger subgroup than in the older subgroup, suggesting

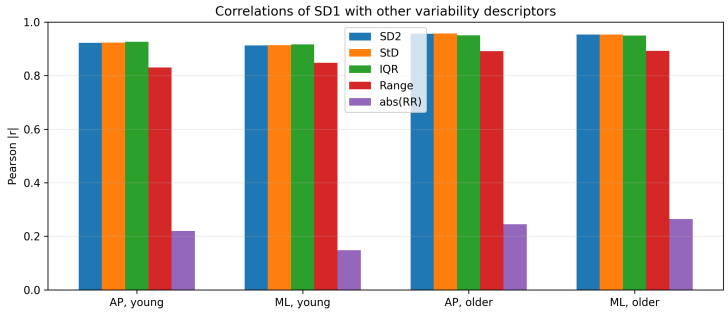


Fig. 8. Absolute Pearson correlations ($|r|$) between SD1 and SD2, StD, IQR, Range, and RR for AP/ML sway in younger and older groups. SD1 shows strong coupling with dispersion descriptors and weaker coupling with RR; age-related differences are shown descriptively.

that short-term variability may be less redundant with long-term and distributive variability measures in younger participants. Because each coefficient is estimated from a finite number of trials per subgroup, we treat these correlation patterns as effect-size summaries and interpret subgroup differences as exploratory unless confirmed by confidence intervals and an explicitly multiple-comparison-controlled inference procedure.

4 Discussion and limitations

The observed deviations from Gaussian displacement statistics indicate that COP quiet-standing trajectories can contain intermittent excursions that are not well captured by variance-only descriptors. From an algorithmic perspective, this matters because several widely used posturographic measures (including diffusion-based scaling and standard dispersion/correlation summaries) can be sensitive to heavy tails and to a small proportion of extreme samples. The surrogate comparison provides a pragmatic baseline: if the Gaussian surrogate preserves the gross scale of motion yet fails to reproduce central concentration or tail behavior, then distribution shape – not only variance – contributes materially to the signal structure.

The central-ellipse fraction descriptor complements tail-focused criteria by quantifying whether sway is centrally concentrated (a sharp peak) or more broadly distributed and structurally non-elliptical in the PCA plane. In practice, this can distinguish trajectories with similar RMS dispersion but different density geometry, which is relevant for classifier design and robustness testing of embedded posturography software. Likewise, differences in coupling patterns among variability descriptors across age groups suggest that "equivalent" variability metrics may not remain interchangeable under aging; this motivates reporting multiple descriptors and explicitly checking correlations rather than assuming redundancy. In particular, the near-perfect SD2–StD coupling and the weaker inverse association with RR suggest that recurrence captures aspects of temporal structure that are not reducible to simple dispersion, reinforcing the value of reporting both dispersion and recurrence-based descriptors.

Several limitations should be acknowledged. First, although the study uses the full HBEDB corpus, the data includes repeated trials per subject across conditions; treating trials as independent can inflate apparent significance. Subject-level aggregation (e.g., within-subject medians) and mixed-effects models would strengthen inferential claims, and future work should emphasize effect sizes with confidence intervals rather than p-value-only interpretation. Second, the analysis is restricted to the HBEDB acquisition protocol (60-s trials sampled at 100 Hz and low-pass filtered at 10 Hz). Distribution shape and recurrence metrics may vary with task demands, recording duration, sampling rate, filtering, and nonstationarity, so generalization to other protocols should be made cautiously. Third, Gaussian surrogates test a specific null model (matched mean vector and covariance); alternative surrogates (e.g., phase-randomized, block-resampled, or AR-matched) may be needed to separate distributional non-Gaussianity from temporal dependence. Finally, recurrence-plot outcomes depend strongly on parameter choices, especially the distance threshold ϵ; routine reporting of parameter sensitivity would improve reproducibility.

Future work should extend the evaluation to additional datasets and stance conditions, include parameter-robust recurrence analysis, and assess whether distribution-shape descriptors improve predictive models for fall risk beyond standard dispersion measures.

5 Conclusions

Principal component analysis (PCA) provided a rotated coordinate system in which the two COP increment components were linearly decorrelated and ordered by variance (Gorban and Zinovyev, 2010). This PCA-based rotation enabled consistent two-dimensional analysis in the decorrelated plane and supported the estimation of bivariate probability density functions for empirical and mean-/covariance-matched Gaussian surrogate distributions (Figure 5).

In this report, we demonstrated the non-Gaussian nature of the actual distributions (see Figs. 1, 2, 3, 4, 5) compared to the Gaussian one using complementary diagnostics. In particular, Shapiro–Wilk testing rejected normality for all AP trials (1930/1930) and for almost all ML trials (1926/1930) at $\alpha = 0.05$, while P–P plots and KDE-based estimates show systematic departures concentrated toward the tails and in the density shape. Non-Gaussianity manifests as skewness, excessive kurtosis, and, primarily, systematic outliers in empirical distributions. The fraction of outliers appears to be relatively small, ranging from 3.5% to 11%. However, one should also account for their sufficiently large deviations from the mean values there.

Nevertheless, the distributions are mostly unimodal, roughly symmetric around the mean, and have similar means, medians, and modes, so the central bulk can appear close to Gaussian even when the overall distribution is not. Therefore, while non-Gaussianity is statistically pervasive across trials, it is largely expressed through tail behavior and central-concentration differences relative to a mean/covariance-matched Gaussian model. In this sense, variance-based modelling assumptions (including the Fractional Brownian motion approach) remain useful for describing the bulk scaling-

dispersion structure, but tail-sensitive analyses and robustness claims should explicitly account for heavy-tailed excursions and distribution shape.

Six descriptors of variability (three linear and three non-linear) are strongly correlated with one another. Subtle age-related hallmarks were observed in some of them (see Figs. 7 and 8). A simple central-occupancy descriptor for bivariate displacement distributions was introduced and compared with standard nonlinear measures, such as the Recurrence Ratio, providing a compact way to quantify distribution-shape differences that are not captured by variance-only summaries.

References

Andò, B., Baglio, S., Manenti, M., Finocchiaro, V., Marletta, V., Rajan, S., Nehary, E. A., Dibilio, V., Zappia, M., Mostile, G. (2025). Investigating performance of an embedded machine learning solution for classifying postural behaviors, *Sensors* **25**, 4262.

Aydın, E. (2017). Postural balance control in women with generalized joint laxity, *Turkish Journal of Physical Medicine and Rehabilitation* **63**, 259–265.

Bao, W., Tan, Y., Yang, Y., Chen, K., Liu, J. (2023). Correlation of balance posturographic parameters during quiet standing with the berg balance scale in patients with parkinson's disease, *BMC Neurology* **23**.

Cit (2003). *Circulatory response to passive and active changes in posture*, Pennsylvania State University.

Collins, J. J., De Luca, C. J. (1993). Open-loop and closed-loop control of posture: A random-walk analysis of center-of-pressure trajectories, *Experimental Brain Research* **95**, 308–318.

Delignlères, D., Deschamps, T., Legros, A., Caillou, N. (2003). A methodological note on nonlinear time series analysis: Is the open-and closed-loop model of collins and de luca (1993) a statistical artifact?, *Journal of Motor Behavior* **35**, 86–96.

Field, A. (2024). *Discovering Statistics Using IBM SPSS Statistics*, 6 edn, Sage Publications.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals, *Circulation* **101**(23), e215–e220.

Golińska, A. K. (2013). Poincaré plots in analysis of selected biomedical signals, *Studies in Logic, Grammar and Rhetoric* **35**, 117–127.

Gorban, A. N., Zinovyev, A. Y. (2010). *Principal Graphs and Manifolds*, Handbook of Research on Machine Learning Applications and Trends, pp. 28–59.

Hudson, D., Wiltshire, T. J., Atzmueller, M. (2022). multisyncpy: A python package for assessing multivariate coordination dynamics, *Behavior Research Methods* **55**, 932–962.

Mandelbrot, B. B., Van Ness, J. W. (1968). Fractional brownian motions, fractional noises and applications, *SIAM Review* **10**, 422–437.

Marwan, N., Webber, C. L., , E., Viana, R. L. (2018). Introduction to focus issue: Recurrence quantification analysis for understanding complex systems, *Chaos An Interdisciplinary Journal of Nonlinear Science* **28**.

Moraes, C. P., Fantinato, D. G., Neves, A. (2021). Epanechnikov kernel for pdf estimation applied to equalization and blind source separation, *Signal Processing* **189**, 108251.

National Safety Council, Home and Community Overview, Injury Facts (2023). `https://injuryfacts.nsc.org/home-and-community/home-and-community-overview/introduction/`

Picoli, S., Santos, E. S. D., Deprá, P. P., Mendes, R. S. (2019). Quantifying postural sway dynamics using burstiness and interevent time distributions, *The European Physical Journal B* **92**(7).

Prieto, T., Myklebust, J., Myklebust, B. (1993). Characterization and modeling of postural steadiness in the elderly: a review, *IEEE Transactions on Rehabilitation Engineering* **1**, 26–34.

Rogers, M. E., Fernandez, J. E., Bohlken, R. M. (2001). Training to reduce postural sway and increase functional reach in the elderly, *Journal of Occupational Rehabilitation* **11**, 291–298. `https://pubmed.ncbi.nlm.nih.gov/11826729/`

Santos, D. A., Duarte, M. (2016a). Human balance evaluation database. `https://physionet.org/content/hbedb/`

Santos, D. A., Duarte, M. (2016b). A public data set of human balance evaluations, *PeerJ* **4**, e2648.

Santos, D. A., Duarte, M. (2016c). A public data set of human balance evaluations.

Shapiro, S. S., Wilk, M. B. (1965). An analysis of variance test for normality (complete samples), *Biometrika* **52**(3-4), 591–611.

Silverman, B. (2018). *Density Estimation for Statistics and Data Analysis*, Routledge.

Treger, I., Mizrachi, N., Melzer, I. (2020). Open-loop and closed-loop control of posture: Stabilogram-diffusion analysis of center-of-pressure trajectories among people with stroke, *Journal of Clinical Neuroscience* **78**, 313–316.

World Health Organization. Falls (2021). `https://www.who.int/news-room/fact-sheets/detail/falls`

Zielinski, W., Kuchar, L., Michalski, A., Kazmierczak, B. (eds) (2018). *Kernel Density Estimation and Its Application*, Vol. 23, ITM Web of Conferences.

Received February 5, 2026 , revised May 7, 2026, accepted June 27, 2026

Baltic J. Modern Computing, Vol. 14 (2026), No. 2, pp. 462–491
https://doi.org/10.22364/bjmc.2026.14.2.10

An Approach for Designing Responsible Privacy Heuristics

Beatriz PONTES DA COSTA REIS, Mohamad GHARIB

University of Tartu, Estonia

{beatriz.pontes.da.costa.reis,mohamad.gharib}@ut.ee

ORCID 0009-0001-1208-2715, ORCID 0000-0003-2286-2819

Abstract. Privacy compliance is a critical requirement for legal entities handling personal data (PD), demanding the integration of protective mechanisms into business workflows and transparency with data subjects (DSs). However, a critical gap persists: DSs struggle to understand privacy information and effectively use available protections. This human-centered challenge demands alignment between organizational processes and individuals' cognitive and behavioral capacities. Privacy heuristics (PHs) can bridge this gap by supporting user decision-making, yet their design is complex, prone to bias, and, if done irresponsibly, may lead to unethical or manipulative outcomes. This paper presents design principles for creating Responsible Privacy Heuristics (RPHs) in usable privacy-aware systems. We demonstrate applicability through online social network examples and validate through an A/B test with 12 users. Results show RPHs matched standard PHs in usability while proving slightly more effective at preventing privacy-invasive choices and fostering informed decision-making, without compromising user autonomy.

Keywords: Usable Privacy, Responsible privacy heuristic, Responsible design, Privacy Engineering, Privacy-aware systems

1 Introduction

In 2009, European Commissioner Meglena Kuneva famously stated that *"Personal data is the new oil of the Internet and the new currency of the digital world"* (Kuneva, 2009). This metaphor has only grown more pertinent, as scholars like Spiekermann et al. (Spiekermann et al., 2015) have highlighted the rise of complex personal data (PD) (also called Personal Information (PI)) markets. They, along with others (Gharib, 2022a), emphasize that PD has become a critical asset, powering services from personalized advertising and recommendation systems to risk analysis. In many contexts, particularly on social networks, the data itself is the product, generated by users and monetized by platforms.

The transformation of PD into a core product and service underscores the critical imperative to embed privacy protections within digital systems (Pattakou et al., 2018). In response, a global regulatory landscape has emerged, codified in laws and regulations such as the European Union's General Data Protection Regulation (GDPR) (European Parliament, 2016), Brazil's General Personal Data Protection Law (LGPD) (D'Oliveira and Cunha, 2024), and Japan's Act on the Protection of Personal Information (APPI) (Iwase, 2019), among others. These frameworks establish legal obligations for organizations to safeguard data subjects (DSs) by preventing various forms of PD mismanagement, including its misuse, excessive processing, improper storage, and unauthorized third-party sharing (Gharib et al., 2021).

Although legal entities handling PD must provide DSs with privacy protection mechanisms and disclose PD processing, the burden of understanding this information and using the mechanisms falls upon DSs (Jacobs and McDaniel, 2022). This creates a significant challenge, as users possess varying levels of digital literacy and often struggle with the legal jargon common in privacy policies and interface cues. This challenge reflects a broader issue: the design of processes involving PD must account for social and human factors. Traditional business processes prioritize efficiency and compliance, but privacy-aware systems must also address the psychological and behavioral dimensions of user interaction (Gharib, 2022b, 2025). Users are not passive endpoints; they are active participants with diverse expectations, preferences, and vulnerabilities. Ignoring this reality can result in user disengagement, eroded trust, and tangible harm.

A promising solution lies in the use of privacy heuristics, which can help users make informed decisions and take appropriate actions (Gharib, 2024). However, the design of these heuristics is inherently complex and prone to bias (Hjeij and Vilks, 2023). More critically, if not designed responsibly, they can unethically manipulate user judgment, leading to decisions that are immoral or socially irresponsible.

This paper aims to mitigate the burden Data Subjects (DSs) face when interacting with privacy mechanisms. To achieve this, we develop an approach centered on design principles for creating and evaluating Responsible Privacy Heuristics (RPHs). These principles guide designers in crafting privacy-aware solutions that empower users, uphold autonomy, and facilitate informed decision-making. This work extends our previous research (Da Costa Reis and Gharib, 2025) by: (1) refining our initial design principles and developing acceptance criteria (AC) for each principle; (2) validating both through expert review; (3) demonstrating the approach on realistic online social network examples; and (4) empirically validating it via A/B testing. Unlike prior work that focuses on identifying dark patterns or usability heuristics in isolation, this work integrates privacy heuristics, ethical design principles, and Design Science Research into a unified framework for designing RPHs.

The rest of this paper is structured as follows: Section 2 outlines the research baseline covering key concepts related to privacy heuristics and an analysis of both ethical and unethical design patterns. Section 3 outlines the methodology used to develop the approach, followed by a detailed description of the approach design in Section 4. Section 5 demonstrates the applicability of this approach, while Section 6 discusses the validation process. Section 7 lists and discusses the threats to the validity of this study, and finally, Section 8 concludes the paper and discusses future work.

2 Baseline

2.1 Heuristics & Privacy Heuristics

Heuristics are cognitive "rules of thumb" or "mental shortcuts" that facilitate faster decision-making (Hertwig and Pachur, 2015; Hjeij and Vilks, 2023). While they do not always guarantee optimality (Hjeij and Vilks, 2023), they are effective problem-solving tools for both well-defined and ill-defined problems by significantly reducing cognitive effort (Gharib, 2024). These shortcuts can be either instinctive (automatic) or deliberate, with experience enabling a transition between the two over time (Hjeij and Vilks, 2023).

Heuristics play a key role in online privacy, particularly in PD disclosure, where they are termed Privacy Heuristics (PHs). Sundar et al. (Sundar et al., 2020) categorize them by context: Personal, Social, and Technological. Complementing this, Marmion et al. (Marmion et al., 2017) propose six classes (e.g., Prominence, Network, Reliability) describing the cognitive principles underlying these decisions. In summary, privacy heuristics simplify complex decisions via mental shortcuts shaped by cues, experience, and biases. Table 1 synthesizes key heuristics from (Sundar et al., 2020; Marmion et al., 2017; Kitkowska, 2023) that influence privacy decision-making.

Table 1. Heuristics that can influence privacy decisions

Affect heuristic. People judge objects or events by associating them with positive or negative feelings.
Anchoring. Under uncertainty, people tend to be biased towards a reference point, or "anchor".
Choice overload. Too many options make people feel overwhelmed and influence their judgment negatively.
Contrast effect. People's decision is influenced by comparing one instance with another, instead of relying on impartial standards.
Framing. People's choice frame is set up in a way to manipulate/control the user's decision.
Instant gratification. People prioritize quick rewards at the expense of future gains.
Loss aversion. People prefer avoiding losses rather than acquiring equivalent gains.
Optimism bias. People tend to underestimate the chances of experiencing negative events and overestimate positive ones.
Social norms. People's behavior is influenced by social norms, that either play a part in guiding or constraining it.
Status quo/Default effect. People tend to favor options that maintain the current state over those that introduce change.
Authority. Recognized brand, institution, or person vouching for security influences disclosure.

2.2 Unethical & ethical design patterns

To understand responsible privacy heuristics, we first examine ethical dimensions of decision-support mechanisms, analyzing both manipulative and ethical design patterns. We begin with unethical approaches before presenting responsible alternatives.

Unethical patterns (dark patterns) were coined by Brignull (Mathur and Mayer, 2021) as "tricks that make you do things you didn't intend to." These patterns are coercive and manipulative, guiding users toward decisions that benefit service providers at the user's expense (Caragay et al., 2024). They exploit biases and heuristics by hiding privacy-preserving options and promoting greater PD disclosure (Potel-Saville and Da Rocha, 2024). Kitkowska (Kitkowska, 2023) organized these patterns into taxonomies. Table 2 presents examples of privacy-deceptive patterns (PDPs), the heuristics they trigger, and their impact on users. While deceptive patterns are well-researched, a notable gap remains in research on what should be done (Caragay et al., 2024).

Table 2: Privacy Deceptive Patterns, heuristics and effects on user (Kitkowska, 2023)

PDP	Heuristic(s)	Effect on user
Privacy Zuckering: is the use of deceptive design or persuasive tactics to manipulate users into sharing more PD than they intend to.	Choice overload, Status quo, Framing.	Users share more PD than they intended and might be unaware of how their PD is being processed.
Bad defaults: user account options are often preconfigured in a privacy-invasive manner (over-sharing) and sometimes with no alternative options available.	Default effect, Status quo, Loss aversion	Same as Privacy Zuckering.
Comparison obfuscation: hinders users from easily comparing privacy policies, data collection practices, or security features across different services.	Anchoring and Optimism bias.	Users adopting privacy-invasive options.
Forced action: users are forced to make choices immediately to use a service.	Instant gratification	Users end up sharing more PD than they intended to keep using a service.
Trick questions: deceive users into making privacy-invading choices with misleading or ambiguous wording.	Framing and anchoring.	Users will be confused and will most likely misinterpret their choices.
Attention diversion: distracts users from privacy-conscious choices by other aspects of the interface.	Anchoring and Framing	Hinders users from properly reflecting on their privacy-related choices.
Confirmshaming: steer users to make specific choices through guilt/shame. May use UI elements to induce a certain emotional state.	Affect, contrast and default effects	Users are manipulated to share more data through guilt or social pressure.
Safety blackmail: users are pressured into less optimal privacy options by implying that failing to do so could result in safety or security risks.	Functional fixedness and instant gratification.	Users end up sharing more PD than they intended to enable their accounts.

Ethical patterns, also known as fair, or responsible patterns, are decision support mechanisms designed to prioritize the user's interests over businesses, enabling them to make informed and unobstructed decisions (Potel-Saville and Da Rocha, 2024). They function as the direct antithesis to dark patterns, which are characterized by user manipulation. The main objective of ethical patterns is user empowerment, achieved through

the transparent and clear presentation of choices. Their design must therefore adhere to principles of succinctness, transparency, and accessibility. A key contribution in this area is the taxonomy by Potel-Saville and Rocha (Potel-Saville and Da Rocha, 2024) (see Table 3), which maps dark patterns (DPs) to corresponding fair patterns (FPs).

Table 3. A taxonomy of dark and corresponding fair patterns that can influence privacy decisions

Dark pattern	Fair pattern
Harmful Default: default settings are against the user's interest.	**Protective Default:** Defaults prioritize user privacy, well-being, and societal good.
Missing Information: Selective disclosure of information.	**Adequate Information:** Clear, sufficient, and relevant information with no overload.
Maze: User path to information, preferences, or choices are made unnecessarily complex.	**Seamless Path:** The user's path to information or choices is equally straightforward whether serving their interest or the provider's.
Push & Pressure: Emotional or time-based triggers pressure user decisions.	**No Pressure:** No manipulative nudges unless they serve user or societal benefits.
More than intended: Users are led through a series of steps that force them to do or give more than they originally intended.	**Free action:** Users are empowered to understand the consequences of their choices—especially regarding spending or data sharing—without unnecessary information overload.
Distorted UX: The UI is designed to mislead or trap users.	**Fair UX:** The UI ensures the clarity, shape, size, and prominence of buttons and icons.

3 Design Science Research Methodology for Developing Responsible Privacy Heuristics

The research approach was developed following the Design Science Research (DSR) methodology (Peffers et al., 2007), which aims to *"improve the state of practice and contribute to design knowledge through the systematic construction of useful artifacts"* (Smuts et al., 2022). DSR not only focuses on the creation of such artifacts but also emphasizes their demonstration, evaluation, and communication. Our approach (illustrated in Fig. 1) aligns with DSR's key steps while further refining some into sub-steps, and described as follows:

1. **Problem identification:** This phase involves recognizing a gap in current practices, a deficiency in existing solutions, or an opportunity for improvement that can be addressed through the design of a novel artifact. The problem must be clearly defined and motivated to establish the necessity and relevance of the research. As discussed earlier, a clear gap exists between the prevalence of manipulative designs and the lack of prescriptive frameworks for ethical alternatives. Therefore, this research is motivated by the need for a structured approach to guide the design and evaluation of RPHs for privacy-aware solutions.
2. **Approach design:** This phase involves the creation of a purposeful artifact (e.g., a method, model) to address the identified problem. Our approach to designing the

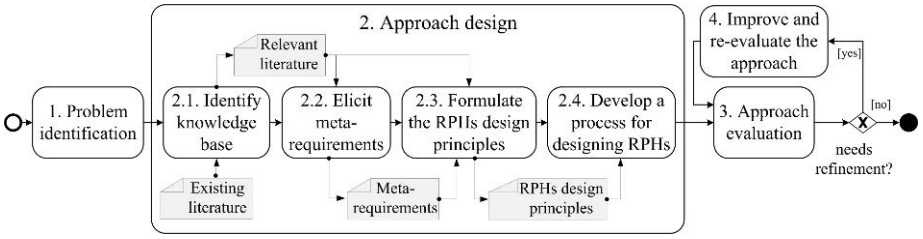


Fig. 1. The process for constructing and evaluating the approach

RPH framework is composed of four sub-steps: *2.1. Identity knowledge base*, *2.2. Elicit Meta-requirements*, *2.3. Formulate the RPHs design principles*, and *2.4. Develop a methodological process for designing RPHs*. The first three sub-steps have been adopted following the method for developing design principles in (Möller et al., 2020), and the last step offers a systematic process for using the approach. The approach design is further described in Section 4.

3. **Approach evaluation:** This phase is critical for validating the utility and efficacy of the proposed artifact. It involves gathering empirical evidence to assess how well the artifact—in this case, our RPH design approach—solves the problem identified in the first phase. This is achieved by defining specific, measurable evaluation criteria and employing appropriate methods to test the artifact against them. Our evaluation aims to assess the approach based on how well it supports the creation of solutions in the problem space, and it is further described in Section 6.
4. **Improve and re-evaluate the approach:** this step mainly focuses on identifying limitations or areas of improvement and refining the approach accordingly.

4 Approach design

This section details the methodological procedures for each step.

1. Identify knowledge base. The approach guides RPH design and evaluation for privacy-aware solutions. Design principles are prescriptive statements that aid requirement-to-design transfer and enable reuse across similar contexts (Cronholm and Göbel, 2018; Möller et al., 2020). In this step, we identified literature on ethical and unethical design patterns applicable to privacy heuristics, summarized earlier.

2. Elicit Meta-requirements. Drawing on literature, we define RPHs as user-empowering, transparent, accessible, and easy-to-understand decision-support mechanisms that respect user autonomy and enable informed privacy choices. They must be grounded in ethical principles (Respect, Beneficence, Non-maleficence (Kisselburgh and Beever, 2022), Justice, Integrity, Social Responsibility (Renaud and Shepherd, 2018)). Based on this, we elicited six Meta-Requirements (MR) listed in Table 4.

While these meta-requirements are intentionally formulated at a high level of abstraction, as they define the design objectives within the DSR process, we extend them with corresponding *Design Considerations* (see Table 4). These provide concrete guidance on how each abstract requirement can be translated into design decisions (e.g.,

avoiding biased defaults, ensuring reversibility of choices), thereby clarifying their role in guiding the derivation of actionable design principles.

Table 4. Meta-requirements for RPHs

Meta Requirement (MR) - Source
MR1. Integrity: Information presented through RPHs must be accurate, truthful, consistent, and free from incomplete representations (Renaud and Shepherd, 2018; Kisselburgh and Beever, 2022).
Design Considerations: Ensure that information is not selectively omitted; claims (e.g., "data is not shared") remain consistent across the interface and are verifiable where applicable
MR2. Non-manipulation: A RPH must not exploit cognitive biases or limitations to steer users toward privacy-invasive decisions (Ahuja and Kumar, 2022).
Design Considerations: Avoid default options that bias toward privacy-invasive choices and eliminate dark patterns such as urgency cues or emotionally loaded language.
MR3. Beneficence and non-maleficence: A RPH should maximize user benefits while minimizing privacy risks and potential harms, ensuring ethical and responsible guidance (Renaud and Shepherd, 2018; Kisselburgh and Beever, 2022).
Design Considerations: Present privacy options with a clear balance between benefits and risks, avoid high-risk defaults, and clearly communicate potential harms.
MR4. Autonomy and control: A RPH should empower user's autonomy and freedom of choice by offering genuine, meaningful, and informed options without implicit coercion (Kisselburgh and Beever, 2022; Ahuja and Kumar, 2022).
Design Considerations: Provide equally accessible options and enable users to easily modify or revoke their choices without unnecessary barriers.
MR5. Context-aware and accessible: A RPH should, when possible, adapt to different contexts by considering risk levels, situational factors, and user diversity (Sundar et al., 2020).
Design Considerations: Provide additional explanations or warnings for sensitive data and ensure accessibility for users with varying levels of digital literacy and contexts of use.
MR6. Regulatory compliant: A RPH should align with applicable data protection regulations (European Parliament, 2016).
Design Considerations: Ensure that consent mechanisms are informed, specific, freely given, granular, and revocable in accordance with applicable regulations (e.g., GDPR).

3. Formulate the RPHs design principles. The RPH design principles were formulated from the literature reviewed in the *Identify knowledge base* step. We analyzed deceptive patterns and their heuristics (Table 2) and prior research (Mathur and Mayer, 2021; Bösch et al., 2016; Gunawan et al., 2022; Ahuja and Kumar, 2022) to understand how user privacy behaviors are influenced. We drew particular insight from Ahuja and Kumar (Ahuja and Kumar, 2022), who identify 25 dark strategies and seven ethical concerns (e.g., compulsion, lack of control) grounded in four autonomy dimensions. Building on this, we formulated design principles aligned with our meta-requirements.

We then derived acceptance criteria (AC) as binary (Yes/No) checklists for each principle (Table 5), demonstrated in Section 5 and evaluated by experts (Section 6.1).

Table 5: Design principles and acceptance criteria for responsible privacy heuristics.

DP1. Neutral: A RPH should present information about privacy choices in a neutral, balanced, and clear manner, avoiding framing that could lead to biased or skewed decisions.
DP1AC1. Are all choices presented with equal prominence? (i.e., Are they displayed in a way to allow users to perceive them as equally relevant?)
DP2. Honesty and clarity: A RPH should ensure that all information presented to users is truthful, clear, and easy to understand.
DP2AC1. Is the privacy-related choice or information presented accurately and comprehensibly to users of different levels of expertise?
DP3. Navigable and actionable information: A RPH should help users easily identify, understand, and act upon privacy-related information.
DP3AC1. Is the path to privacy-related mechanisms easy to navigate to?
DP3AC2. Are actionable privacy mechanisms (e.g., privacy settings) easily recognizable and intuitive to use?
DP4. Pressure-free: A RPH should not impose time constraints, emotional manipulation, or other coercive tactics that pressure users into making privacy decisions. Instead, it should allow users to deliberate freely, ensuring informed and voluntary choices.
DP4AC1. Are users free to make privacy decisions without being subject to: time constraints; exclusive time-limited offers or other alleged financial gains in exchange for PD; coercive tactics exploiting emotional and social factors (e.g., guilt shaming, fear of missing out, and bandwagon effect)?
DP5. Benefit-Risk Balance: A RPH should prioritize user benefits while proactively minimizing potential privacy risks.
DP5AC1. Are the benefits and potential privacy risks associated with a given action clearly communicated?
DP6. Consequences awareness: A RPH should provide feedback related to privacy choices, avoiding obscuring consequences, which could affect the users' decision-making.
DP6AC1. Are the consequences of privacy choices clearly communicated to the user during or after a decision-making process, through real-time feedback or confirmation?
DP6AC2. Is information about the implications of users' privacy choices easy to find?
DP7. Empowering: A RPH should support users to select privacy choices that align with their privacy requirements.
DP7AC1. Does the privacy solution provide intuitive and customizable privacy options that enable users to control, correct, and retract their privacy choices in a way that aligns with their preferences and requirements?
DP8. Context-aware: A RPH should help users assess privacy decisions in context, considering factors such as data sensitivity, purpose of collection and use, recipient identity, and potential risks.
DP8AC1. Does the privacy solution provide users with context-specific information that allows them to assess the sensitivity and risks implied by the type of data being requested (e.g., health, financial, or personal data), the purpose for its use, and the identity of the recipients?
DP9. Situation-aware: A RPH should adapt to different situations to provide relevant, meaningful, and actionable guidance.

DP9AC1. Is privacy guidance provided based on the user's current situation (e.g., location, device type, or task being performed) and interaction context (e.g., signing up for a service vs. sharing a photo)?
DP10. Accessible and inclusive: A RPH should ensure that users, regardless of their abilities or technical expertise, can understand and act upon privacy-related information.
DP10AC1. Is privacy-related information provided in multiple formats (e.g., simple language, assistive technologies, alternative formats) to accommodate users' varying preferences, expertise, sensory needs, and disabilities?
DP11. Regulation Compliant: A RPH must not encourage or lead to violating privacy legislation (e.g., purpose limitation, data minimization).
DP11AC1. Are privacy-related choices and information compliant with the relevant privacy legislation (e.g., GDPR in Europe)?

4. Developing a methodological process for designing RPHs. The methodology for designing RPHs (Fig. 2) guides the design of RPHs, and comprises four key steps:

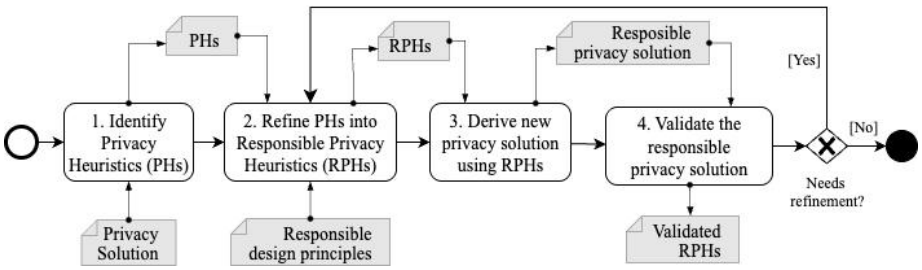


Fig. 2. The methodological process to be followed during the overall RPHs design

1. **Identify core privacy heuristics for usability:** This step takes the privacy solution as input and derives Privacy Heuristics (PH) from Gharib's ten Usable Privacy Heuristics (UPHs) (Gharib, 2024): Visibility, Revocability, Clarity, Expressiveness, Learnability, Minimalist design, Errors, Satisfaction, User suitability, and User assistance. These ensure privacy interactions are intuitive and user-friendly.
2. **Refine PHs into RPHs:** This step transforms PHs into RPHs using the responsible design principles. Designers select relevant principles and consult the acceptance criteria to guide refinement.
3. **Derive a new privacy solution based on RPHs:** Employs the selected RPHs and acceptance criteria to guide the revised design. Designers evaluate the solution using yes-or-no questions; positive responses satisfy the corresponding principles.
4. **Validate the responsible privacy solution:** Evaluates whether RPHs fulfill their purpose via end-user testing, expert reviews, or other assessment techniques, individually or in combination.

5 Applying the approach to an example from the online social network domain

This section demonstrates our approach by redesigning a social media platform's privacy settings—a user's primary tool for controlling how personal data is shared, accessed, and processed. To establish a realistic scope, we analyzed the core privacy functionalities of major Online Social Networks (OSNs), including Facebook, Instagram, LinkedIn, and Reddit. This review identified recurring features and control patterns, which informed the design of our example platform. Generally, OSNs allow users to: create personalized profiles; share content via posts; and interact through reactions, comments, and direct messaging. Table 6 outlines the privacy settings that correspond to these core functionalities in our example.

Table 6. Key privacy settings of OSN based on the scope defined.

Profile Visibility: controls the visibility of profile details (e.g., profile picture, description, birthday, email, friends list, profiles the user follows, tagged content), and activities (e.g., user's posts, and whether their comments and reactions to other's posts show up on their friends feeds).
Account & Security: controls over account details (e.g., change email used for account recovery, change password, enable/disable two-factor authentication); account deletion and temporary deactivation, and active sessions management;
Interaction Preferences: controls how others can interact with DS's profile (e.g., who can tag the DS or comment on their posts, and who can message them), and activity (e.g., who can interact with users' posts, and blocking profiles);
Ad Preferences: control and information over ads customization and manage experience (e.g., what information can be accessed and processed, learn who uses and what data they use on advertisement customization);
Permissions and Policies: control over data access and processing (e.g., service provider and third-party permissions), and policies (e.g., privacy policy, cookie policy, and terms and conditions);

To avoid redundancy among overlapping settings, we focus on the first category from Table 6: *Profile Visibility*. Following our approach, we designed five distinct privacy solutions (available in Appendix A) for this category, each with a unique interface. Their descriptions are listed in Table 7. This paper details the application of our approach to the *Profile Visibility Default Interface*; the full set of applications is available online[1]

Profile visibility default interface. In what follows, we go through the methodological process for the *Profile visibility default interface* privacy solution.

In **Step 1**, we use the default interface of the *Profile Visibility* settings as input, illustrated in Fig. 3a, then we identify the core privacy heuristics that enhance usability. We identified minimalist interventions that provides DSs with essential information for

[1] https://hdl.handle.net/10062/117140

their privacy actions, following **PH6. Minimalist design:** *A DS should be offered relevant information relating to their privacy actions*. At the top of the interface (see Fig. 3a), the DS is presented with a brief description of what these settings enable. DSs are also informed that tapping on a setting will lead them to more information about it—a feature inspired by **PH4. Expressiveness:** *A DS should be guided on privacy while still being able to have freedom of expression*; and **PH1. Visibility:** *A DS should be informed about their privacy choices*. Following this description, the privacy settings are split into two sections with descriptive names that avoid technical terms. This design choice accounts for the fact that DSs have different levels of digital literacy and familiarity with privacy settings, as suggested by **PH9. User suitability:** *DSs should be provided with options considering their diverse levels of skill and experience in security*.

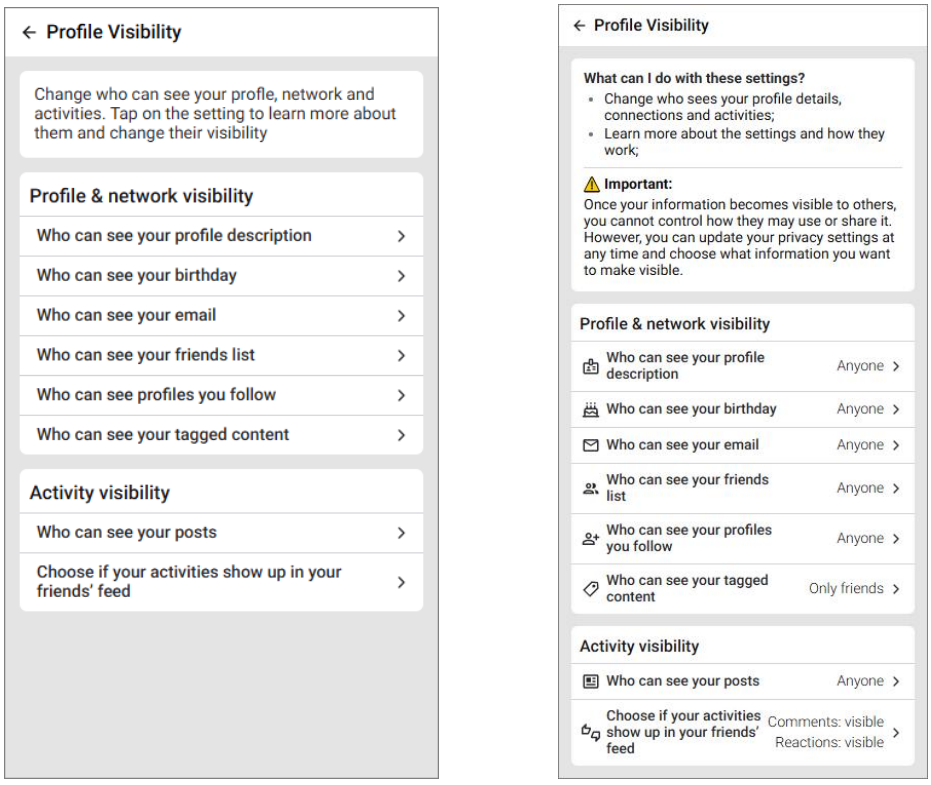
In **Step 2**, we refine the identified PHs into RPHs by applying relevant design principles. This process is documented in Table 8. For each PH, we first show its original form, followed by its refined version. Changes introduced by applying each principle are highlighted in **bold** (e.g., RPH1.1[2]).

With the PHs now refined into RPHs, we proceed to **Step 3** by revisiting the initial design solution. The updated interface, which incorporates these refinements, is presented in Fig. 3b. The following analysis examines the new design, detailing the implemented changes and the specific RPHs that informed them. Specifically, this revision transforms the description into a question to engage DSs. The content is presented in two bullet points to clearly highlight the enabled actions, adhering to **RPH6.1**'s principle of minimal, navigable design. A warning has been added to alert DSs to the privacy risks of making information visible. This aligns with **RPH1.2** by providing guidance, helping them reflect on potential consequences.

[2] To maintain traceability, each RPH retains its originating PH's number. The number after the dot indicates the version, incremented when a new design principle is applied. For example, **RPH1.1** and **RPH1.2** are both refinements of **PH1. Visibility**

Table 7. Privacy solutions used in the demonstration of the approach

Profile visibility default interface: This is the main screen of the Profile Visibility settings. From this interface, users can navigate to individual visibility settings.
Who can see your profile description setting: This interface allows users to control who can view their profile description, which includes information such as workplace, education, and locations.
Who can see your tagged content setting: This interface enables users to manage who can view posts they are tagged in, when accessed through their profile. Note that these posts may still be visible via other sources (e.g., another user's profile).
Who can see your posts setting: This interface allows users to control which groups of people can view the content they post.
Choose if your reactions or comments show up on your friends' feed setting: This interface provides two privacy controls, one for managing the visibility of reactions and another for comments. These controls only affect whether such actions appear on friends' feeds; friends may still see them by visiting the original post.



(a) Profile visibility following PH

(b) Profile visibility following RPHs

Fig. 3. Profile visibility settings following PHs and RPHs

After the description, the two-section structure is retained for consistency, but the settings descriptions have been enhanced with visual icons, and the current status is now immediately visible, removing the need for DSs to open each setting. This addition of at-a-glance visual cues is guided by **RPH6.1**, which enhances navigation through recognizable elements, and **RPH9.1**, which reinforces inclusive and accessible design. When implemented correctly, these elements support DSs in quickly locating and understanding the purpose of each control.

By displaying setting statuses directly on the *Profile Visibility* interface, we enable DSs to quickly review their current privacy choices and assess if changes are needed. This immediate visibility sets clear expectations about the available options and improves engagement, a change grounded in **RPH9.1**, **RPH6.1**, and **RPH1.2**. Furthermore, while the design presents more information, we have carefully balanced the layout to avoid overwhelming the DS or appearing to favor specific privacy actions, in strict adherence to **RPH4.1**'s principle of unbiased design.

The final step involves evaluating the design against the acceptance criteria of the applied principles. Table 9 lists how the AC for each principle has been achieved.

Table 8. Step 2 of the methodological process for profile visibility default interface.

PH1. Visibility	"A DS should be informed about their privacy choices."
RPH1.1 refined by DP9. Situation-aware	"A DS should be **provided with meaningful and actionable guidance, adapted to the interaction context, when** informed about their privacy choices."
RPH1.2 refined by DP8. Context-aware	"A DS should be provided with meaningful, actionable guidance, adapted to the interaction context **and considerate of the data sensitivity**, when informed about their privacy choices, **allowing them to recognize potential privacy risks.**"
PH4. Expressiveness	"A DS should be guided on privacy while still being able to have freedom of expression."
RPH4.1 refined by DP1. Neutral	"A DS should be **provided with unbiased and balanced information about** privacy while still being able to have freedom of expression."
PH6. Minimalist design	"A DS should be offered relevant information relating to their privacy actions."
RPH6.1 refined by DP3. Navigable and actionable privacy	"A DS should be offered relevant**, easy to learn information** relating to their privacy actions **, making sure it is easily recognizable and usable.**"
PH9. User suitability	"DSs should be provided with options considering their diverse levels of skill and experience in security."
RPH9.1 refined by DP10. Accessible and inclusive	"DSs should be provided with **inclusive** options considering their diverse levels of skill **and accessibility needs.**"

Table 9. Profile visibility default interface evaluation with AC.

DP1AC1. - Achieved: *Yes*. All links to the privacy settings follow the same pattern: icon, name, and current status.
DP3AC1. - Achieved: *Yes*. The privacy settings are one click away from the interface, and their links were updated with recognizable visual elements (e.g., icons) as well as the current setting status.
DP3AC2. - Achieved: *Yes*. Icons were chosen to resemble the functionality they are associated with (e.g., a birthday cake icon for birthday visibility). The inclusion of setting statuses upfront helps set user expectations about the controls and whether changes are needed.
DP8AC1. - Achieved: *Yes*. The new design explicitly warns users that others may use their information in unexpected ways. The privacy risks were not explicitly mentioned to avoid pressuring the DS.
DP9AC1. - Achieved: *Yes*. The description at the top and the warning serve to guide the DS on how to use these controls and what to keep in mind, given the interaction context (i.e., control of profile information visibility).
DP10AC1. - Achieved: *Yes*. The enhanced design maintains simple language, avoiding technical terms; utilizes different font weights to create contrast, and visual elements to make the design more engaging and sections of information more identifiable.

6 Validation

This section validates the proposed approach in terms of its design principles and their AC through expert review, and its effectiveness for producing RPHs for usable privacy-aware systems via end-user experiment. Consistent with early-stage Design Science Research (DSR), this evaluation is exploratory in nature. It aims to provide initial evidence of the feasibility and potential usefulness of RPHs, rather than to establish statistically generalizable conclusions.

6.1 Validation of the design principles via experts

To validate the design principles, we engaged privacy experts to complete a structured questionnaire (Available online[3]). The questionnaire evaluated the **clarity** and **applicability** of each design principle, the **validity** of acceptance criteria, and the **completeness** of the proposed design principles and their AC. The questionnaire began with an introduction to the research context and objectives, followed by the evaluation methodology. Experts then assessed each principle and its AC, presented in a table, using a 5-point Likert scale (1=negative, 5=positive). The table included dedicated spaces for specific feedback, and the survey concluded with an open-ended question on completeness. For analysis, we defined three color-coded evaluation categories based on score ranges, summarized in Table 10, to determine satisfactory outcomes and facilitate an at-a-glance interpretation of the results.

Table 10. Evaluation categories for Likert-scale questions.

Color	Score Range	Evaluation
Green	≥ 4	Positive evaluation — the principle or criterion is considered clear, applicable, and valid.
Yellow	≥ 3 and < 4	Neutral to slightly positive — the principle or criterion might require some minor adjustments.
Red	< 3	Negative evaluation — the principle or criterion might need substantial revision.

Two privacy experts completed the questionnaire, and their evaluations are referred to as *Expert A* and *Expert B* for clarity. Table 11 presents the average scores from their questionnaire responses, color-coded according to the evaluation categories defined in Table 10. We interpret the quantitative scores alongside the experts' qualitative feedback where available. The experts evaluated the principles without a detailed demonstration of their practical application, which may have contributed to some lower scores.

DP9. Situation-aware received the lowest applicability score (below 3.0) for applicability (Q2). While *Expert A* gave a low score without justification, *Expert B* (score: 3.0) noted that distinguishing it from **DP8. Context-aware** required clarification. We addressed this by prioritizing *Expert B*'s feedback and providing concrete examples to

[3] https://zenodo.org/records/17508891

Table 11. Average scores from expert evaluations on the Likert-scale questions.

	Q1. Clarity	Q2. Applicability	Q3. Validity
DP1	5.0	3.5	-
DP1AC1	-	-	3.0
DP2	5.0	3.5	-
DP2AC1	-	-	3.5
DP3	5.0	4.5	-
DP3AC1	-	-	3.0
DP3AC2	-	-	4.0
DP4	4.5	4.0	-
DP4AC1	-	-	3.0
DP5	3.5	3.5	-
DP5AC1	-	-	3.5
DP6	4.5	4.0	-
DP6AC1	-	-	4.0
DP6AC2	-	-	4.0
DP7	4.5	4.0	-
DP7AC1	-	-	3.5
DP8	4.5	4.0	-
DP8AC1	-	-	4.0
DP9	3.5	2.5	-
DP9AC1	-	-	3.0
DP10	5.0	4.0	-
DP10AC1	-	-	4.0
DP11	5.0	3.5	-
DP11AC1	-	-	3.5

improve distinction. To address this concern, Table 12 provides explicit differentiation between DP8. and DP9.

Several principles received neutral scores. For **DP1. Neutral**, *Expert B* suggested replacing "displayed" with "designed" in **DP1AC1**, while *Expert A* noted potential cultural gaps in privacy awareness among practitioners. For **DP2. Honesty and Clarity**, *Expert B* recommended minor wording improvements, though we considered some suggestions redundant given **DP10. Accessible and inclusive**.

Other neutral-scoring elements included **DP3AC1**, **DP4AC1**, **DP5. Benefit-risk balance**, and **DP5AC1**. For **DP3AC1**, *Expert A*'s feedback about assessment complexity led us to refine the criterion from "easy to learn and intuitive to use" to "easy to navigate to". Neutral scores for **DP4AC1** likely reflect the inherent subjectivity in evaluating coercive patterns. For **DP5** and **DP5AC1**, we addressed *Expert B*'s suggestion for clarification through practical demonstrations rather than just formal definitions. We additionally observed that DP5. and DP6. are conceptually close and may confuse practitioners. Both address risk communication but differ in timing and function. Consequently, we clarified these distinctions in Table 12 along with the DP8./DP9. differentiation.

DP7AC1, **DP9**, **DP9AC1**, **DP11. Regulation compliant**, and **DP11AC1** also received neutral scores primarily from *Expert A*, who did not provide any feedback to justify these evaluations. *Expert B* rated these positively and suggested evaluating **DP11** alongside other principles to prevent superficial compliance. The feedback provided for **DP9AC1** was already discussed when addressing the lowest score category. We consider the positively evaluated principles satisfactory and will not discuss them further.

Table 12. Differentiation of closely related design principles

Principle	Focus	Example Application	Boundary
DP8. Context-aware	Static characteristics of the interaction environment (data sensitivity, regulatory requirements, platform type)	Displaying a stronger privacy warning when users share health data (high sensitivity) vs. basic profile information (low sensitivity)	Focuses on *what* is being shared and *where*
DP9. Situation-aware	Dynamic, real-time user state and interaction flow (user's current task, decision fatigue indicators)	Simplified options after 5+ consecutive decisions; pause confirmation after rapid, uncharacteristic changes	Focuses on *when* and *how* the user is interacting
DP5. Benefit-risk balance	Trade-off communication between positive outcomes and negative outcomes of a privacy choice	Sharing your location enables nearby friend alerts (benefit) but allows tracking of your movements (risk)	Concerns *content* of what is communicated
DP6. Consequences awareness	Post-decision feedback and long-term implication transparency	After selecting a setting: Your birthday is now visible to 'Everyone'. This information may be used for identity verification or targeted advertising.	Concerns *timing* (during/after) and *mechanism* of communication

Experts also evaluated the principles' collective completeness and adaptability through open-ended questions. When asked, *"Do the current design principles, collectively, cover necessary aspects of responsible privacy? Are they adaptable across different privacy contexts...?"*, *Expert B* affirmed good coverage and suggested emphasizing integration throughout the Privacy by Design cycle. *Expert A* did not directly address completeness but noted in response to the open-ended prompt that regulatory compliance alone does not ensure ethical practice—a concern already reflected in **DP11**.

6.2 Validation of the responsible privacy solution via end-users

We validated the effectiveness of the approach through an A/B test with end-users, comparing a baseline interface (designed with existing PHs) against an enhanced version (designed with our proposed RPHs derived from the original PHs using our methodology). The test used the previously developed Profile Visibility settings, augmented with additional settings from the same group. Related interfaces were designed using Figma[4], a digital tool for design and prototyping. The design was then

[4] https://www.figma.com/

implemented using NextJS, ChakraUI, and Typescript and deployed using Vercel[5]. In what follows, we present the validation objectives, outline the experimental method, and analyze the results.

Objectives and Criteria. This experiment evaluates whether the RPH-based design fosters user autonomy and enables more informed decisions without compromising usability. Consequently, we assess the following criteria: **1. Informed decision:** Compares the user's actual choice against an optimal or reference action. **2. Decision awareness:** Captures evidence of user interaction with informative elements (e.g., hovers, clicks, dwell time). **3. Perceived usability:** Measures how easily participants could use the interface, ensuring the RPH version performs at least as well as the PH baseline. **4. Perceived informed decision:** Assesses whether users felt they had sufficient information to make an informed privacy choice. **5. Perceived autonomy:** Measures the extent to which participants felt free from interface pressure or manipulation. **6. Perceived consequence-awareness:** Evaluates whether the interface helped users understand potential risks and outcomes. Please note that RPH-based design is not intended to restrict disclosure but to support alignment between user intentions and outcomes. Users may deliberately choose to share PD to obtain benefits; therefore, the goal is to ensure that such decisions are informed and deliberate rather than unintentionally influenced by interface design.

Methodology. We recruited 14 participants with diverse backgrounds and varying familiarity with social media privacy settings. Two participants were excluded: one due to a language barrier affecting interface interaction, and another who failed the quality control question requiring acknowledgment that social media can negatively impact privacy. The experiments were moderated and conducted primarily remotely, with a few sessions in person. The researcher initiates screen recording (using OBS Studio[6] for remote sessions; Mac's built-in recorder for in-person). Recording begins after informing the participant which design version they will use. Participants proceed with the task, notifying the researcher upon completion, at which point the recording stops. Each participant interacted with only one design version. The protocol for all sessions was as follows:

1. **Briefing:** Participants access the experiment's
   The Google Form[7], which details the purpose, data collection (demographics, screen recording, post-experiment questionnaire), and procedure. They read instructions at their own pace, and we answer any questions. While the duration was mentioned during recruitment, we reiterate that there is no time pressure. Participants must acknowledge that social media can affect privacy (only "Yes" responses proceed) and provide informed consent before continuing.
2. **Demographic questionnaire:** Participants provide their age range, gender, occupation, education level, and familiarity with social media.
3. **Experiment:** Involves the following steps

[5] https://vercel.com/
[6] https://obsproject.com/
[7] `https://zenodo.org/records/17508891`

   (a) **Scenario:** Participants read a scenario and notify the researcher when ready to begin. The researcher then provides the application link and instructs them to split their screen between the application and the scenario.
   (b) **Access interface:** Participants access the assigned interface version (Version A for the PH baseline or Version B for the RPH-enhanced interface).
   (c) **Review and adjust privacy settings:** Participants are asked to review and select the privacy option that best matches their preference for all four different interfaces. Then, inform the researcher upon task completion.
4. **Post-experiment questionnaire:** Participants return to the Google Form to complete the post-experiment questionnaire, which includes questions related to assess perceived usability, informed decision, autonomy, and consequence awareness.
5. **Respond to any clarifying questions:** the researcher conducts a brief follow-up discussion to clarify specific choices made during the interaction or responses provided in the questionnaire.

Results. We collected data via Google Forms, screen recordings, and post-experiment interviews, which included demographic information, social media habits, and participants' rationale for their choices. To structure our findings, we first present participant demographics, then analyze the privacy settings they selected, discuss post-experiment questionnaire responses, and finally, provide a discussion on the experiment's findings.

Participant Demographics. Our sample consisted of 12 participants; their demographics are summarized in Table 13. We also categorized participants as **active** or **passive** social media users based on their self-described behavior. **Active users** regularly create and share content and interact with others' posts, while **passive users** primarily consume content with minimal interaction or private sharing. Four of the six participants using the PH version were **active** users, compared to only one in the RPH group. All **active** users were women, and all men were **passive** users. Notably, even active users reported conservative sharing habits. Participant assignment to design versions (PH vs. RPH) was balanced by age and education, but not by expertise or social media behavior, which were collected post-experiment. This resulted in the PH group having five women and one man, and the RPH group having two women and four men.

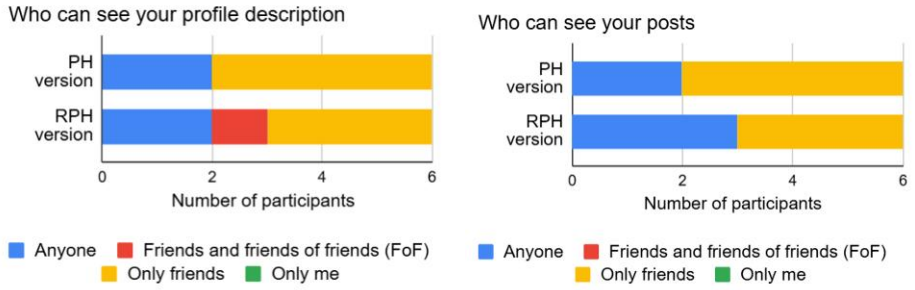
Participants privacy choices. The analysis of participant privacy choices and their interactions provides a multi-faceted comparison between the PH (baseline) and RPH (enhanced) interface versions. We focused on assessing **Informed decision** and **Decision awareness**, and participants were introduced to the following privacy configuration interfaces: 1. *Who can see your profile description?* (see 3a and 3b for PH-based and RPH-based design, respectively). 2. *Who can see your posts?* (see 10a and 10b for PH-based and RPH-based design, respectively). 3. *Who can see your birthday?* (see 11a and 11b for PH-based and RPH-based design, respectively). 4. *Who can see your email?* (see 12a and 12b for PH-based and RPH-based design, respectively). 5. *Who can see your friends' list?* (see 13a and 13b for PH-based and RPH-based design, respectively). 6. *Who can see profiles you follow?* (see 14a and 14b for PH-based and RPH-based design, respectively). 7. *Who can see your tagged content?* (see 15a and 15b for PH-

Table 13. Participants' background information.

#	Age	Gender	Employment	Education	Social media familiarity
1	26-35	Man	Full-time	Bachelor's	> 10 years
2	26-35	Woman	Full-time	Bachelor's	> 10 years
3	26-35	Woman	Homemaker	Master's	> 10 years
4	26-35	Woman	Full-time	Master's	> 10 years
5	18-25	Woman	Student	High school	5-10 years
6	26-35	Woman	Student	Master's	> 10 years
7	26-35	Woman	Part-time	Bachelor's	> 10 years
8	26-35	Man	Full-time	Bachelor's	> 10 years
9	36-45	Man	Full-time	Master's	> 10 years
10	36-45	Woman	Unemployed	Bachelor's	> 10 years
11	18-25	Man	Student	Bachelor's	5-10 years
12	26-35	Man	Full-time	High school	> 10 years

based and RPH-based design, respectively). 8. *Choose if your activities show up in your friends' feeds* (see 16 and 17 for PH-based and RPH-based design, respectively).

For profile and post visibility (Figures 4a and 4b), four participants (two per version) selected *Anyone* for professional reasons. While both groups made this choice, their processes differed: one RPH user deliberately switched to *Anyone* only after confirming optional fields, allowing sensitive data to be excluded. Screen recordings showed RPH users engaged in more deliberation, while a PH user decided quickly without reflection. This indicates the RPH version better promoted thoughtful **decision awareness**. Quantitatively, RPH users selected *Anyone* in 2/6 cases (33.3%) for profile visibility and 2/6 (33.3%) for post visibility, compared to PH users with 2/6 (33.3%) and 1/6 (16.7%), respectively.

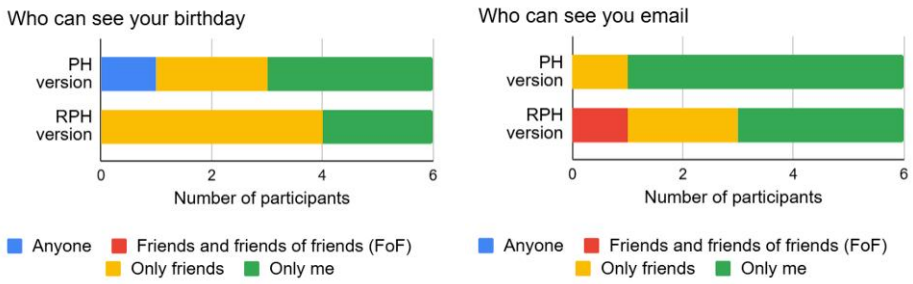


(a) (b)

Fig. 4. (a) who can see your profile description and (b) who can see your posts

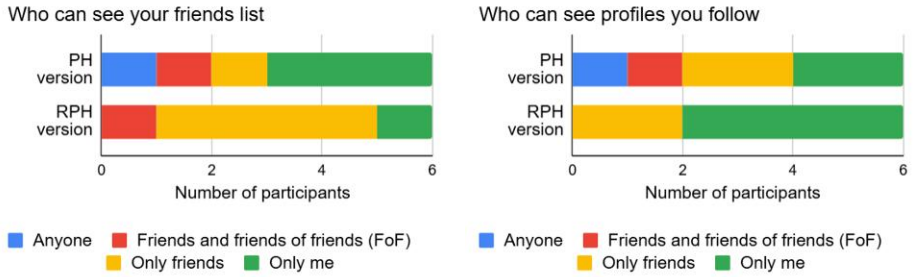
For birthday and email visibility (Figures 5a and 5b), the RPH version demonstrated superior support for informed choices. Notably, no RPH user selected the public **Any-**

one option for their birthday (0/6, 0%), unlike one PH user (1/6, 16.7%) who saw no risk. For email visibility, one RPH user (1/6, 16.7%) changed their selection from **Anyone** to a more private option after engaging with the interface; no PH user made such a change. This informed deliberation is reflected in the longer average time RPH users spent on the setting (M = 18.0s, SD = 6.2) compared to PH users (M = 8.0s, SD = 3.1), yielding a large effect size (Cohen's d = 2.06). For the friends list and followed profiles settings (Figures 6a and 6b), only a PH user selected the **Anyone** option (1/6, 16.7%), citing a professional profile strategy. The RPH version, despite minimal design differences from the PH version (see Figures 13 and 14), resulted in no users making this fully public choice (0/6, 0%).



(a) (b)

Fig. 5. Results from (a) who can see your birthday and (b) who can see your email.

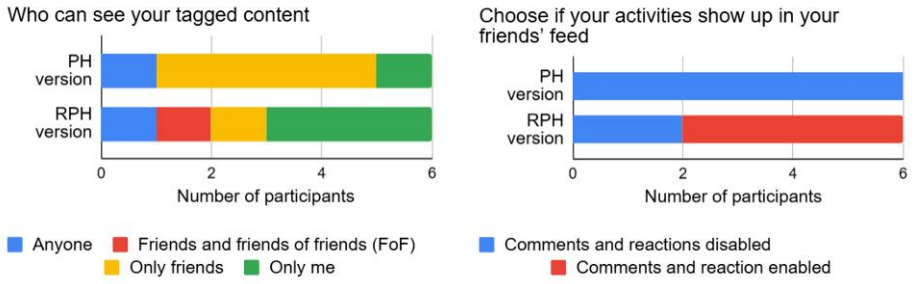


(a) (b)

Fig. 6. (a) who can see your friends list and (b) who can see profiles you follow.

For tagged content and activity feed visibility (Figures 7a and 7b), both versions had one user select **Anyone** for tagged content (1/6, 16.7% per version) based on misconceptions, indicating uninformed decisions. Despite the RPH user spending more time

on the settings (M = 22.4s, SD = 8.1) than the PH user (M = 11.2s, SD = 5.3), the risks were not effectively communicated. For the complex activity feed, the RPH version better maintained user engagement with clearer explanations. These results highlight a need for further refinement of the RPH design to enhance risk communication and decision awareness for these specific settings. Due to the exploratory, underpowered design (N=6 per condition), p-values and confidence intervals are not reported; presented statistics reflect directional trends only.

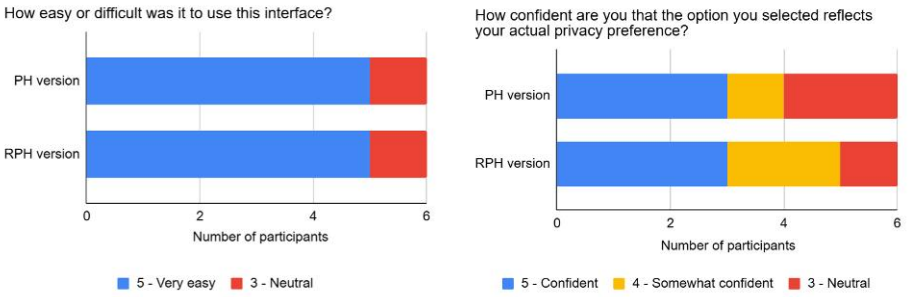


(a) (b)

Fig. 7. (a) who can see your tagged content and (b) choose if your activities show up in your friends' feed.

Post-experiment questionnaire. The questionnaire assessed perceived design aspects. To assess **perceived usability**, participants rated ease of use on a 5-point scale. The RPH version performed equally well as the PH version (Fig. 8a), despite containing more information. However, open-ended feedback revealed that some participants who rated usability as "very easy" still described both interfaces as text-heavy, suggesting information density affects perceived clarity independently of overall ease-of-use ratings. Specific critiques included unclear activity visibility (PH version) and uncertainty about tag functionality (RPH version). Concerning **Perceived informed decision**, measured on a 5-point confidence scale (1=Not confident at all, 5=Confident), the RPH version scored slightly higher than the PH version (Fig. 8b). While all participants agreed the interfaces provided sufficient information, neutral ratings were attributed to external factors: one PH user cited general platform distrust, while an RPH user reported infrequent social media use. One RPH participant reiterated concerns about tag functionality, suggesting a need for clearer descriptions.

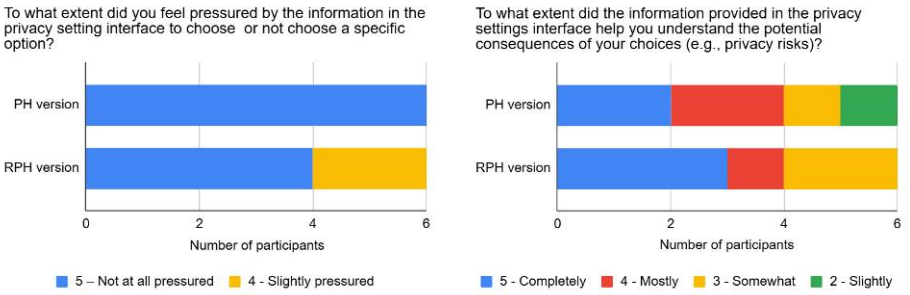
For **Perceived autonomy**, rated on a 5-point pressure scale (1=Very strongly pressured, 5=Not at all pressured), RPH users reported feeling slightly more influenced than PH users (Fig. 9a). This aligns with the RPH's design goal to encourage deliberation, as demonstrated by one user who changed their email setting from **Anyone** to **Friends and friends of friends (FoF)** after engaging with the interface guidance. Concerning **Perceived consequence awareness**, measured on a 5-point scale (1=Not at all, 5=Completely), the RPH version scored higher than the PH version (Fig. 9b). While pre-existing knowledge influenced scores, the RPH design's "soft warnings"



(a) Perceived usability results (b) Perceived informed decision results.

Fig. 8. Perceived usability and informed decision

proved effective: three participants specifically noted the informational cues prompted reflection, with one adopting a more conservative setting directly as a result.



(a) Perceived autonomy results (b) Perceived consequence awareness results.

Fig. 9. Perceived autonomy and consequence awareness

Discussion of results. Overall, the RPH version demonstrated several key advantages. It reduced selections of the most public option (**Anyone**) by 25% compared to the PH version. While both interfaces were equally **Perceived usable**, RPH significantly outperformed PH in fostering **Informed decision**, **Perceived informed decision**, and **Decision awareness**. Users of the RPH version reported higher **Perceived consequence awareness** and showed more deliberate engagement, with one user concretely changing from **Anyone** to **Friends and friends of friends** after deliberation. The increased influence reported in **Perceived autonomy** for RPH users reflects this positive, deliberate engagement rather than coercive pressure.

7 Threats to validity

This research contributes to the development of responsible privacy design, though several potential threats to validity should be acknowledged. Following Wohlin et al. (Wohlin et al., 2012), we classify threats to validity under:

1. **Internal validity** concerns whether the observed effects can be causally attributed to the experimental conditions. A key threat is the potential for participants to alter their behavior when aware of being observed. To mitigate this, we instructed participants to act naturally and assured them they were not being personally evaluated. A second threat is moderator bias, which was minimized by strictly limiting the moderator's role to answering procedural questions, with no guidance provided on interface usage or privacy choices.
2. **External validity** concerns the generalizability of our findings. The approach was evaluated using relatively simple privacy mechanisms in a generic social media context. More complex scenarios, such as cookie consent or privacy policy interfaces, remain unexamined. Future work should assess the approach's applicability across diverse privacy contexts and user groups, and validate its effectiveness in mitigating deceptive patterns within real-world applications. Moreover, the small sample size (n=12) limits statistical power and increases the risk of Type II errors. As a result, the study is not intended to establish statistically significant effects but rather to identify trends and generate hypotheses for future research.
3. **Conclusion validity** concerns the reasonableness of the study's conclusions. Given the exploratory nature of the validation and limited participant diversity, the findings regarding the effectiveness and usability of the responsible privacy heuristics indicate a modest positive impact. These findings should be interpreted as exploratory and preliminary. More extensive studies with larger, more diverse samples would strengthen confidence in these conclusions.

8 Conclusions and future work

This paper presented an approach for designing Responsible Privacy Heuristics (RPHs) to support the creation of ethical and usable privacy mechanisms. Developed through design science research, the contribution comprises 11 design principles with acceptance criteria and a methodological process for application. This provides a practical framework for translating ethical requirements into privacy solution design. Validation via an A/B test demonstrated that the RPH-based design maintained usability while slightly enhancing privacy risk awareness and supporting more informed decision-making, without compromising user autonomy.

Future research should expand on the demonstration of the proposed approach by applying the design principles to a broader range of privacy solutions, ensuring that at least one concrete example is provided for each principle. This would help illustrate their versatility, offer clearer guidance for practitioners, and potentially find gaps and ways to improve the approach. To broaden the range of the proposed approach, future research could focus on the application of the approach to privacy mechanisms, such as cookie consent banners, privacy policies, and other regulatory-driven disclosures. These

mechanisms often involve intricate trade-offs between legal compliance, user comprehension, and business goals, which could help assess the robustness and adaptability of the produced responsible privacy heuristics. Finally, we will investigate the interdependencies between RPHs and system requirements (Gharib and Mirzazada, 2025). Since privacy decisions often affect usability, accessibility, transparency, security, and compliance, understanding these dependencies could help identify conflicts and trade-offs early and support more systematic and human-centered privacy-aware design.

Acknowledgment

This work was supported by the Estonian Research Council grant "Developing human-centric digital solutions" (No. TEM-TA120).

References

Ahuja, S., Kumar, J. (2022). Conceptualizations of user autonomy within the normative evaluation of dark patterns, Ethics and Information Technology **24**(4), 52.
https://link.springer.com/article/10.1007/s10676-022-09672-9

Bösch, C., Erb, B., Kargl, F., Kopp, H., Pfattheicher, S. (2016). Tales from the Dark Side: Privacy Dark Strategies and Privacy Dark Patterns, Proceedings on Privacy Enhancing Technologies **2016**(4), 237–254.
http://c2.com/cgi/wiki

Caragay, E., Xiong, K., Zong, J., Jackson, D. (2024). Beyond Dark Patterns: A Concept-Based Framework for Ethical Software Design, Conference on Human Factors in Computing Systems - Proceedings, ACM, p. 16.
https://dl.acm.org/doi/abs/10.1145/3613904.3642781

Cronholm, S., Göbel, H. (2018). Guidelines supporting the formulation of design principles, ACIS 2018 - 29th Australasian Conference on Information Systems, Vol. 1.
https://www.diva-portal.org/smash/record.jsf?pid=diva2:1269110

Da Costa Reis, B. P., Gharib, M. (2025). Towards an Approach for Designing Responsible Privacy Heuristics, Joint Proceedings of RCIS (Research Challenges in Information Science) Workshops and Research Projects Track, Vol. 3987, pp. 1–15.
https://ceur-ws.org/Vol-3987/paper3.pdf

D'Oliveira, N., Cunha, F. J. A. P. (2024). Brazilian General Data Protection Law (LGPD): the relationship between information policy and information regime, Revista Digital de Biblioteconomia e Ciencia da Informacao **22**.
https://www.scielo.br/j/rdbci/a/DWntpkXMB9GgCPKycFcxtts/?lang=en

Parliament, E. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Da, Official Journal of the European Communities **59**, 1–88.

Gharib, M. (2022a). Privacy and Informational Self-determination Through Informed Consent: The Way Forward, Lecture Notes in Computer Science, Vol. 13106 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 171–184.
https://link.springer.com/chapter/10.1007/978-3-030-95484-0_11

Gharib, M. (2022b). Toward an architecture to improve privacy and informational self-determination through informed consent, Information and Computer Security **30**(4), 549–561.

Gharib, M. (2024). Towards a Heuristic Model for Usable Privacy, Joint Proceedings of RCIS (Research Challenges in Information Science) Workshops and Research Projects Track, CEUR-WS.org, pp. 1–10.
https://ceur-ws.org/Vol-3674/ASPIRING-paper3.pdf

Gharib, M. (2025). From Perception to Protection: A Mental Model-Based Framework for Capturing Usable Security and Privacy Requirements, The 30th Nordic Conference on Secure IT Systems (NordSec25), pp. 465–483.
https://link.springer.com/chapter/10.1007/978-3-032-14782-0_25

Gharib, M., Giorgini, P., Mylopoulos, J. (2021). COPri v.2 — A core ontology for privacy requirements, Data and Knowledge Engineering **133**, 101888.
https://linkinghub.elsevier.com/retrieve/pii/S0169023X2100015X

Gharib, M., Mirzazada, E. (2025). ReInTa: A Novel Requirements Interdependencies Taxonomy, Baltic Journal of Modern Computing **13**(4), 834–861.
https://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/13_4_05_Gharib.pdf

Gunawan, J., Santos, C., Kamara, I. (2022). Redress for Dark Patterns Privacy Harms? A Case Study on Consent Interactions, Proceedings of the 2022 Symposium on Computer Science and Law, Association for Computing Machinery, Inc, pp. 181–194.
https://dl.acm.org/doi/abs/10.1145/3511265.3550448

Hertwig, R., Pachur, T. (2015). Heuristics, History of, International Encyclopedia of the Social and Behavioral Sciences: Second Edition, pp. 829–835.
https://pure.mpg.de/rest/items/item_2139307/component/file_2404607/content

Hjeij, M., Vilks, A. (2023). A brief history of heuristics: how did research on heuristics evolve?, Humanities and Social Sciences Communications **10**(1).
https://www.nature.com/articles/s41599-023-01542-z

Iwase, H. (2019). Overview of the act on the protection of personal information, European Data Protection Law Review **5**(1), 92–98.

Jacobs, D., McDaniel, T. (2022). A Survey of User Experience in Usable Security and Privacy Research, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 13333 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 154–172.
https://link.springer.com/chapter/10.1007/978-3-031-05563-8_11

Kisselburgh, L., Beever, J. (2022). The ethics of privacy in research and design: Principles, practices, and potential, Modern Socio-Technical Perspectives on Privacy, pp. 395–426.
https://library.oapen.org/bitstream/handle/20.500.12657/52825/1/978-3-030-82786-1.pdf#page=392

Kitkowska, A. (2023). The Hows and Whys of Dark Patterns: Categorizations and Privacy, Human Factors in Privacy Research pp. 173–198.
https://library.oapen.org/bitstream/handle/20.500.12657/76226/978-3-031-28643-8.pdf?sequence=1#page=177

Kuneva, M. (2009). Roundtable on online data collection, targeting and profiling.

Marmion, V., Bishop, F., Millard, D. E., Stevenage, S. V. (2017). The cognitive heuristics behind disclosure, decisions, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10539 LNCS, Springer Verlag, pp. 591–607.

Mathur, A., Mayer, J. (2021). What makes a dark pattern... dark? design attributes, normative considerations, and measurement methods, Conference on Human Factors in Computing Systems - Proceedings, Association for Computing Machinery, pp. 1–18.

Möller, F., Guggenberger, T. M., Otto, B. (2020). Towards a Method for Design Principle Development in Information Systems, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 12388 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 208–220. https://link.springer.com/chapter/10.1007/978-3-030-64823-7_20

Pattakou, A., Mavroeidi, A. G., Kalloniatis, C., Diamantopoulou, V., Gritzalis, S. (2018). Towards the design of usable privacy by design methodologies, Proceedings International Workshop on Evolving Security and Privacy Requirements Engineering, ESPRE, pp. 1–8. https://ieeexplore.ieee.org/abstract/document/8501325/

Peffers, K., Tuunanen, T., Rothenberger, M. A., Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems **24**(3), 45–77. http://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302

Potel-Saville, M., Da Rocha, M. (2024). From Dark Patterns to Fair Patterns? Usable Taxonomy to Contribute Solving the Issue with Countermeasures, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 13888 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 145–165.

Renaud, K., Shepherd, L. A. (2018). How to make privacy policies both GDPR-compliant and usable, International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA, pp. 1–8. https://ieeexplore.ieee.org/abstract/document/8551442/

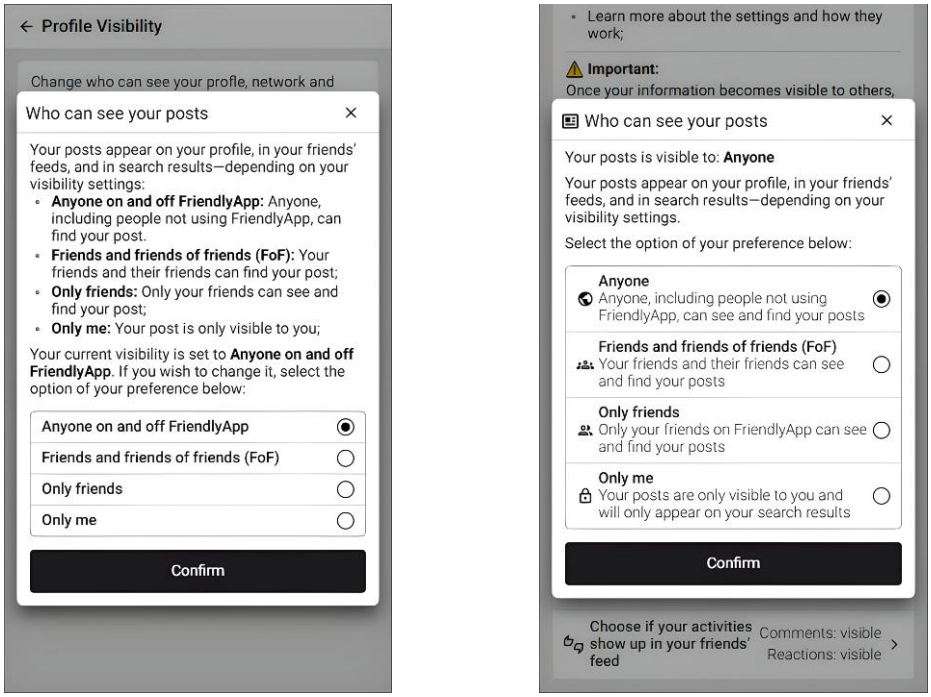
Smuts, H., Winter, R., Gerber, A., van der Merwe, A. (2022). "Designing" Design Science Research – A Taxonomy for Supporting Study Design Decisions, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 13229 LNCS, Springer Science and Business Media Deutschland GmbH, pp. 483–495. https://link.springer.com/chapter/10.1007/978-3-031-06516-3_36

Spiekermann, S., Acquisti, A., Böhme, R., Hui, K. L. (2015). The challenges of personal data markets and privacy, Electronic Markets **25**(2), 161–167. https://link.springer.com/article/10.1007/S12525-015-0191-0

Sundar, S. S., Kim, J., Rosson, M. B., Molina, M. D. (2020). Online Privacy Heuristics that Predict Information Disclosure, Conference on Human Factors in Computing Systems - Proceedings, Association for Computing Machinery, pp. 1–12.

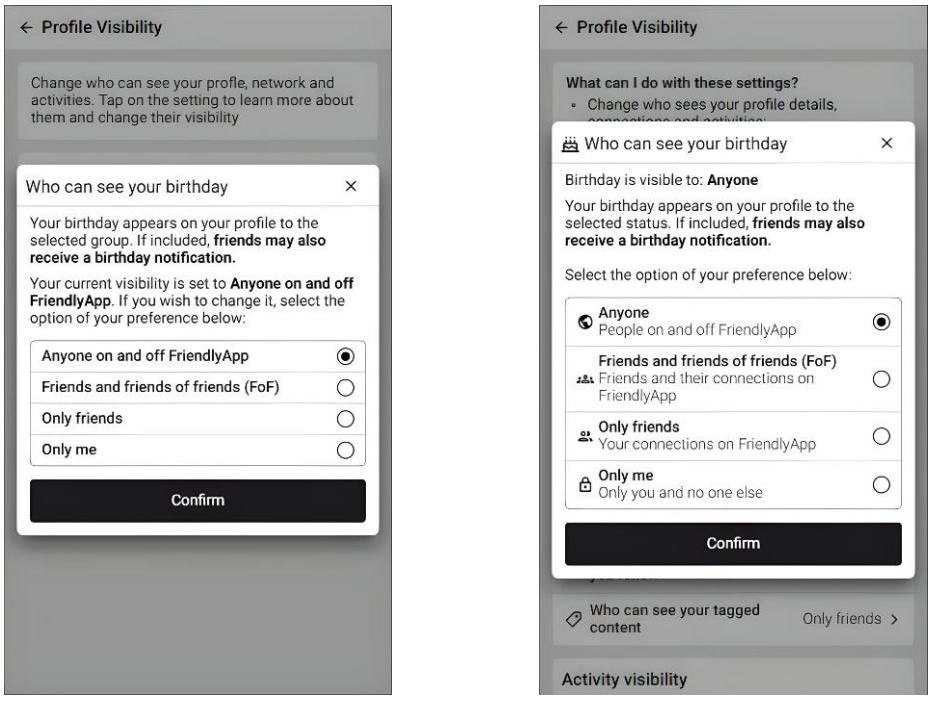
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A. (2012). Experimentation in software engineering, Vol. 9783642290, Springer Science and Business Media.

Appendix A: Responsible privacy solutions used in demonstration and end-user validation



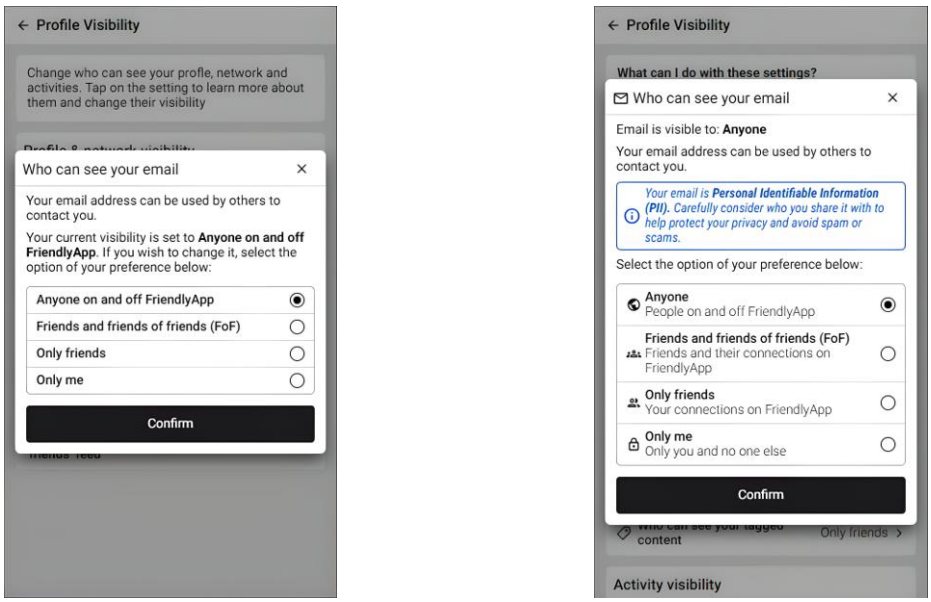
(a) (b)

Fig. 10. *Who can see your posts* (a) PH version and (b) RPH version.



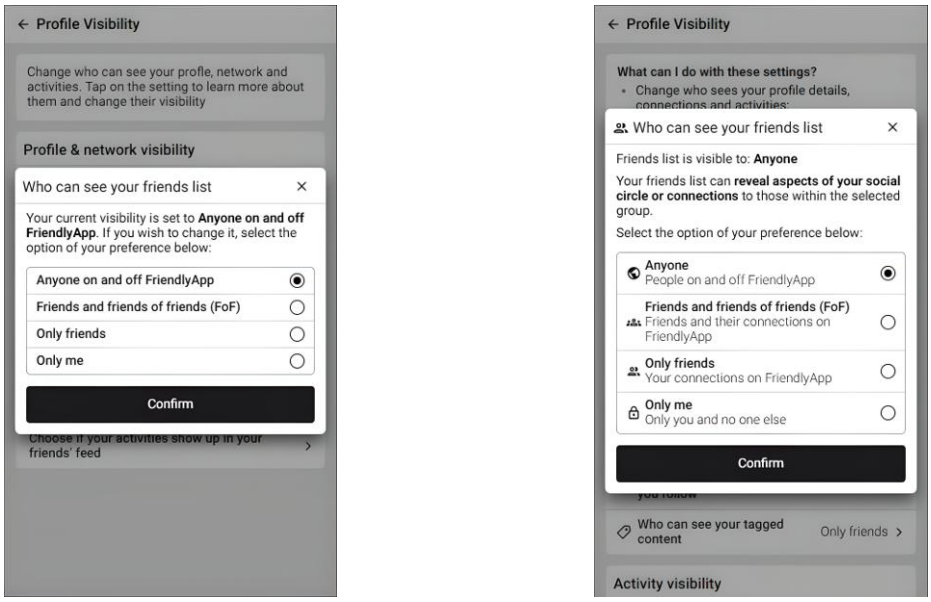
(a) (b)

Fig. 11. *Who can see your birthday* (a) PH version and (b) RPH version.



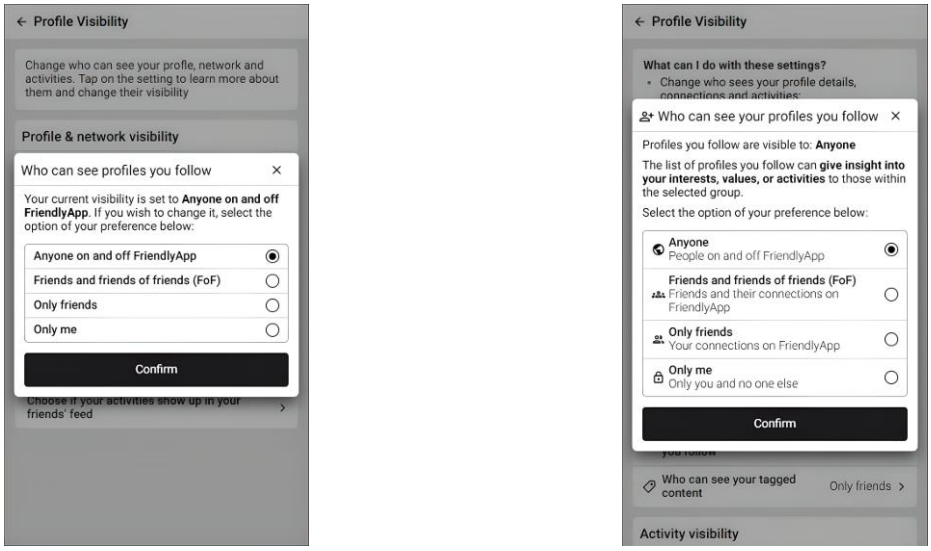
(a) (b)

Fig. 12. *Who can see your email* (a) PH version and (b) RPH version.



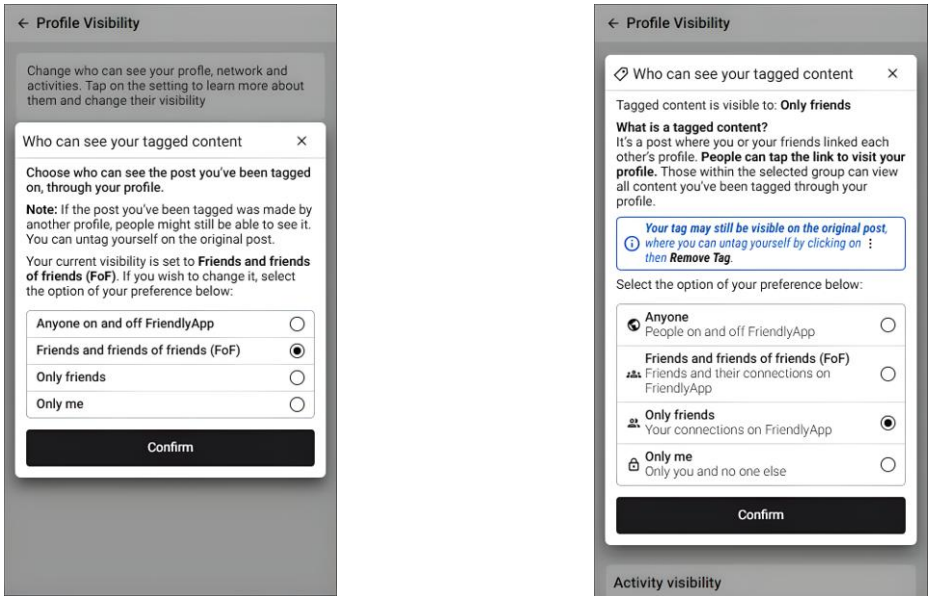
(a) (b)

Fig. 13. *Who can see your friends list* (a) PH version and (b) RPH version.



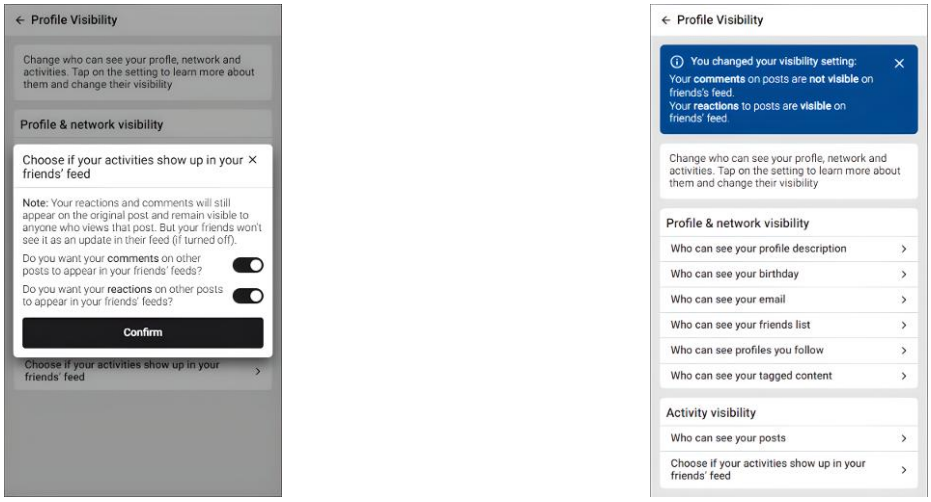
(a) (b)

Fig. 14. *Who can see profiles you follow* (a) PH version and (b) RPH version.



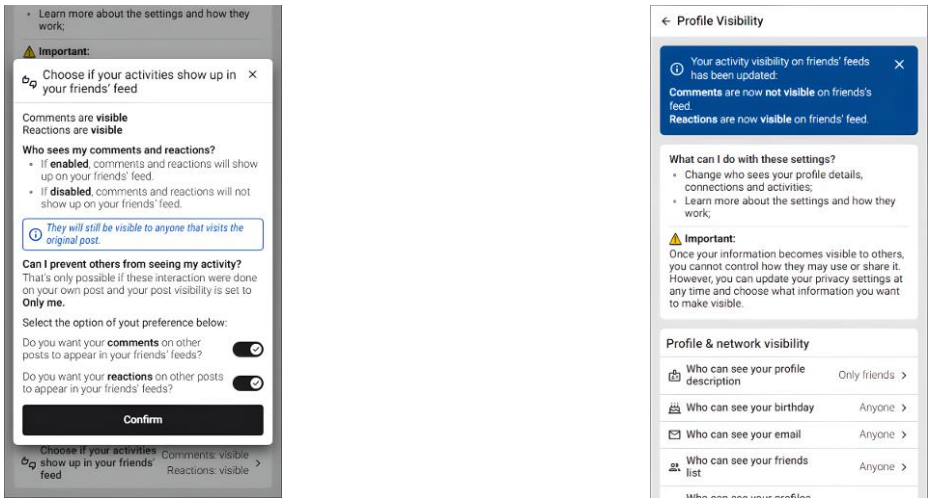
(a) (b)

Fig. 15. Who can see your tagged content (a) PH version and (b) RPH version.



(a) (b)

Fig. 16. *Choose if your activities appear in friends' feed* (a) setting and (b) feedback, RPH version.



(a) (b)

Fig. 17. *Choose if your activities appear in friends' feed* (a) setting and (b) feedback, RPH version.

Received February 19, 2026 , revised April 30, 2026, accepted June 18, 2026