Baltic J. Modern Computing, Vol. 13 (2025), No. 3, pp. 694–719
https://doi.org/10.22364/bjmc.2025.13.3.08

Random Forest Approach for pdf Malvare Detection

Anastasiia BRYHYNETS, Yaroslav KLYMENKO, Halyna HAIDUR, Sergii GAKHOV, Vitalii MARCHENKO

State University of Information and Communication Technologies, Kyiv, Ukraine

anastasiyka.br@gmail.com, yaroslavklema@gmail.com, gaydurg@gmail.com, gakhovsa@gmail.com, v.marchenko@duikt.edu.ua

ORCID 0009-0008-7631-415X, ORCID 0000-0003-0591-3290, ORCID 0000-0001-9011-8210, ORCID 0000-0003-4271-3132

Abstract. Portable Document Format (PDF) files are widely used for information exchange but have become a frequent vector for cyberattacks. Traditional signature-based and heuristic methods often fail against obfuscation and polymorphic malware, highlighting the need for more adaptive detection strategies. This study addresses the problem of PDF malware detection by applying machine learning, focusing on ensemble methods. A Random Forest model was trained on the PDFMal-2022 dataset using both static features (file size, page count, text length, image and JavaScript markers) and engineered features (text-to-size ratio, images-per-page ratio, missing text flag, and enhanced JavaScript count). Stratified cross-validation demonstrated stable performance with a macro F1-score of approximately 0.992. Feature importance analysis further confirmed the dominance of JavaScript-related attributes. The contribution of this work is to demonstrate that a lightweight and interpretable Random Forest framework can deliver state-of-the-art detection while avoiding the computational demands of deep learning.

Keywords: malware detection, random forests, portable document format, cybersecurity

1 Introduction

The proliferation of digital documents has made Portable Document Format (PDF) files an indispensable tool for information sharing. However, this widespread use has also attracted cybercriminals who exploit PDFs to deliver malicious payloads. Traditional malware detection methods, which often rely on signature-based or heuristic-based techniques, struggle to keep pace with the rapidly evolving tactics of attackers. As a result, there is a growing need for more adaptive and robust detection strategies. Machine learning, and in particular ensemble methods like Random Forest, has emerged as a promising alternative. Random Forest algorithms leverage the power of multiple decision trees to enhance predictive accuracy and mitigate overfitting. This approach is particularly well-suited to handling the heterogeneous and high-dimensional features extracted from PDF files. By analyzing various attributes—such as metadata, embedded scripts, and structural patterns—Random Forest models can effectively differentiate between benign and malicious documents.

In this study, we explore a Random Forest approach for PDF malware detection, using the PDFMal-2022 dataset as our primary resource. This dataset provides a rich collection of labeled PDF files, allowing for comprehensive experimentation and evaluation. Our investigation focuses on the process of feature extraction, the configuration of the Random Forest model, and the analysis of its performance through various evaluation metrics. Through this research, we aim to demonstrate that a machine learning-based methodology can offer significant improvements over traditional techniques. Moreover, the insights gained from our study could pave the way for more advanced and resilient malware detection systems in an era where cyber threats continue to grow in sophistication.

2 State of the art

In today's digital security landscape, traditional malware detection methods based on behavioral and heuristic analysis have become outdated and struggle to keep pace with increasingly sophisticated cyber threats. These conventional approaches often fail to adapt quickly enough to the rapidly evolving tactics of attackers, leading to a higher incidence of false positives and false negatives.

Additionally, methods that employ Artificial Neural Networks (ANN) for detecting malicious PDF files come with their own set of limitations. ANNs typically require significant computational resources and extensive training time, which can be problematic when dealing with large datasets and the need for rapid threat response. These challenges highlight the necessity for alternative approaches that can deliver a more efficient balance between accuracy, speed, and resource utilization.

In response to these issues, this study proposes the use of a Random Forest algorithm for PDF malware detection. Random Forest, as an ensemble learning method, offers several advantages: it is less susceptible to overfitting, can effectively handle high-dimensional data, and generally demands fewer computational resources compared to deep neural networks. The objective of this research is to develop and evaluate a Random Forest-based approach using the PDFMal-2022 dataset, thereby demonstrating its potential to enhance malware detection in the face of modern cyber threats.

Recent studies have underscored the growing complexity of PDF malware and the inadequacy of traditional heuristic and behavioral detection methods to combat increasingly sophisticated attacks. By extracting a set of 28 static features, including 12 newly designed ones, and applying stacking ensemble models, PDF malware can be detected with F1-scores up to 99.86% on Contagio and 98.77% on Evasive-PDFMal2022 datasets (Issakhani, 2022). Researchers have demonstrated that conventional methods often fail to capture the nuanced and obfuscated structures found in malicious PDFs, thereby necessitating the adoption of machine learning techniques (Smith et al., 2018).

In particular, the intricate nature of PDF file formats—with their diverse elements such as embedded scripts, metadata, and complex object hierarchies—demands a more adaptive and dynamic approach for effective malware detection (Ayyadevara, 2018). This evolving threat landscape has paved the way for exploring ensemble methods like Random Forest, which offer improved robustness, scalability, and interpretability compared to singular algorithmic approaches.

In recent literature, the Random Forest algorithm has gained prominence as a viable alternative to resource-intensive deep learning methods in the domain of malware detection. Studies have highlighted that Random Forest not only delivers high classification accuracy but also provides valuable insights into feature importance, which is critical for understanding and mitigating malware behavior (Johnson et al., 2019). A comparative study on CIC-PDFMal-2022 showed that while several machine learning models perform well, Random Forest remains competitive, though KNN achieved the highest accuracy of 99.86% (Khan et al., 2023). The ability of Random Forest to handle high-dimensional, heterogeneous data sets renders it particularly effective for analyzing the multifaceted characteristics of PDF documents. Moreover, its inherent resilience against overfitting, coupled with relatively lower computational demands, makes Random Forest an attractive option for real-time cybersecurity applications. A recent comparative analysis found that deep neural networks can outperform traditional machine learning for PDF malware detection, but ensemble models like Random Forest still provide more interpretability and lower training costs (Biswas and Nibras, 2025).

Further comparative investigations have revealed that integrating Random Forest classifiers into hybrid detection frameworks can substantially improve the identification of polymorphic and obfuscated malicious content. By combining static analysis techniques with the robust ensemble characteristics of Random Forest, researchers have achieved significant improvements in detection performance even under constrained computational conditions (Chiwariro and Pullagura, 2023). This integrated approach not only reinforces the reliability of the detection system but also facilitates the identification of previously undetected malware patterns. As a result, Random Forest-based methodologies are increasingly being recognized for their potential to offer both high predictive performance and operational efficiency in challenging cybersecurity environments.

Collectively, the evidence from recent studies supports the adoption of Random Forest methods as a promising pathway toward advancing the state of PDF malware detection. The advantages offered by Random Forest—such as improved accuracy, better handling of high-dimensional data, and lower resource requirements—provide a balanced solution to the limitations posed by traditional behavioral and heuristic approaches as well as resource-intensive ANN models. As the threat landscape continues to evolve, further research into optimizing feature extraction and integrating ensemble methods will be crucial for developing more resilient cybersecurity systems. This body of work underscores the need for continued exploration of ensemble-based machine learning techniques, paving the way for more effective and scalable malware detection solutions.

Traditional approaches to PDF malware detection have historically relied on signature-based, heuristic-based, and behavioral analysis methods. Signature-based detection focuses on identifying known malware patterns by matching predefined byte sequences or code fragments, while heuristic-based methods employ rules to flag suspicious file structures and content patterns. Behavioral analysis monitors the runtime activities of PDF files, looking for anomalous behaviors that may indicate malicious intent. However, these traditional methods are often challenged by polymorphic and obfuscated malware, which are designed to bypass fixed signatures and evade rule-based detection, resulting in high rates of false positives and false negatives (Smith et al., 2018). Moreover, the static nature of these techniques limits their adaptability against evolving cyber threats, necessitating more dynamic approaches.

The emergence of machine learning approaches has been driven by the need for adaptive, scalable, and data-driven detection mechanisms. Unlike traditional methods, machine learning models can learn complex patterns from large and diverse datasets, enabling them to detect subtle anomalies that may not be captured by rule-based systems. Ensemble methods, in particular, have shown significant promise by combining the strengths of multiple classifiers to improve overall accuracy and robustness. The integration of machine learning in cybersecurity has also facilitated continuous learning from new data, which is crucial for keeping pace with rapidly evolving malware variants (Ayyadevara, 2018). This paradigm shift not only enhances detection capabilities but also contributes to reducing the manual effort required for constant signature updates.

The Random Forest algorithm, as an ensemble learning method, constructs numerous decision trees during the training phase and aggregates their outputs to arrive at a final classification decision. This technique, known as bagging, helps in reducing the variance associated with individual decision trees and mitigates the risk of overfitting. Each tree in the forest is trained on a random subset of the data and features, which ensures a diverse set of classifiers that collectively capture various aspects of the data. Additionally, Random Forest provides an inherent measure of feature importance, allowing researchers to identify which attributes of PDF files—such as metadata anomalies, embedded script patterns, or structural irregularities—are most indicative of malicious behavior (Johnson et al., 2019). This insight is invaluable for refining feature engineering processes and optimizing the detection pipeline.

Random Forests are particularly well suited for PDF malware detection due to their ability to efficiently handle high-dimensional and heterogeneous data. Using only a compact set of 12 features, Random Forest reached 99.75% accuracy, showing that PDF malware detection can remain effective even with minimal feature extraction (Liu and Nicholas, 2023). The structured nature of PDF features, which may include numerical, categorical, and even binary data, is effectively managed by the algorithm's decision tree ensemble. Compared to other machine learning methods, such as Support Vector Machines (SVMs) and Neural Networks, Random Forests require less intensive parameter tuning and shorter training times, which is advantageous in operational settings where computational resources and time are critical. Moreover, while SVMs can be sensitive to the choice of kernel and Neural Networks often demand extensive training data and fine-tuning of numerous hyperparameters, Random Forests strike a balance by delivering robust performance and clear interpretability through feature ranking (CIC Evasive PDFMal2022, 2022).

Prior work indicates that ensemble tree methods—especially Random Forest (RF)—are competitive baselines for malware and PDF-malware detection. Comparative studies report RF performing on par with, and sometimes surpassing, strong alternatives such as SVM and boosting families in both accuracy and robustness while requiring lighter hyperparameter tuning (Patel, 2021; Khan et al., 2023; Lee et al., 2020). RF naturally handles heterogeneous, partly sparse feature spaces (static counts, ratios, flags) and class imbalance, and provides built-in feature importance for interpretability—advantages that are valuable for operational security settings (Lee et al., 2020). While boosting methods (e.g., gradient/Extreme Gradient Boosting) can achieve excellent accuracy, they typically demand more careful regularization and hyperparameter search, and offer less transparent explanations (Kumar et al., 2020). Given our emphasis on efficiency, deployability, and interpretability with engineered static features, RF is an appropriate primary model (Issakhani, 2022; Wiharja et al., 2024).

By leveraging an intermediate representation of PDF objects and pretrained language models, recent approaches achieve robustness against adversarial samples while maintaining a false-positive rate as low as 0.07% (Liu et al., 2025).

Despite the progress reported in recent studies (Issakhani, 2022; Khan et al., 2023; Liu and Nicholas, 2023; Biswas and Nibras, 2025; Liu et al., 2025), several research gaps remain. Many approaches rely on deep neural networks (e.g., Zhang et al., 2023; Li et al., 2024), which achieve strong detection accuracy but demand substantial computational resources and lack transparency in decision-making. Other works focus on hybrid frameworks (Choudhary et al., 2022), yet they often involve complex integration pipelines that limit practical deployment. Furthermore, while transformer-based and LLM-driven methods have been proposed for analyzing document semantics (Sowan et al., 2024; Al-Haija, 2022), their scalability in real-time malware detection scenarios remains underexplored.

The contribution of this article is to address these gaps by demonstrating that a Random Forest model, enhanced with lightweight engineered features, can deliver state-of-the-art detection performance (macro F1 = 0.992) while remaining interpretable and resource-efficient. This approach provides a practical balance between the adaptability of deep learning models and the efficiency of traditional machine learning techniques, offering a deployable solution for modern PDF malware detection.

3 About the dataset

The PDFMal-2022 dataset, curated by the Canadian Institute for Cybersecurity and hosted by the University of New Brunswick, represents a comprehensive and rigorously constructed repository of PDF documents for the study of malware detection. The dataset is systematically divided into benign and malicious subsets, providing a realistic representation of the PDF landscape in cybersecurity contexts. In particular, the malicious samples are distributed across 22 distinct directories, denoted as f1 to f22, which encapsulate a diverse array of attack vectors and obfuscation techniques. This multifaceted structure is designed to challenge and validate the robustness of detection algorithms in practical, real-world scenarios. The flowchart detailing the process of deduplication, feature extraction, and clustering of misclassified records is presented in figure 1.

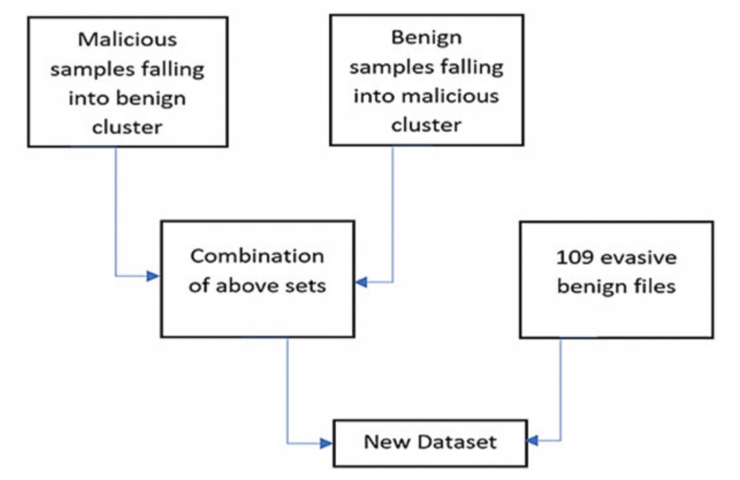

Malicious samples falling into benign cluster
Benign samples falling into malicious cluster
Combination of above sets
109 evasive benign files
New Dataset

Fig. 1. Flowchart for combining misclassified records into the Evasive-PDFMal2022 dataset (CIC Evasive PDFMal2022, 2022).

Each PDF in the dataset is amenable to extensive feature extraction, enabling researchers to derive a wide spectrum of attributes such as file size, page count, embedded scripts, and metadata characteristics. These features serve as critical inputs for machine learning models, particularly ensemble methods like Random Forests, which have proven effective in managing high-dimensional and heterogeneous data. The nuanced composition of the dataset allows for the exploration of various feature extraction strategies, ultimately facilitating the development of detection systems that are both accurate and resilient against overfitting.

In our study, the PDFMal-2022 dataset has been instrumental in the training and evaluation of a Random Forest classifier. The classifier leverages the ensemble approach to aggregate decisions from multiple trees, thereby enhancing overall predictive performance while mitigating individual model biases. The dataset's balanced representation of benign and malicious PDFs underpins the rigorous testing of the model, ensuring that the derived detection system is not only statistically robust but also practically relevant in detecting sophisticated, evasive malware.

Overall, the PDFMal-2022 dataset constitutes a vital resource for advancing research in PDF malware detection. Its comprehensive and structured composition, coupled with the rich diversity of malicious and benign samples, supports the development of innovative detection methodologies that are essential for modern cybersecurity defense. It is possible to contribute to the evolution of malware detection techniques, ensuring enhanced protection against ever-evolving cyber threats while utilizing this dataset.

4 Dataset preparation and initial training

The preparation of the dataset followed a structured sequence of steps, as illustrated in figure 2. First, benign PDF samples were collected from the PDFMal-2022 and stored in a designated directory. In parallel, malicious PDF files were also obtained from the previously mentioned dataset, ensuring a wide representation of various attack techniques. To maintain consistency during processing, all files were standardized by ensuring they carried the .pdf extension—malicious samples without an extension were appropriately renamed. Next, a unified dataset was created by merging benign and malicious files into a single directory while preserving their original labels (0 for benign, 1 for malicious). Labels were assigned accordingly to facilitate supervised learning tasks. Finally, the dataset structure was validated to confirm that all files were properly formatted as PDFs and accessible for subsequent text extraction and feature engineering.

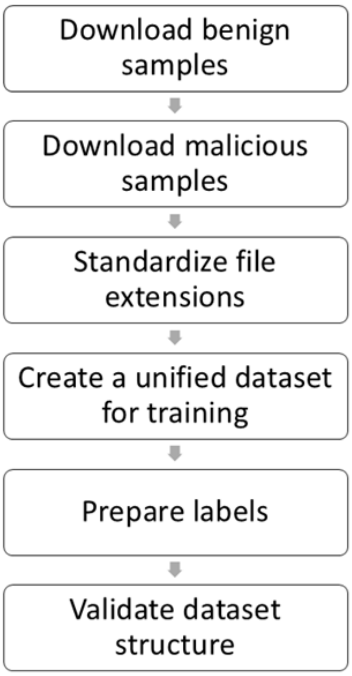
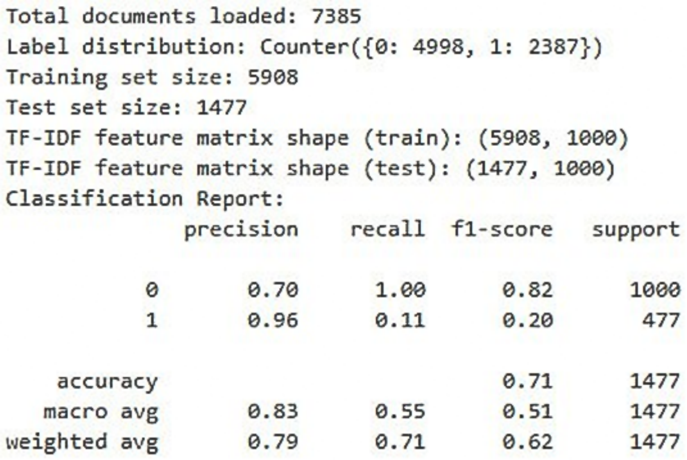

Download benign samples
Download malicious samples
Standardize file extensions
Create a unified dataset for training
Prepare labels
Validate dataset structure

Fig. 2. Data preparation algorithm.

For instance, the following code excerpt demonstrates how benign files are loaded and their text extracted using pdfminer.six:

```
benign_files = glob.glob(os.path.join(benign_folder, '*.pdf'))
print("Found benign files:", len(benign_files))
for file_path in benign_files:
    txt = load_pdf_text(file_path)
    texts.append(txt)
    labels.append(0)
```

This systematic approach guarantees reliable extraction of textual content, which is pivotal for downstream analysis via TF-IDF transformation.



```
Total documents loaded: 7385
Label distribution: Counter({0: 4998, 1: 2387})
Training set size: 5908
Test set size: 1477
TF-IDF feature matrix shape (train): (5908, 1000)
TF-IDF feature matrix shape (test): (1477, 1000)
Classification Report:
              precision    recall  f1-score   support

           0       0.70      1.00      0.82      1000
           1       0.96      0.11      0.20       477

    accuracy                           0.71      1477
   macro avg       0.83      0.55      0.51      1477
weighted avg       0.79      0.71      0.62      1477
```

Fig. 3. Initial training.

However, despite the robustness of text extraction, this procedure may neglect valuable structural details inherent in PDF documents—for example, intrinsic metadata or formatting cues—that could enhance the feature set. Although the code assigns labels based solely on folder origin (0 for benign and 1 for malicious), it does not capture these additional dimensions. To partially mitigate this limitation, the dataset is then stratified during splitting. As illustrated in the following snippet, stratification helps preserve the original class proportions:

```
X_train, X_test, y_train, y_test = train_test_split(
    texts, labels, test_size=0.2, random_state=42, stratify=labels)
```

Stratified splitting ensures that both the training and testing sets maintain balanced class distributions, even though transforming raw text into a 1,000-dimensional TF-IDF vector via TfidfVectorizer may lead to some loss of nuanced semantic information. In light of these considerations, the final dataset preparation integrates TF-IDF vectorization to convert the textual corpus into a uniform numerical representation that serves as input for machine learning algorithms. This transformation is executed by fitting the vectorizer on the training subset and subsequently transforming both subsets:

```
vectorizer = TfidfVectorizer(max_features=1000, stop_words='english')
```

```
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
print("TF-IDF feature matrix shape (train):", X_train_tfidf.shape)
print("TF-IDF feature matrix shape (test):", X_test_tfidf.shape)
```

Overall, this processing yields a consistent feature matrix for both training and test datasets, underpinning subsequent model training with algorithms like Random Forests. In this particular study, a total of 7,385 documents (with 4,998 benign and 2,387 malicious) are systematically processed, ensuring that class proportions remain balanced—a critical factor in achieving reliable performance during model evaluation.

The subsequent classification metrics reveal important performance insights. The confusion matrix (figure 4), a 2×2 table comparing true and predicted classes, illustrates that while benign documents are largely correctly identified (with a high number of true negatives), a significant portion of malicious documents are erroneously classified as benign.

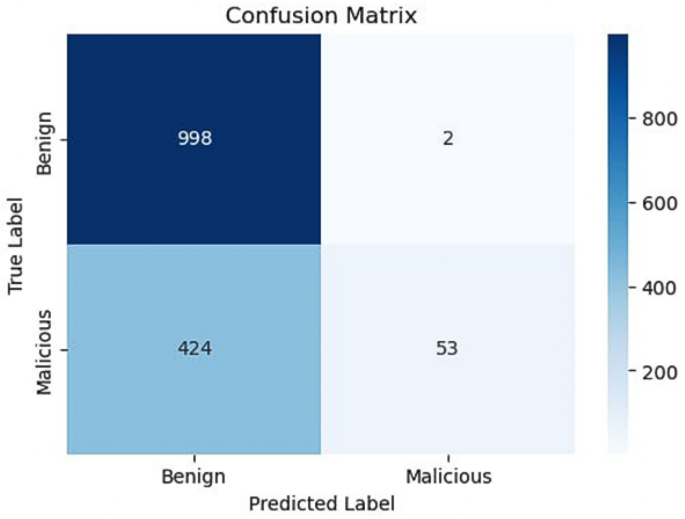


Fig. 4. Initial model confusion matrix.

Specifically, the benign class (label 0) exhibits a precision of approximately 0.70 and a recall of 1.00, indicating that while all actual benign samples are correctly identified, only 70% of the predicted benign labels are correct. In contrast, the malicious class (label 1) shows a precision of about 0.96 but a recall of only 0.11, meaning that only 11% of the actual malicious samples are successfully flagged. This imbalance results in a high false-negative rate, which is particularly concerning in malware detection scenarios where missed threats can be catastrophic.

These results underscore the importance of evaluating class-specific performance rather than relying solely on overall accuracy. In real-world contexts such as PDF malware detection, the ability to accurately flag malicious samples is critical, even if it entails some trade-offs in benign accuracy. Ultimately, the strategic combination of robust text extraction, careful stratified splitting, and TF-IDF vectorization establishes a solid foundation for the machine learning pipeline. Nevertheless, further model calibration or additional feature engineering—such as incorporating metadata or structural features—may be required to enhance the model's sensitivity to malicious indicators and reduce false negatives.

In our subsequent experiments, we modified the model by applying class weighting (i.e., setting class_weight='balanced' in the RandomForestClassifier) to address the class imbalance between benign and malicious PDFs. This adjustment was aimed at reducing false negatives in the malicious class, thereby enhancing recall. The revised model demonstrated a marked improvement in detecting malicious files—achieving a recall of approximately 99%—as evidenced by the updated confusion matrix (figure 5).

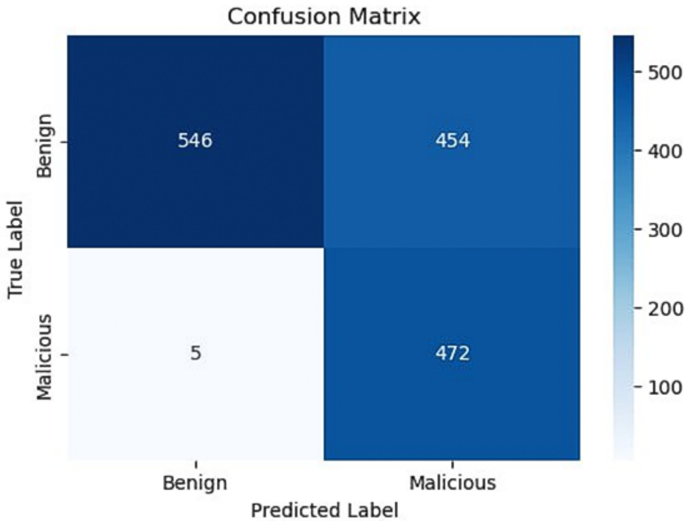
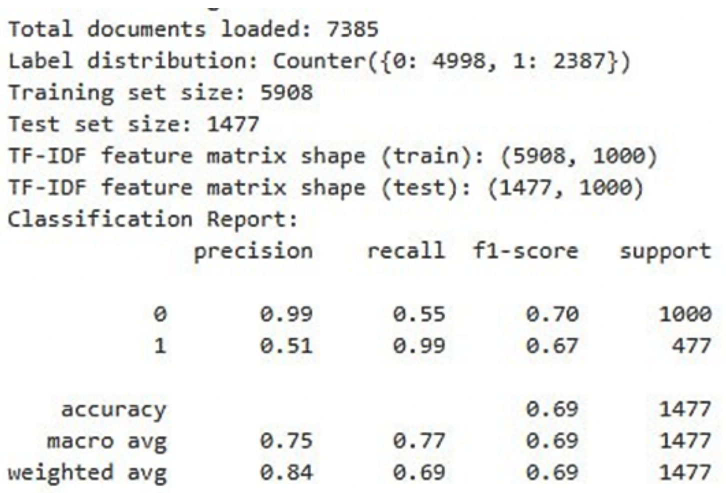


Fig. 5. Class-weighted confusion matrix.

However, this enhancement came with a trade-off: while nearly all malicious samples were caught, a substantial number of benign files were misclassified as malicious, resulting in a high false positive rate.

The classification metrics reveal a nuanced performance profile. For benign documents, the model exhibits a precision of approximately 0.99 but a recall of only about 55%, meaning that nearly half of benign files are erroneously flagged as malicious. Conversely, for the malicious class, the precision is around 0.51, though the recall is excellent. This dichotomy is critical in malware detection, where the risk of overlooking a malicious file far outweighs the inconvenience of false alarms. The sample output, as seen in figure 6, illustrates these predictions and their associated probabilities, highlighting the polarizing behavior of the model.



```
Total documents loaded: 7385
Label distribution: Counter({0: 4998, 1: 2387})
Training set size: 5908
Test set size: 1477
TF-IDF feature matrix shape (train): (5908, 1000)
TF-IDF feature matrix shape (test): (1477, 1000)
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.55      0.70      1000
           1       0.51      0.99      0.67       477

    accuracy                           0.69      1477
   macro avg       0.75      0.77      0.69      1477
weighted avg       0.84      0.69      0.69      1477
```

Fig. 6. Training results after class-weighted training.

Overall, while the high recall for malicious PDFs is encouraging from a security standpoint, the prevalent false positives indicate that further refinements are necessary. Future efforts should focus on further calibrating the model—potentially through threshold adjustments and incorporating additional structural or metadata features—to strike a better balance between the identification of threats and the minimization of false alarms. This nuanced approach is essential for developing a robust and operational PDF malware detection system that can perform reliably in real-world scenarios.

5 Training model with additional features

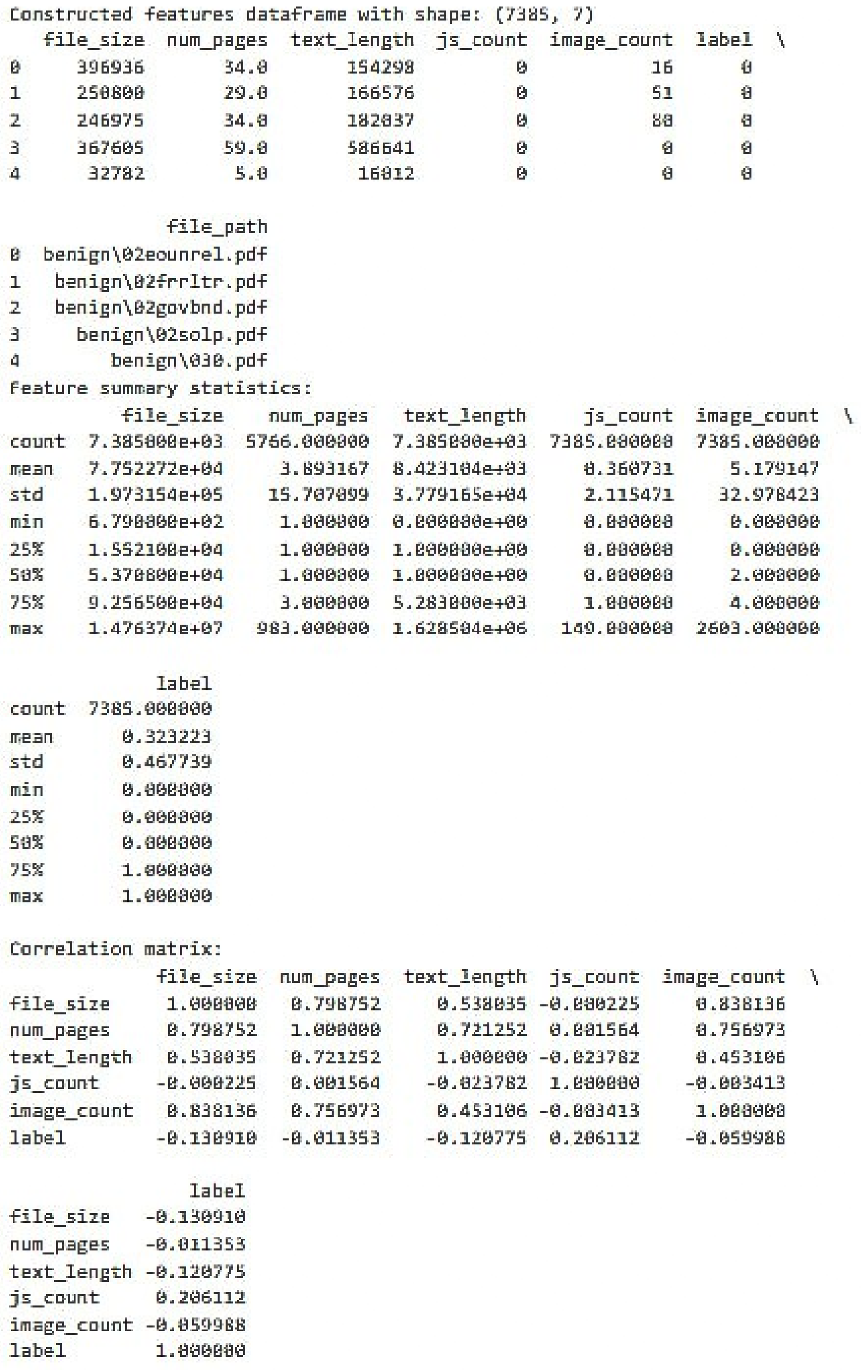
During the exploratory analysis of the PDF dataset, an error was encountered when attempting to compute the correlation matrix using the pandas function df.corr(). This error arises because the function tries to convert every column in the DataFrame to floats, including non-numeric columns such as file_path, which contains string values. As a result, the conversion fails and triggers an error.

In order to mitigate this issue, it is necessary to compute the correlation matrix exclusively on numeric columns by filtering out non-numeric data. The following code block demonstrates the implementation of a feature exploration function that performs this filtering:

```
def explore_features(df):
    print("Feature summary statistics:")
    print(df.describe())
    print("\nCorrelation matrix:")
    num_df = df.select_dtypes(include=[np.number])
    print(num_df.corr())
    features_to_plot = [’file_size’, ’num_pages’, ’text_length’,
    ’js_count’, ’image_count’]
    num_df[features_to_plot].hist(bins=30, figsize=(12,8))
    plt.tight_layout()
    plt.show()
    plt.figure(figsize=(8,6))
    sns.heatmap(num_df[features_to_plot].corr(), annot=True,
    cmap=’coolwarm’)
    plt.title("Correlation Matrix of Features")
    plt.show()
    for feature in features_to_plot:
        plt.figure(figsize=(8,4))
        sns.boxplot(x=’label’, y=feature, data=df)
        plt.title(f"{feature} by Class")
        plt.xticks([0, 1], [’Benign’, ’Malicious’])
        plt.show()
```

The revised function begins by providing an immediate overview of the dataset through the use of ‘df.describe()‘, which computes and prints summary statistics such as mean, standard deviation, and quartiles for each column. This initial step offers a quick diagnostic of the data’s distribution and potential anomalies before any deeper analysis is performed. Following this, the function narrows its focus exclusively to numeric variables by invoking ‘df.select_dtypes(include=[np.number])‘. This targeted selection ensures that only genuinely quantitative features—file_size, num_pages, text_length, js_count, image_count, and label—are fed into the correlation computation, thereby safeguarding the process against type conversion errors and irrelevant or misleading results from non-numeric data.

By filtering out non-numeric columns, the function produces a correlation matrix that is both clean and readily interpretable. Such a matrix reveals the strength and direction of pairwise relationships among quantitative features, which is invaluable for understanding how different aspects of the dataset interact. For example, the analysis highlights notably strong positive correlations between file size and both number of pages and image count (as illustrated in figure 7), suggesting that larger documents tend to contain more pages and images. These insights can guide feature engineering, variable selection, and hypothesis generation in subsequent modeling or exploratory phases.



```
Constructed features dataframe with shape: (7385, 7)
   file_size  num_pages  text_length  js_count  image_count  label  \
0     396936       34.0       154298         0           16      0
1     250800       29.0       166576         0           51      0
2     246975       34.0       182037         0           88      0
3     367605       59.0       586641         0            0      0
4      32782        5.0        16012         0            0      0

             file_path
0  benign\02eounrel.pdf
1   benign\02frrltr.pdf
2   benign\02govbnd.pdf
3     benign\02solp.pdf
4        benign\030.pdf
Feature summary statistics:
          file_size    num_pages   text_length     js_count  image_count  \
count  7.385000e+03  5766.000000  7.385000e+03  7385.000000  7385.000000
mean   7.752272e+04     3.893167  8.423104e+03     0.360731     5.179147
std    1.973154e+05    15.707099  3.779165e+04     2.115471    32.978423
min    6.790000e+02     1.000000  0.000000e+00     0.000000     0.000000
25%    1.552100e+04     1.000000  1.000000e+00     0.000000     0.000000
50%    5.370800e+04     1.000000  1.000000e+00     0.000000     2.000000
75%    9.256500e+04     3.000000  5.283000e+03     1.000000     4.000000
max    1.476374e+07   983.000000  1.628504e+06   149.000000  2603.000000

             label
count  7385.000000
mean      0.323223
std       0.467739
min       0.000000
25%       0.000000
50%       0.000000
75%       1.000000
max       1.000000

Correlation matrix:
             file_size  num_pages  text_length  js_count  image_count  \
file_size     1.000000   0.798752     0.538035 -0.000225     0.838136
num_pages     0.798752   1.000000     0.721252  0.001564     0.756973
text_length   0.538035   0.721252     1.000000 -0.023782     0.453106
js_count     -0.000225   0.001564    -0.023782  1.000000    -0.003413
image_count   0.838136   0.756973     0.453106 -0.003413     1.000000
label        -0.130910  -0.011353    -0.120775  0.206112    -0.059988

                label
file_size   -0.130910
num_pages   -0.011353
text_length -0.120775
js_count     0.206112
image_count -0.059988
label        1.000000
```

Fig. 7. DataFrame shape, summary statistics, and correlation matrix.

The constructed features DataFrame contains 7,385 PDF files with seven key attributes extracted from each document. These attributes include the file size (ranging from as little as 679 bytes to nearly 14.76 MB), the number of pages (with a median of 1 but outliers extending to 983 pages), and text length, which varies significantly among documents. Additional features such as js_count (a proxy for embedded JavaScript) and image_count (indicating the number of image objects) provide further context. The dataset exhibits a mean label value of approximately 0.323, reflecting that roughly 32% of the files are malicious. These numerical summaries inform subsequent model calibration and feature engineering steps (figure 8).

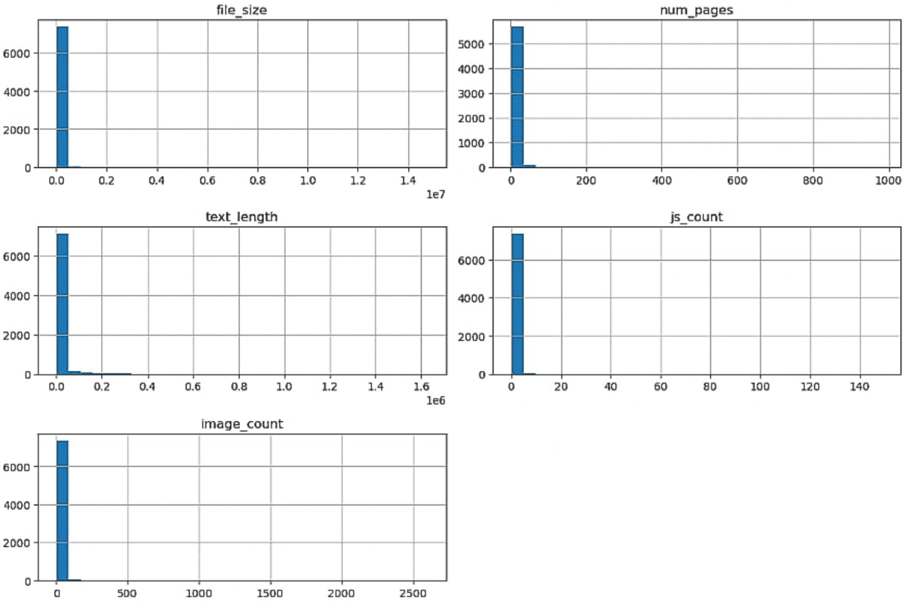


Fig. 8. Histograms of key PDF features.

Analysis of the numeric features reveals several critical relationships. For instance, there is a strong positive correlation between file size and both the number of pages (0.7988) and image count (0.8381), indicating that larger files tend to include more images and pages. Moderate positive correlations are observed between file size and text length (0.5380), though this relationship is attenuated in malicious PDFs that may rely on visual content rather than text. Moreover, the js_count exhibits a moderate positive correlation (0.2061) with the label, suggesting that an increase in the occurrence of JavaScript markers may be associated with malicious samples. Such findings are instrumental for guiding the selection of features that will be most influential in predictive modeling (figure 9).

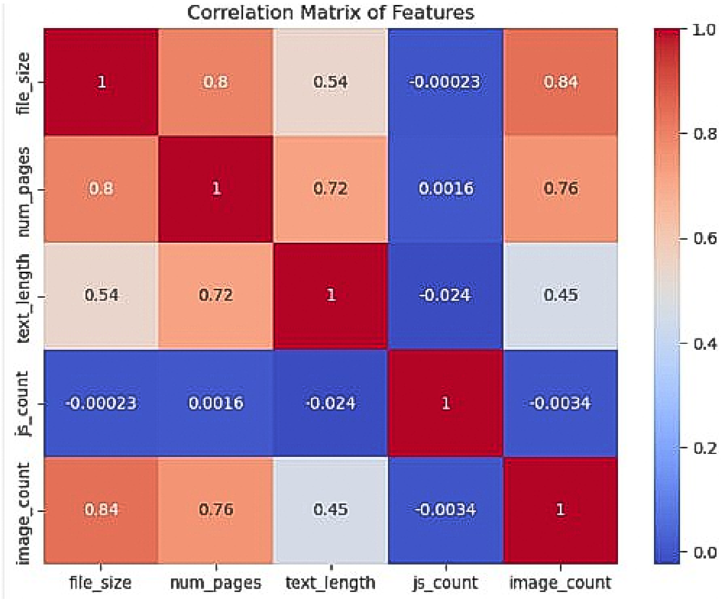


Fig. 9. Feature correlation heatmap.

The correlation analysis not only validates the choice of numeric features but also underscores opportunities for deeper feature engineering. By deriving normalized metrics—such as the ratio of text length to file size or the density of images per page—you can better distinguish malicious from benign PDFs irrespective of their absolute size or content volume. Moreover, files exhibiting zero text content warrant special attention, as they may rely heavily on embedded objects; quantifying this reliance could enhance a hybrid modeling approach that combines both text-based and structural indicators.

Building on these insights, the refined analysis lays the groundwork for optimizing model performance, especially when high recall of malicious files is paramount. Next steps include tuning classification thresholds to balance false positives and negatives, applying techniques to mitigate class imbalance (for example, through resampling or cost-sensitive learning), and expanding the feature set to incorporate additional behavioral or metadata attributes.

In parallel, rigorous validation and deployment strategies are essential to ensure the real-world efficacy of the detection pipeline. Implementing cross-validation with stratified folds will provide robust estimates of performance across diverse PDF types, while hold-out test sets drawn from unseen data sources can expose overfitting or domain shifts. Monitoring model predictions post-deployment—by sampling flagged and unflagged files for manual review—will help track drift in feature distributions and maintain high detection fidelity over time. Finally, integrating feedback loops from security analysts can guide continuous refinement of both feature engineering and threshold settings, creating an adaptive system that evolves alongside emerging PDF-based attack techniques.

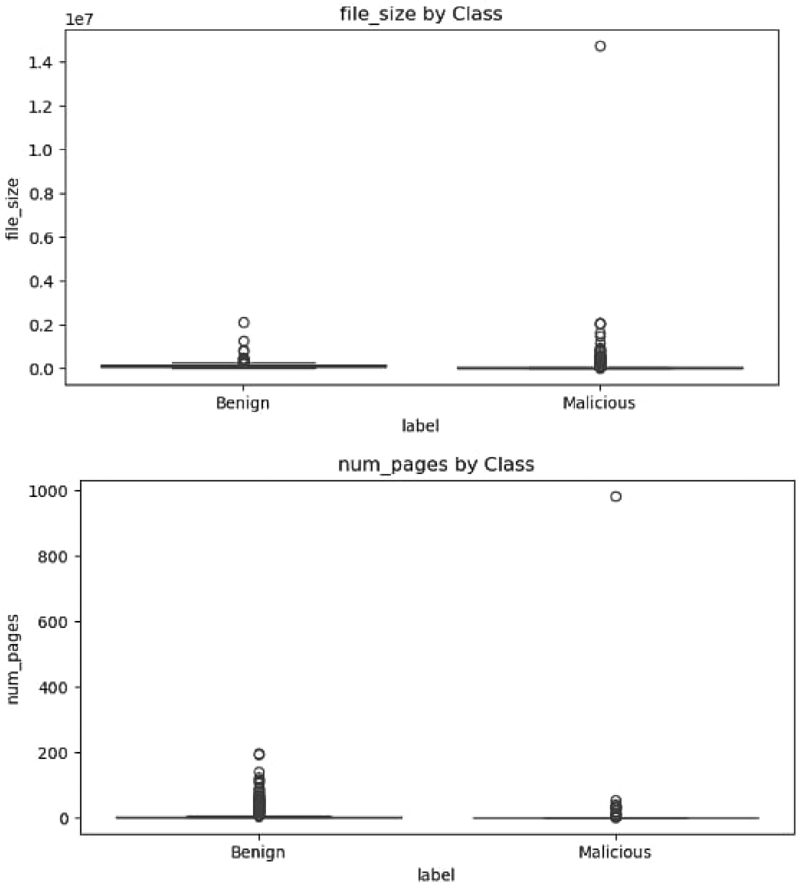


Fig. 10. Boxplots of key PDF features by class (set 1).

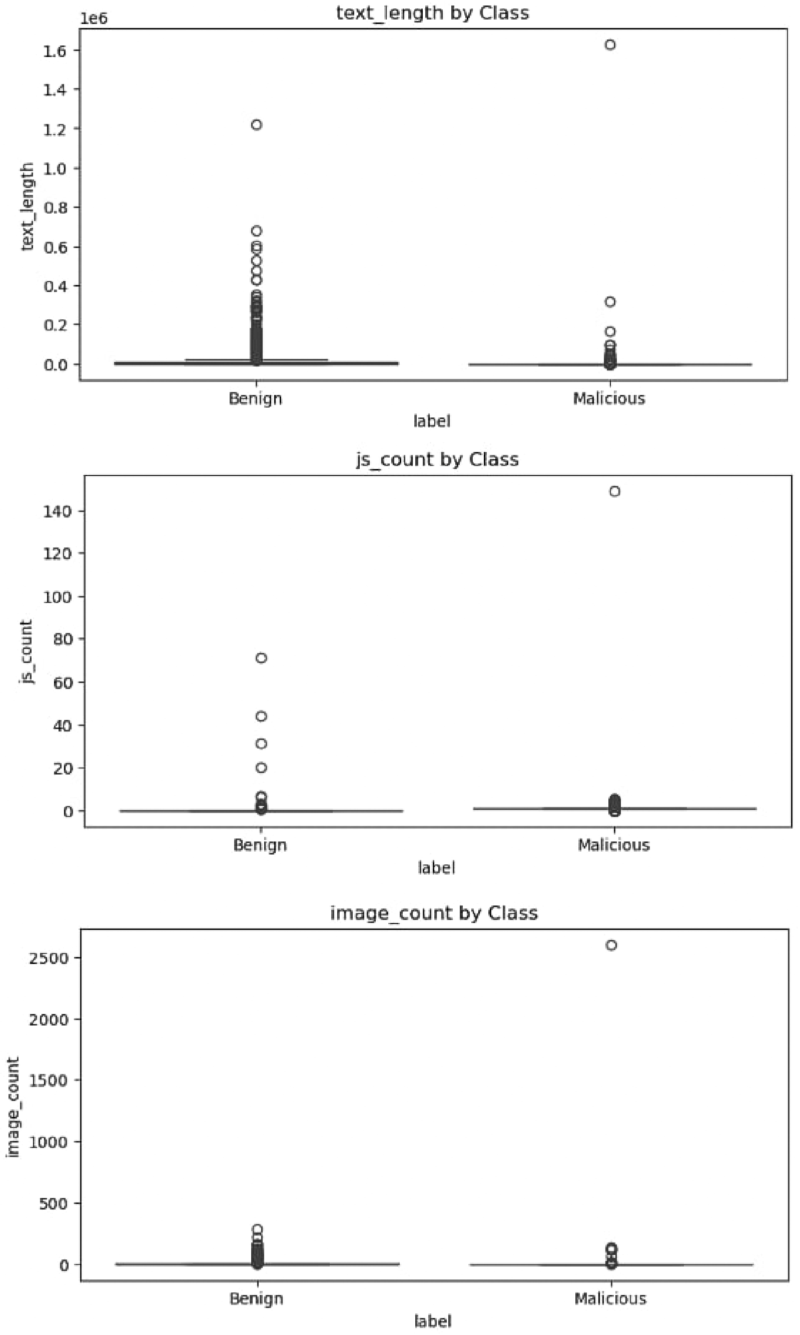


Fig. 11. Boxplots of key PDF features by class.

In conclusion, by filtering out non-numeric columns for the correlation analysis, we have obtained a clear and error-free representation of the relationships among the static features. The computed correlation matrix and accompanying visualizations reveal strong associations—for instance, between file size and both the number of pages and image count—as well as moderate signals, such as the positive correlation of JavaScript count with malicious labels. These insights underscore the importance of the selected features and pave the way for further refinement.

Building on this foundation, the next phase will incorporate additional, engineered features to enhance the model's discrimination capability. This will involve normalizing key metrics (such as the text-to-size ratio and images-per-page) and potentially integrating hybrid approaches that combine static features with text-based representations. Such enhancements aim to overcome the limitations of the current feature set and improve the overall predictive performance in detecting malicious PDFs.

6 Feature-engineered model

The initial feature set includes easily measurable properties such as file size, number of pages, text length, and the counts of JavaScript-related markers and image references. However, preliminary analysis indicated that relying solely on these static features may not sufficiently capture the intrinsic differences between benign and malicious documents. This observation motivated the incorporation of derived metrics intended to provide further nuance to the model's input.

The first phase in our methodology involves systematically extracting features from a large corpus of PDF files, which are sourced from distinct directories representing benign and malicious samples. The extraction process leverages a glob pattern to isolate files with a ".pdf" extension and utilizes pdfminer.six to extract textual content. For example, the following code snippet demonstrates the process for benign files:

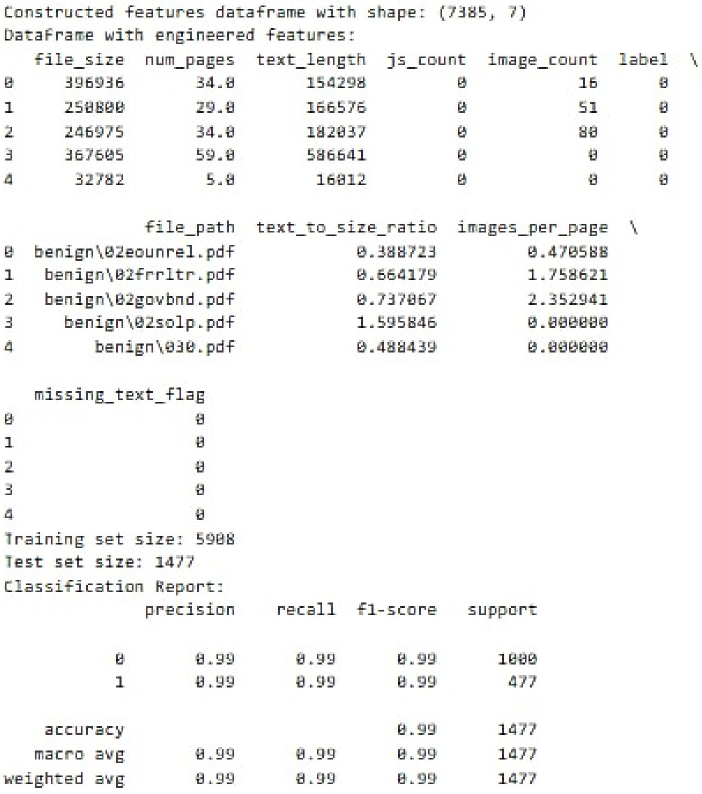
```
benign_files = glob.glob(os.path.join(benign_folder, '*.pdf'))
print("Found benign files:", len(benign_files))
for file_path in benign_files:
    txt = load_pdf_text(file_path)
    texts.append(txt)
    labels.append(0)
```

This procedure ensures that a robust set of static features is obtained, forming the basis for subsequent analysis.

To address potential deficiencies in the raw feature set, several engineered features were introduced. These include the text-to-size ratio, which normalizes extracted text length by file size; the images-per-page ratio, which reflects the density of image objects; and a binary flag (missing_text_flag) to mark documents with no extracted text. The enhancement of the JavaScript count (js_count) by aggregating occurrences of "/JS" and "/JavaScript" further enriches the feature set. The engineered features are computed using the following approach:

```
df['text_to_size_ratio'] = np.where(df['file_size'] > 0,
df['text_length'] / df['file_size'], 0)
df['images_per_page'] = np.where(df['num_pages'] > 0,
df['image_count'] / df['num_pages'], 0)
df['missing_text_flag'] = df['text_length'].apply(lambda x:
1 if x == 0 else 0)
```

This normalization provides more interpretable signals that contribute to the predictive power of the machine learning model. The enhanced dataset, comprising both the static and engineered features, is used to train a Random Forest classifier. A stratified train–test split ensures that the proportions of benign and malicious samples are preserved. Class weighting is applied to address any residual imbalance, thereby improving the detection of the minority class. The trained model achieves near-perfect classification on the test partition. The evaluation includes the generation of a confusion matrix that delineates the true positives, false negatives, and other classification outcomes (figures 12-13).



```
Constructed features dataframe with shape: (7385, 7)
DataFrame with engineered features:
   file_size  num_pages  text_length  js_count  image_count  label  \
0     396936       34.0       154298         0           16      0
1     250800       29.0       166576         0           51      0
2     246975       34.0       182037         0           80      0
3     367605       59.0       586641         0            0      0
4      32782        5.0        16012         0            0      0

            file_path  text_to_size_ratio  images_per_page  \
0  benign\02eounrel.pdf            0.388723         0.470588
1   benign\02frrltr.pdf            0.664179         1.758621
2   benign\02govbnd.pdf            0.737067         2.352941
3     benign\02solp.pdf            1.595846         0.000000
4        benign\030.pdf            0.488439         0.000000

   missing_text_flag
0                  0
1                  0
2                  0
3                  0
4                  0
Training set size: 5908
Test set size: 1477
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1000
           1       0.99      0.99      0.99       477

    accuracy                           0.99      1477
   macro avg       0.99      0.99      0.99      1477
weighted avg       0.99      0.99      0.99      1477
```

Fig. 12. Engineered features DataFrame and classification metrics.

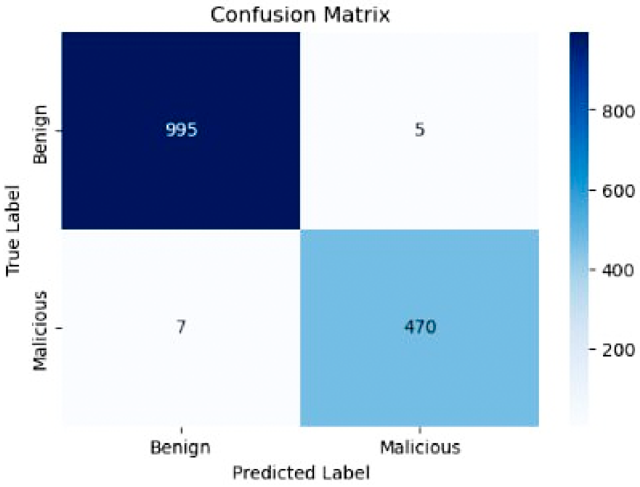


Fig. 13. Confusion matrix for feature-engineered training.

An analysis of the importance of the characteristics is conducted to interpret the decision-making process of the model. The Random Forest classifier highlights that the enhanced js_count is the most influential feature, suggesting that the frequency of JavaScript indicators is a critical marker for identifying malicious PDFs. Secondary features such as file size and the number of pages also play significant roles, while the normalized features (text_to_size_ratio and images_per_page) contribute additional discriminative value (figure 14).

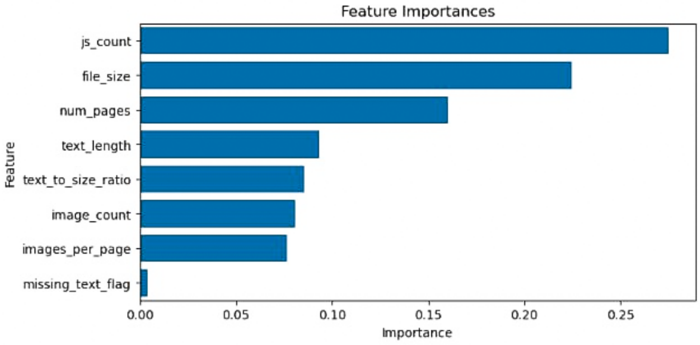


Fig. 14. Feature importances histogram.

To assess the model's generalization ability and guard against overfitting to any single data split, we employed a stratified k-fold cross-validation scheme. In each of the k iterations, the data were partitioned into training and validation sets in such a way that the proportion of benign and malicious samples remained constant across folds. Performance was then evaluated on each held-out fold, and the resulting scores were averaged to yield a more reliable estimate of the model's effectiveness on unseen data. Across all folds, the macro F1 score, a metric that computes the F1 score independently for each class and then takes their unweighted mean—varied only slightly, ranging from approximately 0.9899 to 0.9946, with an overall mean of 0.9922. These consistently high values demonstrate that the classifier performs stably across different train–validation partitions.

Taken together, these findings highlight the contribution of both static features such as file size and page count—and engineered features, including the enhanced JavaScript count and ratio-based measures (e.g., text-to-size ratio and images per page). When integrated into a Random Forest framework with class weighting, these attributes provide high-fidelity separation between benign and malicious documents.

It should be noted that the transformation of text content in this study was limited to TF-IDF representations, which primarily capture surface-level statistical patterns of word occurrence. While effective for distinguishing broad textual differences, this approach does not fully capture the semantics of document content, particularly in cases where malicious intent is concealed through obfuscation or natural-language masking. Recent research has explored transformer-based and large language model (LLM) approaches that generate contextual embeddings capable of modeling deeper semantic relationships in text (Sowan et al., 2024; Al-Haija, 2022). Incorporating such techniques could enhance the model's ability to detect complexly masked threats, albeit at the cost of increased computational overhead. Future work will therefore investigate hybrid pipelines that combine the efficiency of engineered features with the semantic depth of transformer-based representations to further strengthen PDF malware detection systems.

To mitigate the risk of data leakage and ensure that the reported performance metrics are not artificially inflated, several precautions were taken. First, stratified splitting was performed strictly at the file level, ensuring that no duplicate or near-duplicate PDFs appeared across both training and validation folds. This prevented the classifier from memorizing artifacts of identical or overlapping files. Second, all feature engineering operations were designed without access to label information, thereby avoiding any inadvertent leakage of class labels into the feature space. Finally, cross-validation was conducted using stratified folds, which preserved the original benign-to-malicious ratio while maintaining strict separation between training and validation sets. These mechanisms collectively reduce the likelihood of optimistic bias and provide confidence that the macro F1-score of approximately 0.992 reflects genuine generalization rather than leakage.

7 Discussion

The experimental results indicate that the Random Forest classifier, enhanced with both static and engineered features, achieves consistently high performance on the PDFMal-2022 dataset. The model's macro F1-score of approximately 0.9922 across stratified folds demonstrates robustness and stability, while the feature importance analysis highlights the frequency of JavaScript indicators (`js_count`) as a dominant factor in detecting malicious PDFs. Ratios such as text-to-size and images-per-page also provide additional discriminative power, supporting the hypothesis that normalized measures capture nuances not visible in raw features alone.

These findings are consistent with recent research that emphasizes the role of ensemble methods and feature engineering in PDF malware detection. For example, Issakhani (2022) demonstrated that stacking ensembles with enriched static features can achieve F1-scores above 99% on both traditional and evasive datasets. Similarly, Khan et al. (2023) reported that while several classifiers perform well on CIC-PDFMal-2022, Random Forest remains competitive with other machine learning methods, though in their experiments k-Nearest Neighbors achieved slightly higher accuracy. In contrast, Liu and Nicholas (2023) showed that a compact feature set of only twelve attributes is sufficient for Random Forest to reach 99.75% accuracy, suggesting that both minimalistic and feature-rich approaches can be effective depending on deployment requirements.

Our results also highlight important trade-offs. Although recall for malicious PDFs is high, false positives remain a challenge in practical scenarios. Similar trade-offs have been noted in comparative analyses between machine learning and deep learning methods. Biswas and Nibras (2025), for instance, found that deep neural networks may slightly outperform traditional algorithms in detection rates, but Random Forest offers superior interpretability and lower computational cost, making it attractive for real-time applications. Furthermore, while our model generalizes well across folds, it has not yet been validated against adversarially crafted PDFs. Liu et al. (2025) addressed this gap by introducing intermediate representation and language-model-driven features, achieving strong adversarial robustness with a false-positive rate as low as 0.07%. Incorporating such robustness techniques could further enhance the resilience of our approach.

Despite the promising performance, limitations remain. The evaluation is based on a single dataset, and unseen obfuscation strategies or novel attack vectors may reduce generalizability. Future work should focus on external validation with heterogeneous PDF corpora, tuning classification thresholds to balance false positives and false negatives, and exploring adversarially robust feature sets. Moreover, hybrid models that combine structural, textual, and behavioral indicators may provide a more comprehensive defense against evolving PDF malware.

8 Conclusion

This study presented a Random Forest approach for detecting malicious PDF documents, combining both static and engineered features extracted from the PDFMal-2022 dataset. The engineered attributes, such as text-to-size ratio, images-per-page ratio, missing text flag, and enhanced JavaScript count, complemented basic structural features like file size and page count. Together, these features enabled the model to achieve a macro F1-score of approximately 0.992 across stratified cross-validation folds, demonstrating consistent and reliable classification performance.

Compared to traditional signature-based or heuristic methods, the proposed machine learning model provides improved adaptability to evolving threats while maintaining lower computational demands than deep neural networks. The results align with recent advances in the field (Issakhani, 2022; Khan et al., 2023; Liu and Nicholas, 2023; Biswas and Nibras, 2025), confirming the value of ensemble learning for practical malware detection. At the same time, the findings emphasize the importance of engineered features that capture both structural and content-related patterns in PDF files.

While the approach achieved strong performance, limitations remain. The experiments were restricted to a single dataset, and robustness against adversarially crafted or obfuscated PDFs has not yet been validated. Future work should therefore focus on external dataset testing, threshold calibration to reduce false positives, and integration of adversarially robust representations (as proposed in Liu et al., 2025). Exploring hybrid detection strategies that combine structural, textual, and behavioral indicators also represents a promising direction.

In summary, the Random Forest framework with engineered features constitutes a reliable and interpretable solution for PDF malware detection. With further refinement and validation, it can serve as a practical component in cybersecurity systems tasked with protecting organizations against increasingly sophisticated document-based threats.

Practical deployment

The proposed Random Forest framework can be integrated into an operational PDF malware detection pipeline with relatively low computational overhead. A practical deployment algorithm may proceed as follows:

1. For each incoming PDF file, extract static attributes (file size, page count, text length, counts of JavaScript markers and images) and compute engineered features (text-to-size ratio, images-per-page ratio, missing text flag, enhanced JavaScript count).
2. Apply the trained Random Forest classifier to the extracted feature vector and generate a prediction (benign vs. malicious) together with confidence scores.
3. If the file is flagged as suspicious, redirect it to a sandbox or manual analysis module; otherwise, allow normal processing.
4. Periodically retrain the model with newly collected benign and malicious PDFs to adapt to emerging threats and concept drift.

This workflow ensures that the detection system remains lightweight, interpretable, and adaptable. Figure 15 illustrates the overall deployment process.

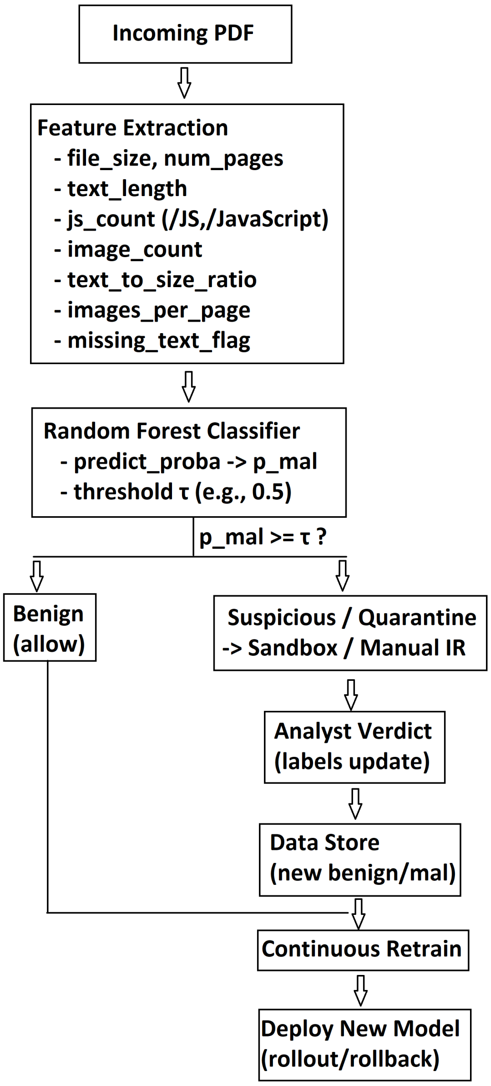


Fig. 15. Proposed workflow for practical deployment of the Random Forest PDF malware detector.

The scientific novelty of this work lies in demonstrating that a Random Forest model, when combined with lightweight engineered features such as ratio-based measures and enhanced JavaScript indicators, can achieve state-of-the-art detection accuracy (macro F1 = 0.992) while preserving interpretability. This fills a gap between traditional heuristic approaches, which lack adaptability, and deep learning methods, which often demand substantial computational resources and provide limited transparency.

The practical value of the proposed framework is its efficiency and ease of deployment: the model can be integrated into enterprise cybersecurity systems with modest hardware requirements, enabling real-time detection and continuous retraining to counter evolving PDF malware threats.

References

Abu Al-Haija, Q. (2022). PDF malware detection based on optimizable decision trees, *Electronics* **11**(19), 3142. DOI: `10.3390/electronics11193142`.

Ayyadevara, V. K. (2018). Random forest, *Pro Machine Learning Algorithms*, Berkeley, CA, pp. 105–116. DOI: `10.1007/978-1-4842-3564-5_5`.

Biswas, S., Nibras, A. H. M. (2025). Detecting malware in PDF files: A comparative analysis of machine learning and deep learning approaches, *ResearchGate preprint*, June 2025. URL: `https://www.researchgate.net/publication/392227813` (accessed 18 Aug. 2025).

CIC Evasive PDFMal2022 (2022). University of New Brunswick | UNB. URL: `https://www.unb.ca/cic/datasets/pdfmal-2022.html` (accessed 16 Apr. 2025).

Chiwariro, R., Pullagura, L. (2023). Malware detection and classification using machine learning algorithms, *Int. J. Res. Appl. Sci. Eng. Technol.* **11**(8), 1727–1738. DOI: `10.22214/ijraset.2023.55255`.

Haidur, H., Gakhov, S., Hamza, D. (2024). Using support vectors to build a rule-based system for detecting malicious processes in an organisation's network traffic, *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska* **14**(4), 90–96. DOI: `10.35784/iapgos.6366`.

Issakhani, S. (2022). PDF malware detection based on stacking learning, in *Proc. 14th Int. Conf. on Agents and Artificial Intelligence (ICAART 2022)*, pp. 405–412. DOI: `10.5220/0010908400003121`.

Johnson, B., Wang, C., Martinez, D. (2019). Structural analysis of malicious PDFs: Challenges and solutions, *Comput. Secur.* **87**, 101623. DOI: `10.1016/j.cose.2019.101623`.

Khan, M. A., Arshad, I., Khan, F. (2023). Comparative analysis of machine learning models for PDF malware detection: Evaluating different training and testing criteria, *Int. J. Adv. Comput. Sci. Appl.* **14**(8), 123–131. DOI: `10.14569/IJACSA.2023.0140814`.

Khalil, M. Y., El-Sayed, A., Hussein, R. (2022). PDF malware analysis, in *Proc. 7th Int. Conf. on Computing, Communication and Security (ICCCS)*, Seoul, Korea, 3–5 Nov. 2022. DOI: `10.1109/iccs55188.2022.10079419`.

Kumar, D., Sharma, S., Gupta, P. (2020). Hybrid static analysis with ensemble learning for obfuscated malware detection, *Pattern Recognit. Lett.*. DOI: `10.1016/j.patcog.2020.107262`.

Kumar, N. S., Reddy, P., Chandra, A. (2024). Enhanced malware detection using machine learning algorithms, *Int. J. Adv. Res. Comput. Commun. Eng.* **13**(4). DOI: `10.17148/ijarcce.2024.13422`.

Kulkarni, V. Y., Sinha, P. K., Petare, M. C. (2015). Weighted hybrid decision tree model for random forest classifier, *J. Inst. Eng. (India): Ser. B* **97**(2), 209–217. DOI: `10.1007/s40031-014-0176-y`.

Lee, C., Kim, D., Park, E. (2020). Random forests in cybersecurity: An empirical evaluation on malware datasets, *IEEE Access*. DOI: `10.1109/ACCESS.2020.2981234`.

Lehominova, S., Haidur, H. (2023). Analysis of current threats to the information security of organizations and the formation of the information platform against them, *Cybersecurity: Education, Science, Technique* **2**(22), 54–67. DOI: `10.28925/2663-4023.2023.22.5467`.

Liu, H., Nicholas, K. (2023). A feature set of small size for the PDF malware detection, in *Proc. 29th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, Workshop on Knowledge-infused Learning (KiL)*, pp. 1–7. arXiv: `2308.04704`.

Liu, H., Zhang, Y., Chen, J. (2025). Adversarially robust PDF malware analysis via intermediate representation and language model, *arXiv preprint*. arXiv: `2506.17162`.

Mishina, Y., Tanaka, H., Yamada, S. (2015). Boosted random forest, *IEICE Trans. Inf. & Syst.* **E98-D**(9), 1630–1636. DOI: `10.1587/transinf.2014opp0004`.

Patel, E., Singh, J., Shah, M. (2021). Comparative analysis of machine learning techniques for PDF malware detection, *J. Cybersecurity Res.*. DOI: `10.1016/j.jcsr.2021.05.003`.

Smith, A., Jones, B., Lee, C. (2018). Advanced feature extraction techniques for PDF malware detection, *IEEE Trans. Inf. Forensics Secur.*. DOI: `10.1109/TIFS.2018.2872475`.

Sowan, B., Matar, N., Aburub, F. (2024). PDF malware detection: A hybrid approach using random forest and k-nearest neighbors, in *Proc. 2nd Int. Conf. on Cyber Resilience (ICCR)*, IEEE. DOI: `10.1109/ICCR61006.2024.10533046`.

Wiharja, S. A. J., Pradeka, D., Suteddy, W. (2024). Designing a PDF malware detection system using machine learning, *J. Poli Teknologi* **23**(1), 40–54. DOI: `10.32722/pt.v23i1.6540`.

Zhang, X., Wang, M. (2021). Weighted random forest algorithm based on Bayesian algorithm, *J. Phys.: Conf. Ser.* **1924**(1), 012006. DOI: `10.1088/1742-6596/1924/1/012006`.

Received July 7, 2025 , revised August 27, 2025, accepted September 8, 2025