XAI-PDF: A Robust Framework for Malicious PDF Detection Leveraging SHAP-Based Feature Engineering

Mustafa Al-Fayoumi
Department of Cybersecurity,
Princess Sumaya University for Technology, Jordan
m.alfayoumi@psut.edu.jo

Qasem Abu Al-Haija
Department of Cybersecurity,
Jordan University of Science for Technology, Jordan
qsabuhaija@just.edu.jo

Rakan Armoush
Department of Data Science,
Princess Sumaya University for Technology, Jordan
rak20190009@std.psut.edu.jo

Christine Amareen
Department of Data Science,
Princess Sumaya University for Technology, Jordan
CHR2020013@std.psut.edu.jo

Abstract: *With the increasing number of malicious PDF files used for cyberattacks, it is essential to develop efficient and accurate classifiers to detect and prevent these threats. Machine Learning (ML) models have successfully detected malicious PDF files. This paper presents XAI-PDF, an efficient system for malicious PDF detection designed to enhance accuracy and minimize decision-making time on a modern dataset, the Evasive-PDFMal2022 dataset. The proposed method optimizes malicious PDF classifier performance by employing feature engineering guided by Shapley Additive Explanations (SHAP). Particularly, the model development approach comprises four phases: data preparation, model building, explainability of the models, and derived features. Utilizing the interpretability of SHAP values, crucial features are identified, and new ones are generated, resulting in an improved classification model that showcases the effectiveness of interpretable AI techniques in enhancing model performance. Various interpretable ML models were implemented, with the Lightweight Gradient Boosting Machine (LGBM) outperforming other classifiers. The Explainable Artificial Intelligence (XAI) global surrogate model generated explanations for LGBM predictions. Experimental comparisons of XAI-PDF with baseline methods revealed its superiority in achieving higher accuracy, precision, and F1-scores with minimal False Positive (FP) and False Negative (FN) rates (99.9%, 100%, 99.89%,0.000, and 0.002, respectively). Additionally, XAI-PDF requires only 1.36 milliseconds per record for predictions, demonstrating increased resilience in detecting evasive malicious PDF files compared to state-of-the-art methods.*

Keywords: *Machine learning, malicious PDF detection, feature engineering, explainable artificial intelligence, shapley additive explanations.*

Received August 3, 2023; accepted December 26, 2023
https://doi.org/10.34028/iajit/21/1/12

1. Introduction

The widespread use of internet-based software and systems has revolutionized personal and professional lives, enhancing convenience and productivity. However, reliance on digital systems also opens the door to cybercriminal exploitation [17, 9, 28]. The demand for robust security measures against such threats has never been greater [1, 6].

PDF files, a common means for sharing and viewing documents, are frequently manipulated by cybercriminals for malware delivery due to their universal usage and inherent feature set [15, 35]. They can support JavaScript and contain links, embedded files, and various media types, making them a convenient medium for hiding and delivering malicious code [49]. Malware concealed within PDF documents can wreak havoc by exploiting vulnerabilities in PDF viewer applications to execute its code, potentially leading to unauthorized access, data theft, and system disruption or even laying the groundwork for broader network breaches [14, 43]. The complexity and sophistication of these attacks are continually evolving, with modern malware capable of intricate evasion techniques and adaptive behavior, including exploiting zero-day vulnerabilities [8, 34, 45].

The growing dependence on digital documents and the increasing sophistication of cyber-attacks underscores the necessity for efficient and effective malware detection mechanisms. These systems must keep pace with the evolving strategies of threat actors, integrating emerging technologies like ML and AI to identify and neutralize threats [2, 22]. The recent transition to remote work, precipitated by the COVID-19 pandemic, has expanded the potential attack surface for cybercriminals, amplifying the relevance of malicious PDF detection [5, 10, 16].

Consequently, research focused on detecting and preventing malicious PDF files is crucial and timely, considering current trends in cybercrime and digital transformation. In recent years, malicious PDF detection has become a significant research area, with various Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) techniques proposed for such threat detection. A fundamental aspect of these methods involves extracting meaningful features from PDF documents to build classifiers [15, 30, 35, 49].

However, the results of these AI, ML, and DL methods still need to be understood. Models are typically "black boxes," making it hard to understand how they arrive at their conclusions. Explainable Artificial Intelligence (XAI) approaches, such as SHAP and Local Interpretable Model-Agnostic Explanations (LIME), have attracted attention in recent studies as a possible solution to this problem. These methods were developed to make AI more transparent by explaining the reasoning behind model outcomes [31, 38, 48].

The existing literature on AI/ML/DL models in malware detection shows high efficiency in performance. However, these models' "black box" nature often makes their outcomes challenging to interpret, causing doubts about their reliability and making debugging more complicated [2, 5, 8, 10, 16, 22, 24, 30, 31, 34, 38, 45, 48]. To address this issue, the integration of XAI techniques into these systems has gained attention, intending to enhance interpretability [40, 42]. While the integration of XAI into malware detection models has shown promise in enhancing the interpretability of AI/ML models, there appears to be a gap in the current research landscape, specifically in detecting and interpreting malicious PDF files. Several studies have examined the application of XAI in general malware detection. However, these have yet to be deeply focused on PDF malware detection [11, 26, 42, 47], which presents unique challenges and characteristics.

Furthermore, the effective utilization of Shapley Additive Explanations (SHAP) for improving model interpretability in PDF malware detection appears to be an under-researched area [32]. Even though some works have applied SHAP in more general malware detection contexts, exploring its particular application in understanding the relevance of features in a PDF-based malware detection model remains limited [4].

Moreover, another noticeable gap is Feature Engineering (Feng), specifically concerning SHAP in PDF malware detection. Though efficient FEng can notably enhance the performance of ML models [39], the existing literature offers limited studies on how to engineer and select the most pertinent features for a SHAP-based PDF malware detection model.

Finally, the comprehensive understanding of the correlation between feature importance, as highlighted by SHAP, and the decision-making process of an ML model for PDF malware detection needs to be adequately covered in the current literature [40]. While FEng, in combination with XAI, has been examined in the domain of Android malware detection [32, 47], comprehensive research into how the same concept could be employed to enhance the performance of PDF malware detection models needs to be improved.

In conclusion, there is a significant gap in the literature related to the efficient use of XAI and SHAP for FEng and model interpretation in the specific context of PDF malware detection. The current study is therefore significant in addressing these gaps by proposing an effective model for malicious PDF detection with FEng based on XAI techniques (SHAP), which could enhance the robustness and explainability of detection models.

In conclusion, AI, ML, and DL have greatly improved in identifying fraudulent PDF files. XAI methods, such as SHAP and LIME, are becoming essential in this field of study because of the growing need for openness and interpretability. These methods help make the created models more reliable and stable. Combining AI/ML/DL malware detection models with XAI methods can help develop more trustworthy and efficient PDF malware detection systems. Finally, the major impacts of our research can be summarized as follows:

- We present an efficient-applicable ML-based framework for detecting malicious PDFs aiming to improve PDF file security by efficiently identifying and isolating potential threats. This solution addresses the challenge of identifying such files thoroughly; the enhancement of detection performance and the provision of significant insights into the decision-making process are accomplished by it.
- We implement improved feature selection and derivation techniques to enhance model accuracy, reduce computational complexity, improve the model recognition for complicated patterns and relationships, and improve its input and performance.
- We integrate XAI approaches to make decision-making transparent and interpretable, fostering trust in its predictions and enabling successful mitigation strategies. The amalgamation of XAI methodologies with feature engineering provides a fresh perspective for future cybersecurity research in the field of cybersecurity.
- We provide extensive experimental and benchmarking results. Our system models were evaluated on a contemporary broad dataset for PDF files, the Evasive-PDFMal2022 dataset. We assessed the models' performance using conventional performance metrics (accuracy, precision, recall, prediction time, and others), which shows the superiority of our model compared to state-of-the-art models.

The finding of this research underscore the originality of the research within cybersecurity, specifically in malicious PDF detection. The remainder of this paper is organized as follows: Section 2 reviewed the most sophisticated state-of-the-art models developed to improve malware detection for PDF files. Section 3, the proposed XAI-PDF Framework, presents the detailed development stages for the proposed XAI-PDF, including the overall XAI-PDF system architecture, the dataset collection, description and engineering, feature construction and selection, learning models and evaluation metrics, and the prediction activities. Section 4 discusses the experimental setup and empirical results. Section 5 discusses the obtained results and benchmark XAI-PDF findings with baseline state-of-the-art models. Finally, Section 6 concludes the paper.

2. Literature Review

This paper examines malicious PDF file detection research and highlights its progress. The literature review shows several methods to this challenge, including traditional ML algorithms and XAI approaches. Due to their growing importance, many researchers use AI, ML, and DL to identify malicious PDF files. These strategies build classifiers by extracting and engineering PDF features. The literature review has been divided into three categories:

1) Conventional malware detection.
2) Malicious PDF detection.
3) XAI for malware detection.

2.1. Conventional Malware Detection

This section discusses malware detection studies and methods. The literature agrees that ML and DL algorithms detect malware. Therefore, we explore several studies on generic malware detection and highlight their unique approaches to addressing this important cybersecurity challenge. For example, Smutz and Stavrou [45] studied applying ensemble classifiers to enhance malware detection. The authors exhibited the efficacy of incorporating diversity in ensemble classifiers to augment the robustness of malware detectors against evasion techniques. The study underscored the advantages of employing a varied range of classifiers, thereby reinforcing the proposition that model diversity can enhance the system's resilience.

Malw D and C by Buriro *et al*. [12] uses a Random Forest (RF) classifier to detect and categorize malware on Windows-based systems. 2,381 features are extracted from each binary file in a publically available BODMAS dataset of 57,293 malware and 77,142 benign samples from 581 families. Malware detection was 99.56%, and categorization was 97.69%. The research's strengths are accuracy, speed, and feature detection. Limitations include dependency on a single dataset, classifier performance unpredictability, and computing difficulties with many features.

Kumar and Das [25] used k-neighbor, Extreme Gradient Boosting (XGB), RF, and LGBM to classify malicious and benign files. The authors used Cuckoo Sandbox static and dynamic analysis on 10,540 samples. Method 1 showed that XGBoost was best for benign datasets (98.1105%), and LGBM was best for malicious datasets (98.2314%). Method 2 produced the best accuracy ratings for benign (98.4325%) and malicious (98.5312%) datasets. This study uses a novel two-level classifier to demonstrate ML's malware detection accuracy. However, it acknowledges potential misclassifications and single-source malware samples that need verification, advocating the development of a behavior-based antiviral platform using ensemble algorithms.

Gorment *et al*. [18], in a thorough literature review, examined ML's function in malware detection. They developed a malware detection taxonomy based on classification approaches, analysis types, and problems from 77 research papers, strengthening the notion of ML's cybersecurity potential. They used Support Vector Machine (SVM), Decision Trees (DT), and N-grams algorithms on a large dataset to show that dataset size, classification method, and analysis type can greatly affect detection accuracy. Despite its significant contributions, the research has drawbacks, including a need for more datasets and feature details. However, it emphasizes the necessity for larger datasets, classification methods, and analysis kinds to comprehend ML's significance in malware detection.

Lu *et al*. [33] suggest merging capsule networks and feature selection to improve malware detection. Their mobile malware study addresses redundant and unneeded detection features. A Correlation Information Decision Matrix (CIDM) feature selection method reduces dimensionality and improves detection model efficiency and accuracy. As a detection model, the capsule network preserves local information. The authors compare their technique to others on a real-world network traffic dataset. Accuracy and recall increased by 9.71% and 20.18%, respectively. The work advances malware detection and mobile security research.

2.2. Malicious PDF Detection

The increasing significance of identifying malicious PDF files has prompted numerous researchers to utilize AI, ML, and DL techniques to counteract these security risks. The core of these methodologies revolves around identifying and retrieving salient attributes from PDF files, which are subsequently employed in creating classifiers. Consequently, the detection of malware embedded in PDF files has been extensively investigated and discussed in academic research. For instance, Smutz and Stavrou [44] use metadata and structural factors to identify malicious PDF files, using the random forests ML algorithm to detect and categorize embedded hazardous code. This study shows how FEng improves malware detection. The research model used 202 features to classify over 5,000 malicious and 100,000 benign PDFs accurately. The model classified ‘opportunistic’ and ‘targeted’ malware with a False Positive (FP) rate of 0.2% or less and a classification rate over 99%. Due to its focus on PDFs, the study's resilience against evasion and mimicry attacks and effectiveness in identifying new malware variants show a significant advancement in ML's use in cybersecurity.

Maiorca *et al*. [35] used a RFs classifier to detect malicious PDF files. They trained their model with 21,146 benign and malicious PDFs and 243 features. PDF Malware Slayer (PDFMS) outperforms other classifiers and competes with commercial antivirus systems, although it has trouble discovering vulnerabilities and may be vulnerable to advanced attackers. The study emphasizes the significance and potential of ML in cybersecurity while also drawing attention to the necessity for additional enhancements.

Corona *et al*. [13] developed Lux0R, a system that employs discriminant analysis of Application Programming Interface (API) references to detect malicious JavaScript code embedded in PDF files. Lux0R uses one-class SVM to detect API fraud. Thousands of PDF malware samples were trained and tested using 500 API references. Lux0R's 98.8% True Positive (TP) and 0.4% FP rates are remarkable. The study assessed the tool's API selection criteria and simulated assault resistance. Their research increases ML for cyber threat detection, despite API extraction mistakes and vulnerability to more complex gradient descent attacks.

Li *et al*. [28] introduced the robust feature extractor FEPDF to detect fraudulent PDFs. The paper discusses a vulnerability attack that evades detection and inserts a malicious template. The authors suggest Feature Extractor for M alicious PDF Detection (FEPDF) as a remedy to current feature extractor constraints. FEPDF searches suspicious segments and extracts features other extractors miss. Antivirus engines and feature extractors showed that FEPDF performed better. Compared to JsUnpack, FEPDF had 97.57% precision, 90.87% recall, and 95.11% accuracy. This research can extract previously overlooked features, detect dangerous JavaScript code, and enhance accuracy over existing feature extractors.

Zhang's paper [49] detects PDF-based malware using ML. MLPdf uses a multilayer perceptron neural network model trained on 105,000 benign and malicious PDF documents. MLPdf performs well with 48 high-quality dataset features, achieving a 95.12% TP rate and 0.08% FP rate. The Multilayer Perceptron (MLP) neural network model detects PDF malware better than eight commercial antivirus scanners. Sayed and Shawkey [41] offered feature selection and classification data mining to detect fraudulent PDF files. An Improved Binary Gravitational Search Algorithm (IBGSA) selected features and RF and DT classifiers classified. On a massive dataset of 22,000 malicious and benign PDF files, the system had 99.77% detection, 99.84% accuracy, and 0.05% FPs. Data mining can detect APTs and mimicked PDF files, and the approach secures PDF files.

Cuan *et al*. [15] suggested studying SVM evasion attacks in PDF virus detection. The gradient-descent attack misled a basic SVM model, and the authors suggested vector component threshold, feature selection, and adversarial learning to make it more resilient. The SVM was trained and evaluated on 10,000 clean and 10,000 malicious PDF files. The analysis was conducted using 21 default features with the PDFID tool developed by Didier Stevens to examine the malicious nature of the PDF files. Combining the threshold and feature selection countermeasures gave the highest performance, with 99.22% accuracy and 99.99% theoretical resistance to gradient-descent attacks.

Jeong *et al*. [22] suggested a convolutional neural network-based method for detecting malicious behaviors in non-executable byte sequences, specifically PDF files. Their convolutional neural network outperforms standard ML algorithms. The proposed network had F1 ratings of 97.68% for benign and 98.61% for malicious classes, and the authors provided insights into the best network architecture for this task. The paper proposes a promising method for detecting malicious actions in non-executable byte sequences using DL techniques. However, it only uses PDF files and relies on Graphical Processing Unit (GPU) performance for training time.

Maiorca *et al*. [34] explore ML-based PDF file detection methods and adversarial assaults. The authors describe PDF file-detecting features and classification ML algorithms. The paper also discusses training and testing datasets and PDF malware detection problems. The paper emphasizes PDF malware detection's present state-of-the-art and potential future research. Also, Li *et al*. [27] enhance the robustness of the K-Nearest Neighbors (KNN) algorithm against adversarial assaults in classifying malicious PDF files. A gradient descent approach generates adversarial samples for a new KNN classifier's training set. Evaluation measures show that this strategy enhances KNN's robustness without losing accuracy. Despite its success, the research's limitations, including a single attack focus and a small dataset with limited features, suggest exploring different approaches and larger datasets to improve classifier robustness.

Kang *et al*. [23] used structural, meta, and content factors to detect malicious PDF documents using ML. Three machine-learning techniques were tested on 3,930 PDF files using 10-fold cross-validation. The RF algorithm with structure and content features had the maximum accuracy of 99.2% in detecting malicious PDF documents and was robust to adversarial attacks.

Image processing and ML are used to detect PDF malware by Corum *et al*. [14]. The authors use ML methods to convert PDF files to grayscale photos, extract image attributes, and identify them as benign or dangerous. The proposed technique outperformed multiple prominent antivirus scanners in the Contagio dataset. The authors found their system more resistant to reverse imitation attacks than the current learning-based strategy. This research reveals how image processing and ML can detect viruses in PDF files.

He *et al*. [19] suggested a two-stage classification technique utilizing a Convolutional Neural Network (CNN) to extract content and structural data from PDF files to detect fraudulent PDFs. The proposed technique had over 98% accuracy, precision, recall, and F1 score. Performance dropped less than 1%, demonstrating robustness. The proposed method may distinguish vulnerabilities used in malicious files and detect other file formats. The work emphasizes feature selection for malware detection and shows that a two-stage classification model with content and structural features can detect harmful PDFs.

Li *et al*. [30] used the Feature-Vector Generative Adversarial Network (fvGAN) model to generate adversarial feature vectors using the Mimicus framework to create malicious PDF files. The proposed method beat conventional PDFrate classifier evasion attacks in evasion rate and execution cost. GANs can produce adversarial feature space samples for evasion and learning. The proposed method improves evasion attempts against ML-based classifiers but may fail in the issue space due to created adversarial samples. In all four attack cases, the suggested approach outperformed earlier evasion attacks in evasion rate and execution cost, reaching 99% in the best case.

Mohammed *et al*. [37] developed a holistic ML and DL strategy for PDF virus detection. They used binary and keyword analysis to create a model that accurately detected PDF viruses. KNN and RF classifiers collected image, audio, and hash features for signal-based malware analysis and a bag-of-words model for PDF structural analysis. Dynamic analysis results were used to create a PDF malware bag-of-words model. Their PDF virus detection method had a 99.92% accuracy rate.

Falah *et al*. [16] use PDFiD, PeePDF, and derived features to detect malicious PDFs. A wrapper function with three ML algorithms and a feed-forward deep neural network determines feature importance. The study found that content-, evasion-, and malice-related variables help identify malicious papers, and adding them to a classifier improves its performance. The approach achieves 98.6% accuracy, 99% precision, 98.3% recall, and 98.6% F1 score on the test dataset. Two feature extraction algorithms focusing on structural features rather than content analysis limit the study. This study can improve detection tools and PDF-based assault countermeasures.

Tay *et al*. [46] tested three adversarial attacks, Mimicry, Mimicry+, and Reverse Mimicry, against two state-of-the-art PDF malware classifiers, Mimic and Hidost. Traditional feature extraction and classification approaches are vulnerable to adversarial attacks, underscoring the need to combine them with ML methods for malware detection and analysis. The classification was done by RF. The study gives valuable insights, but it only uses one classification method.

Abu Al-Haija *et al*. [2] recommend optimizable DT with AdaptiveBoost (AdaBoost) and proper hyperparameters to identify malicious PDF files from benign ones. Evasive-PDFMal2022, a new dataset containing 10,025 PDF records and 37 essential static features extracted from each file, trains and evaluates the model. After rigorous testing, the authors' model's accuracy, sensitivity, and precision exceed 98.80%. The study suggests that the concept can be expanded to provide many detection services in other domains.

Li *et al*. [29] use DL, mutual agreement analysis, and active learning to detect PDF viruses. Contagio provides structural route characteristics. SVM works for small sample classification, and uncertain samples increase model performance. Active learning and mutual agreement analysis add test set samples to the training set. The approach has 96.5% accuracy, 97.3% precision, 95.7% recall, and 96.5% F1. DL-based classifiers, mutual agreement analysis, and active learning improve PDF malware detection.

Issakhani *et al*. [21] suggested a stacking-based learning model detect malicious PDF files and tested it on the Contagio dataset and the newly built evasive PDF malware dataset. The proposed model included three ML algorithms as base learners, and the authors suggested testing their approach on other malware and using DL for feature extraction and model training. The paper proves that the stacking-based learning model can detect fraudulent PDF files.

Adhatarao and Lauradoux [3] present a novel method to detect PDF file origins using coding style traits as a forensic tool. The study trained and tested models using ML methods and 2000 PDF files. A RF model showed 95% accuracy in determining PDF file origins based on coding style factors. The study's benefits include detecting counterfeit papers and identifying PDF file authors. Still, its drawbacks include FPs and the requirement for a big dataset to train the ML model.

Using the Fuzzy Unordered Rule Induction Algorithm (FURIA), Mejjaouli and Guizani [36] offer a new PDF virus detection method. The study preprocesses PDF files using feature extraction to extract object count, typeface usage, and color usage. The FURIA algorithm classifies PDFs as malicious or benign. FURIA outperformed other ML algorithms with 97.6% accuracy. The study's benefits include a novel PDF virus detection method and fuzzy logic to handle PDF file feature ambiguity. Study limitations include FPs and requiring a larger dataset to train the ML model.

Kattamuri *et al.* [24] suggest utilizing swarm optimization and ML to detect malware in Portable Executable (PE) files. The study preprocesses PE files and extracts features like section headers, imports, and exports to train ML models like RF, Naïve Bayes, and SVMs using Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). The RF model with ACO optimization surpassed other ML methods with 99.85% accuracy. Swarm optimization can improve ML models' PE file virus detection.

2.3. XAI Malware Detection

Integrating XAI methodologies with malware detection, particularly within the context of PDF files, represents a developing area of research. Several research studies have recently commenced investigating the integration of XAI with malware detection systems. These studies aim to enhance the interpretability of AI/ML models, thereby increasing their reliability.

Rahman *et al.* [40] concluded the literature on PDF malware detection by analyzing machine and DL models utilizing XAI and SHAP Framework. Malware detection is becoming more transparent and interpretable via the SHAP framework, a game-theoretic method for explaining ML model output. This research aims to interpret ML and DL models. They show the necessity for explainability, notably in malware detection, where comprehending a model's logic can increase its dependability, fairness, and trustworthiness.

Scalas [42] in his dissertation, also developed an explainable ML system for malware analysis and detection, specifically focusing on malicious PDF files. He highlighted the challenges in understanding the reasoning behind ML decisions and how XAI can help elucidate this process. In this study, the model provided detailed explanations of its decision-making process, enhancing the interpretability of malware detection.

Liu *et al.* [32] examined the explainability of ML models in malware detection using SVM, attention-based neural networks, and MLP. They observed that simple ML models could reach 99% accuracy using a large dataset from the Androzoo repository and time-specific information from Google developer documentation. However, temporal discrepancies in the training data highly correlated with accuracy, suggesting that the models were learning temporal differences rather than differentiating malware from benign entities. ML/DL-based systems may be overoptimistic due to their low discernment and explainability. Bose [11] discussed explainability in ML for malware detection, emphasizing XAI's growing importance in interpreting complex ML models' behavior.

Ullah *et al.* [47] used Bidirectional Encoder Representations from Transformers based (BERT) transfer learning and malware feature visualization to detect Android malware. A pre-trained BERT model extracted trained features from huge textual material and used Packet CAPture (PCAP) file byte streams to depict malware graphically. They used Synthetic Minority Over-Sampling (SMOTE) to address class imbalance caused by textual and texture features. Using an ensemble model, they detected and classified malware at 99.16% using the AAGM2017 and MalDroid 2020 datasets. The research shows the potential of DL, transfer learning, and ensemble models in Android malware detection, notwithstanding class imbalance. Kumar and Subbiah [26] created an extratree ML malware detection model. Their model employed 27,239 malware samples from 60,059 February 2017 samples. Static analysis of PE headers and file characteristics yielded 2,351 features for model training. They used SHAP values to build inductive rules to reduce misclassifications. After applying inductive principles, the test dataset had a 98.09% accuracy rate and no FPs or negatives. The study implies better precision, recall, and F1-score. Kumar and Subbiah's [26] model detects malware by improving misclassification management and accuracy.

Drebin-215 dataset, Alani and Awad [4] created PAIRED, a lightweight, explainable, and accurate ML-based Android malware detection system. RF finds apps utilizing 35 static features from 215 features. PAIRED outperforms various state-of-the-art methods with 0.9807 accuracies, 0.9806 F1 score, and 0.7631μs testing duration, and after decreasing Drebin-215, Malgenome-215, and CICMalDroid2020 increased system performance and generalization. Due to its small size, continuous updates may require a robust cloud-based ML model update function and memory/processing reduction. Ogiriki [39] examines machine-learning strategies for malware detection and model explainability. The researchers use RF, DT, Naive Bayes, Logistic Regression (LR), and SVM classifiers using a custom dataset from Virus Total and a testbed emulating a typical operating system. Black box evaluations are followed by moDel Agnostic Language for Exploration and eXplanation (DALEX) explainability analysis. The RF and DT are the most accurate, with 88.47% and 86% accuracy, respectively. The study reveals numerous key factors affecting predictions. The dataset's specificity and model explanation methodologies' limited breadth limit the research's malware detection and model interpretability insights.

3. Proposed XAI-PDF Framework

This study introduces XAI-PDF as a highly efficient and effective system for malicious PDF detection, aiming to boost the detection system's accuracy while minimizing decision-making time. This section presents the methodology to build the SHAP values-based malicious PDF detection method. Figure 1 shows an overview of the proposed method. Mainly, the proposed method involves four phases:

1) The data preparation phase.
2) The models building phase.
3) The derived features phase.
4) The evaluation model phase.

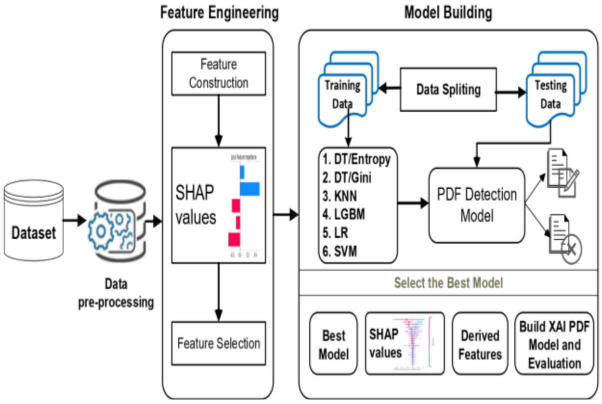


Figure 1. Architecture of XAI-PDF.

During the data preparation phase, the dataset is cleaned by removing unwanted observations, handling missing data, fixing structural errors, and addressing outliers. The main contribution of this study pertains to the application of SHAP values for automating the feature selection process, specifically emphasizing the inclusion of features that significantly influence the decision-making mechanism of the system. During the model-building phase, six distinct classifiers are selected to facilitate the training and testing of the XAI-PDF system, which has been specifically developed for malware detection. The classifiers encompassed in this study are Decision Tree/Entropy (DT/Entropy), Decision Tree/Gini (DT/Gini), KNN, Lightweight Gradient Boosting Machine (LGBM), LR, and SVM.

This phase aims to ascertain the classifier with the highest efficiency level among the six classifiers. A range of methodologies is utilized within the domain of Model Interpretability to enhance the understanding of the prediction mechanism employed by an ML model. The SHAP algorithm is a computational method that calculates SHAP values for each feature included in the model, thereby revealing their influence on predictions. The SHAP algorithm is utilized in this paper as a model-agnostic method for elucidating ML models. Calculating SHAP values entails quantifying the influence of individual model features by comparing the model's performance with and without each feature. This analysis assists in perceiving the extent to which each feature, positively or negatively, contributes to the prediction. During the derived features phase, the classifier with the highest performance is chosen by employing SHAP values. This process enables the discovery of novel and noteworthy attributes for the classifier that exhibits superior performance. Applying XAI techniques, such as SHAP, has improved the interpretability of AI models and optimized their performance.

3.1. Dataset Description

The Canadian Institute for Cybersecurity at the University of New Brunswick (UNB) provided the dataset for this study. The dataset can be accessible at via [20]. The Evasive-PDFMal2022 dataset is utilized to build and evaluate the performance of the proposed XAI PDF malware detection system. The dataset comprises two classes of PDF samples: malicious and benign. Of the 19972 samples, 11040 are malicious, while 8932 are benign. The distinguishing feature of this dataset is the presence of evasive characteristics that make it challenging for conventional machine-learning algorithms to differentiate between malicious and benign samples accurately. The dataset comprises 32 distinct static representative features, comprising 10 general and 22 structural features. These features have been extracted from each PDF file, as illustrated in Figure 2.

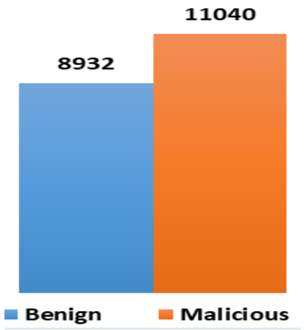


Figure 2. Dataset classes count.

3.2. Data Engineering

The phase of Data Engineering is of utmost importance in developing a robust detection system for malicious PDFs. Its primary focus is acquiring and processing the Evasive-PDFMal2022 dataset, comprising 19972 instances of benign and malicious PDF files. During the initial phase, the dataset is carefully selected from a diverse array of dependable sources and subsequently subjected to preprocessing procedures to remove redundant entries, anomalies, and damaged files, guaranteeing the integrity and excellence of the data for subsequent analytical procedures. Subsequently, the PDF files extract static and dynamic features, encompassing attributes such as file size, encryption status, and metadata, in addition to the execution of JavaScript code, embedded objects, and API calls.

After systematically organizing these features, they are subjected to scaling and normalization techniques to establish a consistent scale and avoid any individual feature dominating the model while undergoing training. The ultimate stage of the Data Engineering phase involves implementing a stratified sampling technique to partition the dataset into distinct training and testing subsets. This approach ensures that the proportion of benign and malicious samples is maintained in both subsets. This measure guarantees that both categories are sufficiently represented during the construction of the model and enables an unbiased assessment of the model's efficacy on novel data. The Data Engineering phase is crucial in establishing a solid foundation for developing ML models that can accurately and effectively identify and mitigate cyber threats associated with malicious PDF files. This involves a thorough and precise collection, preprocessing, and data preparation.

3.3. Feature Construction

The Evasive-PDFMal2022 dataset's malware detection method depends significantly on the input features. To create a customized set of relevant features for this dataset, we perform FEng, which may involve considering features like file size, embedded object count, JavaScript usage, encryption status, and metadata properties. Apart from these, features associated with the document structure, such as the number of pages, fonts used, pdf size, and layout information, could also play a critical role in the detection process. Table 1 comprehensively describes the generated features in the Evasive-PDFMal2022 dataset, highlighting their significance in enhancing the model's ability to identify and classify malicious PDF files accurately.

Table 1. Features descriptions.

F#	Feature Name	Data Type	Description
F1	JavaScript	Object	Number of JavasScript keywords
F2	Metadata size	Float64	Information about the PDF for embedding hidden contents.
F3	StartXref	Object	The number of keywords denotes the start of the Xef table.
F4	OpenAction	Object	The number of OpenAction keywords.
F5	Obj	Object	Number of total objects inside the PDF.
F6	Endobj	Object	The number of indirect objects in a PDF.
F7	XFA	Object	XML Form Architecture to support scripting technologies.
F8	Pdfsize	Float64	The size of a PDF file.
F9	Pageno	Object	Page number of pdf file pages.
F10	Xreflength	Float64	The number of Xref tables.
F11	Images	Object	Number of images in the PDF document.
F12	Text	Object	Presence of text inside the PDF.
F13	Pages	Float64	The number of pages in the PDF file.
F14	Title Characters	Float64	The number of characters in the title.
F15	Isencrypted	Float64	Document Encryption(shows if PDF document is protected or not).
F16	Embedded Files	Float64	The number of embedded files inside the PDF document.
F17	Stream	Float64	The number of sequences of binary data in the PDF.
F18	Endstream	Object	Keywords that denote the end of the streams.
F19	Xref	Object	Xref: Number of Xref tables.
F20	Trailer	Float64	Trailer: Number of trailers inside the PDF.
F21	Encrypt	Float64	The encryption technique used in PDF documents.
F22	ObjStm	Float64	The number of stream objects that contain other objects.
F23	JS	Object	The number of JavaScript codes.
F24	AA	Object	The number of AA keywords: states a specific action upon an event
F25	Acroform	Object	Number of Acroform tags PDF to support scripting technologies
F26	JBIG2decode	Object	Presence of JBig2Decode filter.
F27	Richmedia	Object	The number of embedded media and flash files.
F28	Launch	Object	The number of Launch keywords used to execute a command/program.
F29	EmbeddedFile	Object	The average size of all the embedded media.
F30	Colors	Float64	The different colors that are used in the PDF.
F31	Fine name	Object	The name of the pdf file
F32	Class	Object	Malicious or Benign

3.4. Feature Selection

This step chooses a smaller set of features from the original collection. After that, the smaller sample is used to retrain the classifiers and get new results. The objective is to minimize input features while maintaining relevance and eliminating irrelevance. State-of-the-art explainable AI techniques, including SHAP and LIME, are employed to interpret the results of ML models. These methods highlight the most influential features affecting the model's decision-making process. Local explainable AI examines individual data points, while global explainable AI identifies significant features across all data points for training the model. It is widely known that having more features can result in a more complex ML model, which may be prone to overfitting. Therefore, this study uses SHAP values to select the most important features before implementing the proposed XAI-PDF detection system. SHAP values, a method based on cooperative game theory, are employed to enhance the transparency and interpretability of ML models. In this article, the impact of the 33 features on the performance of the proposed method is measured using SHAP values. Consequently, the features that contribute to the model are selected are input features. The SHAP values analysis in Figure 7 shows that 12 features contribute to the model, i.e., [JavaScript, metadata size, startXref, OpenAction, obj, endobj, XML Forms Architecture (XFA), pdfsize, pageno, Xref, length, images, text]. Therefore, 20 features are removed from the input features list.

3.5. Derived Features Phase

The most effective classifier is chosen based on SHAP values in the derived features phase. This leads to identifying new and more critical features for the top-performing classifier and the standard and highly influential features impacting the model's decision-making process. This approach showcases how interpretable AI techniques like SHAP can guide FEng processes and improve model performance.

3.6. ML Models

In the current study, several ML models were utilized to detect malicious PDFs, such as DT with Entropy, DT with Gini, KNN, LGBM, LR, and SVM. These models contain hyperparameters that have a significant impact on the learning performance of the classifier. The parameters that were used in these models are listed in Table 2.

Table 2. Summary of values for classifiers parameters.

Parameter	Criterion	Random_state
DT/Entropy	Entropy	TRUE
DT/Gini	Gini	TRUE
KNN	Number of Neighbors	FALSE
LGBM	Gini	TRUE
LR	Log-likelihood	TRUE
SVM	Probability	TRUE

3.7. Model Evaluation Phase

The evaluation phase uses five-fold cross-validation to supplement validation. The XAI-PDF model's accuracy, precision, recall, F1-measure, False Positive Ratio (FPR), False Negative Ratio (FNR), and testing duration are evaluated. The measures above are crucial for assessing the model's sensitivity-specificity balance. Each fold evaluates model performance, and the aggregate performance metrics are derived by averaging all experiments.

3.8. Prediction Activities

The effectiveness of the proposed methodology has been examined by conducting two distinct experiments. The main aim of the initial experiment is to evaluate the influence of SHAP values on the effectiveness of the proposed methodology. The suggested XAI-PDF detection system is built in two different ways in this experiment: with and without the SHAP values as a way to choose which features to employ. The SHAP values obtained by deleting eight features from the input list are shown in Figure 10. The results demonstrate that removing these features has an insignificant impact on the model's performance. The results of applying the proposed method with and without the SHAP values are displayed in Table 5.

- Phase 1. Experiment: they have built six classifiers using SHAP values tool feature sets to determine and ascertain the features' importance of the standard features outlined in section 3.2. Furthermore, the model's performance was classified using DT/Entropy, DT/Gini, KNN, LGBM, LR, and SVM. During this procedure, the identification and selection of the most crucial features of the model were undertaken. Utilizing the SHAP values tool was crucial in advancing ML models, resulting in enhanced effectiveness.
- Phase 2. Experiment: experiment 2 aims to enhance the ML model's performance. The SHAP summary plot explains the model's decision-making process and evaluates the model's output. This detected model bias and the need for more data to improve performance. The SHAP summary plot helped find innovative model-improving elements. Various methods were investigated to extract more information from these features or merge them with others to create unique, more informative features. This strategy improved ML models. The dataset was classified again to determine the importance of the derived features from section 3.3.

4. Experiments and Results

This section evaluates the detection system in the context of XAI-PDF based on SHAP values for XAI. Experiments were conducted to evaluate the effectiveness of using SHAP values as a method for selecting features and to evaluate the overall performance of this proposed method. In addition, the proposed method's results were compared to those of recent approaches described in the existing literature.

4.1. Experiments Setup

In evaluating the performance of the proposed XAI-PDF system, all experiments were conducted under controlled and consistent conditions. Experiments are performed on machines with hardware specifications, as listed in Table 3.

Table 3. Experimental environment hardware specifications.

Brand	**ThinkPad E560**
RAM	16GB
HD	250GB SSD
System Processor	Intel(R) Core(TM) i7-6500U CPU © 2.50GHz 2.60
OS	Windows 10 Enterprise

The experiments aimed to examine the effectiveness and potential of the XAI-PDF system and SHAP values in the domains of FEng and interpretability. A set of experiments was conducted to evaluate these aspects rigorously. Additionally, a comparative analysis was conducted to evaluate the performance of our approach when compared to other contemporary methods. A five-fold cross-validation methodology is employed to ensure effective validation in all experiments. The proposed approach involves randomly partitioning the complete dataset into five distinct folds. Each fold is created by dividing the data in a 70:30 ratio. The performance of the model is assessed at each fold [33]. The procedure above is iterated five times, and the aggregate performance metrics are computed by taking the average of the outcomes obtained from the five experiments (folds/iterations).

4.2. Evaluation Metrics

In evaluating the performance of a binary ML-based classifier, four fundamental metrics are employed: TP, True Negative (TN), FP, and FN, which together form the confusion matrix. Figure 3 shows the confusion matrix.

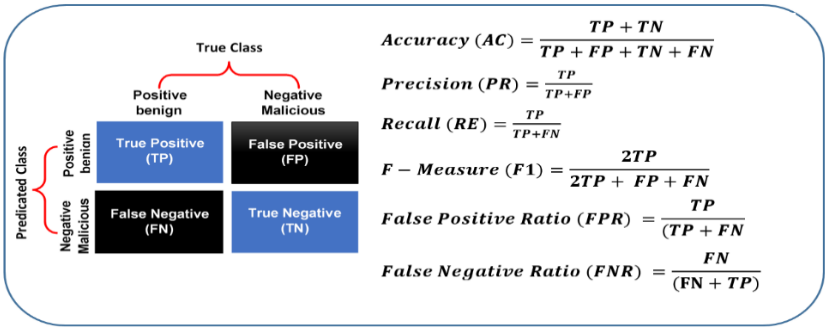

True Class
Positive benign
Negative Malicious
Predicated Class
Positive benign
Negative Malicious
True Positive (TP)
False Positive (FP)
False Negative (FN)
True Negative (TN)
$Accuracy\ (AC) = \frac{TP+TN}{TP+FP+TN+FN}$
$Precision\ (PR) = \frac{TP}{TP+FP}$
$Recall\ (RE) = \frac{TP}{TP+FN}$
$F-Measure\ (F1) = \frac{2TP}{2TP+FP+FN}$
$False\ Positive\ Ratio\ (FPR) = \frac{TP}{(TP+FN)}$
$False\ Negative\ Ratio\ (FNR) = \frac{FN}{(FN+TP)}$

Figure 3. Performance assessment of measurement (confusion matrix).

This study assesses the performance of XAI-PDF using seven parameters: accuracy, precision, recall, F1-measure, FPR, FNR, and testing time. Accuracy represents the proportion of correct predictions relative to all predictions, while precision and recall specifically target the malicious class. The F1 measure computes the harmonic mean of precision and recall. These metrics, contingent on the confusion matrix, are crucial for model evaluation, especially when handling imbalanced data. FPR quantifies the proportion of incorrect positive predictions among actual negatives, and FNR measures the proportion of incorrect negative predictions among actual positives. These ratios are instrumental in evaluating the balance between sensitivity and specificity, particularly in imbalanced datasets. Testing time refers to the duration necessary for a trained classifier to process a single input instance and generate a prediction. These measures' computation depends on the confusion matrix [7].

4.3. Phase 1 Experiment: Feature and Classifier Evaluation

The goal of this section is to:

1) Conduct initial training and testing to obtain preliminary results using all classifiers.
2) Identify crucial features from the feature set described in section 3.4. selection features that carry the most weight in detecting malicious PDFs.
3) Evaluate the performance and accuracy of the classifiers to select the best-performing classifier.

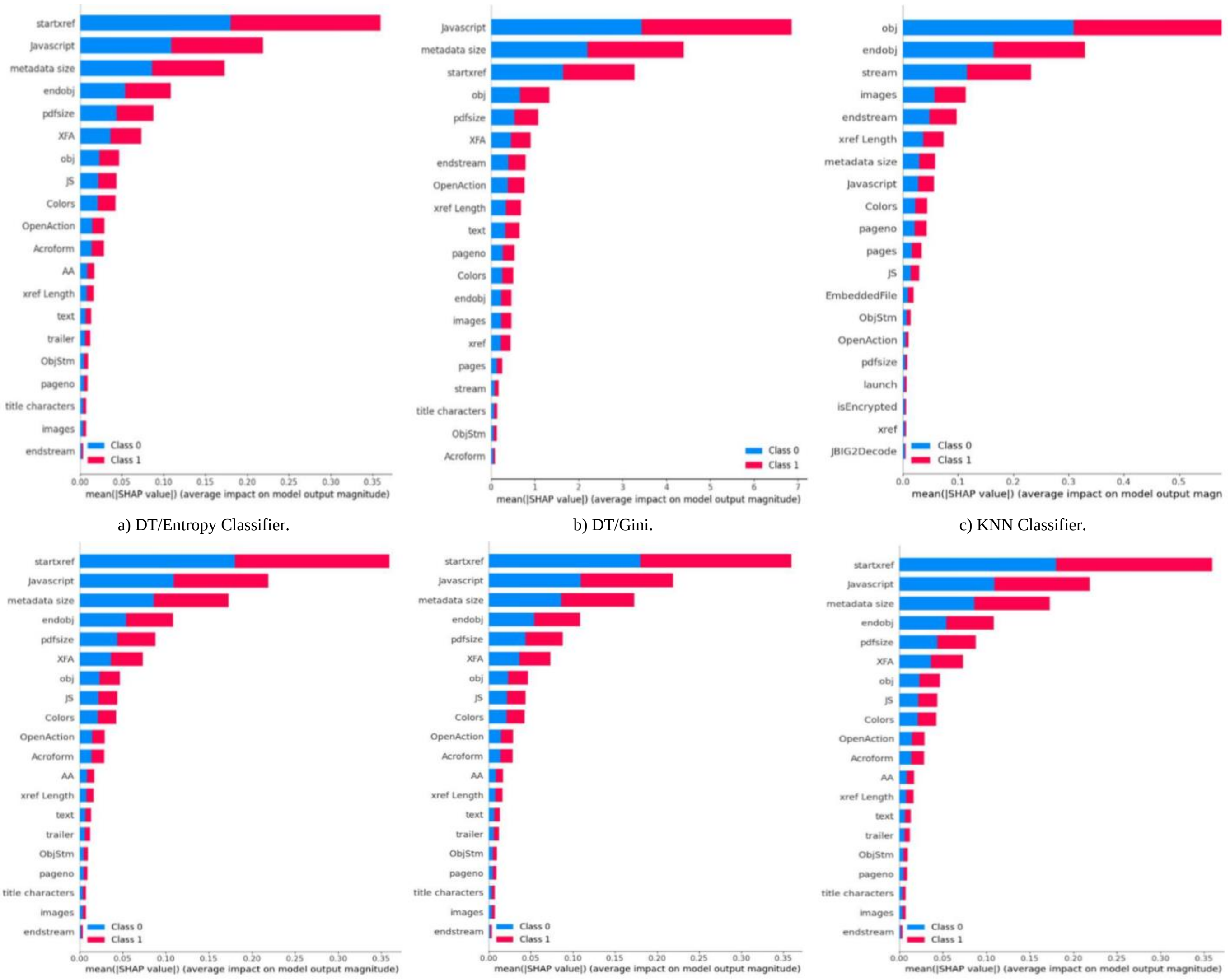
Consequently, this research's primary objective is to develop efficient and interpretable models for malicious PDF detection, utilizing both ML and XAI techniques. Providing predictions with explanations enhances the trust and reliability of AI-based systems. To select the best-performing classifier, six different types of classifiers were considered for training and testing, including DT/Entropy, DT/Gini, KNN, LGBM, LR, and SVM.

The aim was to establish a lightweight solution; therefore, deep neural networks were not used due to their high computational requirements compared to the six selected models. The primary focus of this investigation is the selection of the best-performing classifier by employing SHAP values to determine the most influential features of the proposed classifiers. Consequently, this study offers an in-depth analysis of the ML models illustrated in Figure 4, which are based on the SHAP libraries. Figure 4 displays the results of the SHAP values analysis, which measures the impact of 32 features on the performance of the proposed classifiers used for training and testing the XAI-PDF detection system.

Figure 4 illustrates the SHAP values associated with each feature, representing the shift in model output from our initial expectation to the final prediction made by the model. The features are arranged in ascending order based on their SHAP values, with the features having smaller values grouped towards the bottom of the display beyond the maximum threshold. The red and blue hues symbolize the respective roles played by malware and benign features in this context. A comprehensive set of 32 features was considered, particularly emphasizing the impact of the top 12 features. In Figure 4-a) and (b), the first three features, ‘startXref,’ ‘JavaScript,’ and ‘metadata size,’ have almost the same values. In Figures 4-c) and (d), the KNN and LGBM models have the same results for the features and their importance. In the LR model, Figure 4-e), the most important feature is ‘obj,’ which appeared in other models' top 10 important features. Finally, Figure 4-f) shows the SVM model, which only shows the top 7 features, which are ‘Xreflength,’ ‘metadata size,’ ‘obj,’ ‘endobj,’ ‘pdfsize,’ ‘stream’ and ‘endstream’ while the other feature has very low values, almost 0.

The experimental results for the performance evaluation of the proposed system using six learning models (DT/Entropy, DT/Gini, KNN, LGBM, LR, and SVM) are summarized in Table 4 and Figure 5. The comparative study considers detection accuracy, precision, sensitivity (recall), harmonic mean (F score), FP rate, and detection time. The empirical evaluation reveals that the LGBM model provides higher performance rates for malicious PDF detection than other models. These results also emphasize the superior performance of LGBM, with values of 99.9%, 100%, 99.9%, and 99.9% for accuracy, precision, recall, and F-measure, respectively, along with minimal FP and FN rates. Furthermore, LGBM offers the highest inference speed compared to other models for detecting malicious PDFs, requiring only 1.36 milliseconds per record for predictions compared to other classifiers. The DT/Gini and DT/Entropy classifiers achieved a detection time of 1.79ms and a loss rate of 12%, demonstrating competitive performance rates with low detection time. Furthermore, the KNN classifier provided competitive performance rates of 97.95%, 97.2%, 98.3%, and 97.7% for AC, PR, RE, and F1, respectively. Nevertheless, the KNN model exhibited the lowest performance rate in terms of detection time, displaying a decrease of 78% compared to the top-performing model.

In contrast, LR and SVM exhibited the lowest performance rates, with values of (86.3%, 66.3%), (87%, 92.2%), (82.2%, 28.7%), and (84.5%, 43.8%) for accuracy, precision, recall, and F-measure, with (13.7%, 33.6%), (13%, 7.7%), (17.8%, 71.3%), and (15.4%, 56.2%) loss, respectively. The lowest performance rates were reported for the LR and SVM models, scoring detection time, falling below the best model by 14.25% to 47.13%. Additionally, these models exhibited the worst performance in terms of FN rate, with a more than 71% loss compared to the best model.



a) DT/Entropy Classifier.

b) DT/Gini.

c) KNN Classifier.

d) LGBM Classifier.

e) LR Classifier.

f) SVM Classifier.

Figure 4. Global analysis using SHAP.

Table 4. Experiment (1) result. A comparison of 6 ml algorithms' accuracy, precision, recall, and F1.

Model	AC	PR	RE	F1	FPR	FNR	Error rate	Testing time
DT/ Entropy	0.998	0.999	0.997	0.998	0.001	0.003	0.002	1.79
DT/ Gini	0.999	0.999	0.999	0.999	0.001	0.001	0.001	1.79
KNN	0.979	0.972	0.983	0.977	0.024	0.017	0.019	6.89
LGBM	0.999	1.000	0.999	0.999	0.000	0.001	0.000	1.57
LR	0.863	0.870	0.822	0.845	0.103	0.178	0.123	1.83
SVM	0.664	0.922	0.287	0.438	0.020	0.713	0.240	2.97

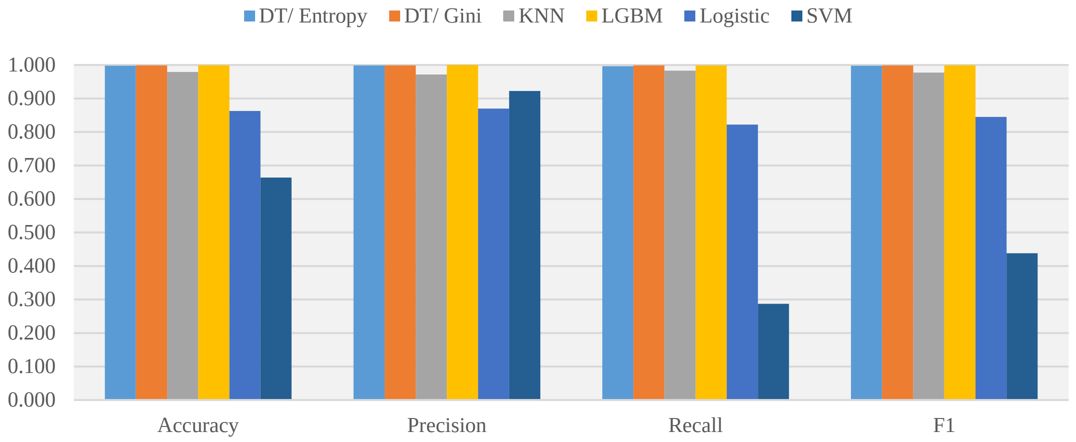


Figure 5. Comparing the performance for several implemneted models to selected the best proposed model.

4.4. Phase 2 Experiment: Derived Features Evaluation

Experiment 2 aimed to develop a more effective ML model by identifying the most important features using the SHAP method, improving model accuracy and interpretability, and making better decisions in various applications. In this experiment, the most effective classifier was identified using SHAP values. In addition to the standard features, a set of features was derived from malicious PDFs.

4.4.1. Derived Features Evaluation

Based on the top-performing LGBM model and the analysis of the most influential features for classifying malicious PDFs, a unique set of features was derived from malicious PDFs. The SHAP summary plot revealed malicious PDFs associated with the JavaScript, js, text, and images features. Figure 6 represents the SHAP summary plot that displays the results of the SHAP values analysis, which measures the impact of 20 features on the performance of the proposed classifiers (LGBM model) used for training and testing the XAI-PDF detection system. The JavaScript feature is one of the most significant features contributing to the LGBM model's decision-making process for detecting malicious PDFs. High values of the JavaScript feature tend to push the LGBM model's output towards a positive prediction for benign files and a negative prediction for malicious files. The SHAP summary plot shows that the high and low values of the JavaScript feature significantly impact the model's output.

The Text feature denotes the existence of textual content within the PDF document. The SHAP summary plot indicates that the inclusion or exclusion of text is a predictive factor for the LGBM model's ability to identify malicious PDFs. However, its influence could be stronger when compared to other features. The SHAP summary plot for the Images feature indicates that its inclusion or exclusion is a predictive factor for the LGBM model's ability to identify malicious PDFs. However, it is noteworthy that the influence of this feature is comparatively not strong compared to other features. The merge of specific features, such as having images and text or JavaScript code and “js” in the file name, may have a stronger influence on model output than the effects of each feature in isolation. The JavaScript_js and images_text feature exhibit substantial predictive power in the LGBM model for identifying malicious PDFs. The identification of new features that have the potential to enhance the performance of the model is of utmost importance. For instance, in cases where a particular feature significantly influences the output of the model, it is advisable to investigate methods for extracting extra information from said feature or amalgamating it with other features to generate new features that are more informative.

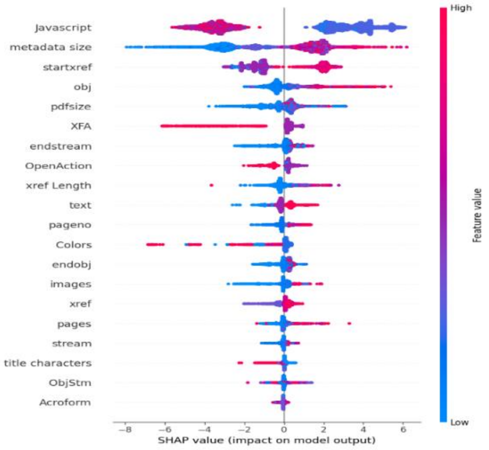


Figure 6. LGBM summary plot.

The selected features were determined based on their performance and significance in the SHAP values, as illustrated in Figure 4-d). The following features were selected. JavaScript, metadata size, startXref, OpenAction, obj, endobj, XFA, pdfsize, pageno, Xref length, images, text, and two extra features were constructed using the best values.

4.4.2. Feature Selection of LGBM after Derived Features

To conduct a more comprehensive assessment of the derived features, we determined the significance of each feature based on the XAI-PDF model's performance, both with and without the inclusion of derived features. This paper aims to establish a lightweight solution. Consequently, an additional experiment is conducted to validate the significance and relevance of the derived features and determine whether they should be considered influential in the final set of features. Simplifying the model and mitigating the risk of overfitting can enhance the model's performance.

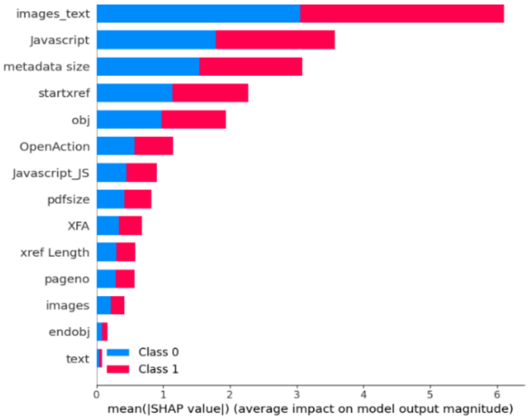


Figure 7. Global analysis using SHAP for LGBM classifier after applied derived features.

Figure 7 presents the SHAP values analysis results, which measure the impact of 20 selected features obtained through the phase 1 experiment. As a result, 14 features with the highest feature importance have resulted based on SHAP values, including (JavaScript, metadata size, startXref, OpenAction, obj, endobj, XFA, pdfsize, pageno, Xref, length, images_text), plus two derived features (JavaScript_js, images_text). Figure 7 displays the SHAP values obtained by removing certain features from the input list, demonstrating that removing these features has an insignificant impact on the model's performance.

4.4.3. Evaluation of Proposed XAI-PDF System

To conduct a more comprehensive assessment of the significance of the derived features, we performed calculations to determine the importance of each feature based on the results obtained from the new classifier. The results of the classification process are presented in Table 5, which demonstrates that integrating the derived features has maintained the performance measures of the classifier without any negative impact on performance. Furthermore, the decision-making time has improved because of this integration. Table 5 summarizes experimental results for the proposed method (XAI-PDF) using the LGBM classifier. The table reveals the performance metrics for LGBM with and LGBM without derived features. Accordingly, the performance evaluation metrics for the proposed method were not affected by removing 20 out of 32 features. In addition, the required time to make a prediction using the proposed method with the SHAP values and derived features is less than that of the proposed method without derived features. Employing SHAP values as a feature selection method improves the time needed to predict records by 13.5%.

Table 5. Experiment (2) results. Comparing XAI-PDF performance for the LGBM model.

Model	CA	PR	RE	F1	FPR	FNR	Error Rate	Time
LGBM without derived feature	0.999	1.000	0.999	0.999	0.000	0.001	0.000	1.57
LGBM with derived feature	0.999	1.000	0.998	0.999	0.000	0.002	0.001	1.36

Finally, the proposed model was validated by utilizing SHAP values to automate the feature selection process, considering only features contributing to the system's decision-making process. Our analysis showed that by implementing this technique, the feature-set size could be reduced by more than 56% while increasing the model's speed. The features with the highest importance were then used to construct a new classifier, resulting in a significant boost in classification performance and improved testing time. Our study demonstrated how XAI techniques, like SHAP, can make AI models more understandable and improve their performance. This section's subtitle is comparison of XAI-PDF with and without derived features techniques. In Figure 8, the summary plot provides a visual representation of the impact of different features on the model's predictions. Each feature is represented by a vertical bar, where the length indicates its importance, and the color reflects its value relative to a baseline. Blue represents malicious values or features that decrease the model's output, which, as seen in the figure, are more toward the negative side. In contrast, red represents benign values or features that increase the model's output and are more on the positive side. The intensity of the color corresponds to the magnitude of the SHAP value. Darker or more intense colors indicate a stronger impact on the prediction.

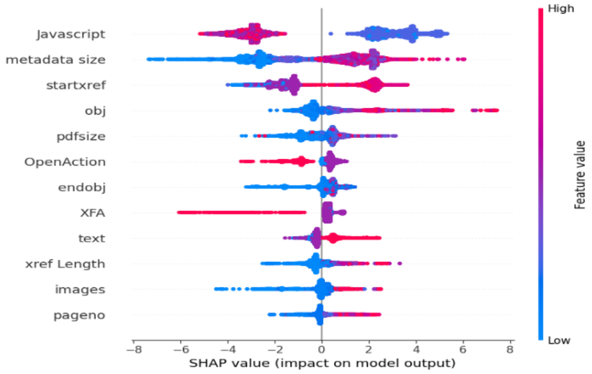


Figure 8. LGBM after summary plot.

5. Discussion and Comparisons

5.1. Model Interpretation and Validation Using XAI

Explaining ML model output uses SHAP. It calculates the model's prediction's feature contribution. These SHAP values reveal how the model makes judgments and which features are most essential for its predictions. SHAP values can also reveal model errors. These cases' SHAP values can reveal the features most responsible for the inaccurate prediction and why. This can help you fix model mistakes and biases. SHAP values and other interpretable AI methods can reveal a model's decision-making process and predictive properties. This can help you find model mistakes or biases and enhance performance. Figure 9 shows each feature's contribution to a prediction. The waterfall plot shows how each feature affects the prediction. Each step in the waterfall plot corresponds to a feature and its importance, and the length of the step indicates the magnitude of the feature's contribution. The 'JavaScript' feature has the highest magnitude of -4.55, and the feature 'images' has the lowest negative magnitude of -0.41. The 'startXref' feature has the highest positive magnitude of +1.9, and 3 other features have a positive magnitude.

In Figure 10, the force plot visually represents the individual feature contributions as arrows, showing the direction and magnitude of their effects on the prediction.' JavaScript,' 'metadata size,' and 'startXref' are the top 3 contributors; their magnitudes are the highest, with the first 2 features as negative contributors and 'startXref' as a positive contributor. All features meet at the -6.89 value, as shown in the figure since most features have a negative magnitude.

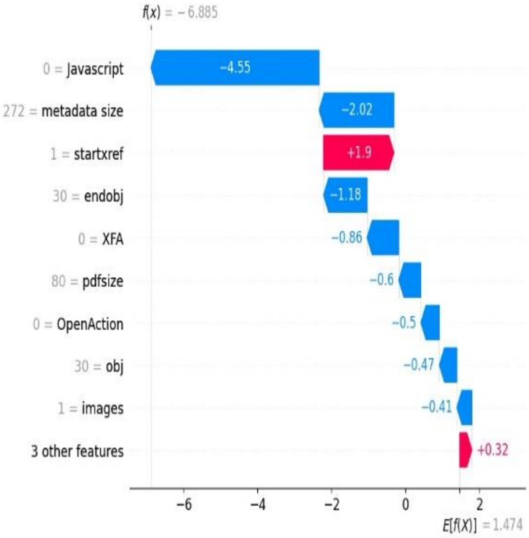


Figure 9. Waterfall plot compares model output to data distribution based on feature SHAP values.

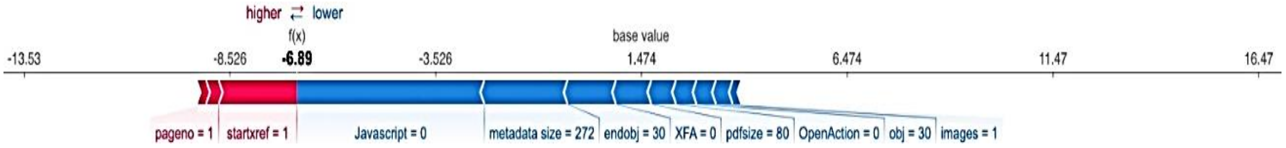
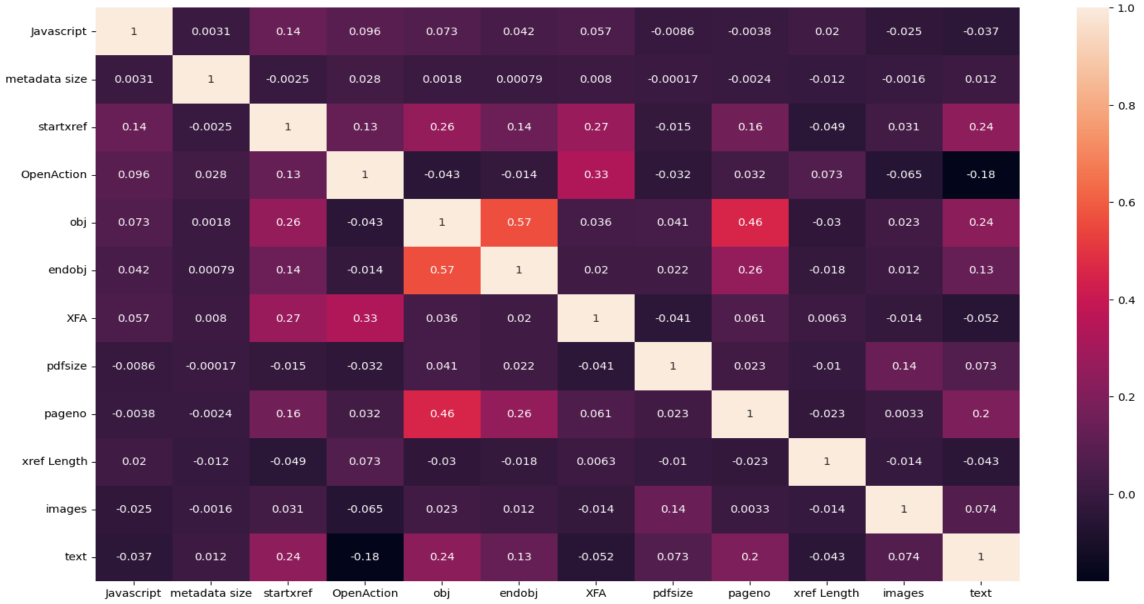


Figure 10. Threshold-dependent feature contribution to a class.

Using a color scale, Figure 11 shows a heat map representing the correlation coefficients between the chosen features. The strength of the correlation is represented by the intensity of the colors, with lighter colors representing positive correlations that lead to 1 and darker colors showing negative correlations that lead to -1. The intensity of the colors also shows the direction of the correlation



	Javascript	metadata size	startxref	OpenAction	obj	endobj	XFA	pdfsize	pageno	xref Length	images	text
Javascript	1	0.0031	0.14	0.096	0.073	0.042	0.057	-0.0086	-0.0038	0.02	-0.025	-0.037
metadata size	0.0031	1	-0.0025	0.028	0.0018	0.00079	0.008	-0.00017	-0.0024	-0.012	-0.0016	0.012
startxref	0.14	-0.0025	1	0.13	0.26	0.14	0.27	-0.015	0.16	-0.049	0.031	0.24
OpenAction	0.096	0.028	0.13	1	-0.043	-0.014	0.33	-0.032	0.032	0.073	-0.065	-0.18
obj	0.073	0.0018	0.26	-0.043	1	0.57	0.036	0.041	0.46	-0.03	0.023	0.24
endobj	0.042	0.00079	0.14	-0.014	0.57	1	0.02	0.022	0.26	-0.018	0.012	0.13
XFA	0.057	0.008	0.27	0.33	0.036	0.02	1	-0.041	0.061	0.0063	-0.014	-0.052
pdfsize	-0.0086	-0.00017	-0.015	-0.032	0.041	0.022	-0.041	1	0.023	-0.01	0.14	0.073
pageno	-0.0038	-0.0024	0.16	0.032	0.46	0.26	0.061	0.023	1	-0.023	0.0033	0.2
xref Length	0.02	-0.012	-0.049	0.073	-0.03	-0.018	0.0063	-0.01	-0.023	1	-0.014	-0.043
images	-0.025	-0.0016	0.031	-0.065	0.023	0.012	-0.014	0.14	0.0033	-0.014	1	0.074
text	-0.037	0.012	0.24	-0.18	0.24	0.13	-0.052	0.073	0.2	-0.043	0.074	1

Figure 11. The absolute mean of the main and interaction effects for the first 14 features.

Figure 12 showcases a matrix of squares, where each square represents the interaction effect between a specific pair of features. Each square's color and intensity indicate the interaction effect's magnitude and direction. The more the effect is on the right side of the line, the higher its magnitude is; as shown in the figure between 'JavaScript' and itself, it has the highest positive magnitude, almost +5, while other features like 'pdfsize' and 'endobj' have almost 0 magnitudes and not very strong interaction since the color of the effect is light.

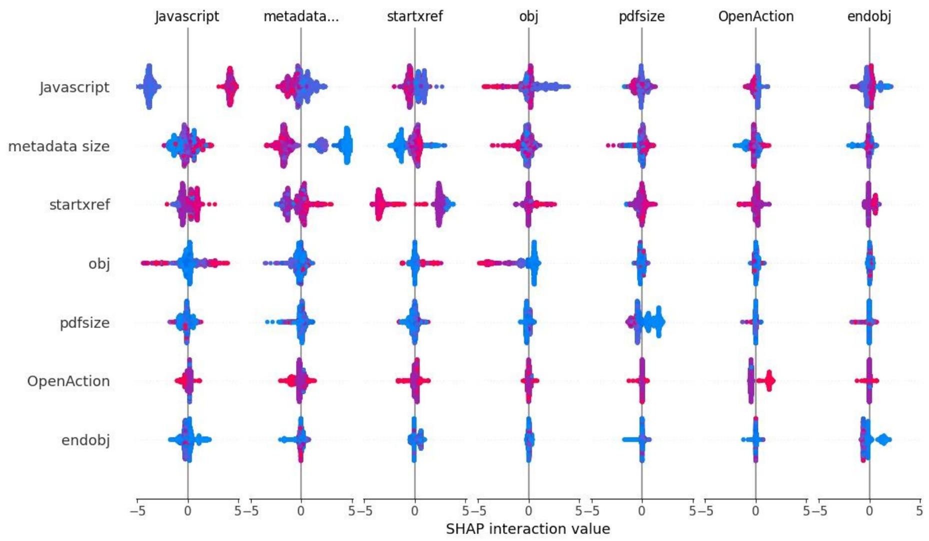


Figure 12. Model output feature interaction.

5.2. Comparisons with Previous Methods

Table 6 shows the proposed performance compared to other approaches from the literature, which shows that it is more resilient in detecting evasive malicious PDF files than state-of-the-art methods. The XAI-PDF model's superior accuracy, precision, recall, F1 score, and reduced rates of FPs and negatives are the foundation for its superior robustness. The model also boasts a decreased decision-making time per record, demonstrating its effectiveness and reliability.

The proposed XAI-PDF model, which combines XAI, PDF file detection, and optimal FEng, outperforms several existing models. The key comparison criteria are accuracy, precision, recall, F1 score, and testing time. The performance of the proposed XAI-PDF model, which combines XAI and PDF malware detection with optimized FEng, outstrips several existing models in the literature. The critical comparison metrics used are accuracy, precision, recall, F1 score, and testing time.

However, including XAI in malware detection, as demonstrated in the PAIRED/RF model by Alani and Awad [4] and the models by Rahman *et al*. [40], represents a fundamental change in the area. In particular, the model created by Alani and Awad [4], which used FEng on 35 characteristics, reported an accuracy of 98.07%. Rahman *et al*.s' research [40] shows how XAI may be used with different ML models Stochastic Gradient Descent (SGD), XGBoost, perception, and ANN) to detect PDF malware, with their Artificial Neural Network (ANN) model demonstrating an accuracy of 99.68%. However, their models did not strongly emphasize the optimal selection and engineering of features, which could have significantly improved their performance.

Our proposed XAI-PDF model brings together the strengths of XAI, PDF malware detection, and a precise feature selection mechanism, leading to an impressive accuracy of 99.9%, precision of 100%, recall of 99.8%, and an F1 score of 99.99%. Compared to the other models in the literature, our model enhances the detection accuracy and improves precision, recall, and F1 score while maintaining a reasonable testing time. Using FEng based on SHAP values in our model has been instrumental in achieving this superior performance.

Table 6. A comparison between the proposed model and other works.

Paper	Year	Model	Detection	FEng	AC	PR	RE	F1	Test Time
Li *et al*. [28]	2017	FEPDF	Mal-PDF Detection	NO	95.11%	97.57%	90.87%	-	-
Zhang [49]	2018	MLPdf/ MLP-NN	Mal-PDF Detection	No	95.12%	-	-	-	-
Falah *et al*. [16]	2021	PDFiD, PeePDF	Mal-PDF Detection	Yes	98.6%	99%	98.3%	98.6%	
Mohammed *et al*. [37]	2021	HAPSSA	Mal-PDF Detection	Yes	99.92%	-	-	-	-
Abu Al-Haija *et al*. [2]	2022	O-DT	Mal-PDF Detection	NO	98.84%	98.80%	98.90%	98.8%	2.174 ms
Alani and Awad [4]	2022	PAIRED/RF	XAI with Mal-PDF	Yes	98.07%	-	-	-	0.7631ms
Rahman *et al*. [40]	2023	SGD	XAI with Mal-PDF	Yes	97.93%	-	-	-	0.92ms
Our model	2023	XAI-PDF	XAI with Mal-PDF	Yes	99.9%	100%	99.8%	99.9%	1.36ms

6. Conclusions and Remarks

An intelligent, trustworthy recognition scheme with enhanced accuracy and minimized decision-making time to identify malicious PDF files (XAI-PDF) has been proposed, developed, evaluated, and reported in this paper. The proposed system characterizes the performance of six ML algorithms: Decision tree-based Entropy criteria (DT/Entropy), Decision tree-based Gini criteria (DT/Gini), KNN, LGBM, LR, and SVM. The proposed model has been evaluated on an inclusive, up-to-date dataset, the Evasive-PDFMal2022 dataset, comprising around 20K examples of benign and malicious PDF files. To ensure the trustworthiness of the optimized classification, the model has been explained using SHAP techniques through a four-phase approach: data preparation, model building, explainability of the models, and derived features (FEng). We identified crucial features while generating new ones by employing the interpretability of SHAP values. This process has led to an improved classification model, demonstrating the profound impact of interpretable AI techniques on overall model performance. The simulation evaluation for the proposed XAI-PDF system has disclosed its superiority scoring notable results for accuracy, precision, and F1-scores with minimal FP and FN rates (99.90%, 100.00%, 99.89%, 0.000 and 0.002, respectively). Besides, XAI-PDF required only 1.36 milliseconds per record for predictions. Finally, the comparisons of XAI-PDF with several baseline methods have demonstrated the increased resilience of XAI-PDF in detecting evasive malicious PDF files compared to state-of-the-art methods. Finally, it works remarks on the following four outcomes inferred from the development and evaluation of the proposed system:

- Implementing SHAP values for feature engineering and interpretability in XAI-PDF optimizes the malicious PDF classifier performance by reducing model complexity without negatively impacting accuracy.
- XAI-PDF leads to better accuracy than ML-based methods using the same input features.
- Using SHAP values for feature engineering and interpretability in XAI-PDF leads to identifying crucial features and generating new ones, resulting in an improved classification model that enhances model performance.
- XAI-PDF is more resilient in detecting evasive malicious PDF files than state-of-the-art methods, as demonstrated by its minimal FP and FN rates and low decision-making time per record.

References

[1] Abu Al-Haija Q., Alohaly M., and Odeh A., "A Lightweight Double-Stage Scheme to Identify Malicious DNS over HTTPS Traffic Using a Hybrid Learning Approach," *Sensors*, vol. 23, no. 7, pp. 1-19, 2023. https://www.mdpi.com/1424-8220/23/7/3489

[2] Abu Al-Haija Q., Odeh A., and Qattous H., "PDF Malware Detection Based on Optimizable Decision Trees," *Electron*, vol. 11, no. 19, pp. 1-18, 2022. https://doi.org/10.3390/electronics11193142

[3] Adhatarao S. and Lauradoux C., "Robust PDF Files Forensics Using Coding Style," *in Proceedings of the 37th IFIP TC-11 International Conference, ICT Systems Security and Privacy Protection*, Copenhagen, pp. 179-195, 2022. https://doi.org/10.1007/978-3-031-06975-8_11

[4] Alani M. and Awad A., "PAIRED: An Explainable Lightweight Android Malware Detection System," *IEEE Access*, vol. 10, pp. 73214-73228, 2022. DOI:10.1109/ACCESS.2022.3189645

[5] Al-Fawa'reh M., Al-Fayoumi M., Nashwan S., and Fraihat S., "Cyber Threat Intelligence Using PCA-DNN Model to Detect Abnormal Network Behavior," *Egyptian Informatics Journal*, vol. 23, no. 2, pp. 173-185, 2022. https://doi.org/10.1016/j.eij.2021.12.001

[6] Al-Fayoumi M. and Abu Al Haija Q., "Capturing Low-Rate DDoS Attack Based on MQTT Protocol in Software Defined-IoT Environment," *Array*, vol. 19, pp. 100316, 2023. https://doi.org/10.1016/j.array.2023.100316

[7] Al Fayoumi M., Al Fawareh M., and Nashwan S., "VPN and Non-VPN Network Traffic Classification Using Time-Related Features," *Computers, Materials and Continua*, vol. 72, no. 2, pp. 3091-3111, 2022. https://doi.org/10.32604/cmc.2022.025103

[8] Al-Fayoumi M., Elayyan A., Odeh A., and Al-Haija Q., "Tor network Traffic Classification Using Machine Learning Based on Time-Related Feature," *in Proceedings of the 6th Smart Cities Symposium, IET Conference*, Bahrain, pp. 92-97, 2022. DOI: 10.1049/icp.2023.0354

[9] Al-Fayoumi M., Alwidian J., and Abusaif M., "Intelligent Association Classification Technique for Phishing Website Detection," *The International Arab Journal of Information Technology*, vol. 17, no. 4, pp. 488-469, 2020. https://doi.org/10.34028/iajit/17/4/7

[10] Baz M., Alhakami H., Agrawal A., Baz A., and Khan R., "Impact of Covid-19 Pandemic: A Cybersecurity Perspective," *Intelligent Automation and Soft Computing*, vol. 27, no. 3, pp. 641-652, 2021. https://doi.org/10.32604/iasc.2021.015845

[11] Bose S., Towards Explainability in Machine Learning for Malware Detection, Ph.D Dissertation, Florida State University, 2020.

https://diginole.lib.fsu.edu/islandora/object/fsu:776810

[12] Buriro A., Buriro A., Ahmad T., Buriro S., and Ullah S., "MalwD and C: A Quick and Accurate Machine Learning-Based Approach for Malware Detection and Categorization," *Applied Science*, vol. 13, no. 4, pp. 1-14, 2023. https://doi.org/10.3390/app13042508

[13] Corona I., Maiorca D., Ariu D., and Giacinto G., "Lux0R: Detection of Malicious PDF-Embedded JavaScript Code through Discriminant Analysis of API References," *in Proceedings of the Workshop on Artificial Intelligent and Security Workshop*, pp. 47-57, Arizona, 2014. https://doi.org/10.1145/2666652.2666657

[14] Corum A., Jenkins D., and Zheng J., "Robust PDF Malware Detection with Image Visualization and Processing Techniques," *in Proceedings of the 2nd International Conference on Data Intelligence and Security*, Texas, pp. 108-114, 2019. DOI: 10.1109/ICDIS.2019.00024

[15] Cuan B., Damien A., Delaplace C., and Valois M., "Malware Detection in PDF Files Using Machine Learning," *in Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, Porto, pp. 412-419, 2018. https://www.scitepress.org/Link.aspx?doi=10.5220/0006884704120419

[16] Falah A., Pan L., Huda S., Pokhrel S., and Anwar A., "Improving Malicious PDF Classifier with Feature Engineering: A Data-Driven Approach," *Future Generation Computer Systems*, vol. 115, pp. 314-326, 2021. https://doi.org/10.1016/j.future.2020.09.015

[17] Ferrag M., Maglaras L., Argyriou A., Kosmanos D and Janicke H., "Security for 4G and 5G Cellular Networks: A Survey of Existing Authentication and Privacy-Preserving Schemes," *Journal of Network and Computer Applications*, vol. 101, pp. 55-82, 2018. https://doi.org/10.1016/j.jnca.2017.10.017

[18] Gorment N., Selamat A., Cheng L., and Krejcar O., "Machine Learning Algorithm for Malware Detection: Taxonomy, Current Challenges and Future Directions," *IEEE Access*, vol. 11, pp. 141045-141089, 2023. DOI: 10.1109/ACCESS.2023.3256979

[19] He K., Zhu Y., He Y., Liu L., Lu B., and Lin W., "Detection of Malicious PDF Files Using a Two-Stage Machine Learning Algorithm," *Chinese Journal of Electronics*, vol. 29, no. 6, pp. 1165-1177, 2020. https://doi.org/10.1049/cje.2020.10.002

[20] Index of /CICDataset/CICEvasivePDFMal2022/Dataset, http://205.174.165.80/CICDataset/CICEvasivePDFMal2022/Dataset/, Last Visited, 2023.

[21] Issakhani M., Victor P., Tekeoglu A., and Lashkari A., "PDF Malware Detection Based on Stacking Learning," *SciTePress*, vol. 1, pp. 562-570, 2022. DOI: 10.5220/0010908400003120

[22] Jeong Y., Woo J., and Kang A., "Malware Detection on Byte Streams of PDF Files Using Convolutional Neural Networks," *Security and Communication Networks*, vol. 2019, pp. 1-10, 2019. https://doi.org/10.1155/2019/8485365

[23] Kang A., Jeong Y., Kim S., and Woo J., "Malicious PDF Detection Model against Adversarial Attack Built from Benign PDF Containing JavaScript," *Applied Science*, vol. 9, no. 22, pp. 1-17, 2019. https://doi.org/10.3390/app9224764

[24] Kattamuri S., Penmatsa R., Chakravarty S., and Madabathula V., "Swarm Optimization and Machine Learning Applied to PE Malware Detection towards Cyber Threat Intelligence," *Electron*, vol. 12, no. 2, pp. 1-25, 2023. https://doi.org/10.3390/electronics12020342

[25] Kumar D. and Das S., "Machine Learning Approach for Malware Detection and Classification Using Malware Analysis Framework," *International Journal Intelligent Systems Applications Engineering*, vol. 11, no. 1, pp. 330-338, 2023. https://ijisae.org/index.php/IJISAE/article/view/2543/1126

[26] Kumar R. and Subbiah G., "Explainable Machine Learning for Malware Detection Using Ensemble Bagging Algorithms," *in Proceedings of the 14th International Conference on Contemporary Computing*, Noida, pp. 453-460, 2022. https://doi.org/10.1145/3549206.3549284

[27] Li K., Gu Y., Zhang P., An W., and Li W., "Research on KNN Algorithm in Malicious PDF Files Classification under Adversarial Environment," *in Proceedings of the 4th International Conference on Big Data and Computing*, Guangzhou, pp. 156-159, 2019. https://doi.org/10.1145/3335484.3335527

[28] Li M., Liu Y., Yu M., Li G., Wang Y., and Liu C., "FEPDF: A Robust Feature Extractor for Malicious PDF Detection," *in Proceedings of the IEEE Trustcom/BigDataSE/ICESS*, Sydney, pp. 218-224, 2017. DOI:10.1109/Trustcom/BigDataSE/ICESS.2017.240

[29] Li Y., Wang X., Shi Z., Zhang R., Xue J., and Wang Z., "Boosting Training for PDF Malware Classifier Via Active Learning," *International Journal of Intelligent Systems*, vol. 37, no. 4, pp. 2803-2821, 2022. https://doi.org/10.1002/int.22451

[30] Li Y., Wang Y., Wang Y., Ke L., and Tan Y., "A Feature-Vector Generative Adversarial Network for Evading PDF Malware Classifiers,"

Information Sciences, vol. 523, pp. 38-48, 2020. https://doi.org/10.1016/j.ins.2020.02.075

[31] Lin Y. and Chang X., "Towards Interpreting ML-Based Automated Malware Detection Models: A Survey," *arXiv Preprint*, vol. arXiv:2101.06232v1, pp. 1-39, 2021. https://doi.org/10.48550/arXiv.2101.06232

[32] Liu Y., Tantithamthavorn C., Li L., and Liu Y., "Explainable AI for Android Malware Detection: Towards Understanding Why the Models Perform So Well?," *in Proceedings of the IEEE 33rd International Symposium on Software Reliability Engineering*, North Carolina, pp. 169-180, 2022. DOI:10.1109/ISSRE55969.2022.00026

[33] Lu K., Cheng J., and Yan A., "Malware Detection Based on the Feature Selection of a Correlation Information Decision Matrix," *Mathematics*, vol. 11, no. 4, pp. 1-17, 2023. https://doi.org/10.3390/math11040961

[34] Maiorca D., Biggio B., and Giacinto G., "Towards Adversarial Malware Detection: Lessons Learned from PDF-Based Attacks," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1-36, 2019. https://doi.org/10.1145/3332184

[35] Maiorca D., Giacinto G., and Corona I., "A Pattern Recognition System for Malicious PDF Files Detection," *in Proceedings of the 8th International Conference of Machine Learning and Data Mining in Pattern Recognition*, Berlin, pp. 510-524, 2012. https://doi.org/10.1007/978-3-642-31537-4_40

[36] Mejjaouli S. and Guizani S., "PDF Malware Detection Based on Fuzzy Unordered Rule Induction Algorithm (FURIA)," *Applied Science*, vol. 13, no. 6, pp. 1-13, 2023. https://doi.org/10.3390/app13063980.

[37] Mohammed T., Nataraj L., Chikkagoudar S., Chandrasekaran S., and Manjunath B., "HAPSSA: Holistic Approach to PDF Malware Detection Using Signal and Statistical Analysis," *in Proceedings of the IEEE Military Communications Conference MILCOM*, San Diego, pp. 709-714, 2021. DOI:10.1109/MILCOM52596.2021.9653097

[38] Nwakanma C., Ahakonye L., Njoku J., Odirichukwu J., Okolie S., and Uzondu C., "Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review," *Applied Sciences*, vol. 13, no. 3, pp. 1-29, 2023. https://doi.org/10.3390/app13031252

[39] Ogiriki I., Machine Learning Models Interpretability for Malware Detection Using Model Agnostic Language for Exploration and Explanation, Master Thesis, Rowan University, 2022. file:///C:/Users/user/Desktop/ELEC5200_6200%20Performance.pdf

[40] Rahman T., Ahmed N., Monjur S., Haque F., and Hossain M., "Interpreting Machine and Deep Learning Models for PDF Malware Detection using XAI and SHAP Framework," *in Proceedings of the 2nd International Conference Innovation in Technology*, Bangalore, pp. 1-9, 2023. DOI:10.1109/INOCON57975.2023.10101116

[41] Sayed S. and Shawkey M., "Data Mining Based Strategy for Detecting Malicious PDF Files," *in Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE*, New York, pp. 661-667, 2018. DOI:10.1109/TrustCom/BigDataSE.2018.00097

[42] Scalas M., Malware Analysis and Detection with Explainable Machine Learning, Ph.D Thesis, Università degli Studi di Cagliari, 2021. https://iris.unica.it/retrieve/e2f56eda-0586-3eaf-e053-3a05fe0a5d97/tesi%20di%20dottorato_Michele%20Scalas.pdf

[43] Singh P., Tapaswi S., and Gupta S., "Malware Detection in PDF and Office Documents: A Survey," *Information Security Journal: A Global Perspective*, vol. 29, no. 3, pp. 134-153, 2020. https://doi.org/10.1080/19393555.2020.1723747

[44] Smutz C. and Stavrou A., "Malicious PDF Detection Using Metadata and Structural Features," *in Proceedings of the 28th Annual Computer Security Applications Conference*, Florida, pp. 239-248, 2012. https://doi.org/10.1145/2420950.2420987

[45] Smutz C. and Stavrou A., "When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors," *in Proceedings of the Network and Distributed System Security Symposium*, San Diego, pp. 1-15, 2016. DOI:10.14722/ndss.2016.23078

[46] Tay K., Chua S., Chua M., and Balachandran V., "Towards Robust Detection of PDF-Based Malware," *in Proceedings of the 12th ACM Conference on Data and Application Security and Privacy*, Maryland, pp. 370-372, 2022. https://doi.org/10.1145/3508398.3519365

[47] Ullah F., Alsirhani A., Alshahrani M., Alomari A., Naeem H., and Shah S., "Explainable Malware Detection System Using Transformers-Based Transfer Learning and Multi-Model Visual Representation," *Sensors*, vol. 22, no. 18, pp. 1-22, 2022. https://doi.org/10.3390/s22186766

[48] Younisse R., Ahmad A., and Abu Al-Haija Q., "Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP)," *Big Data and*

Cognitive Computing, vol. 6, no. 4, pp. 1-20, 2022. https://doi.org/10.3390/bdcc6040126

[49] Zhang J., "MLPdf: An Effective Machine Learning Based Approach for PDF Malware Detection," *arXiv Preprint*, arXiv:1808.06991v1, pp. 1-6, 2018. https://doi.org/10.48550/arXiv.1808.06991

Mustafa Al-Fayoumi received a BSc degree in Computer Science from Yarmouk University, Irbid, Jordan, in 1988. He earned an MSc degree in Computer Science from the University of Jordan, Amman, Jordan, in 2003. In 2009, he successfully completed his Ph.D. with distinction in Computer Science from the Faculty of Science and Technology at Anglia University, UK. Currently, he is an Associate Professor of Computer Science/Cybersecurity and the Chair of the Cybersecurity Department at Princess Sumaya University for Technology (PSUT). He authorizes more than 50 scientific research papers His research interests include Cybersecurity, Cryptography, Identification and Authentication, Wireless and Mobile Networks Security, E-Application Security, Machine Learning, and other related topics.

Qasem Abu Al-Haija received his Ph.D. from Tennessee State University (TSU), USA, in 2020. He is an Assistant Professor at the Department of Cybersecurity, Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan. He authorizes more than 100 scientific research papers and book chapters. His research interests include Artificial Intelligence (AI), Cybersecurity and Cryptography, the Internet of Things (IoT), Cyber-Physical Systems (CPS), Time Series Analysis (TSA), and Computer Arithmetic. Recently, he was listed as one of the world's top 2% of scientists list released publicly by Stanford University and Elsevier Publisher.

Rakan Armoush graduated with distinction, receiving a Bachelor's degree in Data Science and Artificial Intelligence from Princess Sumaya University for Technology (PSUT) in 2022. During his time at PSUT, he showcased exceptional leadership capabilities as the head of the Data Science Club, fostering a dynamic environment for aspiring data scientists. Currently holding a position at Microsoft, he utilizes his expertise in Data Science and AI to drive innovation.

Christine Amareen is a 4th year Data Science and Artificial Intelligence BSc student at Princess Sumaya University for Technology (PSUT), Amman, Jordan. Her research interests include cyber security, machine learning, data mining, data science, machine learning in the medical field and other related topics. She is currently working on a research paper for Osteoporosis Prediction using Machine Learning and Statistical Models. Her commitment to advancing knowledge and exploring innovative solutions makes her a promising researcher in her field.