electronics

Article

PDF Malware Detection Based on Optimizable Decision Trees

Qasem Abu Al-Haija [1,*], Ammar Odeh [2] and Hazem Qattous [3]

[1] Department of Cybersecurity, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan
[2] Department of Computer Science, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan
[3] Department of Software Engineering, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan
[*] Correspondence: q.abualhaija@psut.edu.jo

Abstract: Portable document format (PDF) files are one of the most universally used file types. This has incentivized hackers to develop methods to use these normally innocent PDF files to create security threats via infection vector PDF files. This is usually realized by hiding embedded malicious code in the victims' PDF documents to infect their machines. This, of course, results in PDF malware and requires techniques to identify benign files from malicious files. Research studies indicated that machine learning methods provide efficient detection techniques against such malware. In this paper, we present a new detection system that can analyze PDF documents in order to identify benign PDF files from malware PDF files. The proposed system makes use of the AdaBoost decision tree with optimal hyperparameters, which is trained and evaluated on a modern inclusive dataset, viz. Evasive-PDFMal2022. The investigational assessment demonstrates a lightweight and accurate PDF detection system, achieving a 98.84% prediction accuracy with a short prediction interval of 2.174 µSec. To this end, the proposed model outperforms other state-of-the-art models in the same study area. Hence, the proposed system can be effectively utilized to uncover PDF malware at a high detection performance and low detection overhead.

Keywords: portable document format (PDF); machine learning; detection; optimizable decision tree; AdaBoost; PDF malware; evasion attacks; cybersecurity

Citation: Abu Al-Haija, Q.; Odeh, A.; Qattous, H. PDF Malware Detection Based on Optimizable Decision Trees. *Electronics* **2022**, *11*, 3142. https://doi.org/10.3390/electronics11193142

Academic Editors: Jungong Han and Ahmed Abu-Siada

Received: 6 September 2022
Accepted: 28 September 2022
Published: 30 September 2022

1. Introduction

A piece of harmful code that has the potential to damage a computer or network is referred to as malware. As conventional signature-based malware detection technologies become useless and unworkable, recent years have seen a significant increase in malware. Malware developers and cybercriminals have adopted code obfuscation techniques, which reduce the efficiency of defensive mechanisms against malware [1,2].

Malware classification and identification remain a challenge in this decade. This is largely because advanced malware is more sophisticated and has the cutting-edge ability to remain hidden or change its code or behavior to behave more intelligently. As a result, outdated detection and classification methods are less useful today. As a result, the focus has shifted to machine learning for better malware identification and categorization [3,4].

Malicious PDF software is one of the common hacking methods [5]. Forensic research is hampered by the difficulty of separating harmful PDFs from large PDF files. Machine learning has advanced to the point where it may now be used to detect malicious PDF documents to assist forensic investigators or shield a system from assault [6]. However, adversarial techniques have been developed against malicious document classifiers. Precision-manipulation-based hostile examples that have been carefully crafted could be misclassified. This poses a danger to numerous machine-learning-based detectors [7,8]. For particular attacks, various analysis or detection methods have been provided. The threat posed by adversarial attacks has not yet been fully overcome. Figure 1 depicts a PDF document's header, body, cross-reference table (xref), and trailer components [9].

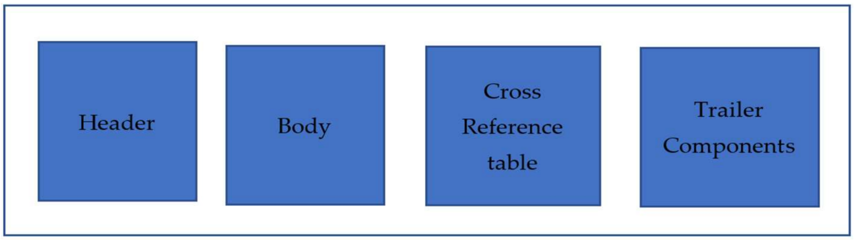


Figure 1. Structure of a PDF file.

The interpreter format version that will be utilized is specified in the header. The PDF's body defines its content and includes text blocks, fonts, pictures, and file-specific metadata. The document's content is contained in a group of PDF elements. These things can fall under one of four categories: Booleans, strings, streams, and numbers [10].

An analyst or analysis tool may use static, dynamic, or hybrid malware analysis techniques (Figure 2) [11]. Static analysis techniques examine the sample without running the code and rely on the file attributes, such as the code structure. In analytical methods, dynamism executes the code to observe its behavior, such as the program network operations [12].

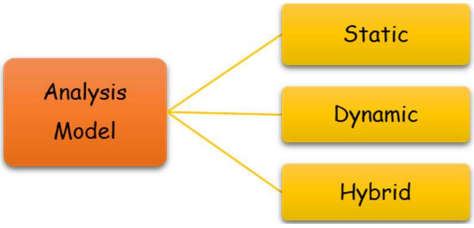


Figure 2. Structure of a PDF file.

Adopting advanced evasion and obfuscation techniques to mask dangerous runtime behavior makes static analysis vulnerable. It is insufficient to undertake static analysis alone in the current security environment. Any attacker serious about their campaign will obfuscate and encrypt their code, typically undetectable by static analysis.

On the other hand, dynamic approaches are more resistant to code obfuscation, making them more effective against sophisticated viruses [13]. To avoid harm, dynamic techniques must run the virus in a secure, sandboxed environment. Whether it believes the malware is running in a sandbox or not, an adversary may change the virus's behavior to obstruct the malware analysis process [14,15]. While static analysis is frequently quick, dynamic analysis is typically slow and difficult. Hybrid analysis refers to the combining of the two methodologies. This is more efficient against sophisticated malware than either of the two ways, but it also takes more time and requires a more involved analysis process [16].

In this paper, we present a new detection system that can analyze PDF documents to identify benign PDF files from malware PDF files. The proposed system uses the AdaBoost decision tree with optimal hyperparameters [17], which is trained and evaluated on a modern inclusive dataset, viz. Evasive-PDFMal2022. The investigational assessment demonstrates a lightweight and accurate PDF detection system, achieving a 98.84% prediction accuracy with a short prediction interval of 2.174 μSec. To this end, the proposed model outperforms other state-of-the-art models in the same study area. Hence, the proposed system can effectively uncover PDF malware at a high detection performance and low detection overhead.

The rest of this paper is organized as follows: Section 2 presents a systematic and inclusive review of the recent related articles in the same field of study. Section 3 provides the modeling architecture for the malware PDF detection system. Section 4 presents and discusses the performance and experimental evaluation results. Lastly, Section 5 provides the concluding remarks.

2. Literature Review

Deep learning methods, particularly Deep Neural Networks (DNN), have become popular in academic and industrial areas [18]. Their applications can be found in various fields, including malware analysis. On resource-demanding tasks such as speech recognition, natural language processing, and picture recognition, DNN performs well. However, it has been demonstrated that machine-learning-based systems' categorization is susceptible to hostile settings with cutting-edge evasion attempts [19].

For the identification of malware, supervised machine learning has been frequently used. Several detectors that use this technology were created specifically for PDF files in the past ten years. Choosing whether any unknown PDFs should be classified as harmful or benign is the main objective of machine learning detectors for malicious document identification. Such systems can work by examining data retrieved from the document's content or structure. Their general process flow is depicted in Figure 3, which comprises three main sections [20]: pre-processing analyzes PDF files and provides access to data essential for detection.

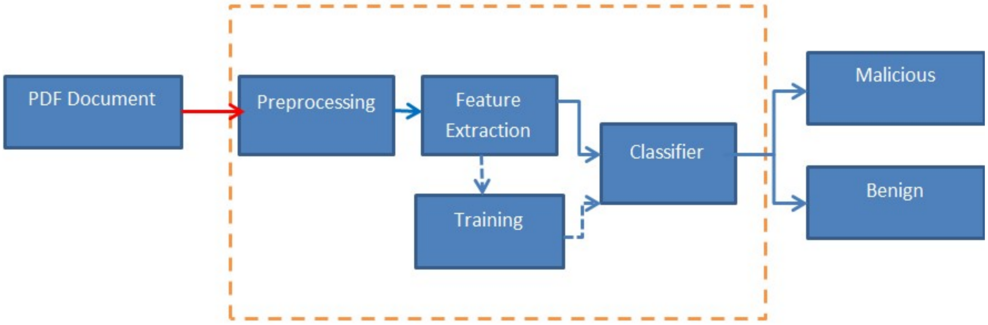


Figure 3. General process flow of machine learning techniques.

The information is transformed into a normalized vector as part of the feature extraction process. To ensure an accurate prediction, the classifier chooses the best learning algorithm for training and modification to acquire improved parameters. Because the quality of the features may have a distinct impact on the prediction performance, feature extraction is crucial [21].

An integrated method for malware detection that uses static and dynamic features was introduced by [22]. Combining static and dynamic features has improved identification accuracy compared to using static or dynamic approaches separately. According to the findings, the support vector machine (SVM) learning method is the most effective at classifying data. However, in addition to improvements in false positive (FP) and false negative (FN) rates, the random forest (RF) also improved the accuracy [23]. The classification findings show that dynamic analysis is superior to code-based static approaches. In comparison to static approaches, the dynamic method is more accurate. The integrated strategy improves detection accuracy, in line with the study goal.

In [24], the authors proposed a brand-new embedded malware detection system based on statistical anomaly detection techniques. This is the first anomaly-based malware detection method to pinpoint the infection location within an infected file. The suggested Markov n-gram detector outperforms existing detectors in terms of detection rate. Additionally, when used with current commercial off-the-shelf antivirus (COTS-AV) software, the suggested detector can offer very low false-positive rates due to its capacity to locate embedded malware.

A non-signature-based technique that examines the byte-level file content has been proposed by [25]. Such a method offers inherent resistance to typical obfuscation strategies, particularly those that use repacked malware to hide signatures. This study has found that infected and benign files differ fundamentally, even at the byte level. Thirteen unique statistical and information-theoretic features computed on 1, 2, 3, and 4 g of each file block are used in the proposed approach, which has a rich feature set.

In [26], the authors introduce a framework for machine-learning-based robust detection of fraudulent documents. The suggested method is based on elements taken from the document's structure and metadata. The study demonstrates the suitability of certain document attributes for malware identification and the resilience of these features against new virus strains using real-world datasets. The analysis phase shows that the ensemble classifier Random Forests, which randomly chooses features for each distinct classification tree, produces the highest detection rates.

In this investigation, two main data sources were used. The first is the widely used Contagio data collection [27], which is intended for testing and studying signatures. This source of datasets was chosen because it has many papers classified as malicious and benign, including a sizable proportion from targeted attacks. This source offers a few document sets. The second collection comes from the network monitoring of a sizable university campus. These files were taken from Simple Mail Transfer Protocol (SMTP) and Hypertext Transfer Protocol (HTTP) traffic.

Authors in [28] devised a method to identify a set of features extracted using extant tools and derive a new set of features from improving PDF maldoc detection and extending the lifespan of existing analysis and detection technologies. The derived features are evaluated with a wrapper function that uses three fundamental supervised learning [29] algorithms (Random Forests, C5.0 Decision Trees, and two-class Support Vector Machines) and a feed-forward deep neural network to determine how important the features are. Finally, a new classifier is built using features of the highest significance, dramatically improving classification performance with less training time. The results were confirmed using sizable datasets from VirusTotal.

The authors of [30] present a brand-new technique for pinpointing an ensemble classifier's data struggles. When enough individual classifier votes conflict during detection, the ensemble classifier's prediction is demonstrated to be incorrect. Without the need for extra external ground truth, the suggested technique, ensemble classifier mutual agreement analysis, enables the discovery of numerous types of classifier evasions.

Using PDFrate, a PDF malware detector, the authors test the proposed strategy and demonstrate that the great majority of predictions can be generated with high ensemble classifier agreement using data from an entire network and our methodology. Nine targeted mimicking situations from two recent bouts of research are among the classifier evasion efforts typically assigned an unclear outcome, indicating that the classifier cannot provide a reliable forecast for these data [31]. To demonstrate the approach's broad applicability, the author tested it against the Drebin Android malware detector, where most special attacks were correctly predicted as uncertain. The proposed method can be applied more broadly to reduce the potency of attacks on support vector machines made via gradient descent and kernel density estimation. The most crucial element for enabling ensemble classifiers' diversity-based evasion detection is feature bagging.

The authors in [32] introduce Lux0R, sometimes known as "Lux 0n discriminant References", a unique and portable method for identifying fraudulent JavaScript code. The suggested approach is based on characterizing JavaScript code through references to its API, which includes functions, constants, objects, methods, keywords, and attributes natively recognized by a JavaScript Application Programming Interface (API). The suggested methodology uses machine learning to identify a subset of API references that are indicative of dangerous code and then uses those references to identify JavaScript malware. It has been said that the selection mechanism is "safe by design" against evasion using mimicking assaults. Identifying dangerous JavaScript code in PDF documents is the relevant application domain that the author focuses on in this work.

This technique can obtain outstanding malware detection accuracy even on samples that exploit previously unknown vulnerabilities, i.e., for which there are no instances in training data. Finally, an experimental evaluation of Lux0R's resistance to mimicking attacks based on feature augmentation is performed.

This work [33] presents a novel approach that combines a feature extractor module closely related to the structure of PDF files with a powerful classifier. This technique has shown to be more efficient than most commercial antivirus programs and other cutting-edge research tools for detecting dangerous PDF files. Furthermore, because of its adaptability, it can be used to enhance the efficiency of an antivirus that is already installed or as a stand-alone program.

It performs significantly better than Wepawet, a potent instrument created by academics. Wepawet has been created to detect various threats, including malicious PDF files, but the developed program is focused on detecting PDF attacks.

It can be further enhanced by assessing the proposed system's resilience to new vulnerabilities and enhancing the parsing procedure. The suggested tool might also be a component of a multi-classifier system, where each classifier focuses on identifying particular dangers. Making our security systems stronger against a wider range of dangers and providing them the ability to anticipate new threats is a challenge for the future as attacker tactics advance.

The authors in [34] discovered the flaws in the existing feature extractors for PDFs by reviewing them and examining how the malicious template was implemented. The authors then created a powerful feature extractor called FEPDF (feature extractor-PDF), which can extract features that conventional feature extractors might overlook and capture realistic information about the elements in PDFs. The authors produced many brand-new malicious PDFs as samples to test the current antivirus engines and feature extractors. The findings demonstrate that several current antivirus engines could not recognize the new harmful PDFs, but FEPDF can extract the crucial elements for enhanced hazardous PDF detection.

This study [35] demonstrates the typical K-nearest neighbor (KNN) classification algorithm's weaker resistance in adversarial environments by using the gradient-descent attack method to alter the malicious samples in the test set to evade detection by the classifier. The authors provide a method in which the created adversarial samples are added to the training set, followed by the usage of the training set to create a new KNN classifier and test their robustness against various attack strengths.

Finally, the tests demonstrate that the robustness of the KNN classifier may be greatly increased without impacting the generalization performance of the KNN classifier by including the adversarial samples produced by gradient-descent attacks on the training set.

A new data-mining-based approach is provided by the authors of [36], introduced for identifying fraudulent PDF files. There are two stages to the proposed algorithm: feature selection and classification. The feature selection step is utilized to choose the ideal amount of features extracted from the PDF file to achieve a high detection rate and a low false-positive rate with little computational cost. According to experimental data, a suggested algorithm can achieve a 99.77% detection rate, 99.84% accuracy, and 0.05% false-positive rate.

It can perform better by comparing the suggested algorithm against antivirus programs from CalamAV, TrendMicro, MacAfee, and Symantec. The suggested algorithm is based on data mining techniques, which gives it the edge over antivirus software in detecting harmful PDF files that have never been seen before. Consequently, the suggested method can better identify advanced persistent threats (APTs).

Using a gradient-descent (GD) approach, the naive SVM used by the authors in [37] was easily deceived by us. The authors also devised defenses against this assault by setting a threshold over each considered feature.

This allowed the suggested method to thwart practically all gradient-descent attacks. Next, fewer features were chosen so that features used in the gradient-descent assault could be removed. This reduces the attack's viability even further at the expense of the SVM's precision [38]. The authors also suggested employing adversarial learning to train the SVM using gradient-descent-forged PDF files and repeating the procedure to decrease the likelihood that the gradient-descent attack will succeed. After only three cycles, the SVM exhibited resistance to attacks using gradient-descent techniques.

Authors in [39] offer in-depth analyses of PDFs' JavaScript content and structure. They created a rich feature set in JavaScript that includes content features such as object names, keywords, and readable strings, as well as the structure and metadata features such as the file size, version, encoding method, and keywords. It is challenging to create hostile examples when features are diverse because machine learning algorithms are resistant to tiny alterations. To reduce the risk of adversarial assaults, analysts create detection models employing black-box types of models with structure and content properties. The authors created the adversarial attack to verify the suggested model. Additionally, we gather wholesome documents with various JavaScript codes for the foundation of the hostile samples.

The PDF files used in this study comprise 9000 benign and 11,097 malicious document files gathered by the Contagio malware dump between November 2009 and June 2018 [39]. The malware samples are provided via the Contagio malware dump site. From the website, researchers can obtain samples of malware. The samples cover a large amount of time. The authors gathered 115 clean files with JavaScript files separate to develop an adversarial assault for the validation. The authors, except for encrypted files, successfully implanted harmful software into 101 clean files.

In terms of machine learning methods, the authors discovered that while most conventional machine learning algorithms perform adequately for malware detection, they perform worse for adversarial samples, except for the random forest algorithm. Due to this transferability, the random forest algorithm may perform well.

The author in [40] methodically put forth several guiding concepts to select features to decrease the capacity for escape while retaining high accuracy. These guidelines are followed for extracting features and training a two-stage classifier. The experimental findings demonstrate that our model performs superbly in accuracy, generalization capability, and robustness. It can also differentiate between the vulnerability used in malicious files.

The author introduced a strategy to identify the software that created a PDF file [41] based on coding style: specific patterns that specific PDF producers only produce. Additionally, they looked at the coding practices of 900 PDF files created by 11 distinct PDF producers on three different operating systems. A set of 192 rules that can be used to identify 11 PDF manufacturers has been acquired by the authors. We used 508,836 PDF files from scientific preprint sources to test our identification method. The tool used has a 100% accuracy rate for identifying specific producers. Overall, it still detected well (74%). To understand how online PDF services operate and detect inconsistencies, utilize the provided tool. Lastly, Table 1 summarizes the important reviewed related research.

In [42], the authors provided a thorough summary of the approaches currently used for malicious document identification. The foundational tools that are frequently employed in detection approaches were covered. Various methods were categorized based on the chosen features and the static/dynamic analysis.

A new distance metric to bound robustness features was proposed in [43]. The proposed model maintained 99.74% accuracy and a 0.56% false-positive rate while achieving 99.68% and 85.28% verified robust accuracy (VRA) for the insertion and deletion properties, respectively. The findings demonstrated that training security classifiers with verified robustness attributes is a promising direction for raising the bar for unrestrained attackers.

In [44], the importance of doubtful samples is confirmed, and a detection model based on active learning is used with those uncertain samples. A small number of information-rich test set samples, or so-called doubtful samples, are chosen to supplement the training set during each assessment epoch, gradually enhancing the classifier's performance. Compared to conventional retraining methods, the authors significantly cut the time needed for training. This study uses a mutual-agreement-analysis-based active learning approach. The authors employed the Hidost model with active learning and mutual agreement analysis as the criterion for selecting doubtful samples, enabling us to use a smaller training set and improve classification performance.

Authors in [45] proposed a new model called OPEM. This hybrid malware detector combines the frequency of occurrence of operational codes (statically collected) with details of an executable execution trace (dynamically obtained). The authors demonstrated that, when used independently, this hybrid strategy improves the performance of both approaches.

The authors of [46] suggested a revolutionary learning-based method for detecting PDF malware through processing and visualizing images. Grayscale graphics are created from the PDF files (byte and Markov plots). The different visual qualities of the images are then retrieved using image features such as Keypoint descriptors and texture features.

A unique evasion technique based on a feature vector generative adversarial network (fvGAN) was proposed in [47] to target a learning-based malware classifier. This work generates adversarial feature vectors in the feature space using the fvGAN and then converts those feature vectors into actual adversarial malware samples. The findings demonstrate that the fvGAN model has a high evasion rate within a constrained time. The proposed strategy has also been contrasted with two currently used attack algorithms, Mimicry and GD-KDE. The findings show that the proposed technique performs better regarding both the evasion rate and execution cost.

In [48], the authors studied three well-known attacks—Mimicry, Mimicry+, and Reverse Mimicry—to compare how well they evade classifiers in Hidost and Mimicus. The findings demonstrate that Mimicry and Mimicry+ are successful in avoiding models in Mimicus but not in Hidost, while Reverse Mimicry is successful in avoiding both Mimicus and Hidost models.

Due to the pervasive usage of attack channels such as documents, malware continues to pose a danger to cybersecurity. These infection vectors conceal harmful code from the victim users, making it easier to infect their computers through social engineering methods [49].

Table 1. Summary of reviewed related research.

Ref.	Model	Datasets	Analysis Method	Advantages	Limitation
[22]	SVM, RF	997 virus files and 490 clean files	Hybrid	The high accuracy rate for static, dynamic, and combined techniques.	Very small dataset
[24]	Markov n-gram	37,000 malware and 1800 benign	Static	The Markov n-gram detector offers higher detection and false-positive rates than the other embedded malware detection method currently in use.	An evasion test is not available.
[25]	(J48) classifier	VX Heavens Virus Collection [50]	Static	The proposed model may identify the malware file's family, such as virus, trojan, etc.	An evasion test is not available.
[26]	RF	Contagio [27]	Static	Even though the training set, classification technique, and document features are known, the classifier is resistant to mimicking attacks.	Evasion is much more challenging because classification depends more evenly on many parameters.
[28]	RF, C5.0 Decion Tree (DT), and 2-class SVM	Contagio [27] + VirusTool [51]	Static	It gives us a thorough grasp of how these selected features affect classification, and this will improve the training time.	All datasets provided by VirusTotal are benign, and this will make decisions biased.
[30]	ensemble classifier (random sampling/bagging)	Contagio [27]	Dynamic	Using real data	It does not examine any potential embedded PDF payload.
[32]	Heuristic-based	Contagio [27]	Dynamic	More resistant to code obfuscation	Any API extraction mistakes could compromise the accuracy of the detector.
[10]	Bayesian, SVM, J48, and RF	Contagio [27]	Static	Multi-classifier system	Not efficient with different types of embedded malicious codes in PDF files
[35]	KNN	Generated Dataset	Static	It drastically lowers false negatives and improves detection accuracy by at least 15%.	An evasion test is not available.
[36]	heuristic search, RF, AND DT	Generated Dataset	Static	Identifying advanced persistent threats	It was not tested against evasion techniques and mimicry attacks.
[37]	Naive SVM	Dump [52]	Static	Prevent gradient-descent attacks	Slower than other algorithms
[39]	RF, SVM, and NB	Contagio malware dump between November 2009 and June 2018 [44]	Static	Adequately for malware detection	Not detect adversarial samples
[40]	Convolutional Neural Network (CNN)	VirusTotal	Static	Robustness against evasive samples	Can not detect adversarial samples
[41]	Coding style	HAL dataset [53]	Static	Trust generation process for PDF files	Time-consuming: the complexity depends on the file size.

Table 1. *Cont.*

Ref.	Model	Datasets	Analysis Method	Advantages	Limitation
[42]	Distance metric in the PDF tree structure	Contagio [27]	Static	Verified robust accuracy	Time-consuming due to insertion and deletion of the tree
[43]	Active Learning boost	Generated Dataset	Static	Reducing the training time consumption	Not all outcomes are predictable
[44]	Supervised machine learning	Generated Dataset	Hybrid	Both approaches (Dynamic and Static) enhance performance when run separately.	Time consumption
[45]	Image Visualization	Contagio [27]	Static	Robust to resist reverse mimicry attacks	An evasion test is not available.
[46]	feature-vector generative adversarial network (fvGAN)	generate realistic samples	Static	High evasion rate within a limited time	The complexity depends on the file size, and a 135-dimensional real vector represents each PDF file
[47]	Machine Learning methods and traditional malware analysis procedures	Contagio Malware Dump, PRA Lab.	Hybrid	High-performance results in malware detection and analysis	Time consumption

3. Proposed Classification System

Portable Document Format (PDF) files are one of the most universally used file types. Like other files such as dot-com files, PNG, and Bitcoin, hackers can find means to use these normally harmless PDF files to create security threats via malicious code PDF files. This results in PDF Malware and requires techniques to identify benign files from malicious files. PDF documents have been seized and exploited as a vector for malicious activities. Abundant PDF readers and software are affected incessantly, such as CVE-2018-14442, CVE-2017-10994 in Foxit Reader, and CVE-2018-8350 in Microsoft Windows PDF Library [42]. Recent intelligent detection systems are developed via machine/deep learning techniques [54,55] and blockchain/cryptocurrency techniques [56].

In this section, we present the proposed detection system used to analyze the PDF files to provide insights into the detection model, which classifies the PDF files into either benign or malware. Figure 4 provides the inclusive architecture for the proposed detection system from the input phase to the output phase.

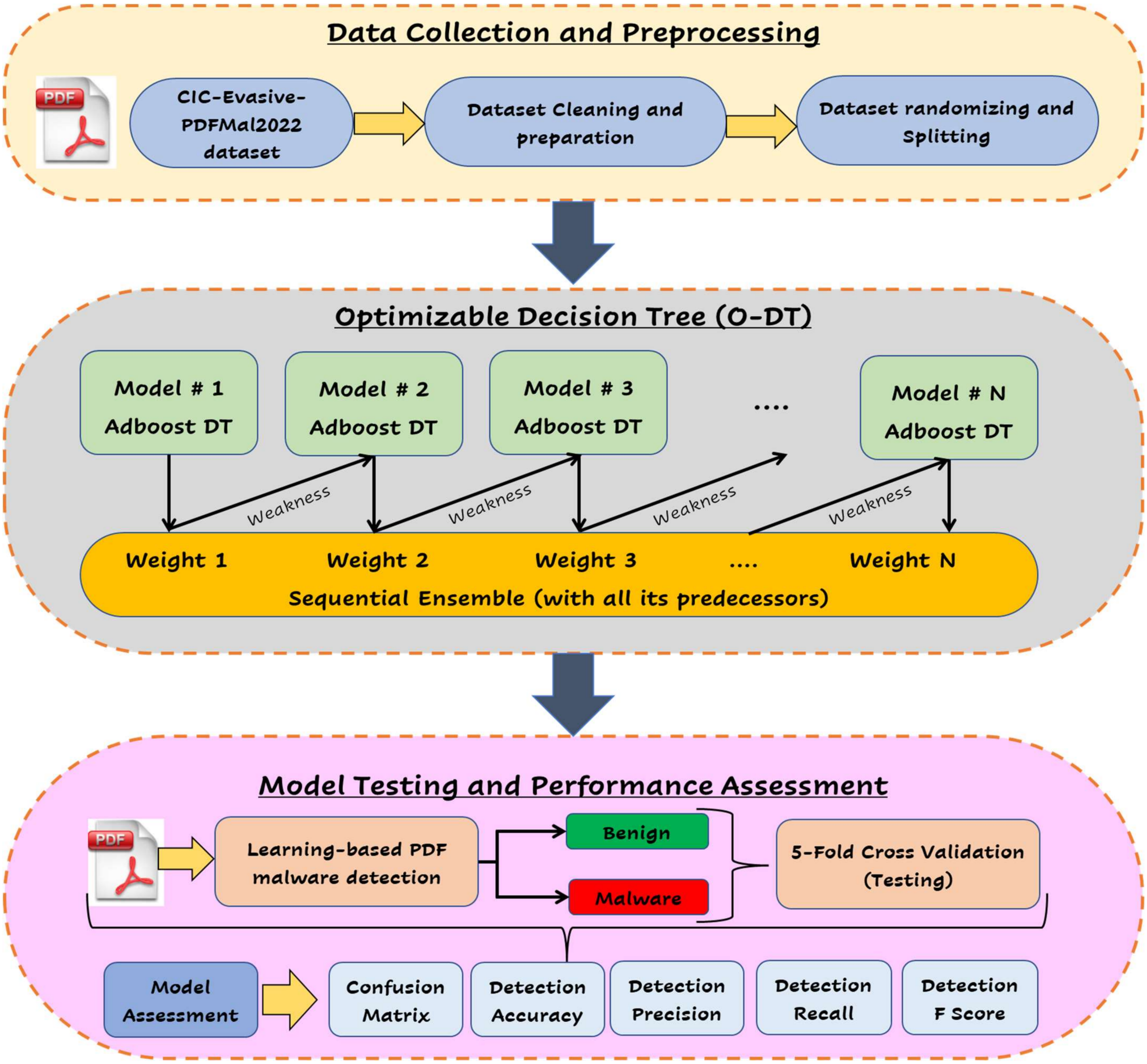

Data Collection and Preprocessing
PDF
CIC-Evasive-PDFMal2022 dataset
Dataset Cleaning and preparation
Dataset randomizing and Splitting
Optimizable Decision Tree (O-DT)
Model # 1 Adboost DT
Model # 2 Adboost DT
Model # 3 Adboost DT
....
Model # N Adboost DT
Weakness
Weakness
Weakness
Weakness
Weight 1
Weight 2
Weight 3
....
Weight N
Sequential Ensemble (with all its predecessors)
Model Testing and Performance Assessment
PDF
Learning-based PDF malware detection
Benign
Malware
5-Fold Cross Validation (Testing)
Model Assessment
Confusion Matrix
Detection Accuracy
Detection Precision
Detection Recall
Detection F Score

Figure 4. Proposed model architecture.

3.1. Data Collection and Preprocessing

Due to their portability, convenience, and dependability, PDF files are the most commonly used document format for several services and applications. However, this reputation and these features of PDFs have attracted black hackers to harness them in various means. Indeed, a variety of significant PDF features can be exploited by attackers to produce a malicious payload. In this paper, we employ a new and thorough PDF dataset, viz. Evasive-PDFMal2022,hat comprises 10,025 records distributed as 4468 benign records and 5557 malicious records. In addition, Evasive-PDFMal2022 comprises 37 significant static features, including 12 general features and 25 structural features extracted from each PDF file [57]. Examples of features include PDF size, title characters, encryption, metadata size, page number, header, image number, text, object number, font objects, number of embedded files, and the average size of all the embedded media.

The collected data of Evasive-PDFMal2022 is imported via MATLAB 2021 to be processed and prepared for use with supervised learning algorithms. Once imported, the dataset originally available as a comma-separated value (.CSV) file format is hosted by MATLAB as a table of records and features. After that, the data tables undergo a number of data cleaning processes, such as fixing incorrect/incomplete records and removing duplicate/erroneous records. Finally, the data are divided into training (70%), validation (10%), and testing (20%) using k-fold cross-validation (with k = 5), as illustrated in Figure 5 below. According to the figure, 20% of the dataset is split out for the final validation of the model. In comparison, 80% of the dataset is used to train and validate the model for several folds. At each fold (say five folds), new random splitting for the 80% is 70% for training and 10% for validation, resulting in five different folds of training and validation sets. For each fold, the model is evaluated, and the final overall performance result is the average of the results attained at all folds. To sum up, for our dataset, ~7000 samples are used for training (70%), ~1000 samples are used for validation (10%), and ~2000 samples are used for testing (20%)

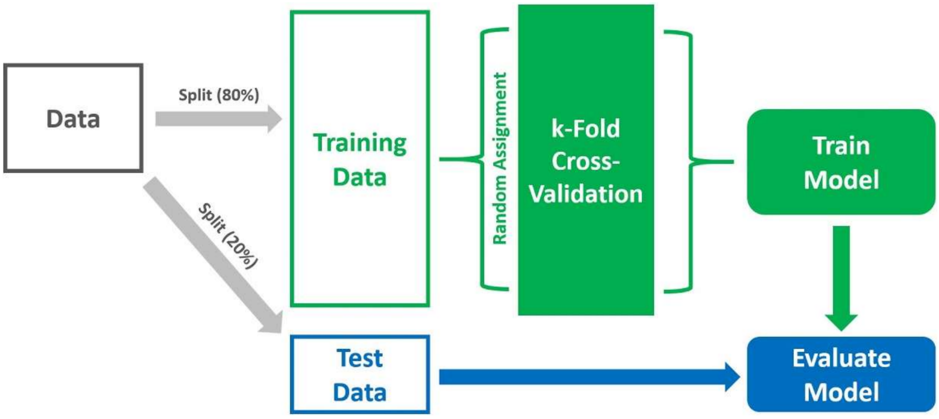


Figure 5. Validation policy: k-fold cross validation.

3.2. Optimizable Decision Tree (O-DT) Model

A decision tree (DT) algorithm is a non-parametric supervised learning method used to perform classification and regression tasks. DT mainly makes use of a probability tree that facilitates the decision making of a specific process and can predict the value of a target variable. For example, the need to decide between two project investment ideas can be done through the decision tree. The main idea of DT is to build a model to learn the decision rules inferred from the data features, which can be used later to make decisions and predictions. An optimizable decision tree (O-DT) is a decision tree that makes use of optimal parameters and hyperparameters to build a detection system by trying a predefined search space for different hyperparameters.

We employed the AdaBoost algorithm in this work to build our decision tree model with various hyperparameter options. AdaBoost (ensemble adaptive boosting ML method) is a boosting approach in which weights are re-allocated to each example, with higher weights allocated to incorrectly classified examples, which helps decrease bias and variance in the learning process. Figure 4 shows how boosting is used in the AdaBoost DT by employing a number of learners expanding sequentially. Apart from the first learner, every successive learner is cultivated from formerly cultivated learners (weak learners are transformed into strong learners). To sum up, Table 2 presents the final optimized hyperparameters for developing O-DT.

Table 2. Optimized parameters for the development of O-DT.

Factor	Description
Preset	Optimizable Tree
Learning algorithm	AdaBoost Tree
Split criterion	Twoing rule
Surrogate decision splits	Off
Maximum number of splits	6704
Optimizer	Random Search
Iterations	30
Training time limit	False
Feature Selection	All features used in the model, No PCA
Cost function	Minimum Classification Error

3.3. Model Testing and Evaluation

Model testing and evaluation is a crucial process for understanding the performance of a machine learning model and gaining more insights into the model's strengths and weaknesses. In this research, we have tested the model using a 5-fold cross-validation and testing dataset (~2000 samples) and evaluated its performance accordingly. Standard assessment metrics have been used to assess the efficacy of the detection model during the training, validation, and testing phases. Figure 6 summarizes the standard performance assessment indicators utilized in this work.

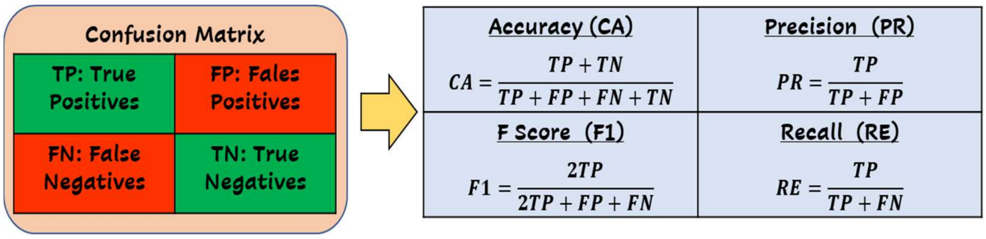


Figure 6. Standard performance assessment indicators.

4. Results and Analysis

This section presents the performance evaluation results for the proposed PDF malware detection system in various indicators. In addition, a comparison with state-of-the-art models is conducted. To begin, Figure 7 trace the trajectory of minimum classification error (MCE) during the training iterations of the optimizable decision trees. According to the figure, the initial recorded MCE after the first iteration is 3.4%, recording a maximum classification accuracy of 96.6%. After that, the MCE sharply decreased toward the minimum MCE hyperparameters only after three learning iterations recording an MCE of 1.3% and classification accuracy of 98.7%. Then, the MCE trajectory continued to slightly decrease towards the best point hyperparameters, where it saturated after iteration 13, recording a 1.16% of MCE with 98.84% of classification accuracy, and it remained constant toward the end of the learning process (30 iterations).

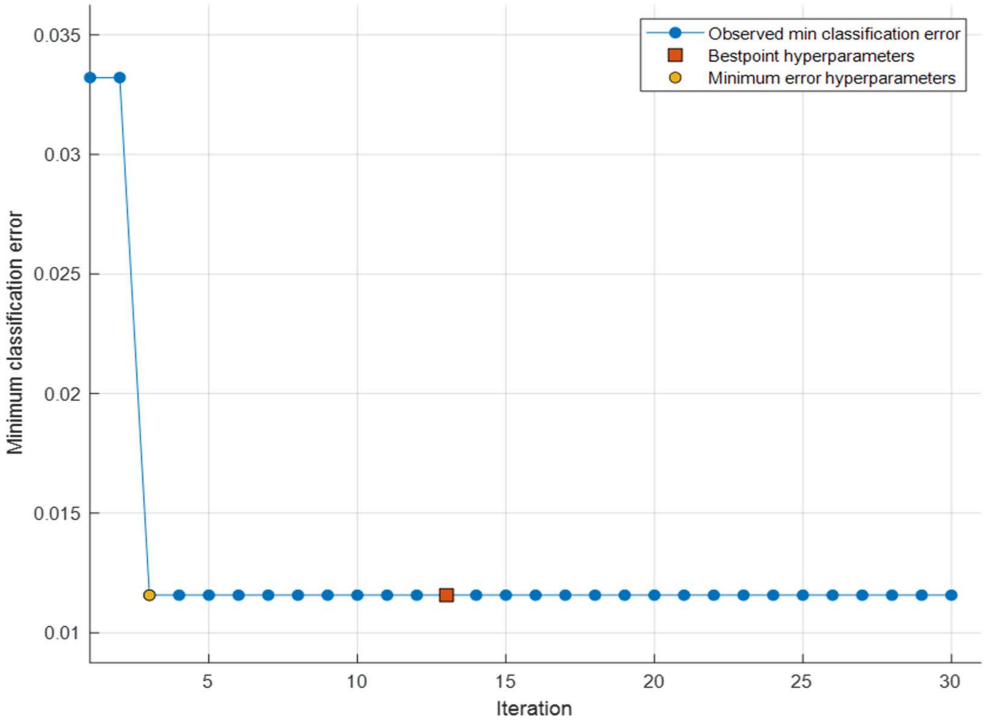
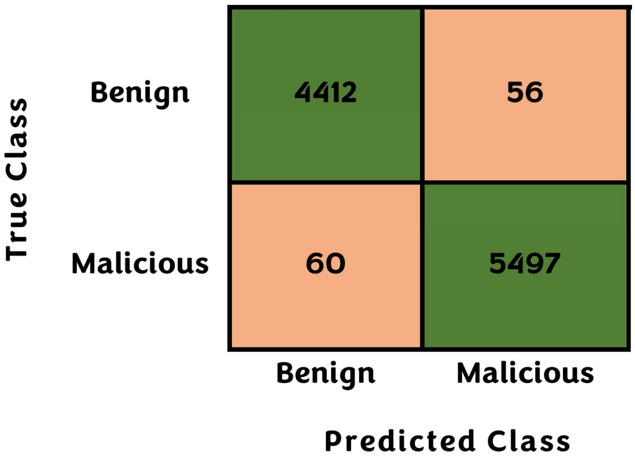


Figure 7. Minimum classification error vs. learning iterations.

In addition, Figure 8 demonstrates the binary confusion matrix results for the proposed PDF malware detection system employing the optimizable decision trees. The presented matrix is composed of blocks: (i) the top left block, which represents the True Positive (TP), expresses the number of samples that are, in reality, benign samples, and which the ML model predicted as benign samples. The Number of TP results in this matrix = 4412. (ii) The top right block, which represents the False Positive (FP), expresses the number of samples that are benign samples, and which the ML model predicted as Malicious samples. The Number of FP results in this matrix = 56. (iii) The bottom left block, which represents the False Negatives (FN), expresses the number of samples that are, in reality malicious samples, and which the ML model predicted as benign samples. The Number of FN results in this matrix = 60. (iv) the bottom right block, which represents the True Negatives (TN), expresses the number of samples that are, in reality malicious samples, and which the ML model predicted as malicious samples. The Number of TN results in this matrix = 5497.



True Class	Benign	Malicious
Benign	4412	56
Malicious	60	5497
	Predicted Class	

Figure 8. Binary confusion matrix results.

Moreover, Table 3 provides a summary of the obtained performance indication results for the proposed PDF malware detection system in terms of various indicators, including Incorrectly Predicted Samples (IPS), Correctly Predicted Samples (CPS), Total Number of Samples (TNS), Prediction speed (PS), Prediction Time (PT), Training time (TT), Prediction Accuracy (CA), Prediction area under the curve (AUC), Prediction Sensitivity/Recall (RE), Prediction Precision (PR), Prediction harmonic average/F-Score (F1), and Balanced Accuracy (BCA). The attained results exhibit high efficiency and detectability for the proposed system, scoring high-performance factors exceeding 98.80% for the system accuracy, sensitivity, and precision.

Table 3. Summary of experimental evaluation factors for the proposed system.

Factor	Value	Factor	Value
IPS	116 samples	CA	98.84%
CPS	9909 samples	AUC	99.00%
TNS	10,025 Samples	RE	98.90%
PS	~ 460,000 obs/sec	PR	98.80%
PT	2.174 µSec	F1	98.85%
TT	11.848 s	BCA	98.95%

Lastly, Table 4 contrasts the performance of our proposed model with several other existing models in the same field of study. The table compareseight8 PDF malware detection systems employing diverse learning models, including Zhang. et al. [58] employing a multi-layer perceptron neural network (MLP-NN), Jiang et al. [34] employing a semi-supervised learning algorithm (Semi-SL), Li et al. [59] employing an intelligent tool known as JSUNPACK, Nissim et al. [60] employing the support vector machine technique (SVM), Mohammed et al. [61] employing deep ResNet-50 convolutional neural network (ResNet-50 CNN), Nataraj et al. [62] employing random forest classifier (RFC), Lakshmanan et al. [63] employing voting ensemble classifier (VEC) that uses random forest classifier (RFC) and k-nearest neighbor (kNN), and Cohen et al. [64] employing support vector machine technique (SVM). In addition to the learning model factor, we have considered four performance factors to compare with existing models: accuracy, precision, sensitivity, and F-Score. Overall, the proposed system is superior in all evaluation factors, with noticeable performance for the other models based on the SVM technique.

Table 4. Comparison with state-of-the-art models in the same area of study.

Ref.	Model	Accuracy	Precision	Sensitivity	F Score
Zhang. et al. [58]/2018	MLP-NN	-	-	95.12%	-
Jiang et al. [34]/2021	Semi-SL	94.00%	-	-	-
Li et al. [59]/2017	JSUNPACK	95.11%	97.57%	90.87%	94.10%
Nissim et al. [60]/2014	SVM-Margin	-	-	97.70%	-
Mohammed et al. [61]/2021	ResNet-50 CNN	89.56%	-	-	-
Nataraj et al. [62]/2020	RFC	96.94%	-	-	-
Lakshmanan et al. [63]/2020	VEC	95.93%	-	-	-
Cohen et al. [64]/2019	SVM-Margin	-	-	96.90%	-
Proposed	O-DT	98.84%	98.80%	98.90%	98.80%

5. Conclusions and Remarks

Due to the worldwide trend toward digital transformation and remote work, the demand for digital documentation has significantly increased. This increase in the use of digital documents has been obviously accompanied by a counter increase in malware development that can threaten user files and machines. PDF files are among the most commonly used digital files worldwide, which makes them highly vulnerable to a wide range of threats and malicious codes. Such infection vectors (developed by the hackers) hide embedded malicious code in the PDF documents to infect the victims' machines. This results in PDF malware and requires techniques to identify benign files from malicious files. Therefore, a new intelligent system for PDF malware detection is proposed, developed, and evaluated in this paper. The proposed system utilized a high-performance machine learning model employing optimizable decision trees with the AdaBoost algorithm. The proposed system was trained and evaluated on a new inclusive dataset for PDF documents known as Evasive-PDFMal2022. The simulation outcomes showed the superiority of the proposed system in terms of detection accuracy, precision, sensitivity, F-Score, and detection overhead. To this end, the proposed model outperforms other state-of-the-art models in the same study area. The proposed model can be generalized and applied to provide several detection services in various areas [65,66]. In short, the contribution of this work can be summarized as follows:

- A comprehensive machine-learning-based model for analyzing PDF documents to identify the malicious PDF files from benign files.
- The proposed model makes use of optimizable decision trees with the AdaBoost algorithm and optimal hyperparameters.
- The proposed model relies on the utilization of a new dataset (Evasive-PDFMal2022) composed of 10,025 records distributed and 37 significant static features (general and structural features) extracted from each PDF file.
- The experimental results proved the efficiency of the proposed PDF detection system, realizing a 98.84% prediction accuracy with a short prediction interval of 2.174 µSec.
- The discussion indicated some gaps in the current state-of-the-art methods which can provide directions for future research.

In future, some other important and commonly used documents can be considered in the malicious detection process such as office documents (.docx, .xlxs, pptx., … etc.). In addition, deeper neural networks can be employed in case of accurate changes in the malware that requires deep learning algorithms such as convolutional neural network (CNN) and long short-term memory (LSTM).

Author Contributions: Conceptualization, Q.A.A.-H.; methodology, Q.A.A.-H. and A.O.; software, Q.A.A.-H., A.O. and H.Q.; validation Q.A.A.-H., A.O. and H.Q.; formal analysis, Q.A.A.-H.; investigation, Q.A.A.-H. and A.O.; resources, Q.A.A.-H.; data curation, Q.A.A.-H.; writing—original draft preparation, Q.A.A.-H., A.O. and H.Q.; writing—review and editing, Q.A.A.-H., A.O. and H.Q.; visualization, Q.A.A.-H., A.O. and H.Q.; project administration, Q.A.A.-H.; funding acquisition, Q.A.A.-H., A.O. and H.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data associated with this research can be retrieved from: https://www.unb.ca/cic/datasets/pdfmal-2022.html, accessed on 19 September 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ndibanje, B.; Kim, K.H.; Kang, Y.J.; Kim, H.H.; Kim, T.Y.; Lee, H.J. Cross-method-based analysis and classification of malicious behavior by API calls extraction. *Appl. Sci.* **2019**, *9*, 239. [CrossRef]
2. Abu Al-Haija, Q.; Al Badawi, A.; Bojja, G.R. Boost-Defence for resilient IoT networks: A head-to-toe approach. *Expert Syst.* **2022**, e12934. [CrossRef]
3. Ali, M.; Shiaeles, S.; Bendiab, G.; Ghita, B. MALGRA: Machine learning and N-gram malware feature extraction and detection system. *Electronics* **2020**, *9*, 1777. [CrossRef]
4. Faruk, M.J.H.; Shahriar, H.; Valero, M.; Barsha, F.L.; Sobhan, S.; Khan, M.A.; Whitman, M.; Cuzzocrea, A.; Lo, D.; Rahman, A.; et al. Malware detection, and prevention using artificial intelligence techniques. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021.
5. Ghanei, H.; Manavi, F.; Hamzeh, A. A novel method for malware detection based on hardware events using deep neural networks. *J. Comput. Virol. Hacking Tech.* **2021**, *17*, 319–331. [CrossRef]

6. Atkinson, S.; Carr, G.; Shaw, C.; Zargari, S. Drone forensics: The impact and challenges. In *Digital Forensic Investigation of Internet of Things (IoT) Devices*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 65–124.
7. Liu, C.; Lou, C.; Yu, M.; Yiu, S.M.; Chow, K.P.; Li, G.; Jiang, J.; Huang, W. A novel adversarial example detection method for malicious PDFs using multiple mutated classifiers. *Forensic Sci. Int. Digit. Investig.* **2021**, *38*, 301124. [CrossRef]
8. Al-Haijaa, Q.A.; Ishtaiwia, A. Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2021**, *11*, 1688–1695. [CrossRef]
9. Livathinos, N.; Berrospi, C.; Lysak, M.; Kuropiatnyk, V.; Nassar, A.; Carvalho, A.; Dolfi, M.; Auer, C.; Dinkla, K.; Staar, P. Robust PDF document conversion using recurrent neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021.
10. Wiseman, Y. Efficient embedded images in portable document format. *Int. J.* **2019**, *124*, 129–138. [CrossRef]
11. Ijaz, M.; Durad, M.H.; Ismail, M. Static and dynamic malware analysis using machine learning. In Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019.
12. Chakkaravarthy, S.S.; Sangeetha, D.; Vaidehi, V. A malware analysis, and mitigation techniques survey. *Comput. Sci. Rev.* **2019**, *32*, 1–23. [CrossRef]
13. Abdelsalam, M.; Gupta, M.; Mittal, S. Artificial intelligence assisted malware analysis. In Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, Virtual Event, 28 April 2021.
14. Or-Meir, O.; Nissim, N.; Elovici, Y.; Rokach, L. Dynamic malware analysis in the modern era—A state of the art survey. *ACM Comput. Surv.* **2019**, *52*, 1–48. [CrossRef]
15. Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. *Appl. Sci.* **2022**, *12*, 5015. [CrossRef]
16. Wang, W.; Shang, Y.; He, Y.; Li, Y.; Liu, J. BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. *Inf. Sci.* **2020**, *511*, 284–296. [CrossRef]
17. Abu Al-Haija, Q.; Al-Saraireh, J. Asymmetric Identification Model for Human-Robot Contacts via Supervised Learning. *Symmetry* **2022**, *14*, 591. [CrossRef]
18. Al-Haija, Q.A.; Gharaibeh, M.; Odeh, A. Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning. *AI* **2022**, *3*, 303–317. [CrossRef]
19. Yang, L.; Ciptadi, A.; Laziuk, I.; Ahmadzadeh, A.; Wang, G. BODMAS: An open dataset for learning based temporal analysis of PE malware. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 21–27 May 2021.
20. Maiorca, D.; Biggio, B. Digital investigation of pdf files: Unveiling traces of embedded malware. *IEEE Secur. Priv.* **2019**, *17*, 63–71. [CrossRef]
21. Wu, Y.X.; Wu, Q.B.; Zhu, J.Q. Data-driven wind speed forecasting using deep feature extraction and LSTM. *IET Renew. Power Gener.* **2019**, *13*, 2062–2069. [CrossRef]
22. Shijo, P.; Salim, A. Integrated static and dynamic analysis for malware detection. *Procedia Comput. Sci.* **2015**, *46*, 804–811. [CrossRef]
23. Al-Haija, A.Q. Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. Front. *Big Data* **2022**, *4*, 782902. [CrossRef]
24. Shafiq, M.Z.; Khayam, S.A.; Farooq, M. Embedded malware detection using Markov n-grams. In Proceedings of the International Conference on Detection of Intrusions and Malware and Vulnerability Assessment, Paris, France, 10–11 July 2008.
25. Tabish, S.M.; Shafiq, M.Z.; Farooq, M. Malware detection using statistical analysis of byte-level file content. In Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics, Paris, France, 28 June 2009.
26. Smutz, C.; Stavrou, A. Malicious PDF detection using metadata and structural features. In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando, FL, USA, 3–7 December 2012.
27. Contagio, M.P. 2011. Available online: http://contagiodump.blogspot.com/2010/08/malicious-documents-archive-for.html (accessed on 2 September 2022).
28. Falah, A.; Pan, L.; Huda, S.; Pokhrel, S.R.; Anwar, A. Improving malicious PDF classifier with feature engineering: A data-driven approach. *Future Gener. Comput. Syst.* **2021**, *115*, 314–326. [CrossRef]
29. Al-Haija, Q.A.; Nasr, K.A. Supervised Regression Study for Electron Microscopy Data. In Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 18–21 November 2019; pp. 2661–2668. [CrossRef]
30. Smutz, C.; Stavrou, A. When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. In Proceedings of the The Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 21–24 February 2016.
31. Abu Al-Haija, Q. A Stochastic Estimation Framework for Yearly Evolution of Worldwide Electricity Consumption. *Forecasting* **2021**, *3*, 256–266. [CrossRef]
32. Corona, I.; Maiorca, D.; Ariu, D.; Giacinto, G. Lux0r: Detection of malicious pdf-embedded javascript code through discriminant analysis of API references. In Proceedings of the 2014 Workshop on Artificial Intelligence and Security Workshop, New York, NY, USA, 7 November 2014.
33. Maiorca, D.; Giacinto, G.; Corona, I. A pattern recognition system for malicious pdf file detection. In Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition, New York, NY, USA, 15–19 July 2012.

34. Li, M.; Liu, Y.; Yu, M.; Li, G.; Wang, Y.; Liu, C. FEPDF: A robust feature extractor for malicious PDF detection. In Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia, 1–4 August 2017; pp. 218–224. [CrossRef]
35. Li, K.; Gu, Y.; Zhang, P.; An, W.; Li, W. Research on KNN algorithm in malicious PDF file classification under adversarial environment. In Proceedings of the 2019 4th International Conference on Big Data and Computing, Guangzhou, China, 10–12 May 2019.
36. Sayed, S.G.; Shawkey, M. Data mining-based strategy for detecting malicious PDF files. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security, and Privacy in Computing and Communica-tions/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018.
37. Cuan, B.; Damien, A.; Delaplace, C.; Valois, M. Malware detection in pdf files using machine learning. In Proceedings of the SECRYPT 2018-15th International Conference on Security and Cryptography, Porto, Portugal, 26–28 July 2018.
38. Badawi, A.A.; Al-Haija, Q.A. Detection of money laundering in bitcoin transactions. In Proceedings of the 4th Smart Cities Symposium (SCS 2021), Online Conference, Bahrain, 21–23 November 2021; pp. 458–464. [CrossRef]
39. Kang, A.R.; Jeong, Y.S.; Kim, S.L.; Woo, J. Malicious PDF detection model against adversarial attack built from benign PDF containing javascript. *Appl. Sci.* **2019**, *9*, 4764. [CrossRef]
40. He, K.; Zhu, Y.; He, Y.; Liu, L.; Lu, B.; Lin, W. Detection of Malicious PDF Files Using a Two-Stage Machine Learning Algorithm. *Chin. J. Electron.* **2020**, *29*, 1165–1177. [CrossRef]
41. Adhatarao, S.; Lauradoux, C. Robust PDF files forensics using coding style. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Copenhagen, Denmark, 13–15 June 2022; Springer: Berlin/Heidelberg, Germany, 2022.
42. Singh, P.; Tapaswi, S.; Gupta, S. Malware detection in pdf and office documents: A survey. *Inf. Secur. J. A Glob. Perspect.* **2020**, *29*, 134–153. [CrossRef]
43. Chen, Y.; Wang, S.; She, D.; Jana, S. On training robust {PDF} malware classifiers. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Berkeley, CA, USA, 12–14 August 2020; pp. 2343–2360.
44. Li, Y.; Wang, X.; Shi, Z.; Zhang, R.; Xue, J.; Wang, Z. Boosting training for PDF malware classifier via active learning. *Int. J. Intell. Syst.* **2022**, *37*, 2803–2821. [CrossRef]
45. Santos, I.; Devesa, J.; Brezo, F.; Nieves, J.; Bringas, P.G. Open A static-dynamic approach for machine-learning-based malware detection. In Proceedings of the International Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions, Ostrava, Czech Republic, 5–7 September 2012; Springer: Berlin/Heidelberg, Germany, 2013; pp. 271–280.
46. Corum, A.; Jenkins, D.; Zheng, J. Robust PDF malware detection with image visualization and processing techniques. In Proceedings of the 2019 2nd International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 28–30 June 2019; pp. 108–114.
47. Li, Y.; Wang, Y.; Wang, Y.; Ke, L.; Tan, Y.A. A feature-vector generative adversarial network for evading PDF malware classifiers. *Inf. Sci.* **2020**, *523*, 38–48. [CrossRef]
48. Tay, K.Y.; Chua, S.; Chua, M.; Balachandran, V. Towards Robust Detection of PDF-based Malware. In Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, Baltimore, MD, USA, 24–27 April 2022; pp. 370–372.
49. Maiorca, D.; Biggio, B.; Giacinto, G. Towards adversarial malware detection: Lessons learned from PDF-based attacks. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–36. [CrossRef]
50. VX Heavens Virus Collection, VX Heavens Website. Available online: http://vx.netlux.org (accessed on 21 July 2022).
51. Available online: https://www.virustotal.com/gui/home/upload (accessed on 19 September 2022).
52. Contaigo, 16,800 Clean and 11,960 Malicious Files for Signature Testing and Research. 2013. Available online: http://contagiodump.blogspot.com/2013/03/16800-clean-and-11960-malicious-files.html (accessed on 19 September 2022).
53. Available online: https://hal.archives-ouvertes.fr/ (accessed on 19 September 2022).
54. Abu Al-Haija, Q.; Al-Dala'ien, M. ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks. *J. Sens. Actuator Netw.* **2022**, *11*, 18. [CrossRef]
55. Abu Al-Haija, Q.; Al Badawi, A. High-performance intrusion detection system for networked UAVs via deep learning. *Neural Comput. Appl.* **2022**, *34*, 10885–10900. [CrossRef]
56. Odeh, A.; Keshta, I.; Al-Haija, Q.A. Analysis of Blockchain in the Healthcare Sector: Application and Issues. *Symmetry* **2022**, *14*, 1760. [CrossRef]
57. PDF Dataset. CIC-Evasive-PDFMal2022. Canadian Institute for Cybersecurity (CIC). 2022. Available online: https://www.unb.ca/cic/datasets/pdfmal-2022.html (accessed on 1 June 2022).
58. Zhang, J. MLPdf: An Effective Machine Learning Based Approach for PDF Malware Detection. Cryptography and Security (cs.CR). *arXiv* **2018**, arXiv:1808.06991.
59. Jiang, J.; Song, N.; Yu, M.; Chow, K.P.; Li, G.; Liu, C.; Huang, W. Detecting Malicious PDF Documents Using Semi-Supervised Machine Learning. In *Proceedings of the Advances in Digital Forensics XVII. Digital Forensics 2021, Virtual Event, 1–2 February 2021*; IFIP Advances in Information and Communication Technology; Peterson, G., Shenoi, S., Eds.; Springer: Cham, Switzerland, 2021; Volume 612. [CrossRef]
60. Nissim, N.; Cohen, A.; Moskovitch, R.; Shabtai, A.; Edry, M.; Bar-Ad, O.; Elovici, Y. ALPD: Active Learning Framework for Enhancing the Detection of Malicious PDF Files. In Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference, Washington, DC, USA, 24–26 September 2014; pp. 91–98. [CrossRef]

61. Mohammed, T.M.; Nataraj, L.; Chikkagoudar, S.; Chandrasekaran, S.; Manjunath, B. Malware detection using frequency domain-based image visualization and deep learning. In Proceedings of the 54th Hawaii International Conference on System Sciences, Grand Wailea, HI, USA, 5–8 January 2021; p. 7132.
62. Nataraj, L.; Manjunath, B.S.; Chandrasekaran, S. Malware Classification and Detection Using Audio Descriptors. U.S. Patent 11244050B2, 4 June 2020.
63. Nataraj, L.; Mohammed, T.M.; Nanjundaswamy, T.; Chikkagoudar, S.; Chandrasekaran, S.; Manjunath, B.S. OMD: Orthogonal Malware Detection using Audio, Image, and Static Features. In Proceedings of the MILCOM 2021–2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA, 29 November–1 December 2021.
64. Cohen, A.; Nissim, N.; Wu, J.; Lanzi, A.; Rokach, L.; Elovici, Y.; Giles, L. Sec-Lib: Protecting Scholarly Digital Libraries From Infected Papers Using Active Machine Learning Framework. *IEEE Access* **2019**, *7*, 110050–110073. [CrossRef]
65. Al-Haija, Q.A.; Saleh, E.; Alnabhan, M. Detecting Port Scan Attacks Using Logistic Regression. In Proceedings of the 2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Casablanca, Morocco, 15–17 March 2021; pp. 1–5. [CrossRef]
66. Abu Al-Haija, Q.; Krichen, M. A Lightweight In-Vehicle Alcohol Detection Using Smart Sensing and Supervised Learning. *Computers* **2022**, *11*, 121. [CrossRef]