Journal of Cyber Security

Tech Science Press

DOI: 10.32604/jcs.2023.042501

ARTICLE

Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training and Testing Criteria

Bilal Khan[1], Muhammad Arshad[2] and Sarwar Shah Khan[3,4,*]

[1]Department of Computer Science, City University of Science and Information Technology, Peshawar, Pakistan

[2]Department of Computer Software Engineering, University of Engineering and Technology, Mardan, Pakistan

[3]Department of Computer and Software Technology, University of Swat, Swat, Pakistan

[4]Department of Computer Science, IQRA National University, Swat, Pakistan

*Corresponding Author: Sarwar Shah Khan. Email: sskhan0092@gmail.com

Received: 01 June 2023 Accepted: 03 August 2023 Published: 21 August 2023

ABSTRACT

The proliferation of maliciously coded documents as file transfers increase has led to a rise in sophisticated attacks. Portable Document Format (PDF) files have emerged as a major attack vector for malware due to their adaptability and wide usage. Detecting malware in PDF files is challenging due to its ability to include various harmful elements such as embedded scripts, exploits, and malicious URLs. This paper presents a comparative analysis of machine learning (ML) techniques, including Naive Bayes (NB), K-Nearest Neighbor (KNN), Average One Dependency Estimator (A1DE), Random Forest (RF), and Support Vector Machine (SVM) for PDF malware detection. The study utilizes a dataset obtained from the Canadian Institute for Cyber-security and employs different testing criteria, namely percentage splitting and 10-fold cross-validation. The performance of the techniques is evaluated using F1-score, precision, recall, and accuracy measures. The results indicate that KNN outperforms other models, achieving an accuracy of 99.8599% using 10-fold cross-validation. The findings highlight the effectiveness of ML models in accurately detecting PDF malware and provide insights for developing robust systems to protect against malicious activities.

KEYWORDS

Cyber-security; PDF malware; model training; testing

1 Introduction

Recent years have seen a sharp rise in sophisticated assaults using maliciously coded documents as file transfers increase. Executable files that are attached to emails or webpages can be dangerous, as most Internet users are aware. Nevertheless, the papers are a useful tool for distributing malware because people are ignorant of them. The major attack vector for malware that has been detected is the PDF, which is much adaptable than other document formats. Malicious PDF files frequently contain JavaScript or binary scripts that take advantage of security weaknesses to do damaging actions [1]. There are uncountable PDF files online. Some are not as innocuous as one may think. In reality, PDF files may contain a wide range of items, such as JavaScript or binary code. These

things might occasionally be dangerous. Since Portable Document Format files can include a variety of harmful material, including embedded scripts, exploits, and malicious URLs, it can be difficult to detect malware in them. A reading flaw might be used by malware software to try to infect a machine [2]. Adobe Acrobat Reader discovered a huge number of vulnerabilities in 2017. Every reader has particular flaws, and a malicious PDF file could be able to exploit them [3]. Offices frequently use the PDF file format due to its great efficiency, reliability, and interaction. The emergence of more advanced, non-executable file-based attack technologies and techniques has made PDF security more challenging because spiteful PDF files are the commonly explore infection vectors in hostile circumstances [4,5]. PDF malware detection is very important due to several reasons including:

Protection against Malicious Content: PDF files are often utilized for document sharing and can include a variety of embedded content types, including JavaScript, links, and multimedia components. These characteristics can be used by malicious actors to embed malware into PDF files, potentially making them a vehicle for virus delivery. Finding PDF malware helps users avoid unintentionally accessing or running dangerous files [6].

Preventing Exploits: Vulnerabilities in PDF reader software and other applications that work with PDF files can be exploited using PDF files. Malicious PDFs may include exploits that use security flaws to access systems without authorization or run malicious malware. For computer systems and networks to remain secure, these attacks must be found and stopped [7].

Protecting Sensitive Information: PDF files are frequently used to store and distribute sensitive information, such as financial information, intellectual property, or personal particulars. This sensitive information may be stolen or leaked by malware that is included in PDF files, which might result in monetary loss, data breaches, or identity theft. Protecting the security and integrity of sensitive data is made easier by finding and eliminating malware from PDF files [8].

Attacks Using Social Engineering: To deceive users into opening infected PDF files, malicious actors frequently utilize social engineering tactics. These files could have alluring subject lines or messages, or they might be presented as actual papers. Finding PDF malware shields consumers from these social engineering scams and guards against the potential loss of money, reputation, or operational efficiency [9].

System Security Overall: Malware attacks can have serious effects on the safety and functionality of computer systems. System crashes, data damage, unauthorized access, and the installation of new malware are all possible consequences of malware. Maintaining the overall security and stability of computer systems and networks involves finding and eliminating PDF malware [10].

The motivation for this research stems from the need to develop effective methods for protecting against sophisticated attacks using PDF files. The authors highlight the importance of PDF malware detection for several reasons. Firstly, detecting malware in PDF files helps protect users from unintentionally accessing or running dangerous files, safeguarding them against potential harm. Secondly, vulnerabilities in PDF reader software and other applications can be exploited through malicious PDF files, making it crucial to identify and prevent such attacks. Thirdly, PDF files often contain sensitive information that can be stolen or leaked by malware, leading to financial loss, data breaches, or identity theft. Detecting and eliminating malware from PDF files helps protect the security and integrity of sensitive data. Lastly, malicious actors often use social engineering tactics to trick users into opening infected PDF files, and detecting PDF malware can mitigate the risks associated with such attacks.

Keeping all these important in mind, researchers have proposed a variety of models to distinguish numerous attacks connected to PDF files as a result of the growth of ML technology in recent years

[11,12]. However, this study presents the analysis of various ML models which are "Average One Dependency Estimator (A1DE), K-Nearest Neighbor (KNN), Support Vector Machine (SVM) [13], Naive Bayes (NB), and Random Forest (RF)" [14]. Based on F1-Score, precision, recall and accuracy, these models are contrasted. The primary objective of this study is to develop a malware detection model capable of safeguarding systems against harmful actions caused by PDF viruses.

The remaining sections of this study are organized as follows: The literature study is summarized in Section 2, the technique is covered in Sections 3, and 4 the inquiry is concluded in Section 5.

2 Literature Review

Using countless ML and DL models, several varieties of research have been managing on the identification of PDF malware. Kang et al. described the use of the PDF in 2019 [15]. They gave a thorough analysis of the JavaScript structure and content in the PDF with embedded XML. They then build a variety of features, such as configuration encoding methods for material and variables like file size, keywords, versions, and JavaScript readable strings. Information about file size, category, and content properties, additionally item names, keywords, and JavaScript readable strings. The approaches to training resilient PDF malware classifiers utilizing observable robustness features were described by Chen et al. in 2019 [16]. For instance, with no substance on how countless pages of innocuous forms are included in the document, the classifier must identify PDF malware as harmful. They demonstrate how to accurately evaluate the worst-case behavior of a malware classifier concerning particular robustness properties.

In several studies, ML approaches have been utilized to develop classifiers for PDF malware. Two prior initiatives that focused on the hazardous JavaScript that was presented in Portable Document Format malware were Wepawet [17] and Laskov et al. [18].

Based on the lexical features of JavaScript scripts as well as functions, constants, objects, techniques, and keywords, Khitan et al. [19] proposed attributes. Zhang et al. [20] merged the JavaScript object count, page count, and stream filtering data with the PDF structure, entity characteristics, metadata information, and content statistics. Following the revelation that malicious JavaScript functions differently from legitimate JavaScript code. Liu et al. [21] suggested a context-aware approach. This approach involves utilizing the original JavaScript code as input to the "eval" function to open the PDF file, while closely monitoring for any unusual behavior based on the given instructions.

According to Herrera-Silva et al. [22], Cyberattacks using ransomware have increased over the past ten years, causing great concern among organizations. It's critical to develop novel and enhanced techniques for detecting this type of virus. This work employs machine learning and dynamic analysis to identify the ransomware signatures that are always evolving using a few dynamic variables. This study can be utilized to identify current and even novel versions of the threat because the majority of the characteristics are shared by a variety of ransom ware-affected samples.

Dhalaria et al. present a hybrid method for detecting and classifying Android malware [23]. The proposed method combines static and dynamic analysis techniques to effectively identify malicious applications and classify them into different malware families. The authors train machine learning models for malware detection and family classification using features taken from both the static and dynamic behaviours of Android apps. Experimental results demonstrate the effectiveness of the hybrid approach in accurately detecting and classifying Android malware, thereby contributing to the field of mobile security and aiding in the prevention of malicious activities on Android devices. However, Deore et al. presented a novel approach for detecting malware using a Faster Region Proposals

Convolutional Neural Network (FRCNN) [24]. The proposed MDFRCNN model aims to increase the accuracy and efficiency of malware detection by effectively identifying and classifying malicious regions within digital content. The authors conduct experiments and evaluate the performance of their model using various datasets, demonstrating its effectiveness in detecting malware in real-world scenarios.

3 Methodology

This study focuses on the comparison of various ML models and model training criteria to find a better solution for PDF malware detection. ML models include A1DE, NB, KNN, RF, and SVM while training criteria include the percentage splitting with 70% and 30% for training and testing respectively, and 10-fold cross-validation. The overall methodology is presented in Fig. 1.

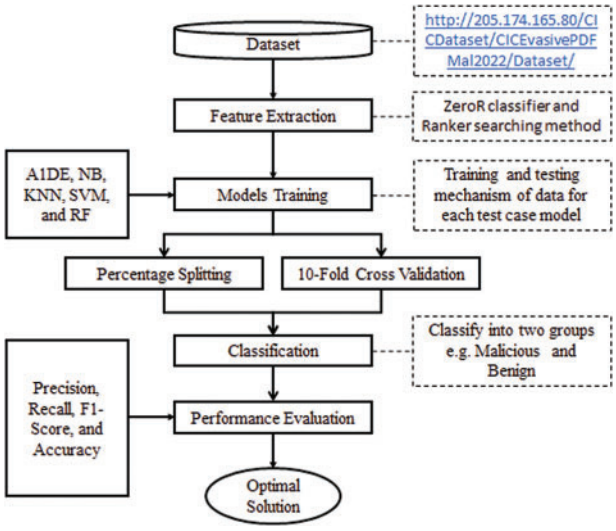


Figure 1: Methodology flow chart

3.1 Dataset Explanation and Preprocessing

We have collected the PDF Malware detection dataset from Canadian Institute for Cyber-security: https://www.unb.ca/cic/datasets/pdfmal-2022.html. The dataset has 33 characteristics, 32 of which are independent, and 1 of which is dependent. The first 11 characteristics were eliminated since they had no effect during the analysis stage. These characteristics are collectively referred to as general features, and they comprise the following information: "Encryption, metadata size, page number, header, picture number, text, object number, font objects, number of embedded files, and average size of all embedded media are all factors to consider". The data is cleaned and no need for further preprocessing.

For further analysis, there is a need to select some features that are best suited for the analysis. To this end, we select some features from Structural features which define the PDF file in terms of its structure, which necessitates more thorough processing and reveals information about the PDF's general framework.

We have employed Classifier Attribute Evaluator techniques employing the ZeroR classifier and the Ranker searching method for retrieving such functions. For accuracy estimation, the number of folds used is 5. The selected features are ranked as:

Selected attributes: “21,7,8,10,6,5,4,3,2,9,11,20,18,19,12,17,16,15,14,13,1: 21”.

The following attributes are present, in that order: Colours, encrypt, JS, XFA startxref, trailer, xref, endstream, stream, endobj, launch, OpenAction, AA, EmbeddedFile, JBIG2Decode, Acroform, pageno, ObjStm, Javascript, RichMedia, and obj. Table 1 represents the selected features with their descriptions.

Table 1: Selected features and description

S. No.	Feature	Description
1	Xref	The stream’s size, as harmful code may be concealed within streams.
2	Trailer	Number of trailers inside the PDF.
3	Pageno	Malicious PDF files often include fewer pages—often just one blank page—because they don’t care how their material is presented.
4	Stream	This displays how many binary data sequences there are in the PDF.
5	Encrypt	This function indicates whether or not the PDF file is password-protected.
6	Objstm	Streams that contain additional objects.
7	Endstream	Keywords that signify the streams’ termination.
8	JS	Several objects encompassing Javascript code.
9	Obj	This might be a sign of an attempt to obfuscate.
10	Javascript	This shows how many things include Javascript code, the most common type of characteristic.
11	AA	Specifies a specific action upon an incident.
12	OpenAction	When a PDF file is opened, this property specifies what action should be done. The majority of commonly encountered malicious PDF files employ this feature in combination with Javascript.
13	endobj	PDFs enable a wide range of obfuscation methods, including string obfuscations in hex, octal, etc., that are frequently employed in evasion strategies.
14	Acroform	Form fields in PDF files created with Acrobat contain scripting that hackers might use against you.
15	Startxref	Numerous keywords that include “startxref” indicate the location of the Xref table’s start.
16	JBIG2Decode	JBig2Decode is a well-liked filter for encoding hazardous data. What items have the most number of nested filters? Nested filters can impede decoding and suggest evasion.
17	Richmeddia	The number of rich media keywords shows the amount of flash and embedded media.
18	Launch	The phrase “launch” refers to the act of executing a command or programme.
19	EmbeddedFile	It is possible for PDF files to attach or embed other things, such as word documents, photos, and more, which can be used maliciously.
20	XFA	XFAs, an XML form architecture that permits scripting technologies that attackers may exploit, are found in some PDF files.

(Continued)

Table 1 (continued)

S. No.	Feature	Description
21	Color	Many color schemes are used in the PDF.
22	Class	Categorize as benign or malicious.

Due to its mobility, PDF has become the most often used document format throughout time. Unfortunately, the ubiquity of PDFs and their sophisticated capabilities have made it possible for attackers to use them in a variety of ways. An attacker can take advantage of several crucial PDF properties to spread a malicious payload. This dataset, which comprises 10,019 records in which 5551 malicious and 4468 are benign which incline to evade the mutual significant features discovered in every class, collects these malicious data and information.

3.2 Model Training and Performance Evaluation

This study focuses on two different types of testing analysis; one is based on percentage splitting of the dataset where we have used 70% for training and the rest 30% for testing, and the second method is K-fold cross-validation in which we have selected the value of K as 10. This study presents a comparison of these testing methods. The ML techniques are compared using standard evaluation metrics such as F1-score, precision, recall, and accuracy. These measures can be calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

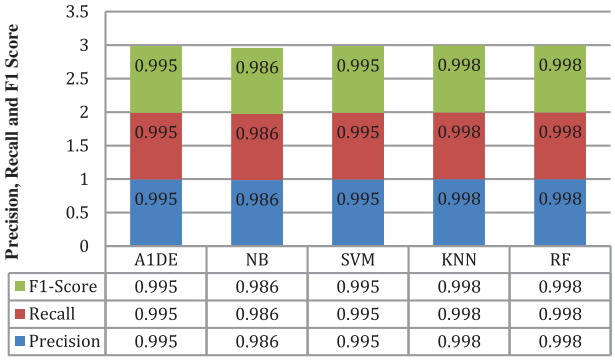
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Here, the true positive values are presented with TP, false positive values are presented with FP, while TN and FN present the values of true negative and false negative calculations.

4 Results Analysis and Discussion

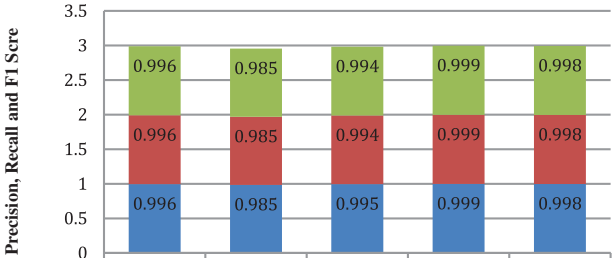
This section presents the outcomes achieved via the aforementioned ML models including A1De, NB, SVM, KNN, and RF. These models are evaluated using f1-score, precision, recall, and accuracy. This study also focuses on different types of testing criteria which are percentage splitting and K-fold cross-validation. For percentage splitting, we have used 70% percent for training and the remaining 30% for testing while in the K-fold, we have selected the value of K as 10. Fig. 2 presents the precision, recall, and f1-score of each employed technique using the first testing criterion which is 70% and 30% for training and testing respectively; however, Fig. 3 presents the same using the second testing criterion which is 10-fold cross-validation. Considering the testing criteria, this analysis illustrates that 10-fold cross-validation is better to utilize for testing instead of 70% and 30% for training and testing.

Moreover, it also shows that KNN outperforms other employed models on 10-fold cross-validation. Using the percentage splitting criterion, KNN and RF both show the same performance.



	A1DE	NB	SVM	KNN	RF
F1-Score	0.995	0.986	0.995	0.998	0.998
Recall	0.995	0.986	0.995	0.998	0.998
Precision	0.995	0.986	0.995	0.998	0.998

Figure 2: Precision, recall, and accuracy analysis using percentage splitting criterion



	A1DE	NB	SVM	KNN	RF
F1-Score	0.996	0.985	0.994	0.999	0.998
Recall	0.996	0.985	0.994	0.999	0.998
Precision	0.996	0.985	0.995	0.999	0.998

Figure 3: Precision, recall, and accuracy analysis using 10-fold cross-validation

Figs. 4 and 5 separately present the accuracy analysis of each employed model using both testing criteria which are percentage splitting and K-fold cross-validation. In both cases, KNN outperforms another employed model with better accuracy is 99.8499% using percentage splitting testing criteria and 99.8599% on K-fold cross-validation criteria. This analysis shows that in the current scenario, K-fold cross-validation is the better training and testing criteria to train the model. Based on a variety of input values, ML models predict output values. One of the simplest ML algorithms is KNN, which is typically employed for classification. It categorizes the data point based on how its neighbors are categorized. Lazy Learner (Instance-based learning) is another name for KNN. It learns nothing throughout the training period. No discriminative function is generated using the training data. In other words, no training is required. It only draws learning from the stored training dataset for making real-time predictions. The KNN technique is much quicker than other utilized technique since other models require training while the KNN technique does not require training before producing predictions, and fresh data may be incorporated effortlessly without affecting the techniques accuracy.

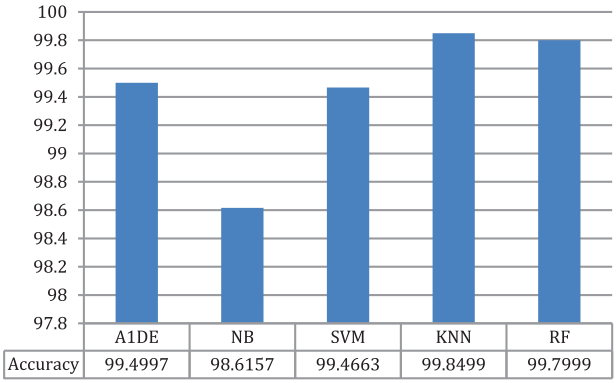


Figure 4: Accuracy analysis using percentage splitting testing criteria

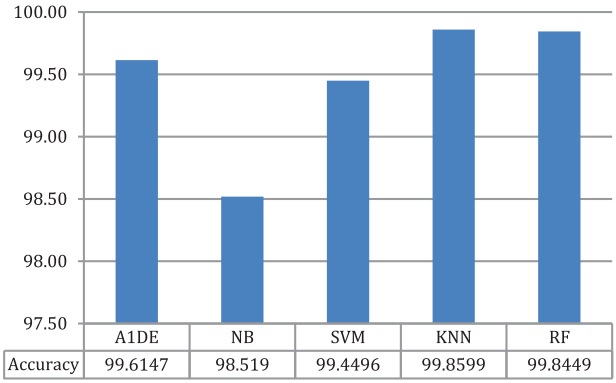


Figure 5: 10 accuracy analysis using 10-fold cross-validation testing criteria

Overall, the findings reveal that the ML models consistently outperform both assessment methods, demonstrating their efficiency in correctly predicting the target variable. Because the data partitioning and model training differ across the two techniques, the accuracy values produced by percentage splitting and 10-fold cross-validation are only slightly different. However, the models' excellent accuracy ratings demonstrate their potential for accurate predictions in the present situation.

The paper contributes to the field of cyber-security by providing insights into the effectiveness of ML models for detecting malware in PDF files. The comparative analysis and performance evaluation contribute to the development of robust systems for protecting against malicious activities associated with PDF malware. The study demonstrates the effectiveness of ML models in accurately detecting PDF malware and provides insights for developing robust systems to protect against malicious activities. The findings suggest that KNN is a promising model for PDF malware detection, but further research and experimentation may be required to validate and improve the results.

5 Conclusion and Future Work

In this research paper, we conducted a comparative analysis of machine learning (ML) models for PDF malware detection, focusing on the A1DE, NB, SVM, KNN, and RF models. We utilized a dataset obtained from the Canadian Institute for Cyber-security and employed two testing criteria: percentage splitting and 10-fold cross-validation. Our evaluation was based on precision, recall,

F1-score, and accuracy metrics. The results showed that KNN outperformed the other models, achieving an accuracy of 99.8599% using 10-fold cross-validation. These findings highlight the effectiveness of ML models in accurately detecting PDF malware and provide insights for developing robust systems to protect against malicious activities in PDF files. The research contributes to enhancing cyber-security measures by providing a reliable model for PDF malware detection, which can help in preventing the proliferation of sophisticated attacks through maliciously coded documents.

Moreover, the paper has some limitations such as a limited dataset and the lack of comparison with other approaches; its methodology, performance evaluation, and comparative analysis contribute to its validity. However, further research and validation using diverse datasets and comparison with alternative methods are necessary to strengthen the findings.

Moving forward, further research in the field of PDF malware detection should focus on several key areas. First, investigating deep learning methods like convolutional neural networks and recurrent neural networks (RNNs) may improve the performance and accuracy of the models. Additionally, incorporating natural language processing (NLP) techniques to analyze the textual content within PDF files could provide valuable insights for malware detection. Moreover, the development of real-time detection systems that can analyze PDF files on the fly and detect emerging threats in a timely manner would be highly beneficial. Lastly, collaboration between researchers, industry professionals, and cyber-security organizations is crucial to gather large-scale, diverse datasets for training and testing purposes, ensuring the models are robust and effective against a wide range of PDF malware variants.

Declarations: We, hereby declare that the research paper titled "Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training and Testing Criteria," submitted for publication, is my original work and has not been submitted elsewhere for any academic or non-academic purpose. We confirm that all the sources used in this research paper have been appropriately cited and acknowledged. Any references, data, or ideas obtained from other authors are duly attributed and documented in the references section.

Acknowledgement: We would like to express our sincere gratitude to all those who contributed to the successful completion of this research paper, "Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training and Testing Criteria". We are thankful to the research team members, Bilal Khan (BK), Muhammad Arshad (MA), and Sarwar Shah Khan (SSK), for their collaboration and valuable insights throughout the research process. We extend our appreciation to the institutions that supported this research work. We are deeply grateful to the Canadian Institute for Cyber-security for providing the dataset that formed the foundation of our analysis. Their contributions have been instrumental in enabling us to conduct this study on PDF malware detection and assess the performance of various ML models. Our heartfelt appreciation goes to all the individuals and institutions that reviewed and provided constructive feedback on this research paper. Your valuable input helped improve the quality and rigor of our work. Without the collective efforts and support of all these individuals and organizations, this research paper would not have been possible. Thank you all for your invaluable contributions.

Funding Statement: No specific grant from a funding organization supported this research.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: BK; data collection: BK and MA; analysis and interpretation of results: SSK and BK; draft

manuscript preparation: MA and SSK. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this research paper, "Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training and Testing Criteria," is obtained from the Canadian Institute for Cyber-security and is publicly available for research purposes. The dataset can be accessed at the following URL: https://www.unb.ca/cic/datasets/pdfmal-2022.html.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. S. Jeong, J. Woo and A. R. Kang, "Malware detection on byte streams of pdf files using convolutional neural networks," *Security and Communication Networks*, vol. 2019, pp. 144–152, 2019.

[2] B. Cuan, A. Damien, C. Delaplace and M. Valois, "Malware detection in PDF files using machine learning," in *ICETE 2018-Proc. of the 15th Int. Joint Conf. on e-Business and Telecommunications*, Porto, Portugal, vol. 2, pp. 412–419, 2018.

[3] A. Falah, S. R. Pokhrel, L. Pan and A. de Souza-Daw, "Towards enhanced PDF maldocs detection with feature engineering: Design challenges," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 41103–41130, 2022.

[4] W. Xu, Y. Qi and D. Evans, "Automatically evading classifiers," in *Proc. of the 23rd Annual Network and Distributed System Security Symp.*, San Diego, California, vol. 2016, no. February, pp. 21–24, 2016.

[5] S. Sibi Chakkaravarthy, D. Sangeetha and V. Vaidehi, "A survey on malware analysis and mitigation techniques," *Computer Science Review*, vol. 32, pp. 1–23, 2019.

[6] F. J. Abdullayeva and S. S. Ojagverdiyeva, "Multicriteria decision making using analytic hierarchy process for child protection from malicious content on the internet," *International Journal of Computer Network & Information Security*, vol. 13, no. 3, pp. 52–61, 2021.

[7] B. Wickman, H. Hu, I. Yun, D. Jang, J. W. Lim *et al.,* "Preventing Use-After-Free attacks with fast forward allocation," in *30th USENIX Security Symp. (USENIX Security 21)*, Vancouver, Canada, pp. 2453–2470, 2021.

[8] M. Templ and M. Sariyar, "A systematic overview on methods to protect sensitive data provided for various analyses," *International Journal of Information Security*, vol. 21, no. 6, pp. 1233–1246, 2022.

[9] W. Syafitri, Z. Shukur, U. Asma'Mokhtar, R. Sulaiman and M. A. Ibrahim, "Social engineering attacks prevention: A systematic literature review," *IEEE Access*, vol. 10, pp. 39325–39343, 2022.

[10] H. Ahmad, I. Dharmadasa, F. Ullah and M. A. Babar, "A review on C3I systems' security: Vulnerabilities, attacks, and countermeasures," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–38, 2023.

[11] W. Li, W. Meng, Z. Tan and Y. Xiang, "Design of multi-view based email classification for IoT systems via semi-supervised learning," *Journal of Network and Computer Applications*, vol. 128, pp. 56–63, 2019.

[12] Y. Li, X. Wang, Z. Shi, R. Zhang, J. Xue *et al.,* "Boosting training for PDF malware classifier via active learning," *International Journal of Intelligent Systems*, vol. 37, no. 4, pp. 2803–2821, 2022.

[13] S. S. Khan, M. Khan, S. Technology and R. Naseem, "Challenges in opinion mining, comprehensive," *A Science and Technology Journal*, vol. 33, no. 11, pp. 123–135, 2018.

[14] T. Tsafrir, A. Cohen, E. Nir and N. Nissim, "Efficient feature extraction methodologies for unknown MP4-malware detection using machine learning algorithms," *Expert Systems with Applications*, vol. 219, pp. 119615, 2023.

[15] A. R. Kang, Y. S. Jeong, S. L. Kim and J. Woo, "Malicious PDF detection model against adversarial attack built from benign PDF containing javascript," *Applied Sciences*, vol. 9, no. 22, pp. 4764, 2019.

[16] Y. Chen, S. Wang, D. She and S. Jana, "On training robust {PDF} malware classifiers," in *29th USENIX Security Symp. (USENIX Security 20)*, Boston, USA, pp. 2343–2360, 2020.
[17] M. Cova, C. Kruegel and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," in *Proc. of the 19th Int. Conf. on World Wide Web*, North Carolina, USA, pp. 281–290, 2010.
[18] P. Laskov and N. Šrndić, "Static detection of malicious JavaScript-bearing PDF documents," in *Proc. of the 27th Annual Computer Security Applications Conf.*, Orlando, Florida, USA, pp. 373–382, 2011.
[19] S. J. Khitan, A. Hadi and J. Atoum, "PDF forensic analysis system using YARA," *International Journal of Computer Science and Network Security*, vol. 17, no. 5, pp. 77–85, 2017.
[20] J. Zhang, "MLPdf: An effective machine learning based approach for PDF malware detection," pp. 1–6, 2018. [Online]. Available: https://arxiv.org/pdf/1808.06991.pdf
[21] D. Liu, H. Wang and A. Stavrou, "Detecting malicious javascript in pdf through document instrumentation," in *2014 44th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, Atlanta, USA, pp. 100–111, 2014.
[22] J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic feature dataset for ransomware detection using machine learning algorithms," *Sensors*, vol. 23, no. 3, pp. 1053, 2023.
[23] M. Dhalaria and E. Gandotra, "A hybrid approach for android malware detection and family classification," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 174–188, 2020.
[24] M. Deore and U. Kulkarni, "Mdfrcnn: Malware detection using faster region proposals convolution neural network," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 146–162, 2022.