Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Pdf Malware Detection: Explainable Machine Learning Modelling Pdf Malware Detection

Kotrike Samula Christeena [1], Garima Sinha [2]

[1] Master's Student, Computer Science and Engineering, Jain Deemed-to-be University
[2] Garima Sinha, Computer Science and Engineering, Jain Deemed-to-be University

Abstract:
The issue regarding the sharing of file in PDF format is one issue that the digital technology has made a stride in the lead and thus, viruses have come to infect computers considering that the software is so widespread. The project titled as Detecting Malware in PDFs: Advancing Machine Learning Models with Interpretability Assessment is in fact executed since it does not just create machine learning algorithms capable of identifying malware concealed within PDF files but also analyses them in terms of effectiveness. The taking of the Kaggled corpus with labelling of the poisonous and non-poisonous PDFs being the core of the data is considered one of the significant components of the experiment. Additionally, RF, C5.0, J48, SVM, AdaBoost, DNN, GBM, and KNN are some of the techniques that will be subject to practical testing. The last two goals are to attain optimal detection rates, and simultaneously to provide a compromise of what the model has decided and give the researcher the clues on what made the model arrive at this decision. The machine learning techniques will be implemented and the project will enable the aggregation of greater cybersecurity solutions that are a community protection against the threats that could have initially surpassed via PDF files and thus their spread is being intercepted.

Keywords: PDF malware detectors, ML, RF, SVM, DNN, explainability, cybersecurity, rogue PDF, classification algorithms, Kaggle Dataset.

1. Introduction

There has been an approval of the use of PDFs as the main format of the documents in the last few years due to its benefits, such as easy handling, customizing and protection of the document and rights of the writer. Therefore, the high frequency of use makes them susceptible to hackers who will take advantage of the vulnerability of the complete system and creep in malicious files in PDF documents. Presently, the conventional signature-based malware detection systems are not usually in a position to notice these threats which continue to evolve in nature. In that regard, the project which is dedicated to the development of ML-based solutions, in its turn, includes the detection of malware in PDF files along with the intelligibility of decisions among the key outcomes.

The main aspect of the study is an open-source dataset that is provided by Kaggle and is accompanied by labeled and unlabeled PDFs; thus, it simplifies the process of identifying the good and the bad. Overall, there will be eight ML algorithms used namely, RF, C5.0, J48, SVM, AdaBoost, DNN, GBM and KNN. The various methods in issue of accuracy in Malware detection are to be studied and the explainability

methods are likely to be integrated in to clarify the prediction made by the models and in the process, this will help the security professionals to have the reasoning behind a given file being termed either as a malicious file or a safe file. This novel study will be the first of its kind in enjoying high level of detection accuracy as well as explainability as such, it can in effect improve the current security provisions, thus, preventing the threat that the PDF documents may be posing in the first place. Further, the result of the explanation analysis will offer a substantial contribution to the current state of mistrust and secrecy when it comes to using machine learning with the aim of malware detection. Finally, the project is also seeking to provide a solution that will fulfill the needs of being fast, easy to scale, and clear in its functionality.

1.1.Objective of Project

Vast ML-based system of detecting malwared PDF files. This shall encompass utilization and analysis of various kinds of algorithms that are the ML algorithm like RF, C5.0, J48, SVM, AdaBoost, DNN, GBM and KNN in the undertaking of identifying an infected or clean PDF file. The project criteria will be to ascertain that the utmost possibility is achieved of the top accuracy of interpretation, and that the models are worth understanding, and that there should be clarity in the decision-making procedure. According to the authors, the accuracy and explainability will enable the project to offer a powerful method of detecting and removing risks in PDF files that, in its turn, will enhance the level of security caution against cyber-attacks. In addition, the project will examine the efficiency of these models with the support of a large variety of measures and the incorporation of the most fruitful models within a convenient framework to reveal malicious codes in time, accordingly, preventing the privacy of the information and ensuring the security of the virtual space.

1.2.Scope

The core concept of the given project is to develop and test novel machine learning algorithms that will be able to identify whether a PDF file has malware or not [10]. To be more exact, it will apply many classifiers to implement the project, such as the ones mentioned above RF, C5.0, J48, SVM, AdaBoost, DNN, GBM, and KNN as well as the ML algorithms of the Kaggle dataset comprised of labelled PDFs. The primary element of success of the project is the accuracy in detection, however, simultaneously, the model will also be capable of defending its conclusions meaning that the user will have the right to know the logic of certain classifications. [11]. Some of the major activities will consist of data preparation, teaching and appraisal of the model on how to compare the models based on their accuracy. The result will be the creation of an app that is capable of identifying malware in PDF documents real-time thus will contribute to the security measures, as well as, give an insight into the process of decision-making through the clarifying features of the models [12] hence will help the project by not introducing any malware into the system and will not require a significant integration with the current security systems.

1.3.Problem of Statement

The popularity of the PDF files, together with their option to embed various types of content, has seen them become the most popular method of malware distribution. But with the development of malware, the conventional security protocols do not work and therefore cannot detect and neutralize computer virus

that is concealed in the PDF files [13]. And, in fact, the project under discussion is the very detection of such mechanisms as the Machine Learning algorithms that are applied to screen the PDFs into the category of either safe or harmful. The process of manual analysis of PDFs is a stressful one because of the sheer quantity but what is more is that malware methods are evolving faster; therefore, detection software that is not automated is imperative. This project will aim at developing an effective, high-performance, and straightforward ML model that not only will encourage smarter malware detection capabilities, but also will improve the general defence system of cybersecurity.

1.4.Motivation

The rising use of PDF file in sharing documents has unfortunately made them to be one of the main avenues where malware attacks can occur. We need to discover an existence of malware in such files in case we need to save confidential information and ensure safety of the cyber world. Conventional security tools cannot sometimes detect attacks because the attackers employ advanced means. This issue is to be addressed in the project Detecting Malware in PDFs: Advancing Machine Learning Models with Interpretability Assessment, in which the sophisticated machine learning algorithms are applied onto the labeled PDF dataset at Kaggle to address this issue [16]. Instead of only achieving a high detection accuracy, we will aim to achieve explainability of the models and in the process provide a more holistic picture of the situation.

2. Related works

Detection of malware on PDF files has become an important aspect of the cybersecurity feature.[17]. Numerous studies have already been carried out and carried out to investigate the application of machine learning practices as one of the approaches to increasing the rate of detection of PDF malware. The literature review is well structured and shows the methods together with their drawbacks in five subtitles. PDF malware exploits the weaknesses of PDF readers and utilization of scripts to execute the illegal operations [18]. The attackers make use of JavaScript, compressed files, and encryption in order to reduce their detection rates. According to Laskov et al. (2021), it is one of the methods of avoiding signature-based detection systems that the criminals employ by implementing evasion methods used by the hackers [19]. A majority of the literature conducted in recent past has reported that the development of PDF malware is yet to take a significant turn towards the encryption and coding as the means of hiding the malicious payload thus becoming an even greater challenge to detect.

Application of machine learning (ML) models to detect malware in PDF files is a subject that is limited to the analysis of its persistent and dynamic properties. As an example, the classifier Random Forest and SVM were exhibited as effective in the procedure of identifying and classify malicious versus non-malicious PDF files with reasonably decent accuracy by Saxe and Berlin (2022) [20]. Under the fixed analysis, the characteristics picked are metadata, object counts and presence of JavaScript whereas under dynamic analysis, an environment that is sandboxed is utilized to monitor the behavior of the execution [21]. The problem of feature selection is regarded as a significant step toward the enhancement of a model work. The features of presence are considered static (JavaScript presence, the number of embedded files, and the type of object), and the behavioral features are the patterns of execution and the count of API calls. Such works as the one conducted by Maiorca et al. (2021) promote the use of Information Gain and PCA

to determine the key characteristics, thereby resulting in the further enhancement of both the accuracy and the interpretability. The provision of explanations is considered to be one of the main aspects of the trust-building process and consequently results in the adoption of the solution within the sphere of cybersecurity. The attempt of researchers today to interpret the decisions of models with the assistance of SHAP and LIME can be termed as the overriding trend currently. Here, Ribeiro et al. (2022) showed how LIME could make predictions of the classifiers to appear as human explanations [23]. However, the issue is not complete, since among the challenges are adversarial attacks, which keep on evolving evasion strategies, and class imbalance that the machine learning models must contend with. One of the new directions of adversarial techniques should be developed in future research.

The breakthrough in the field of PDF malware detection has been reached due to machine learning models. Random Forest, SVM, and deep learning remain only some of the various approaches that have already shown their value, but model interpretability as well as resilience to the adversarial attack remains a significant challenge [25] The future work efforts need to focus on feature selection optimization, model development resilient to adversarial attack, and explainable AI usage in cybersecurity solutions.

3. Methodology

The machine learning-based malware-detection system of its kind (based on PDF files) employs the most Ferrari-esque machine learning methods to categorize the said files as malware and non-malware files. In addition to this, it is supported by a huge amount of data offered in Kaggle with labeled PDF examples of both the classes. It further uses the most common ML algorithms, including RF, C5.0, J48, SVM, AdaBoost, DNN, GBM, and KNN to classification. The whole thing is carried out in such a way that ultimately every approach is rated according to the performance and efficacy in terms of detection of malware. The trustworthiness and transparency in the decision-making process of the suggested system through the proposed ML models is one of the key System. This is attained by explainable AI techniques and any classification that has been made by the system is brought to light to the users and hence, they are able to enhance trust and reliability.

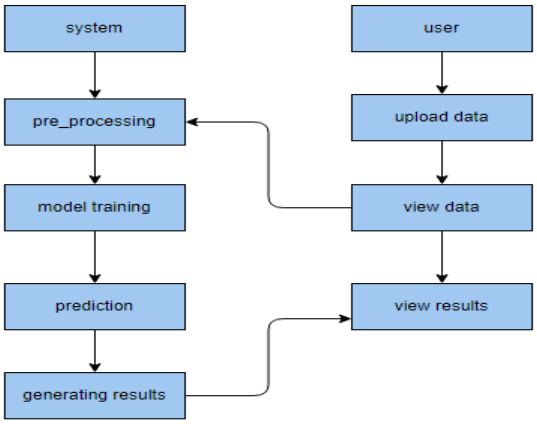

Fig. 1. Block diagram of the proposed system

3.1.Data Collection

The first task that is carried out during this project is an assortment of labelled PDF files, this is a set that will be accessed at Kaggle, and is a mix of malicious and benign Files. The information concerning the appearance of the JavaScript feature, the number of objects, and the information of an embedded file is used to construct the dataset using the subject data of PDFs that are not limited to metadata attributes. The labels score the malware (1) and the non-malware (0), and such a kind of scoring enables the implementation of the supervised learning. Close examination of the data is made such that it consists of changing malware families and these are different attack vectors i.e. obfuscation techniques, shellcode embedding and exploitation by using JavaScript.

3.2.Data Preprocessing

The main factor in considering the quality and further improving of the model is grounded on the preprocessing of data. The data step is cleaned to remove any missing data and any redundant data. Features engineering takes place and Data features such as number of JavaScripts, embedded objects and entropy measurements are computed. The features are then rescaled with scaling technique (MinMax or standardScaler) where all the numeric features are scaled to the same level hence making the model to be more efficient.

3.3.Model Training

Detection of malware documents of PDF files is as a result of training several machine learning models. ML algorithms that will be used would consist of the classifiers, which in this case are RF, C5.0, J48, SVM, AdaBoost, DNN, GBM and KNN. In this study, the size of data split will be 80/20 with the 80 data division attending to training and the remaining 20 percent to testing to ascertain how closely this model will become general. In order to avoid the overfitting and that of tuning the hyperparameters then the Cross-validation method (5-fold or 10-fold) will be used. The learning is achieved through the modeling based on the best parameter of practices- Hyperparameter tuning method is either the Grid Search or the Bayesian Optimization method that is directed towards the best performance. Since one of the requirements in applications of cybersecurity depends on the ability to view how the decision was made, explainability methods e.g. SHAP and LIME can be used not only to reveal the importance of the features, but also view the mechanism used to get to the final decision. The trained models are eventually tested on various measures such as accuracy which are considered as one of the most commonly used to offer sound system of detection.

4. Implementation

4.1.Support Vector Machine

The use SVM is a supervised learning method which is applicable both in classification and regression but mostly on the classification problems. The final aim is to magnify the margin that is the distance between the closest data points (support vectors) and the decision border. The widening of this margin improves the abilities of the model to extrapolate on novel and unknown data to ensure that it works. In addition, SVM has the capacity to solve a unique overlay as well as non-linear classification facts with the help of a collection of different kernel functions that include linear, polynomial, radial basis functional

(RBF), and sigmoid kernels. These kernels are used to lower the mapping By mapping the input space to a more dimensional space feature space that can be separated linearly.

Model Architecture:

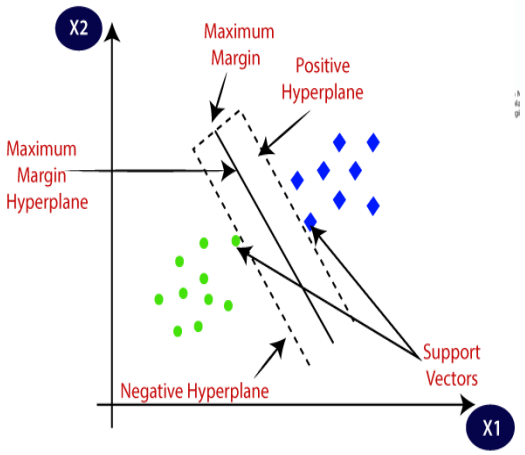

Fig. 2. Architecture of SVM

4.2.Random Forest

RF is a great group learning algorithm, and it is utilized to improve accuracy and consistency of decision tree. Its main purpose is to create a robust predictor model, which will combine more than a decision tree and will combine their answers to make more precise and generalized predictions. It can be applied to regression and classification problems, the problem of overfitting single decision trees can be solved by RF using an introduction of randomness: first, by bootstrapping (selection with replacement) individual subsets of the learning data to form different trees, and second, by merely randomly selecting the subset of features used at each split when forming a tree, so that the trees of the forest are not significantly correlated with each other, and thus the resulting model is better generalized and can withstand noise.

Model Architecture:

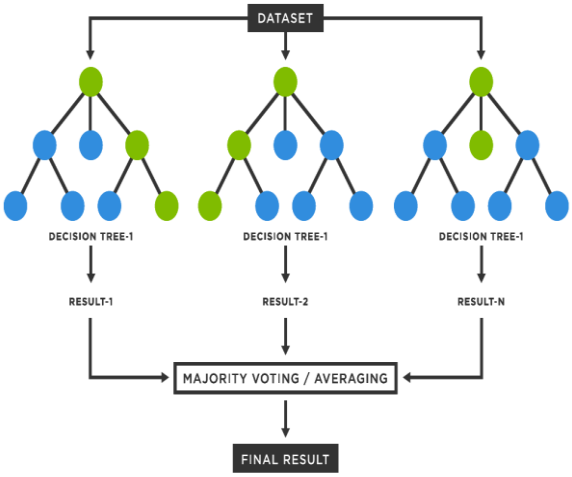

Fig. 3. Architecture of RF

4.3.Adaptive Boosting

AdaBoost ensemble learning algorithm is a tool of increasing the effectiveness of the weak classifier by intensifying the outputs of the them many times of them successively. The main goal of AdaBoost is the cause is to utilize a group of Weak learners, which will generally be decision stumps, together to strong classifier by focusing on the hard to classify ones. AdaBoost places more weight on the misclassified samples, compared with the bagging techniques, such as the Random Forest, which uses independent models to model the data, therefore, subsequent classification places more emphasis on them. AdaBoost reduces the classification error by focusing the model on the difficult to classify examples and this is done iteratively. The technique has a low bias and variance hence is very practical in both the categorical and continue activity and is mostly applied on the categorical activity.

Model Architecture:

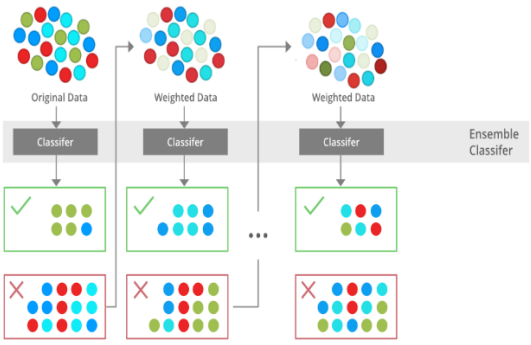

Fig. 4. Architecture of AdaBoost

4.4.Gradient Boosting Machine

GBM is a suitable ensemble learning algorithm that tries to enhance predictive accuracy through a series of weak model making and reduction of error through gradient descent. The main objective of GBM is to build a good predictive model through a series of synthesis of a number of weak learners that are usually decision trees. Before the introduction of bagging or traditional boosting, GBM is an optimizer, which uses a loss function through gradient descent to minimize the residual errors of the already existing models. GBM applies to categorical and on-going activities. It is carried out by generating a combination of predictive performance and high-flexibility of alternative loss functions either by refining predictions, bias reduction and minimization of bias.

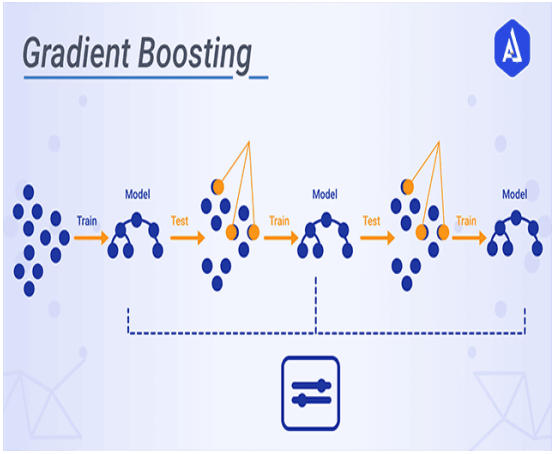

Fig. 5. Architecture of GBM

4.5.K-Nearest Neighbours

KNN algorithm is a machine learning algorithm, instance based and non-parametric algorithm via which categorical and continuous tasks could be carried out, although categorization or prediction of the outcome of a datum point with references to the most popular of the nearest K features in the feature space is the main aim of KNN algorithm. The principles of KNN are that two data that are similar to one another will tend to be near to one another in the particular space. In contrast to most machine learning models, KNN does not build an explicit model at all, but rather just stores the entire training set, which is then used to make predictions based upon it by computing the similarity Metrics that could be Euclidean distance, Manhattan distance or Minkowski distance. KNN finds extensive application in image recognition, recommendation system and anomaly detection, flexibility and effectiveness in multi-class classification problems as it is quite few in number. Nevertheless, it is also computationally inexpensive in case of large datasets involved and distance between query point and all the training points to be calculated.

Model Training Architecture

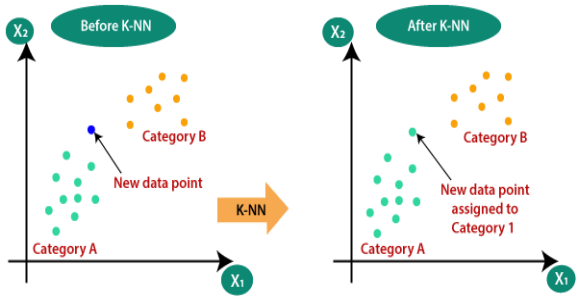

Fig. 6. Architecture of KNN

4.6.Deep Neural Networks

The primary aim of DNN is to detect the complex patterns and representations of the data with multiple layers of neural affiliations. Contrary to the traditional machine learning models, which have a set of features with functions specified manually, DNNs automatically define features with hierarchical functions, this is the reason why it is especially widespread in systems that are attempting to estimate a complex function by minimizing a loss function by implementing iterative optimization algorithms, often gradient descent and backpropagation. The use of multiple layers and non-linear activation functions makes DNNs more effective in high-level abstractions and, therefore, enables their use with large and extremely high-dimensional data.

Model Architecture:

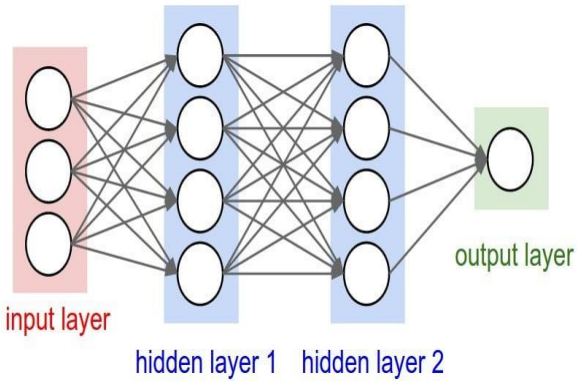

Fig. 7. Architecture of DNN

5. Result

The diagram of the comparison between the accuracy of the different ML algorithms allows inferring the results of the individual models. The two models are also on the peak accuracy which implies that they are both fairly good when applied to the given dataset. To reduce overfitting and have a better performance, they employ ensemble method of running many or more than one decision trees which lead to the predictions being strong and broad. In addition, AdaBoost is doing quite well, a step short of Gradient Boosting and Random Forest, thus, indicating the robustness of boosting techniques in transforming powerless learners by giving them more opportunities. The SVM and KNN on the other hand are slightly lagging behind with slightly lower accuracy. The hyperparameter and the Kernel are quite important in the determination of the performance of SVM and the latter can even be fine-tuned to result in a better performance. KNN is extremely simple and efficient when the data KNN cannot be effective in the case of a small dataset but high data dimension thus the accuracy of KNN would not be as good. But the least surprising is the fact that the accuracy of DNN is the lowest as compared to the rest of the models. This may be due to the fact that the insufficient of training data is an issue, or the insufficiency of network architecture, or excessive tuning of hyperparameters.

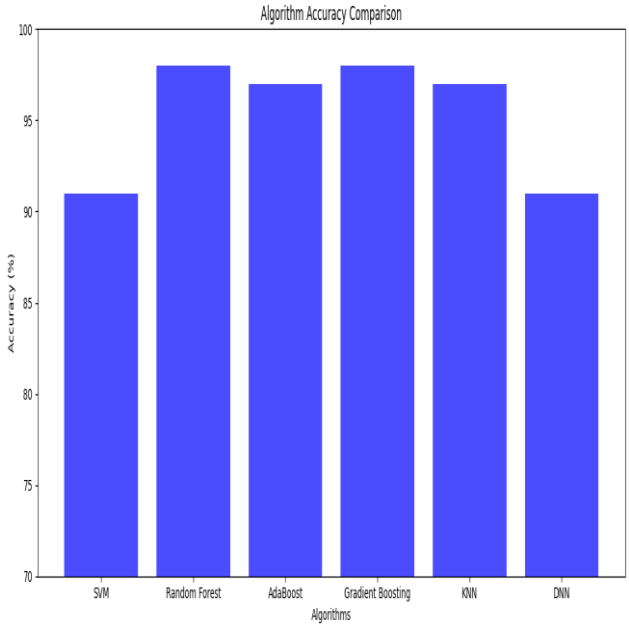

Fig. 8. Comparison plot for all models.

6. Conclusion

The project shows how the different ML algorithms would identify the occurrence of malicious contents in PDF files. The high rates of detection, along with the necessity to have the model that can be explained, are the outcomes of the use of the Random Forest, SVM, DNN, and other models. The doubled attention in this instance not only enhances the openness and credibility of the detecting procedure but also adds to the simplification of the detecting procedure therefore simplifying the way that the cybersecurity specialists comprehend and react to the dangers that can be introduced. The effective merging of the explainability with the detection models is an important innovation in the domain of cybersecurity since the strategy results in the credible and understandable strategy under the conditions of the PDF malware attacks prevention.

7. Future enhancement

The next step in the evolution of this work can be to implement the further enhanced explainability algorithms like SHAP or LIME that can be applied to generate even more understandable deep learning models (DNN) that are often opaque. Moreover, the increased dataset and more varied and recent PDF samples will help the model to be more robust to the changes that were implemented to the malware tactics and, as a result, its predictability will be improved. Better still, the system can be further streamlined by also adding the feature of real-time detection and automation of the model updating process according to the newly emerging threats. Last, but not the least, it is possible to consider the prospect of creating hybrid models that would involve the use of machine learning, which would not only provide a more holistic approach to the detection of PDF malware; it would also enhance the cybersecurity.

References

1. Abu Al-Haija, Q., Odeh, A., & Qattous, H. (2022). PDF Malware Detection Based on Optimizable Decision Trees. Electronics 2022, Vol. 11, Page 3142, 11(19), 3142. https://doi.org/10.3390/ELECTRONICS11193142
2. Alam, S., Horspool, R. N., Traore, I., & Sogukpinar, I. (2021). A framework for metamorphic malware analysis and real-time detection. Computers and Security, 48, 212–233. https://doi.org/10.1016/J.COSE.2014.10.011
3. Alshamrani, S. S. (2022). Design and Analysis of Machine Learning Based Technique for Malware Identification and Classification of Portable Document Format Files. Security and Communication Networks, 2022(1), 7611741. https://doi.org/10.1155/2022/7611741
4. Aslan, O., & Samet, R. (2020). A Comprehensive Review on Malware Detection Approaches. IEEE Access, 8, 6249–6271. https://doi.org/10.1109/ACCESS.2019.2963724
5. Han, K. S., Lim, J. H., Kang, B., & Im, E. G. (2021). Malware analysis using visualized images and entropy graphs. International Journal of Information Security, 14(1), 1–14. https://doi.org/10.1007/S10207-014-0242-0
6. Hossain, G. M. S., Deb, K., Janicke, H., & Sarker, I. H. (2024). PDF Malware Detection: Toward Machine Learning Modeling With Explainability Analysis. IEEE Access, 12, 13833–13859. https://doi.org/10.1109/ACCESS.2024.3357620
7. Islam, R., Tian, R., Batten, L. M., & Versteeg, S. (2023). Classification of malware based on integrated static and dynamic features. Journal of Network and Computer Applications, 36(2), 646–656. https://doi.org/10.1016/J.JNCA.2012.10.004
8. Kang, A. R., Jeong, Y. S., Kim, S. L., & Woo, J. (2020). Malicious PDF detection model against adversarial attack built from benign PDF containing javascript. Applied Sciences (Switzerland), 9(22). https://doi.org/10.3390/APP9224764
9. Komatwar, R., & Kokare, M. (2021). A Survey on Malware Detection and Classification. Journal of Applied Security Research, 16(3), 390–420. https://doi.org/10.1080/19361610.2020.1796162
10. Liu, C., Lou, C., Yu, M., Yiu, S. M., Chow, K. P., Li, G., Jiang, J., & Huang, W. (2021). A novel adversarial example detection method for malicious PDFs using multiple mutated classifiers. Forensic Science International: Digital Investigation, 38. https://doi.org/10.1016/J.FSIDI.2021.301124
11. Livathinos, N., Berrospi, C., Lysak, M., Kuropiatnyk, V., Nassar, A., Carvalho, A., Dolfi, M., Auer, C., Dinkla, K., & Staar, P. (2021). Robust PDF Document Conversion using Recurrent Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 35(17), 15137–15145. https://doi.org/10.1609/AAAI.V35I17.17777
12. Li, Y., Wang, X., Shi, Z., Zhang, R., Xue, J., & Wang, Z. (2022). Boosting training for PDF malware classifier via active learning. International Journal of Intelligent Systems, 37(4), 2803–2821. https://doi.org/10.1002/INT.22451
13. Maiorca, D., & Biggio, B. (2020). Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware. IEEE Security and Privacy, 17(1), 63–71. https://doi.org/10.1109/MSEC.2018.2875879

14. Maiorca, D., & Biggio, B. (2021). Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware. IEEE Security and Privacy, 17(1), 63–71. https://doi.org/10.1109/MSEC.2018.2875879
15. Maiorca, D., Giacinto, G., & Corona, I. (2022). A pattern recognition system for malicious PDF files detection. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7376 LNAI, 510–524. https://doi.org/10.1007/978-3-642-31537-4_40
16. Mao, Z., Fang, Z., Li, M., & Fan, Y. (2022). EvadeRL: Evading PDF Malware Classifiers with Deep Reinforcement Learning. Security and Communication Networks, 2022. https://doi.org/10.1155/2022/7218800
17. Muir, N. (2020). Working with Files and Folders. Windows® 7 Just the Steps™ for Dummies®, 25–35. https://doi.org/10.1002/9781118257562.CH3
18. Shijo, P. V., & Salim, A. (2021). Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804–811. https://doi.org/10.1016/J.PROCS.2015.02.149
19. Singh, P., Tapaswi, S., & Gupta, S. (2020a). Malware Detection in PDF and Office Documents: A survey. Information Security Journal: A Global Perspective, 29(3), 134–153. https://doi.org/10.1080/19393555.2020.1723747
20. Singh, P., Tapaswi, S., & Gupta, S. (2020b). Malware Detection in PDF and Office Documents: A survey. Information Security Journal, 29(3), 134–153. https://doi.org/10.1080/19393555.2020.1723747
21. Souri, A., & Hosseini, R. (2022). A state-of-the-art survey of malware detection approaches using data mining techniques. Human-Centric Computing and Information Sciences, 8(1). https://doi.org/10.1186/S13673-018-0125-X
22. Šrndić, N., & Laskov, P. (2021). Hidost: a static machine-learning-based detector of malicious files. Eurasip Journal on Information Security, 2016(1). https://doi.org/10.1186/S13635-016-0045-0
23. Ucci, D., Aniello, L., & Baldoni, R. (2020). Survey of machine learning techniques for malware analysis. Computers and Security, 81, 123–147. https://doi.org/10.1016/J.COSE.2018.11.001
24. Wiseman, Y. (2021). Efficient Embedded Images in Portable Document Format (PDF). International Journal of Advanced Science and Technology, 124, 129–138. https://doi.org/10.33832/IJAST.2019.124.12
25. Zhang, J. (2022). MLPdf: An Effective Machine Learning Based Approach for PDF Malware Detection. http://arxiv.org/abs/1808.06991