Malicious PDF detection Based on Machine Learning with Enhanced Feature Set

Suleiman Y. Yerima
Cyber Technology Institute
De Montfort University
Leicester, United Kingdom
syerima@dmu.ac.uk

Abul Bashar
Department of Computer Engineering
Prince Mohammad Bin Fahd University
Al-Khobar, Kingdom of Saudi Arabia
abashar@pmu.edu.sa

Ghazanfar Latif
Department of Computer Science
Prince Mohammad Bin Fahd University
Al-Khobar, Kingdom of Saudi Arabia
glatif@pmu.edu.sa

***Abstract*—PDF is one of the most popular document file formats due to its flexibility, platform independence and ability to embed different types of content. Over the years, PDF has become a popular attack vector for spreading malware and compromising computer systems. Existing signature-based defense systems have extremely high recall rates, but quickly become obsolete and ineffective against zero-day attacks, which makes them easy to circumvent by malicious PDF files. Recently, Machine Learning (ML) has emerged as a viable tool to improve discovery of previously unseen attacks. Hence, in this paper we present enhanced ML-based models for the detection of malicious PDF documents. We develop an approach for ML-based detection with static features derived from PDF documents leveraging existing tools and propose new, previously unused features to enhance the performance of the ML-based classifiers. Our investigative study is conducted on the recently published Evasive-PDFMal2022 dataset, which was used to evaluate seven ML classifiers based on our proposed method. The EvasivePDFMal2022 dataset consists of 4,468 benign samples and 5,557 malicious PDF samples. The results of the experiments show that our proposed approach with the enhanced features enabled improved accuracies in five out of seven of the classifiers that were evaluated. The results demonstrate the potential of the new features to increase the robustness of feature-based PDF malware detection.**

***Index Terms*—Malicious PDF detection; Static analysis; Feature engineering; Machine learning; Evasive PDF malware dataset.**

I. INTRODUCTION

Portable Document Format (PDF), being a popular and standardized file sharing format has been exploited by cyber criminals for various types of attacks including phishing and malware proliferation. Its widespread use across the globe, coupled with the flexibility to embed different types of content makes it an attractive vessel for delivering such attacks. Moreover, unlike executable files which are generally treated with caution, most users perceive PDF documents as benign and safe due to their proliferation in our daily digital transactions. The fact that the PDF format is complex, coupled with its susceptibility to a wide range of attacks, makes the detection of malicious PDF documents a challenging task. With the proliferation of PDF-based malware, traditional signature-based detection systems become unsustainable due to the need for constant update of the underlying knowledge base to keep up with the discovery of new attacks. Moreover, signatures are easily defeated by obfuscation, polymorphism and other forms of evasive behavior.

Manual analysis provides a reliable and information-rich approach to discover malicious PDF files. This is made possible by the availability of static and dynamic analysis tools. Static analysis tools can be used to scan PDF documents to reveal its structure and components and gain in-depth information about the file without executing it. Some of the well-known manual analysis tools for PDF investigation include PDFiD [1], PDFwalker [2], PeePDF [3], Origami, and PhoneyPDF [4].

The trend of utilizing ML approaches for automated detection of PDF malware emerged to overcome the limitations of manual and signature-based detection approaches. Researchers have proposed ML-based detection systems such as Slayer [5], Lux0R [6], Slayer Neo [7], HIDOST [8] and several more. In continuation of the trend of ML-based PDF malware detection, this paper proposes an effective system that employs an enhanced feature set to improve the performance and resilience of ML classifiers. The main contributions of this paper are as follows:

- We present a ML-based system for the detection of malicious PDF based on static structural features and anomaly-based features. Our proposed approach is a unique one that integrates anomaly-based features with structural features to improve performance.
- We introduce novel anomaly-based features and provide some insights into their derivation process and potential impact on the performance of the ML-based classifiers.
- Additionally, we demonstrate the performance gains that are achievable with the novel features, by undertaking experiments using the newly released Evasive-PDFMal2022 dataset.

The rest of our paper is organized as follows: In section II, we describe the PDF file format, followed by related work in section III. Section IV presents the methodology of our approach, and our new proposed features. Section V discusses our experiments and results, with conclusion in section VI.

The 14th IEEE International Conference on Computational Intelligence and Communication Networks, CICN 2022, Al-Khobar, Saudi Arabia, 4-6 December 2022 (Accepted version)

II. Structure of a PDF file

PDF was invented by Adobe in 1993 and has since become a de facto standard for sharing documents. It was created as a versatile format for sharing text, images, rich media etc. in a consistent way regardless of the software or hardware platform. It supports third-party technologies such as JavaScript and ActionScript. In 2008, the PDF format was standardized into an open format known as ISO 32000-1:2008 [9]. A PDF document has a typical structure as shown in Fig. 1, consisting of four parts:

- **The header**: which contains information about the PDF file version, according to the ISO standard.
- **The body**: which contains a number of objects that define the operations to be performed by the file, as well as the embedded data which could be images, text, scripting code etc. The contents displayed to the user is typically contained within this section. Operations such as decryption or decompression of data may be defined within an object and will generally take place while the file is being rendered. A PDF file can be updated after initial creation, and this prompts a new body and cross-reference section to be appended at the end.
- **The cross-reference (x-ref) table**: This section provides a list of the offset of each object inside the file to be rendered by the reader application. This allows for random access of any object within the file. Since the PDF standard allows for incremental updates to a document, this is enabled by the presence of the x-ref table. Extra x-ref tables and trailers are appended at the end of the document when it is updated.
- **The trailer**: This section contains a special object called the trailer. It shows how the document viewer should render the file by pointing it to the object identified by the /Root tag, which is the first object to be rendered. It also contains the offset of the start of the x-ref table. At the end of this section, there is an end of file string '%%EOF' which is the last line of the file.

Technically, a PDF file can be seen as a graph of objects that instructs the reader about the operations it has to perform to visualize the file contents to the user [10]. When a PDF reader displays a file, it begins from the trailer object and parses each indirect object referenced by the cross-reference table, and at the same time decompresses the data, so that all pages, images, texts and other components of the PDF file are progressively rendered.

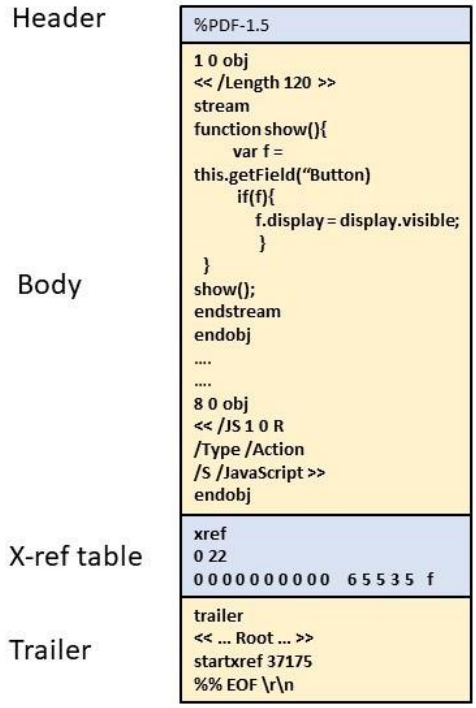

Fig. 1. The sections of a typical PDF file

III. Related work

The application of ML-based approaches to identify malware in PDF files have been successful in the recent years. One such research is presented by Torres and Santos in [11], where they aimed to verify whether using ML techniques for malware detection in PDF documents embedded with JavaScript was effective reinforcement for traditional AV solutions. They presented comparison results between different supervised ML algorithms: Support Vector Machines (SVM), Random Forests (RF) and Multi-Layer Perceptron (MLP), which yielded 50%, 92%, and 96% accuracies respectively. Their experiments were based on 995 training samples, 217 validation samples and 500 testing samples, collected from private shared malware repository.

In [12], a detection system to analyze PDF documents to identify benign from malicious PDF files was proposed. The proposed system makes use of AdaBoost decision tree with optimal hyper-parameters, trained and evaluated on the Evasive-PDFMal2022 dataset [13] (which is also used in our work). Their experiments demonstrate a lightweight PDF detection system that achieved 98.4% prediction accuracy with 98.80% precision, 98.90% sensitivity and 98.8% F1-score.

In [14], Falah et al. presented a PDF maldoc classification system where features were extracted using PDFiD and PeePDF. After extracting keyword features and structural features from both tools, they added a derived set of features from malicious document heuristics. A feature selection step was applied to identify important features, and after selecting the top 14 features, ML classifier accuracy was improved to 97.9% (with precision: 98.6%, recall: 97.4% and F1-score: 0.98). Jiang et al. [15] presented a semi-supervised ML method for detecting malicious PDF documents. Their approach extracts structural features as well as statistical features based on entropy sequences using the wavelet energy spectrum. A random subsampling strategy is employed to train multiple sub-classifiers. Experimental results demonstrate that their method yields an accuracy of 94% despite using training data with just 11% labeled malicious samples.

Zhang proposed MLPdf, an approach based on MLP neural network model in [16], for detection of PDF based malware. The model used a group of high quality features extracted from two real world datasets comprising 105,000 benign and malicious PDF documents. The model was shown to significantly outperform eight well known commercial anti-virus scanners, yielding a true positive rate of 95.12% with low false positive rate of 0.08%.

In [17], the authors proposed two models for PDF malware detection. The first one is a Convolutional Neural Network (CNN) model, while the second one is an ensemble model based on SVM with three different kernels. They utilized a total of 30,797 benign and malicious documents from VirusTotal and the Contagio dataset. Feature extraction was based on tree-based PDF file structure for the CNN model, and n-gram with Object content encoding for the ensemble SVM model. The ensemble model yielded an accuracy of 97.3% while the CNN model obtained 99.93% accuracy. They also demonstrated the robustness of their approach on adversarial samples generated using Mimicus.

Corum, Jenkins and Zheng [18] proposed PDF malware detection approach using image visualization techniques, where various image features representing the distinct visual characteristics of PDF malware and benign files were extracted. They evaluated the performance using Contagio PDF dataset, showing the viability of their approach for PDF malware detection. They also evaluated their models for reverse mimicry attacks showing improved robustness over the PDF Slayer approach. They considered both byte plot and Markov plot visualization approaches with various image processing techniques used in extracting features to train RF, K-Nearest Neighbor (KNN) and Decision Tree (DT) classifiers. The best method (byte plot + Gabor Filter + Random Forest) achieved an F1-score of 99.48%.

In [19], a framework for evasive PDF malware detection based on Stacking Learning was proposed. It uses a set of 28 static features consisting of general and structural features. The model was evaluated on the Contagio dataset yielding an accuracy of 99.89% and F1-score of 99.86%. The authors also evaluated the system on their newly generated Evasive-PDFmal2022 dataset [13] where they achieved 98.69% accuracy and 98.77% F1-score respectively.

Bazzi and Onozato proposed a dynamic analysis approach to automatically detect malicious PDF files in [20]. ML is used to process the report generated by a dynamic analysis system. A sandbox environment is used to automate the analysis of the submitted file. This involves using Cuckoo sandbox to open a submitted file using a designated viewer and logging all observed activities in the report. This was used as part of a larger IDS/IPS solution. The study utilized 6,000 samples for training and 10,904 samples for testing. LibSVM was used to create a classification model that achieved 97.45% accuracy based on three features. Other works on PDF malware detection include [21]–[27]

The above mentioned summary of the recent studies related to PDF malware detection by application of ML approaches, motivates us to implement and evaluate our proposed enhanced features-based approach on the Evasive-PDFMal2022 dataset.

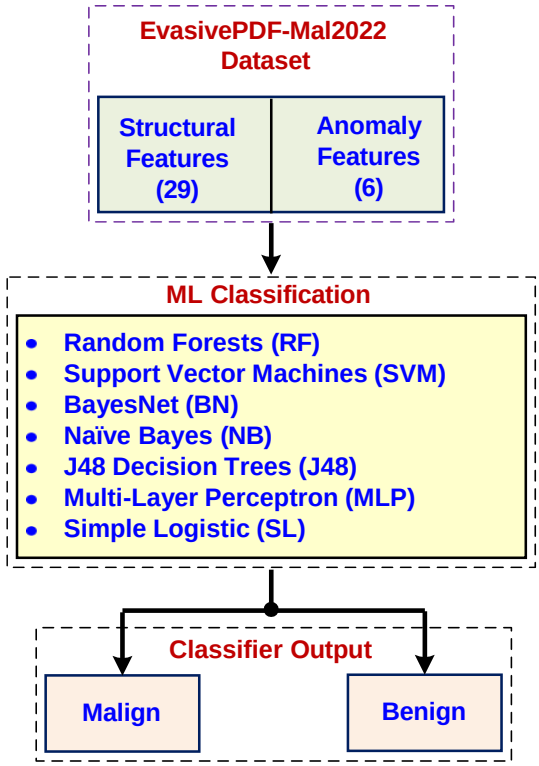

Fig. 2. Proposed enhanced features-based approach.

IV. Methodology

In this section we will discuss our proposed approach to automated ML-based malicious PDF detection by employing an enhanced feature set consisting of 35 features in total (29 Structural features and 6 Anomaly features). These features are extracted statically from the corpus of labeled files that constitutes the training set. The resulting data is fed into the ML classification algorithms to learn the distinguishing characteristics of benign and malicious PDF files thus enabling the prediction and classification of unlabeled PDF files as benign or malign, as shown in Fig. 2. The procedure and resources utilized in building the ML-based detectors are discussed in the following sub-sections.

A. Dataset

The dataset used for the study in this paper is a recently generated evasive PDF dataset (Evasive-PDFMal2022) [13] which was released by Issakhani et al. [19]. This dataset has been generated as an improved version of the well-known Contagio PDF dataset which has been utilized extensively in previous works. According to [19], the Contagio dataset had several drawbacks which include: (a) High proportion of duplicate samples with very high similarity, which was estimated as 44% of the entire dataset. (b) Lack of sufficient diversity of samples

TABLE I
INITIAL FEATURE SET CONTAINING 29 STRUCTURAL FEATURES

Feature name	Description
pdfsize	The size of the PDF file in kilobytes
metadata size	The size of the metadata
pages	Number of pages in the document (not from keyword)
title characters	Number of characters in the title of the file
isEncrypted	Whether or not the file is encrypted (not from keyword)
embedded files	Indicates presence of embedded file (not from keyword)
images	Indicates whether the document contains images
text	Indicates whether the document contains text
obj	Count of obj tags found
endobj	Count of endObj tags found
stream	Count of stream tags found
endstream	Count of endstream tags found
xref	Number of xref tables present
trailer	Number of trailers present
startxref	Count of xref start indicator
/Page	Number of pages in the PDF document
/Encrypt	Document has DRM or needs a password to be read
/ObjStm	Number of object streams that can contain other objects
/JS	Number of JS objects
/JavaScript	Number of JavaScript objects
/AA	Automatic action to be performed upon an event
/OpenAction	Automatic action to be performed on viewing document
/Acroform	Contains traditional forms authored in Adobe Acrobat
/JBIG2Decode	Indicates if the PDF document uses JBIG2 compression
/RichMedia	Contains embedded Flash or embedded media
/launch	Counts launch actions
/EmbeddedFile	Number of EmbeddedFile keywords found
/XFA	Keyword for XML Forms Architecture.
/Colors	Indicates the number of colours present in the file

within each class of the dataset. Thus, the new dataset aimed to address the flaws found with Contagio dataset and provide a more realistic and representative dataset of the PDF distribution. It consists of 10,025 PDF file samples with no duplicate entries (4468 benign and 5557 malicious).

B. Feature extraction of structural feature set

The ML-based detectors proposed in this paper employs 35 features: 29 structural features and 6 features that are based on anomalies i.e. observed deviation from expected characteristics of harmless files. The structural features correspond to those used in previous works, however the anomaly-based features are novel features introduced to improve the robustness of the ML-based detectors. In order to extract the features, we extended the open source PDFMalyzer tool available from [28]. PDFMalyzer is based on PDFiD and PyMuPDF and provided the capability to extract the 29 structural features. Using Python scripts to extend the PDFMalyzer tool, we were able to extract the 6 new anomaly-based features and combine them with the 29 structural features into a feature vector to represent each of the PDF file samples. The initial feature set of 29 structural features are listed in the Table I.

C. Enhancing the structural feature set with new features

Structural features are related to the characteristics of the name object present in the PDF file [10]. The advantage of the use of such features is their ability to detect different types of embedded contents that can enable malware detection (e.g., JavaScript, ActionScript). However, these keywords could be missed if a deliberate attempt has been made to evade their detection, e.g. through obfuscation, or due to errors from the analysis tools employed for the feature extraction. This motivated us to derive new features for improved robustness. We further elaborate on the anomaly-based features as follows.

When a PDF file is directly modified by a user (e.g. using an external tool), a new x-ref table and trailer are appended to the file. Thus, a PDF file that has been manually updated will typically have more than one trailer and x-ref table. Hence the features /trailer, /xref, and /startxref features should have more that one occurrence in the feature set for a benign file. Conversely, having only one occurrence of those features in the feature vector is an anomaly. We therefore defined two new features (mal_trait1 and mal_trait2) based on observing the number of /trailer, /xref or /startxref occurrences (which are typically the same) in combination with keyword features that are indicative of possible malicious content. These indicators of malicious content include (a) presence of JavaScript (b) the presence of one or more embedded files. The anomaly based features are explained below:

- **mal_trait1**: This is a new feature we introduced to indicate when /xref, /trailer and /startxref are only found once in the PDF file and the presence of JavaScript is also detected. This could be an indication of injection or embedding of JavaScript code with an automated tool (such as Metasploit), since the values of the three keywords do not suggest user modification.
- **mal_trait2**: This is a new feature we introduced to indicate when /xref and /trailer and /startxref are only found once in the PDF file and the presence of an embedded file is detected (JavaScript may or may not be present). This could also be an indication of injection or embedding of another file within the PDF file using an automated tool (such as Metasploit), since the values of the three keywords do not suggest user modification.
- **mal_trait3**: This is a new feature we introduced to search for the presence of both embedding files and JavaScript code in the PDF file. The intuition behind this feature is that the JavaScript code can be used to launch a malicious embedded file.
- **diff_obj**: This feature captures anomalies in involving the opening and closing tags of objects in a PDF file as described in [14]. Each object in the file is expected to have an opening tag (obj) and a corresponding closing tag (endObj). A difference in the occurrences of the opening and closing tags indicates possible file corruption (usually a missing closing tag). This is an obfuscation technique designed to bypass some parsing tools that strictly conform to PDF standards. On the other hand the file will still be rendered correctly by the PDF readers, thus enabling the intended malicious activity to occur.
- **diff_stream**: This also captures anomalies similar to diff_obj by observing the occurrences of ‘stream’ and ‘endStream’ which are the opening and closing tags of

TABLE II
CLASSIFIER PERFORMANCE WITH ORIGINAL FEATURES (10-FOLD CROSS VALIDATION RESULTS)

	Precision Mal/Ben	**Recall** Mal/Ben	**F1** Mal/Ben	**Accuracy** (%)
RF	0.997 / 0.995	0.995 / 0.997	0.996 / 0.996	99.6
SVM	0.999 / 0.771	0.753 / 0.999	0.859 / 0.870	86.48
BNet	0.972 / 0.951	0.959 / 0.967	0.965 / 0.959	96.25
NB	0.876 / 0.875	0.899 / 0.847	0.888 / 0.861	87.58
J48	0.993 / 0.991	0.993 / 0.991	0.993 / 0.991	99.2
MLP	0.911 / 0.951	0.962 / 0.887	0.936 / 0.918	92.77
SL	0.929 / 0.916	0.933 / 0.912	0.931 / 0.914	92.36

TABLE III
CLASSIFIER PERFORMANCE WITH ENHANCED FEATURES (10-FOLD CROSS VALIDATION RESULTS)

	Precision Mal/Ben	**Recall** Mal/Ben	**F1** Mal/Ben	**Accuracy** (%)
RF	0.998 / 0.994	0.995 / 0.998	0.997 / 0.996	99.65
SVM	0.993 / 0.81	0.811 / 0.993	0.893 / 0.892	89.27
BNet	0.982 / 0.953	0.961 / 0.978	0.961 / 0.975	96.86
NB	0.817 / 0.919	0.948 / 0.738	0.878 / 0.819	85.38
J48	0.988 / 0.981	0.985 / 0.985	0.986 / 0.986	98.5
MLP	0.986 / 0.958	0.965 / 0.983	0.975 / 0.97	97.31
SL	0.988 / 0.924	0.935 / 0.985	0.961 / 0.954	95.75

stream objects. According to [14], this evasive technique of omitting a stream object tag aims to corrupt the file in a way that it will still be rendered by readers but will confuse the parsing tools.

- **mal_traits_all**: This is a new composite feature we introduced that helps to identify files that exhibit one or more of the above 5 anomalous features. The intuition behind this is to create a robust feature that will maintain its relevance even if new techniques evolve to defeat a subset of the new features. For instance, the ability to obfuscate the /trailer, /xref, or /startxref values may produce errors in capturing mal_trait1 and mal_trait2 features or make them obsolete in future. However, mal_traits_all will still remain relevant in the presence of such obfuscation given that it is a compound feature. Also, the failure of extraction tools could lead to missing or erroneous values for some of the standard features. The composite feature therefore provides an indicator that has resilience against the occurrence of such errors.

V. EXPERIMENTS AND RESULTS

In this section, we present the results of the experiments performed to examine the effect of the new features on classifier performance. We already explained how the features provide resilience against some obfuscation and extraction errors, in the previous section. However, we are also interested in quantifying the impact of the new features on the performance of ML models. To do this we ran a baseline experiment with 7 models built using the original 29 features. The classification algorithms used include: Random Forest (RF), Support Vector Machine (SVM), Bayes Net (BNet), Naive Bayes (NB), J48 Decision tree (J48), Multi-layer Perceptron (MLP), and Simple Logistic (SL). We ran a second set of experiments with a set of new models built from the enhanced set of 35 features.

A. Original feature set results

Table II presents the 10 fold cross validation results of 7 ML classifiers trained using the original 29 structural features extracted from the PDF samples in the dataset. RF had the highest F1-scores for malware (0.996) and benign (0.996), as well as the best overall accuracy of 99.6%. With the exception of NB and SVM, all the other classifiers recorded overall accuracy above 92%.

B. Enhanced feature set results

Table III presents the 10 fold cross validation results of 7 ML classifiers trained using the enhanced set with 35 features. RF still maintained its position as the top classifier. There was only a slight improvement in overall accuracy, which is expected due to the fact that the baseline performance was already quite high to begin with. With other classifiers, i.e. SVM, MLP and SL, there was more significant improvement in performance, which can be attributed to the addition of the new features. This can be seen in Figure 3, where the overall accuracies for enhanced and original feature sets are depicted for the 7 classifiers. The overall accuracy of Bayes Net only increased to 96.85% from 96.25% with the addition of the new features. For NB and J48, however, there is a slight drop in overall performance. These results demonstrate the efficacy of the new features.

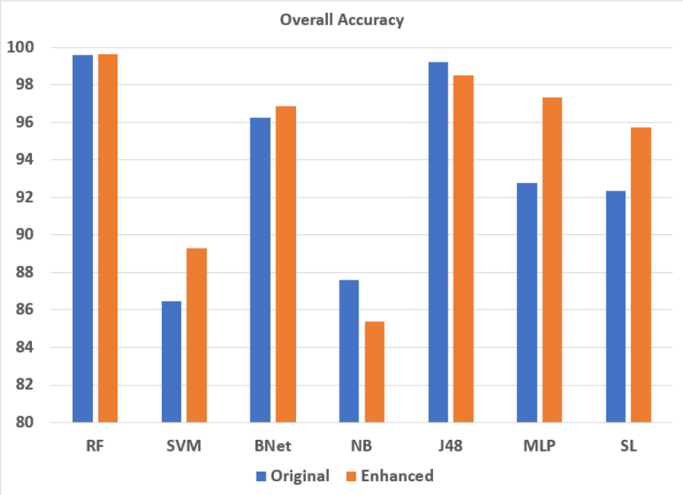

Fig. 3. Overall accuracy for enhanced and original features set with the various ML classifiers

C. Comparison with existing works

Since the dataset we used in this paper is relatively new, only few reported results currently exist in the literature for direct comparison. Papers [12] and [19] utilized the same dataset, and we summarize the results from those papers compared to ours in Table IV. From the table, it can be seen that our approach outperforms the existing works that were based on the same

dataset we used in our study. The best accuracy performance with RF and our enhanced feature set was 99.65%, which is higher than the 98.84% reported in [12], and the 98.69% overall accuracy reported in [19].

TABLE IV
COMPARISON WITH RELATED WORKS

	Precision	Recall	F1	Accuracy
Paper [12]	98.8%	98.9%	98.85%	98.84%
Paper [19]	98.88%	98.87%	98.77%	98.69%
Our approach	99.7%	99.7%	99.7%	99.65%

VI. CONCLUSION AND FUTURE WORK

In this paper we have introduced some novel features to create an enhanced feature set, with the aim of improving the robustness of machine learning based PDF malware detection systems that employ static features. We have discussed in detail how the features were derived, and also presented empirical evidence of their effectiveness by means of experiments with seven machine learning classifiers. We utilized the new EvasivePDFMal2022 dataset for our experiments. The best results from our enhanced feature set approach was achieved by Random Forest with 99.7% F1-score and 99.65% accuracy, which is better than existing works that proposed PDF malware detection solutions based on the same dataset. In future work, we aim to investigate the resilience of the enhanced feature set against different types of adversarial attacks.

ACKNOWLEDGEMENTS

This research is supported by the 2022 Cybersecurity research grant from the Cybersecurity Center at Prince Mohammad Bin Fahd University, Al-Khobar, Saudi Arabia.

REFERENCES

[1] D. Stevens. PDF Tools. [Online]. Available: https://blog.didierstevens.com/programs/pdf-tools/ [Last accessed: 25 Sept., 2022]

[2] G. Gdelugre. PDF Walker: Frontend to explore the internals of a PDF document with Origami. [Online]. Available: https://github.com/gdelugre/pdfwalker [Last accessed: 25 Sept., 2022]

[3] D. Stevens. peepdf - PDF Analysis Tool. [Online]. Available: https://eternal-todo.com/tools/peepdf-pdf-analysis-tool [Last accessed: 25 Sept., 2022]

[4] K. Bandla. phoneyPDF: A virtual PDF analysis framework. [Online]. Available: https://github.com/kbandla/phoneypdf [Last accessed: 08 Nov., 2022]

[5] D. Maiorca, D. Ariu, and I. Corona, "A pattern recognition system for malicious pdf files detection," *Machine Learning and Data Mining in Pattern Recognition*, pp. 510–524, 2012.

[6] I. Corona, D. Maiorca, D. Ariu, and G. Giacinto, "Detection of malicious pdf-embedded javascript code through discriminant analysis of api references," in *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, 2014, pp. 47–57.

[7] D. Maiorca, D. Ariu, and G. Giacinto, "A structural and content- based approach for a precise and robust detection of malicious pdf files," in *Information Systems Security and Privacy (ICISSP), 2015 International Conference on, IEEE, 2015*, 2015, pp. 27–36.

[8] N. Srndic and P. Laskov, "Hidost: A Static Machine-learning-based Detector of Malicious Files," *Eurasip J. Inf. Secur.*, December 2016.

[9] ISO. ISO 32000-1:2008 Document management — Portable document format — Part 1: PDF 1.7. [Online]. Available: https://www.iso.org/standard/51502.html [Last accessed: 30 Oct., 2022]

[10] D. Maiorca and B. Biggio, "Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware," *IEEE Security Privacy magazine, Special Issue on Digital Forensics*, Nov - Dec 2017.

[11] J. Torres and S. Santos, "Malicious pdf documents detection using machine learning techniques - a practical approach with cloud computing applications." in *The 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, 2018, pp. 337–344.

[12] Q. A. Al-Haija, A. Odeh, and Q. Hazem, "PDF Malware Detection Based on Optimizable Decision Trees," *Preprints*, September 2022.

[13] CIC. PDF Dataset: CIC-Evasive-PDFMal2022. [Online]. Available: ttps://www.unb.ca/cic/datasets/PDFMal- 2022.html [Last accessed: 25 Sept., 2022]

[14] A. Falah, L. Pan, S. Huda, S. R. Pokhrel, and A. Anwar, "Improving Malicious PDF Classifier with Feature Engineering: A Data-Driven Approach," *Future Generation Computer Systems*, vol. 115, pp. 314–326, February 2021.

[15] J. Jiang, N. Song, M. Yu, C. Liu, and W. Huang, "Detecting malicious pdf documents using semi-supervised machine learning," in *Peterson, G., Shenoi, S. (eds) Advances in Digital Forensics XVII. DigitalForensics 2021. IFIP Advances in Information and Communication Technology*, vol. 612, 2021.

[16] Z. Jason, "MLPdf: An Effective Machine Learning Based Approach for PDF Malware Detection," *Cryptography and Security (cs.CR). arXiv:1808.06991 [cs.CR]*, 2018.

[17] M. Albahar, M. Thanoon, M. Alzilai, A. Alrehily, M. Alfaar, M. Alghamdi, and N. Alassaf, "Toward Robust Classifiers for PDF Malware Detection," *Computers,Materials Continua*, vol. 69, no. 2, 2021.

[18] A. Corum, D. Jenkins, and J. Zheng, "Robust PDF Malware Detection with Image Visualization and Processing Techniques," in *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, 2019, pp. 1–5.

[19] M. Issakhani, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Pdf malware detection based on stacking learning," in *The 8th International Conference on Information Systems Security and Privacy (ICISSP 2022),*, 2022, pp. 562–570.

[20] A. Bazzi and O. Yoshikuni, "Automatic Detection of Malicious PDF Files Using Dynamic Analysis," in *JSST 2013 International Conference on Simulation Technology, Tokyo, Japan*, 2013, pp. 3–4.

[21] H. V. Nath and B. Mehtre, "Ensemble learning for detection of malicious content embedded in pdf documents," in *2015 IEEE International conference on signal processing, informatics, communication and energy systems (SPICES)*, 2015, pp. 1–5.

[22] M. Xu and T. Kim, "PlatPal: Detecting Malicious Documents with Platform Diversity," in *USENIX Security Symposium*, 2017, pp. 1–5.

[23] P. Singh, S. Tapaswi, and S. Gupta, "Malware detection in PDF and office documents: a survey," *Inf SecurJ: A Glob Perspect*, vol. 29, no. 3, September 2020.

[24] Y. Chen, S. Wang, D. She, and S. Jana, "On training robust pdf malware classifiers," in *29th USENIX Security Symp*, 2020, pp. 1–5.

[25] J. Jeong, Y. Woo and A. Kang, "Malware detection on byte streams of pdf files using convolutional neural networks," *Security and Communication Networks*, pp. 1–9, 2019.

[26] 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, May 2017.

[27] M. Li, Y. Liu, M. Yu, G. Li, Y. Wang, and C. Liu, "FEPDF: A Robust Feature Extractor for Malicious PDF Detection," in *2017 IEEE Trustcom/BigDataSE/ICESS*, 2017, pp. 1–5.

[28] A. H. Lashkari. PDFMALyzer. [Online]. Available: https://github.com/ahlashkari/PDFMalLyzer [Last accessed: 25 Sept., 2022]