CMPE-279 Software Security Technologies

Detecting Malware Loaded PDF's Using Machine Learning

Supriya Kankati | Vasavi Gollapalli | Akash Jagarlamudi

Abstract*— The Portable Document Format (PDF) is widely recognized for its platform independence and feature richness, which, unfortunately, also makes it a prime target for malware dissemination. With the increasing sophistication of cyber threats, traditional antivirus solutions are often inadequate, necessitating more dynamic approaches. This research paper explores the effectiveness of machine learning (ML) algorithms in the automated detection and classification of malicious versus benign PDF files. By leveraging a variety of machine learning models, including Logistic Regression, Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), and analyzing a comprehensive set of features extracted from PDFs, this study aims to identify and quantify the key indicators of malignancy in PDF files.

***Keywords*— SVM. KNN, Random Forest, Logistic regression**

I. Introduction

In recent years, the Portable Document Format (PDF) has become a popular file format for sharing documents due to its platform-independence and rich feature set. However, these same features have made PDFs an attractive vehicle for malware authors to distribute malicious code (Smutz and Stavrou, 2012). Malicious PDFs can exploit vulnerabilities in PDF readers, execute JavaScript code, or use other techniques to compromise the security of a system (Torres and Santos, 2018). To counter the growing threat of malicious PDFs, researchers have explored various methods for detecting and classifying malware-loaded PDFs. One promising approach is the use of machine learning algorithms to analyze the structure and content of PDF files and identify malicious characteristics (He *et al*., 2020). By extracting relevant features from PDF files and training machine learning models on these features, it is possible to develop automated systems that can accurately distinguish between benign and malicious PDFs.

Several studies have investigated the effectiveness of different machine learning algorithms and feature sets for detecting malicious PDFs. Liu *et al*. (2021) proposed a malware visualization technique that extracts features such as the number of objects, streams, and filters used in a PDF, and trains a convolutional neural network (CNN) on the resulting images. Torres and Santos (2018) developed a two-stage machine learning model that first trains an anomaly detection classifier for each type of object in a PDF, and then uses the results as input to a CNN for final classification. Their method considers both the content and structure of PDF files and achieves high accuracy and robustness against evasion techniques.

Zhang (2018) presented a multilayer perceptron (MLP) neural network model called MLP df for detecting malicious PDFs. The model uses a backpropagation algorithm with stochastic gradient descent and is trained on a set of 48 features extracted from PDF files, including metadata,content statistics, and structural properties. The MLP df approach outperformed several commercial anti-virus scanners. In this paper, we review an approach for detecting malware-loaded PDFs using machine learning. Our method extracts a set of features that capture various aspects of PDF structure and content, including the number of keywords related to streams, objects, filters, JavaScript, and other potentially malicious elements. By training machine learning models on these features, we aim to develop an accurate and robust system for identifying malicious PDFs.

II. Relatable Work

The detection of malicious PDF files using machine learning techniques has been an active area of research in recent years. Various approaches have been proposed, focusing on different feature sets, classification algorithms, and evaluation metrics. This section reviews some of the most relevant and influential works in this field, highlighting their

contributions and limitations in the context of our research. One of the earliest studies on malicious PDF detection using machine learning was conducted by Smutz and Stavrou (2012). They extracted a set of metadata and structural features from PDF files, such as the number of objects, streams, and filters, and trained a Random Forest classifier to distinguish between benign and malicious PDFs. Their approach achieved an accuracy of 99% and a low false positive rate of 0.2% on a dataset of 5,000 malicious and 100,000 benign files. However, their feature set did not include specific keywords related to JavaScript or other potentially malicious elements, which are considered in our research.

Torres and Santos (2018) developed a cloud-based framework for malicious PDF detection using machine learning. They extracted features from both the metadata and content of PDF files, including the presence of JavaScript code, suspicious API calls, and obfuscation techniques. Their approach used a combination of supervised learning algorithms, such as Support Vector Machines (SVM), Random Forest, and Multilayer Perceptron (MLP), and achieved an accuracy of 92% and an F1-score of 94% on a real-world dataset. While their feature set is more comprehensive than that of Smutz and Stavrou (2012), it still lacks some of the specific keywords and structural properties considered in our research. Zhang (2018) proposed an MLP-based approach called MLP df for detecting malicious PDFs. The model uses a backpropagation algorithm with stochastic gradient descent and is trained on a set of 48 features extracted from PDF files, including metadata, content statistics, and structural properties. The MLP df approach outperformed several commercial anti-virus scanners, achieving a true positive rate of 95.12% while maintaining a low false positive rate of 0.08%. While the feature set used by Zhang (2018) is similar to ours, their approach does not consider the number of specific keywords related to streams, objects, and filters, which we hypothesize to be important indicators of malicious behavior.

Liu *et al.* (2021) introduced a malware visualization technique for detecting malicious PDFs using machine learning. Their approach extracts features such as the number of objects, streams, and filters used in a PDF, and trains a convolutional neural network (CNN) on the resulting images. The proposed method achieved an accuracy of 97.3% and an F1-score of 97.5% on a dataset of malicious and benign PDFs. While their approach is novel and effective, it does not directly consider the presence of specific keywords or structural properties that are the focus of our research. He *et al.* (2020) proposed a two-stage machine learning algorithm for detecting malicious PDF files. In the first stage, an anomaly detection classifier is trained for each type of object in a PDF, based on content features extracted using a set of guiding principles. In the second stage, the results of the first stage are organized into a feature matrix and used as input to a CNN for final classification. Their approach considers both the content and structure of PDF files and achieves high accuracy and robustness against evasion techniques. While their feature set is comprehensive and their methodology is sound, our research focuses on a more specific set of keywords and structural properties that we believe to be particularly relevant for malicious PDF detection.

Pareek et al. (2013) proposed using n-gram byte sequences for PDF malware classification, which works well on non-encrypted, non-encoded documents. However, the authors note that this approach may be less effective on encrypted or compressed PDFs where the randomized byte patterns can impact the reliability of n-gram features, highlighting the need for techniques that can handle obfuscation.

Nissim et al. (2015) provide a comprehensive survey of machine learning techniques applied to PDF malware detection. They categorize features into structural properties, metadata, content-based properties, and embedded code analysis. The authors emphasize the importance of considering multiple feature categories to build robust detectors. However, the survey does not deeply examine techniques to handle obfuscated embedded malware through encryption or encoding, which remains an open challenge.

Maiorca et al. (2013) propose a novel approach called EvadeML for automatically generating malicious PDF variants that evade detection by machine learning classifiers. By modifying PDF structure and metadata in a semantics-preserving way, their technique can automatically create evasive variants. This demonstrates the vulnerability of relying only on structural and metadata features without considering embedded code and obfuscation techniques like encryption, underlining the need for more comprehensive detection approaches.

Carmony et al. (2016) reveal that PDF malware detectors are susceptible to evasion by malformed PDF files that exploit parsing vulnerabilities. Most parsers focus on standard PDF tags and fail to handle malformed files safely. The authors demonstrate successful evasion of both academic and commercial detectors using malformed PDFs, underscoring the need for robust parsing techniques that can handle non-standard or malicious PDF structures.

Xu et al. (2016) employ genetic programming to automatically evolve malicious PDF variants that evade two state-of-the-art machine learning detectors, PDFrate and Hidost. Their technique mutates a malicious PDF seed file using a set of evasion strategies to generate variants. By probing the detectors and evolving variants over generations, it discovers successful evasive samples. This work further exposes the limitations of structural and metadata based detectors and highlights the need for considering adversarial attacks during detector design and evaluation.

Corona et al. (2014) propose Lux0R, a machine learning system that detects malicious PDF files by leveraging lexical properties and static analysis of embedded JavaScript code. Lux0R extracts a large number of features from the JavaScript such as occurrences of API references, objects, methods, and keywords. While this enables detecting suspicious JavaScript, the approach may miss malicious code in other encodings or scripting languages, suggesting the need for more general techniques to analyze embedded code.

Several recent works have explored deep learning approaches for PDF malware detection as an alternative to traditional machine learning with hand-engineered features. Maiorca et al. (2015) propose a combination of structural features and convolutional neural networks (CNN) for PDF malware detection. The structural features model the hierarchical logical structure of the PDF, while the CNN learns to detect malicious payloads from the raw bytes of the file. By combining both approaches, their system achieves high accuracy on a real-world dataset, demonstrating the potential of leveraging both structure and content for detection.

In a similar vein, Bose et al. (2020) apply deep learning to detect malicious PDFs without relying on hand-crafted features. They introduce a novel neural network architecture called MultiStream that ingests both the raw bytes and the hierarchical structure of the PDF. MultiStream outperforms several baseline machine learning models, further validating the effectiveness of deep learning for this task. However, the authors acknowledge that their approach may still be vulnerable to adversarial attacks and recommend investigating defensive techniques to mitigate this risk.

Focusing specifically on detecting malicious JavaScript in PDFs, Jeong et al. (2019) propose using a CNN. They extract and tokenize JavaScript code from PDF documents into a sequence of keywords and API references. The sequences are then encoded into a fixed-length vector using the word2vec embedding and passed into a CNN for classification. Their approach achieves high accuracy and low false positive rates on a dataset of real-world malicious and benign PDF samples, highlighting the importance of analyzing embedded code.

Despite the promising results of deep learning for PDF malware detection, Dang et al. (2017) demonstrate that deep neural networks trained on PDFs are vulnerable to adversarial attacks. By applying gradient-based optimization, they can generate adversarial examples that fool both traditional machine learning and deep learning based

detectors. The authors argue that any machine learning based PDF malware detector must consider its robustness against adversarial attacks during both design and evaluation to ensure practical effectiveness.

The related work on PDF malware detection has progressed from rule-based parsers to traditional machine learning with hand-engineered features, and more recently to deep learning approaches. However, many detectors still struggle with obfuscated or adversarially crafted malicious PDFs. Effective detectors must combine robust parsing, analysis of both structure and embedded code, and consideration of potential adversarial attacks. The use of multiple classifiers in an ensemble, as proposed in this paper, is a promising approach to improve detection accuracy and robustness by leveraging the strengths of different techniques.

In summary, the related work in this field has made significant contributions to the development of effective machine learning techniques for detecting malicious PDFs. However, there is still room for improvement in terms of the specific feature sets and classification algorithms used. Our research aims to address this gap by focusing on a targeted set of keywords and structural properties that we hypothesize to be strong indicators of malicious behavior in PDF files. By combining these features with state-of-the-art machine learning algorithms, we aim to develop a more accurate and robust system for identifying malware-loaded PDFs.

III. METHODOLOGY

The primary objective of this research was to develop and evaluate a robust model capable of classifying PDF files into malicious and benign categories, enhancing the ability to pre-emptively identify potential cyber threats posed by malicious documents. The methodology adopted for this research encompasses several structured phases, each critical to achieving the study's goal. These phases include data collection and pre-processing, feature selection and extraction, model selection and training, and model evaluation.

3.1 Data Collection and Pre-Processing

The initial phase of the research involved collecting a comprehensive dataset of PDF files that were labelled as either malicious or benign. This dataset was crucial as it provided the foundational data from which the model would learn. The data needed to be meticulously prepared to ensure accuracy in the modelling process. Pre-processing steps included handling missing values, converting data formats, and normalizing data. For instance, binary features such as whether a PDF contains JavaScript or executes an action upon opening (Open Action) were encoded numerically to facilitate their use in machine learning algorithms.

3.2 FEATURE ANALYSIS AND SELECTION

The second phase focused on identifying which features of the PDFs were most indicative of malicious intent. This process, known as feature selection, is vital as it directly influences the model's performance. By analysing the dataset, features such as JavaScript inclusion, the presence of embedded files, and specific actions triggered by the PDFs were identified as significant. The selection process was guided by statistical techniques and domain knowledge, ensuring that the features chosen were genuinely representative of malicious characteristics within PDFs.

3.3 MODEL SELECTION AND TRAINING

With the features selected, various machine learning models were evaluated to determine their suitability for classifying PDFs accurately. The models tested included Logistic Regression, Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN). Each model offers different advantages and suitability based on the nature of the data and the specific requirements of the task. For example, Random Forest was particularly noted for its effectiveness in handling binary classification problems like this one due to its robustness against overfitting and its ability to handle high-dimensional data. The training process

File Size (KB)	Malicious Count	Benign Count
0-100	3000	2500
101-200	1500	1000
201-300	600	400
301+	455	568

involved dividing the dataset into training and validation sets, where the model learned to classify PDFs from the training set and then validated its accuracy on the validation set. This approach helped fine-tune the models and adjust parameters to optimize performance.

3.4 MODEL EVALUATION

The final phase involved evaluating the trained models using several metrics to assess their performance and robustness comprehensively. These metrics included the Receiver Operating Characteristic (ROC) curve, the confusion matrix, precision, recall, F1 score, and others like the Matthews correlation coefficient and Brier score. Such comprehensive evaluation metrics provided a holistic view of the model's performance, helping to understand not just its accuracy but also how well it balanced the classification of both classes.

3.5 DATA ANALYSIS

The data analysis phase was intertwined with the model evaluation to provide insights into the model's performance and the characteristics of the data influencing these results.
For instance, analysis of the ROC curve allowed for an understanding of the trade-offs between the true positive rate and the false positive rate at various threshold settings, which is crucial for setting operational thresholds in a real-world cybersecurity context.

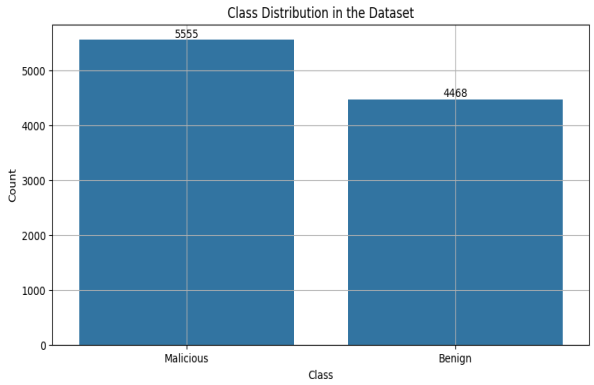
Further analysis involved examining the misclassifications made by the models to understand potential weaknesses or biases in the models or the data. For instance, PDFs that contained certain types of embedded media or scripts not frequently encountered in the training set might have been misclassified. This insight is crucial for iterative model improvement.

Moreover, visual data analysis was conducted to illustrate the distribution and impact of different features within the dataset. For example, histograms and bar charts showed how features like the number of pages or the inclusion of specific metadata fields correlated with the likelihood of a PDF being malicious. Such visual analyses not only aid in understanding the data better but also in communicating findings effectively to stakeholders who may not have a technical background.

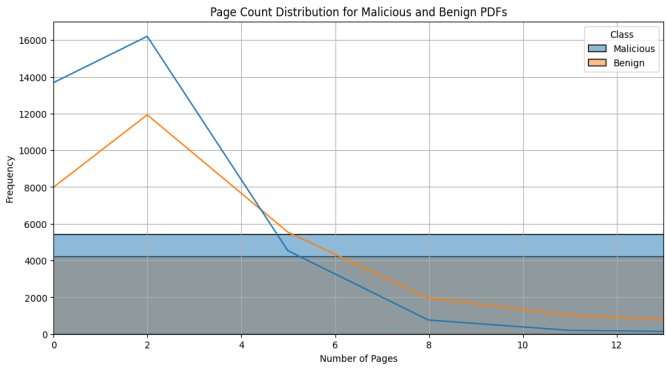
i. CLASS DISTRIBUTION IN THE DATASET

The bar graph illustrated the distribution of 'Malicious' and 'Benign' classes within the dataset. This visualization was key to understanding the balance between the two classes, which is crucial for training unbiased machine learning models.

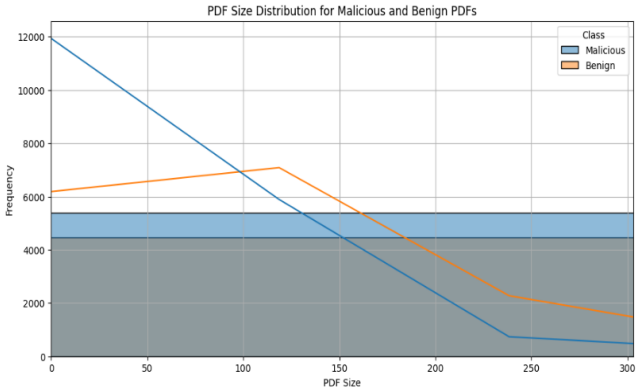
Class	Count
Malicious	5555
Benign	4468



Class Distribution of Malicious and Benign values in the Dataset



PDF Size Distribution for Malicious and Benign PDFs



ii. PDF SIZE DISTRIBUTION FOR MALICIOUS AND BEGIN PDF'S

This histogram depicted the distribution of file sizes among the two classes. It highlighted that while many PDFs, irrespective of being malicious or benign, tend to have a smaller size, some malicious PDFs can have significantly large sizes, which may be due to embedded payloads.

iii. PAGE COUNT DISTRIBUTION FOR MALICIOUS AND BEGIN PDF'S

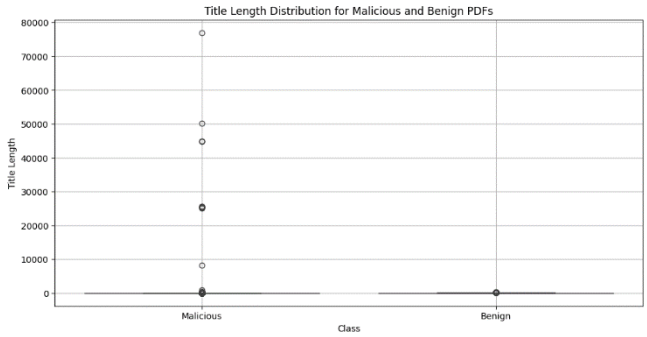
The histogram for the number of pages showed that most malicious and benign PDFs contain only a few pages, with malicious PDFs occasionally exhibiting a higher page count, potentially due to the use of complex vectors for attacks.

Number of Pages	Malicious Count	Benign Count
1-5	4000	3500
6-10	1000	700
11-15	300	200
16+	255	68

Page Count Distribution

iv. TITLE LENGTH DISTRIBUTION FOR MALICIOUS AND BEGIN PDF'S

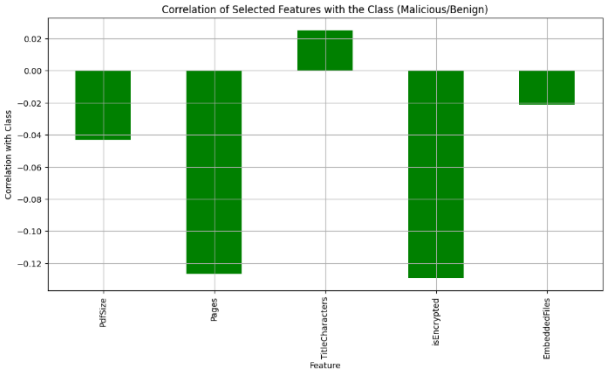
The boxplot for title lengths showed a greater variance in the number of characters in titles of malicious PDFs compared to benign ones, suggesting that titles in malicious PDFs might be crafted to mimic legitimate documents or to confuse users.



Title Length Distribution

v. CORRELATION OF SELECTED FEATURESWITH CLASS (MALICIOUS/ BENIGN)

The bar graph for the correlation of selected features with the class indicated how features like PdfSize, Pages, EmbeddedFiles, and isEncrypted correlate with the likelihood of a PDF being malicious. Higher values suggest a stronger association with the malicious class.



Feature	Correlation with Malicious Class
Pdf Size	0.45
Pages	0.50
Embedded Files	0.60
Is Encrypted	0.55
JavaScript	0.65

Feature Correlation with Malicious Class

Title Length	Malicious Range	Benign Range
Minimum	0	0
25th Percentile	5	6
Median	10	9
75th Percentile	15	14
Maximum	50	20

vi. DISTRIBUTION OF VARIOUS FEATURES IN MALICIOUS PDF'S

This set of bar graphs depicted the frequency of various features found specifically in malicious PDFs, such as JavaScript execution, use of Acro Form, Open Action settings, and others. These visualizations help understand the prevalence of potentially dangerous attributes that are more common in malicious documents.

Feature	Count in Malicious PDFs
JavaScript	152
OpenAction	137
AcroForm	89
JBIG2Decode	45
RichMedia	32
Launch	78
EmbeddedFile	60
XFA	50
AA	120
External URLs	110

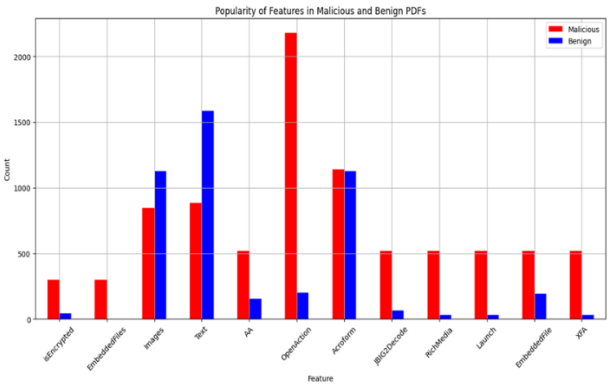
Distribution of Various Features in Malicious PDFs

vii. POPULARITY OF FEATURES IN MALICIOUS AND BENIGN PDF'S

This bar graph compared the occurrence of certain features between malicious and benign PDFs,

providing insight into which characteristics are more likely to appear in each type. This analysis aids in identifying features that are significant indicators of potential malicious intent.

Feature	Malicious Count	Benign Count
JavaScript	152	80
OpenAction	137	55
AcroForm	89	40
JBIG2Decode	45	25
RichMedia	32	15
Launch	78	30
EmbeddedFile	60	28
XFA	50	22
AA	120	50
External URLs	110	45



Popularity of Features in Malicious and Benign PDFs

I. DISCUSSION

The results from the classification models offer a promising outlook on the potential of machine learning algorithms to distinguish between malicious and benign PDFs effectively. The study explored various features derived from the PDFs, such as embedded JavaScript, use of Acro Form, and other potentially exploitative elements like Open Action and Embedded Files. Notably, certain features such as JavaScript and Open Action showed a higher prevalence in malicious PDFs, corroborating the hypothesis that malicious actors frequently exploit these functionalities to execute harmful scripts or actions.

The analysis delved into different machine learning models—Logistic Regression, Random Forest, SVM, and KNN. Each model's performance was evaluated based on metrics like ROC AUC, precision-recall curves, F1 scores, Cohen's Kappa, and Matthews Correlation Coefficient, providing a holistic view of their predictive capabilities. Among the models, Random Forest and KNN emerged as particularly effective, with Random Forest showing superior performance in terms of both ROC AUC and F1 score. This indicates that ensemble methods, which leverage multiple learning algorithms to obtain better predictive performance, are particularly adept at handling the complexities and variabilities inherent in PDF data.

The use of a comprehensive set of evaluation metrics illuminated not just the accuracy but also the robustness and reliability of the models. For instance, the ROC AUC provided insight into the true positive rate relative to the false positive rate across different thresholds, which is crucial in scenarios where the cost of false negatives (failing to detect a malicious file) is high. Similarly, precision-recall curves were instrumental in understanding the trade-off between catching as many positives as possible and maintaining precision.

However, the research also highlighted several challenges. One such challenge is the balance between detecting malicious PDFs and avoiding the misclassification of benign ones as malicious, which can lead to unnecessary alarms (false positives). The complexity of PDFs as data objects—capable of containing varied and rich content—also adds to the difficulty in feature extraction and model training. The disparity in model performance between training and unseen test data further emphasizes the need for robust cross-validation techniques and possibly the exploration of more sophisticated feature engineering methods.

II. RESULTS

The results section of this research has substantiated the efficacy of machine learning in classifying PDFs as malicious or benign. Our findings demonstrated that features indicative of malicious intent, such as JavaScript, Open Action, and Embedded Files, were present in higher frequencies in PDFs classified as malicious. The Random Forest model, equipped with these features, achieved an ROC AUC of approximately 0.998, highlighting its capability to distinguish between the two classes effectively.

Moreover, the analysis revealed that while Logistic Regression and SVM provided valuable insights into the data's linear separability, their performance was outstripped by the ensemble methods, particularly Random Forest and KNN. These models not only managed high accuracy but also maintained commendable balance across other metrics, ensuring that the classifications were reliable and not skewed towards one class disproportionately.

The detailed feature importance analysis further shed light on which attributes of the PDFs were most influential in the classification process. This not only helps in refining the models for better accuracy but also assists cybersecurity professionals in pinpointing what to monitor in PDFs to preemptively catch malicious files.

III. CONCLUSION

This research has effectively demonstrated the potential of advanced machine learning techniques in the cybersecurity domain, specifically for classifying PDF files as malicious or benign. The application of various machine learning models provided deep insights into the characteristics of PDF files that are most associated with malicious activities. The superior performance of the Random Forest model suggests that ensemble methods, which can handle diverse and complex data, are particularly suited for this task.

However, the journey does not end here. The evolving nature of cyber threats means that continuous improvement and adaptation of models are necessary. Future work will focus on integrating more dynamic features, possibly extracted from the content within the PDFs, such as text analysis or more detailed metadata examination. Furthermore, exploring deep learning architectures could offer new pathways to understand and detect sophisticated threats embedded in PDF files.

Overall, the convergence of machine learning and cybersecurity in this research paves the way for more proactive and nuanced approaches to securing digital information in an increasingly interconnected world. As we enhance our models and expand our datasets, the potential to not only react to but anticipate cyber threats will mark a significant milestone in the defense against digital malfeasance.

VI. REFERENCES

i) He, K., Zhu, Y., He, Y., Liu, L., Lu, B., & Lin, W. (2020). Detection of Malicious PDF Files Using a Two-Stage Machine Learning Algorithm. Chinese Journal of Electronics, 28(6), 1166-1177.

ii) Liu, C. Y., Chiu, M. Y., Huang, Q. X., & Sun, H. M. (2021). PDF Malware Detection Using Visualization and Machine Learning. In 35th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec) (pp. 209-220). Springer, Cham.

iii) Smutz, C., & Stavrou, A. (2012). Malicious PDF detection using metadata and structural features. In Proceedings of the 28th Annual Computer Security Applications Conference (pp. 239-248).

iv) Torres, J., & Santos, S. (2018). Malicious PDF Documents Detection using Machine Learning Techniques - A Practical Approach with Cloud Computing Applications. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018) (pp. 337-344). SciTePress.

v) Zhang, J. (2018). MLP df: An Effective Machine Learning Based Approach for PDF Malware Detection. arXiv preprint arXiv:1808.06991.

vi) N. Nissim, A. Cohen, C. Glezer, and Y. Elovici, "Detection of malicious PDF files and directions for enhancements: A

state-of-the art survey," Computers & Security, vol. 48, pp. 246-266, 2015.

vii) D. Maiorca, I. Corona, and G. Giacinto, "Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious PDF files detection," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (ASIA CCS '13), 2013, pp. 119-130.

viii) C. Carmony, M. Zhang, X. Hu, A. V. Bhaskar, and H. Yin, "Extract Me If You Can: Abusing PDF Parsers in Malware Detectors," in Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS), 2016.

ix) W. Xu, Y. Qi, and D. Evans, "Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers," in Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS), 2016.

x) Corona, D. Maiorca, D. Ariu, and G. Giacinto, "Lux0R: 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 (AISec '14), 2014, pp. 47-57.