

An Interpretable Hybrid Model for PDF Malware Detection Using CNN and Vision Transformer

Emmanuel Kwesi Baah 

Department of Computer Science

Lancaster University Ghana

Accra, Ghana

b.emmanuelkwesi@lancaster.edu.gh

Dominic Asamoah 

Department of Computer Science

Kwame Nkrumah University of Science and Technology

Kumasi, Ghana

dasamoah.cos@knust.edu.gh

Frederick Ledi

Department of Computer Science

Kwame Nkrumah University of Science and Technology

Kumasi, Ghana

fredledi25@gmail.com

Anthony Kingsley Sackey 

Department of Computer Science

Ho Technical University

Ho, Ghana

asackey@htu.edu.gh

Eric Agyekum Nana Boamah 

Department of Computer Science

Kwame Nkrumah University of Science and Technology

Kumasi, Ghana

eanboamah@st.knust.edu.gh

Richard Kofi Doh 

Department of Computer Science

Kwame Nkrumah University of Science and Technology

Kumasi, Ghana

rkdoh1@st.knust.edu.gh

Abstract—In recent years, malware has become more prevalent just as the internet has become more complex and ubiquitous. Malware is expanding at an alarming rate, and PDF documents are a prominent attack surface due to their programmable components and hardware independence. Various approaches have been employed to protect PDF(portable document format) files from malware attacks. Older approaches, such as dependency-based algorithms, addressed structural anomalies but struggled to handle black-box threats. Other existing techniques are known to be less efficient and perform poorly against new vulnerabilities. This study presents an efficient and novel hybrid solution (Convolutional Neural Network(CNN) + Vision Transformer (ViT)) for combating PDF malware using the PDFRep dataset. The study identified key players in detecting PDF malware, achieving an average of 0.99 across receiver operating characteristics, accuracy, recall, precision, and F1 score. Furthermore, the hybrid model obtained relatively low false-positive and false-negative rates. According to the results, the solution proved to be an improvement over others for detecting PDF malware.

Index Terms—PDF Malware Detection, Convolutional Neural Networks (CNN), Vision Transformers (ViT), Hybrid Deep Learning Model, PDFRep Dataset

I. INTRODUCTION

Malware is a broad spectrum of computer programs intended to interfere with, harm, or obtain unauthorized access to digital networks and systems [1]. With a variety of modes of operation, malware poses a grave danger in the digital world [2], [3]. Malware uses infection vectors such as documents, which rely on specific actions to infect the victim host through concealed and embedded malicious code [4], [5].

Consequently, the most researched infection vectors are malicious Portable Document Format (PDF) documents [5]. PDF malware is the practice of inserting malicious code into PDF documents to exploit flaws in software readers or lure users into performing destructive actions [6], [7]. PDF files are the primary conduits for document sharing due to their adaptability and consistent formatting across platforms [7]. Unfortunately, PDF files have become a major source of malware propagation [5]. This is because, in an effort to make PDF files more flexible and hardware-independent for sharing text and multimedia, Adobe incorporated programmable elements into its structure [7]. Thus, its structure and complexity afford attackers, to a certain degree, the means to hide, modify, or inject foreign code [5], [8]. Additionally, due to the growing number of unwary users and the limitations of contemporary antivirus software in identifying these injected foreign codes, attackers can potentially exploit victims through social engineering. Attackers have also exploited vulnerabilities in third-party applications used to access and read PDF documents. Singh et al. [7] assert that serious flaws in widely used document readers, such as Adobe Acrobat Reader, are still being discovered. Ultimately, it is well known that most detection techniques currently in use rely on signatures or heuristics, which are considered less efficient and perform poorly against new vulnerabilities [9], [10]. As attackers continue to refine their methods and make them more complex, efficient, and proactive solutions to protect PDF files against these evolving threats and breaches are necessary. Maiorca

et. al [11] concurred that ML algorithms have played crucial roles in delivering effective solutions against such threats. On the other hand, these solutions have not only lagged in accuracy and reliability due to the level of false alarms they generate, but have also mostly failed to reveal key players in the identification of PDF malware [12]–[14]. Furthermore, Liu et al. [15][15] suggested that the effectiveness of ML solutions in such contexts largely depends on the quality of the predictors and the dataset used. In this light, this study proposes to leverage ML to introduce a novel, efficient, and informative solution against PDF malware.

II. RELATED LITERATURE

There is a growing demand for improving detection models, as cyberattacks have become increasingly sophisticated and PDF files have become a common target. Previous research has demonstrated the security risks associated with PDF files due to threats such as adversarial evasion and obfuscation techniques, as well as their inherent nature [16], [17]. Several studies have addressed this issue using keyword analysis, adversarial resiliency, abstract interpretation, and image databases. There is still a need for greater scalability, robustness, and accuracy, especially when faced with evolving obfuscation techniques and adversarial counterattacks. With advances in AI, CNNs, and Vision Transformers, these models can address the issues mentioned earlier and increase accuracy.

A. Overview of the PDF and Its Malware

A PDF file is a tree-like collection of linked objects that consists of the header, body, cross-reference table, and trailer [15]. The first part contains information about the version of the PDF and is introduced with the percentage symbol. The body is the primary portion of the PDF file, comprising objects that specify all its functions and features. The elements of the file displayed to the user are in this body, with each section being presented by an object that contains various types of content, including images and fonts [5], [18]. The cross-reference table organizes the PDF file by listing its indirect objects and their locations in the file. It also allows random access by the user to the file's objects [17]. Finally, the trailer provides a link to the cross-reference table, allowing applications to locate it easily. Again, it contains an object of the dictionary type that bears relevant information about the file, such as its author, title, and details on encryption [15], [18]. Liu and Nicholas [15] confirmed that whenever a PDF file is modified, a new version number is generated and a new body, trailer, and cross-reference table are added to the original file. However, eliminating objects from the prior version may result in errors since the cross-reference table establishes a rigid boundary for every object. As a result, rather than eliminating features, an attacker is inclined to add them. Figure 1 briefly illustrates the structure of the PDF file as described by Maiorca et. al [11].

B. Approaches Employed Against PDF Malware

Over the years, the persistent challenge of effectively detecting PDF malware has led to a steady progression of

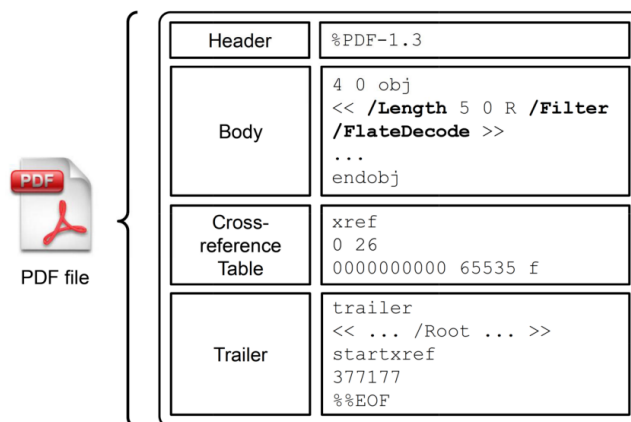


Fig. 1. The structure of the PDF [11]

research in the field, resulting in more advanced techniques for detecting PDF malware. Advanced machine learning (ML) and hybrid techniques have been applied to improve the accuracy, scalability, and robustness of detection. ML methods are typically evaluated based on the type of features they are engineered to detect, particularly meaningful attributes in the PDF structure metadata, embedded JavaScript properties, or file content patterns. They are strong in identifying patterns using statistical modeling and work well under known threats [8]. However, they are less adaptable to obfuscation or new attack types, as all the features must be handcrafted. ML algorithms have shown poor performance for real-time applications and problems involving large feature sets, as they are computationally expensive for large datasets [19]. It is also known that ML algorithms tend to fail in cases where some features are correlated, thus limiting the utility of this classifier in scenarios where PDF parts dynamically interact with each other [20]. On the other hand, deep learning (DL) algorithms are known to adapt to dynamic threats, as learning occurs directly through raw data [21]. DL models, such as Convolutional Neural Networks (CNN), also yield good performance by processing high-dimensional inputs [22]. Transformers also possess self-attention capabilities and complex deep learning architectures, making them particularly suitable for context-dependent relationships. Therefore, from each part of the input features, like the interdependencies between PDF objects, it can efficiently compute while maintaining a certain level of relevance. Additionally, it can be utilized for very large-scale processing and is increasingly used for detecting malicious PDFs. However, to have a sufficiently robust model for malware detection, hybrid models can be used to leverage the unique capabilities of each model and consequently mitigate the impact of other variables. These systems are particularly useful in addressing zero-day threats, as they combine supervised learning techniques with anomaly detection [23].

III. METHOD

This study considers a hybrid of the CNN and Vision Transformer (ViT) architectures to significantly and positively impact the detection of PDF malware. We achieve this by employing the explainability framework SHAP to unveil key insights from the proposed detection solution. Figure 2 summarizes the steps for this study.

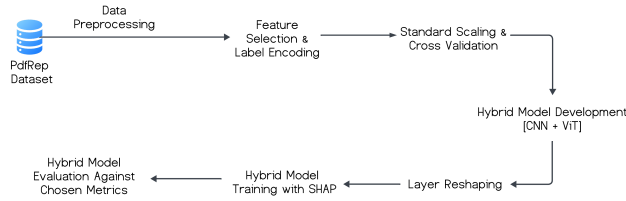


Fig. 2. Research Conceptual Framework

A. The PDFRep Dataset

The PDFRep dataset was employed to train and evaluate the hybrid model proposed in this study. In agreement with its creators, Liu et al. [15], the effectiveness of models largely depends on the data employed to train and test the model. The dataset for this study is suitable for the study as it is recent, extensive, and entails a comprehensive collection of malware samples and PDF files from the Contagio (18,210), CIC (28,791), and VirusShare (290,670). It further includes 155,583 benign samples from Govdocs. Overall, the study utilized a total of 322,761 and 170,477 adverse and benign PDF samples.

B. Feature Selection and Label Encoding

The study began by separating the features from the target variables and setting them as input and output data, respectively, for model training. Irrelevant columns were dropped. The LabelEncoder utility was used to convert categorical data into integers and map them accordingly.

C. Standard Scaling and Cross-validation

The features were normalized and scaled to prevent the dominance of certain features and mitigate biases. A cross-validation pipeline is set up, where the number of splits or folds is determined, shuffling is enabled, and the random number generator is also defined. This technique is employed to adequately and fairly resample the data by splitting it into defined subsets, also known as k-folds. For a repeated k-fold amount of time, the model is trained on the previous fold (k-1) and is tested on the remaining folds.

D. Hybrid Model Development

The study proceeded to define the hybrid solution, which consisted of the CNN and ViT architectures. With CNN's ability to extract features and ViT's strength in capturing contextual relationships, a combination of both holds great potential in combating PDF malware.

E. The CNN Framework

This is a variant of DL, primarily used for processing structured data to extract hierarchical features automatically [24], [25]. It constitutes layers including pooling, convolutional, and fully connected layers, each with a distinct role. The convolutional layer typically employs kernels to extract features and patterns from input data. More of these layers can be added to determine more intricate patterns in data. The pooling layers help reduce computational complexity and mitigate overfitting [26]. Finally, Alzubaidi et al. [27] highlighted that there exist dense layers that, through the features learned, make predictions that are displayed by the output layer of the architecture. Figure 3 summarizes the architecture of CNN.

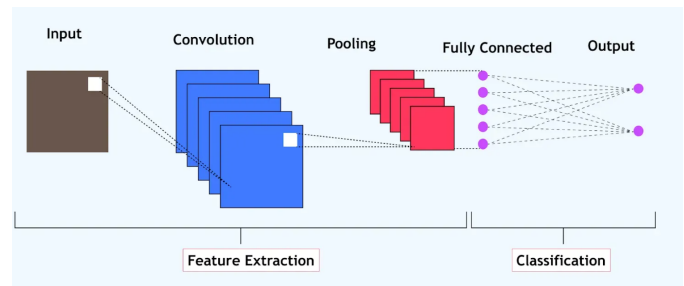


Fig. 3. The CNN Framework

F. The ViT Framework

This architecture primarily operates by dividing an input image or data into patches of fixed sizes, where each of them is flattened into a vector and embedded into a low-dimensional space. The sequence of these embeddings serves as input for the Transformer, which constitutes the self-attention and positional encoding mechanisms [28], [29]. The former mechanism ensures the capture of complex global contexts and long-range dependencies by weighing the relevance of various patches relative to one another [30]. This feature is what gives it the upper hand over CNN [29]. Figure 4 shows the architecture of the Transformer as defined by Vaswani et al. [30].

To initiate the development of the hybrid model, the shape of the input data, along with the number of output classes, was defined. With the defined input shape, the input layer was specified. The extraction of data patterns was achieved through the definition of two one-dimensional convolutional layers, each with 32 and 64 filters, respectively. Both layers have a kernel size of 3 and employ the ReLU activation function, with L2 regularization applied to penalize large weights and mitigate overfitting. Additionally, after each of these layers, a one-dimensional max pooling layer was added to down-sample the input representation. The data coming from this CNN is reshaped to ensure compatibility with the recipient ViT architecture. This is converted into a three-dimensional tensor of 64 features, with a defined sequence length and batch size. A ViT architecture with a multi-head attention layer of 4 heads and attention key and query dimensions of 64 is defined.

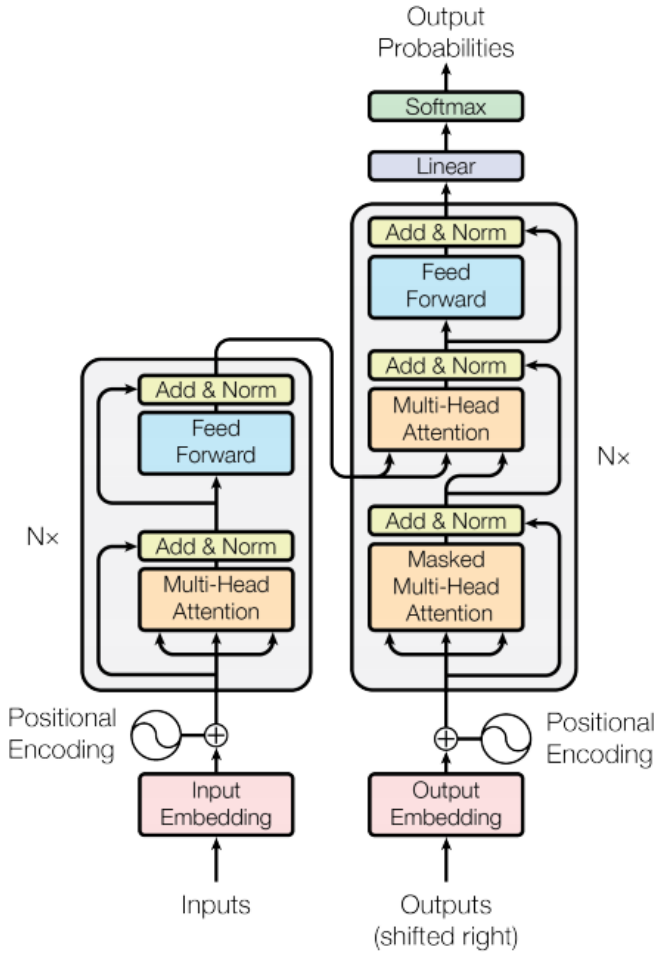


Fig. 4. The Transformer framework [30]

This framework was introduced to determine the intricate contextual relationships in the reshaped data accurately. Again, normalization and global average pooling layers are added afterward for improved stability during training and to reduce the sequence dimensions. Finally, the architecture is fully connected with a first dense layer of 128 neurons with both ReLU activation and L2 regularization mechanisms. A second dense layer with the same number of output classes and the SoftMax activation function is defined. Ultimately, a dropout layer with a rate of 0.5 is introduced to mitigate overfitting further. Both the input and output layers are combined and set up for training. In developing the hybrid architecture, the AdamW optimizer and learning rate parameters were defined. These were to discourage heavyweight values and control step sizes to minimize loss and overfitting. Moreover, the model was compiled with the Sparse Categorical Cross-entropy along with the accuracy metric to efficiently gauge the variance in the true class and the predicted probability distribution.

G. Layer Reshaping and Hybrid Model Training with SHAP

This section highlights the addition of a channel dimension to the training and validation datasets to ensure compatibil-

ity with the development architecture. Additionally, an early stopping mechanism was added that monitors the validation loss and returns the best weights. A waiting feature was set to stop training should the validation loss not improve. All this was to combat model overfitting further. Finally, the model is trained and validated on the appropriate data split at a specified epoch. The batch size during training was also set to 128. For validation, this study also employed the SHAP explainer framework to assess the power of features affecting the prediction.

H. The SHAP framework

It stands for Shapley Additive exPlanations. This framework assigns a value to each feature based on its significance in a prediction, thereby shedding insight into overall feature contributions. Its operation is derived from a cooperative game theory as stated by [31]. These values, known as the Shapley values, compute each feature's contribution to every possible subset for every prediction made. The Shapley value takes $v : 2^N \rightarrow \mathbb{R}$ as its input and produces attributions s_i per player $i \in N$ that sum up to $v(N)$ [32]. The Shapley value of a player i is given in equation 1.

$$S_i = \sum_{S \subseteq N_i} \frac{|S|! \cdot (|N| - |S| - 1)!}{N!} [v(S \cup i) - v(S)] \quad (1)$$

IV. RESULTS AND DISCUSSION

A. Model Performance

According to Table I, the hybrid model consistently yielded high scores across all metrics. We observed that the model was, on average, highly accurate in its predictions, as revealed by its recall, F1-score, and accuracy scores, despite a slight dip in precision. Additionally, the model proved reliable and reassuring because of its scores in FNR and FPR. Such low scores for both indicators proved that it could tell between the classes and avoid misclassifying them. According to the results, the model is well-suited for practical integration with real-world systems and applications.

TABLE I
AVERAGE HYBRID MODEL PERFORMANCE ON THE PDFREP DATASET

Average Recall	0.9910
Average Precision	0.9880
Average Accuracy	0.9910
Average F1 Score	0.9900
FNR	0.0110
FPR	0.0060

This study tackled the detection of PDF malware as a binary classification approach. The performance of the hybrid model against both classes is revealed in Table I. It can be observed that the model performed well in identifying instances of either class and correctly placing them. This great performance was also consistently replicated across the indicators in Table II.

The study further visualizes the performance of the model during training and validation, as shown in Figures 5 and 6.

TABLE II
AVERAGE BINARY CLASSIFICATION REPORT

Class	F1-score	Precision	Recall	Accuracy
0	0.9867	0.9798	0.9938	0.9907
1	0.9928	0.9967	0.9890	0.9907
Macro avg	0.9898	0.9882	0.9914	0.9907
Weighted avg	0.9907	0.9908	0.9907	0.9907

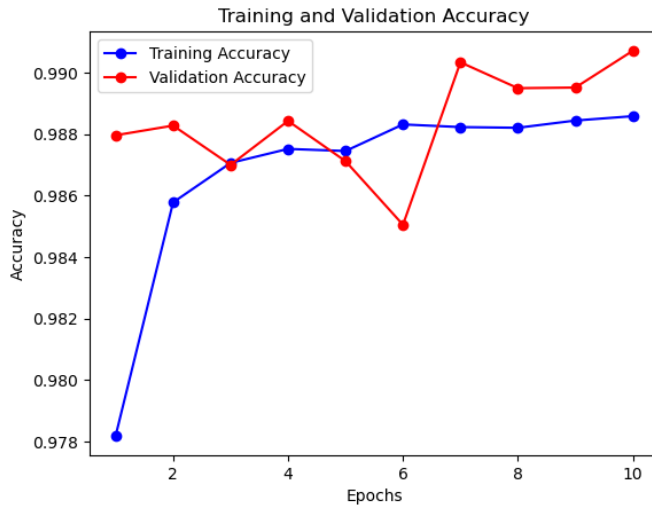


Fig. 5. Validation and Training Accuracies

The model exhibited a sharp rise in accuracy during training and maintained it throughout the last epoch. Yet, it is seen to have struggled in between the epochs, which goes to prove its learning process. Figure 5 illustrates the accuracy of the model during the validation phase. With short rises and falls during the beginning epochs and a drop at the sixth, the model was seen to struggle in applying what it had learned to new data. However, there was a sharp rise after the sixth epoch, which is maintained or even bettered at the tenth epoch. It can be concluded that the hybrid model, throughout the epochs, learned to classify new instances in the validation split at a higher accuracy. For the losses in training, the model did quite well. It drastically reduced losses from the first epoch and consistently maintained a steady decrease to the last. Moreover, Figure 6 illustrates that the model effectively reduced losses during the validation process. Overall, it is observed that the model struggled between the fourth and sixth epochs but subsequently made the correct predictions, resulting in minimal losses afterward.

Further projections of the strength of the model are presented in Figures 7 and 8. These figures reveal that the model is highly accurate in detecting data instances of the malicious class due to the well-learned patterns in features and a deep understanding of the context. Both figures show the hybrid model had peak levels of performance across recall, precision, accuracy, and F1 in both malicious and benign classes.

The hybrid solution obtained a perfect mean ROC with an area of 1. This reveals more about the overall performance of

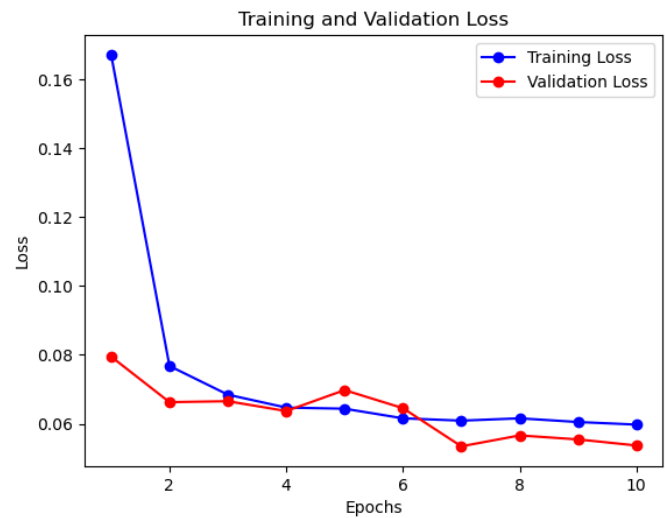


Fig. 6. Validation and Training Losses

the model. Figure 9 defines the trade-off between the TPR and FPR along defined thresholds. Should the model have obtained an area less than 0.5 (random guessing), it would have signified that it was inverted and unreliable. The computation of SHAP values in the proposed binary classification approach gives detailed explanations of the model's prediction for both classes.

The illustration in Figure 10 presents the confusion matrix score of the introduced CNN + ViT model. This is a key metric to further view the performance of the model.

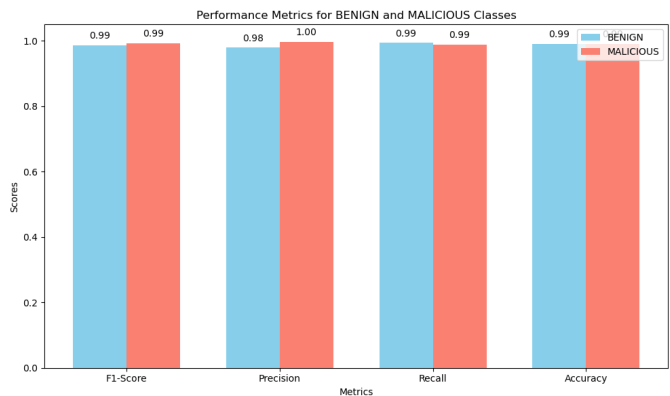


Fig. 7. Hybrid model binary classification output

The figure shows that the model correctly predicted the most negative (34125) and positive (63605) data instances. Yet, a relatively small set of these instances was misclassified and should be improved.

Figure 11 presents a summary plot, where features are listed vertically, and the average magnitude of the SHAP values is plotted horizontally. These features are ranked in descending order of importance (which is defined by the mean SHAP values) in driving predictions. The blue and red bars represent the average SHAP values when classes 1 and 0 are

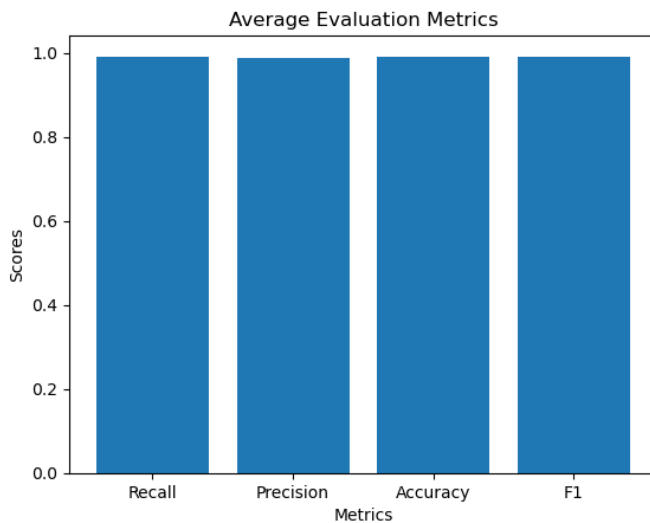


Fig. 8. Average evaluation of the hybrid model

predicted, respectively. It can be concluded that the first three factors from the top significantly influence the prediction of both classes, with `avg_degree` being the most crucial feature. According to the relative lengths of the bars, class 1 is significantly more affected by the `avg_degree`. At the same time, the other is somewhat more affected by the “density” variable, although it also has an impact on both class predictions. The `degree_assortativity` feature is influential in model predictions, but has a more balanced effect on both classes. Additionally, the `avg_clustering_coefficient`, `avg_children`, and `avg_shortest_path` provide a noticeable impact on predictions but are not as significant as the top-ranked features. The `num_edges`, `depth`, `var_children`, and `num_nodes` variables, however, make little to no contribution to the model decisions.

From the plot, `density`, `avg_shortest_path`, and `avg_children` proved to be dominant red or negative contributors. However, the `avg_degree` was strong enough with the base value to predict in favor of Class 1. This study further visualizes the highest-ranked feature to understand its relationship with its SHAP value, as shown in Figure 12. This SHAP dependence plot provides insights into the `avg_degree` feature and its relationship with the output of the model, as well as its interaction with `density`. The horizontal axis in the figure represents the actual values of this feature across the dataset, while the vertical axis represents its impact (SHAP values) on predictions. Each dot in the plot represents an individual observation from the dataset. A color bar is included to measure the density value concerning the chosen feature. Deducing from Figure 12, most low SHAP values (negative or close to zero) of `avg_degree` have little or no positive impact on the prediction. Additionally, a lower density further reduces the SHAP value. As the SHAP values of `avg_degree` increase, it indicates that this has a positive impact on the prediction. Again, with higher density, these values increase more. Yet, there exists an outlier where a high `avg_degree` value corresponds to a low

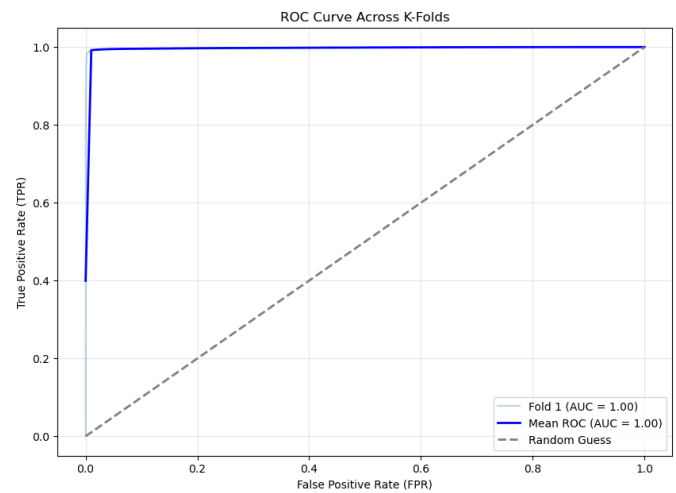


Fig. 9. ROC of the Proposed Model

SHAP value. Overall, it can be inferred that the `avg_degree` feature, on average, has a positive impact on the prediction as its SHAP value increases. The `avg_degree` feature was found to have a greater positive influence at higher densities, but its effect was diminished, and SHAP values declined at lower densities. Finally, the SHAP values of `avg_degree` rose and remained continuously positive, indicating that the model primarily relied on it to make decisions.

Furthermore, this study utilizes the SHAP force plot, as shown in Figure 13, to visualize and explain how the hybrid model arrives at a specific prediction for an instance. This combines SHAP values to illustrate how each feature contributes to pushing the prediction of the model toward one class or the other.

Firstly, the base value, which denotes the model’s average prediction across the dataset without any features, was defined. Then, the SHAP values quantify the contribution of each feature, assigning positive and negative values that push the prediction toward the necessary classes. Essentially, the prediction begins at the base value, with contributors in red pushing the prediction downward (toward Class 0) and those in blue pushing it upward (toward Class 1), as shown in Figure 13.

B. Overall Comparison with Related Works

This section compares the performance of the hybrid solution to that of existing solutions. For ease of reference, Table III presents the various authors of these solutions and reports their outcomes.

V. DISCUSSION

Torres and De Los Santos [33] developed a privacy-enforced framework based on machine learning for detecting malicious PDF documents, with a focus on those containing embedded JavaScript code. The dataset consisted of 1,712 PDF samples collected from various sources. Table III demonstrates that the proposed hybrid solution outperformed theirs across all

TABLE III
THE PROPOSED HYBRID MODEL AGAINST OTHER SOLUTIONS

Authors	Recall	F1 score	FPR	Precision	Accuracy
Torres & De Los Santos [33]	0.9670	0.9600	-	-	0.9600
Chen et al. [13]	-	-	0.0056	-	0.9970
Falah et al. [6]	(PDFiD) 0.9670 (PeePDF) 0.9630	0.9750 0.9740	-	0.9840 0.9860	0.9740 0.9730
Abu Al-Haija et al. [16]	0.9890	0.9880	-	0.9880	0.9880
Alshamrani [12]	-	0.9860	-	-	-
Hossain et al. [34]	0.9910	0.9910	-	0.9910	0.9910
Proposed Model	0.9910	0.9900	0.0060	0.9910	0.9910

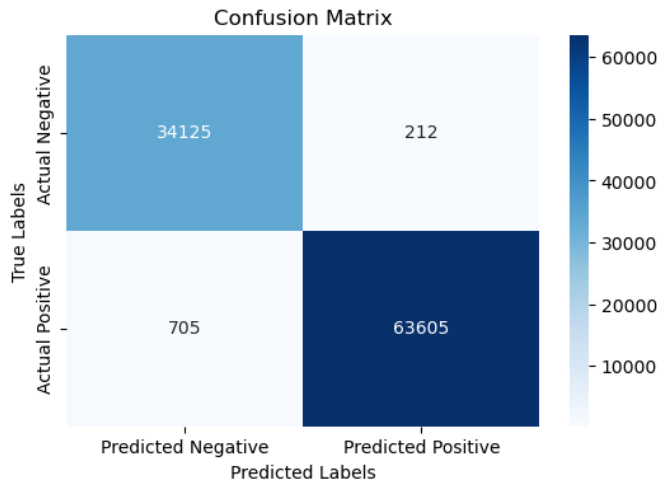


Fig. 10. Average confusion matrix score

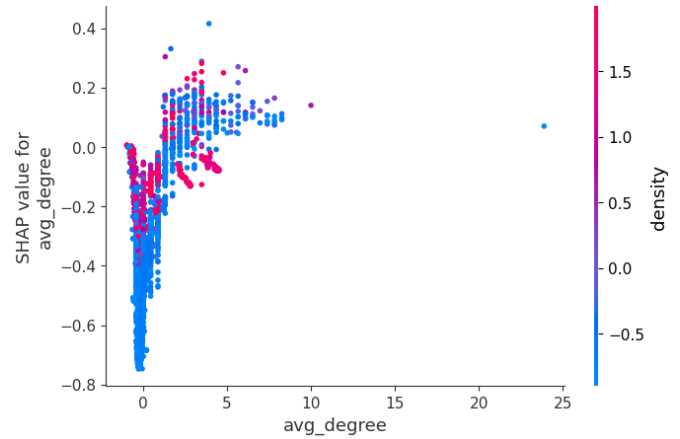


Fig. 12. SHAP Dependence Plot

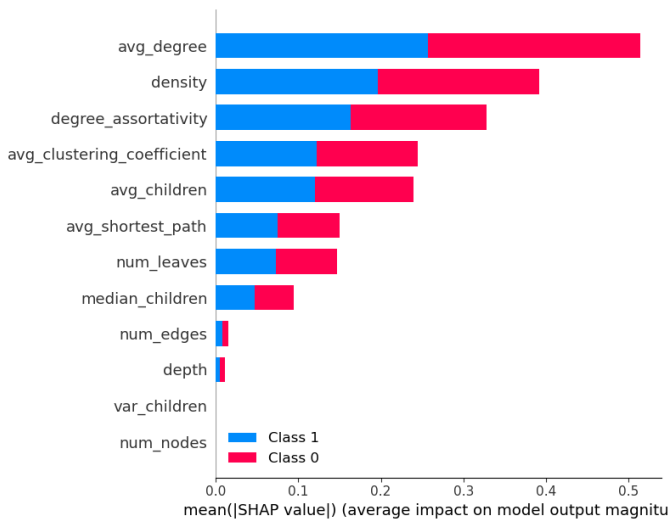


Fig. 11. SHAP summary bar plot

metrics, including the ROC (0.98). Chen et al [13] employed a novel distance metric for PDF tree structures and state-of-the-art verifiably robust training methods. However, Chen et al [13] only addressed simple robustness properties and ignored more complex ones that could have led to stronger security

guarantees. Additionally, their most optimal solution, Robust A+B, is seen to have performed negligibly better than the proposed one, despite using a few metrics for its evaluation.

Falah et al. [6] enhanced feature engineering and data-driven mechanisms to combine features from PDFiD and PeePDF analysis tools, along with derived complementary features. With samples from VirusTotal, they evaluated their approach, with RF standing as the most efficient. However, the proposed CNN+ViT architecture steadily outperformed it. The same can be said for the works of Abu Al-Haija et al. [16] and Alshamrani [12], even though Alshamrani [12] barely had enough metrics to compare with. The dataset employed by Abu Al-Haija et al. [16] is relatively older, thus providing the proposed solution with an additional edge. Hossain et al. [34] developed a dataset comprising 15,958 PDF samples from which features were extracted for training ML algorithms. The random forest(RF) classifier performed very well on the combined feature set, as seen in Table III. However, because it was trained and assessed using a larger and more recent dataset as well as a greater variety of metrics, the suggested approach is objectively superior. Liu and Nicholas [15] presented an approach that utilized a minimalist feature set based on the graph-based structural characteristics of PDF files. Their approach is seen to have done considerably well. Yet the proposed study has led to the discovery of even fewer sets of features that can yield even better results across the

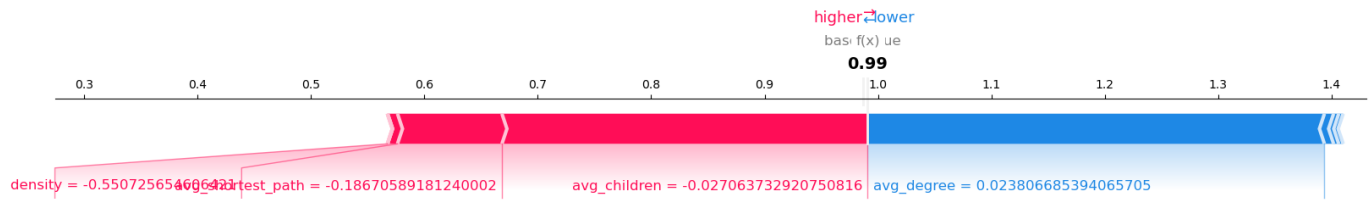


Fig. 13. SHAP Force Plot

necessary metrics.

A. Limitations and Future Works

This study utilized only the PDFRep dataset for both model training and evaluation phases. Although various mechanisms against model overfitting were employed, the sample size of the dataset was maintained without attempts to augment benign samples to reduce present bias. Also, deploying and improving the proposed solution in real systems are the key next steps for this study. With the insights on predictor contributions streamlined, focus will be given to the most influential ones to maximize feature engineering and reduce computational overheads. Finally, the single outlier identified in the dependence plot may be worth further investigation.

VI. CONCLUSION

The study introduced a solution against PDF malware, which proved to be reliable and consistent in making accurate predictions. This is assured by the low FPR score and the perfect ROC area score it recorded. Several SHAP plots were employed to visualize and help explain how the proposed hybrid model arrived at its predictions. It was revealed that the *avg_degree*, *density*, and *degree_assortativity* features were very potent in the model's decision-making process in classifying data instances. Some other features, such as *num_edges* and *depth*, were found to make only a minor contribution. Additionally, it was found that each feature had a greater impact on one class than on the other. The study also proved the relationship between specific features and how the rise in one made the other more positively influential. This further demonstrates that the model primarily relied on specific predictors to make decisions.

REFERENCES

- [1] A. AliAhmad, D. Eleyan, A. Eleyan, T. Bejaoui, M. F. Zolkipli, and M. Al-Khalidi, "Malware detection issues, future trends and challenges: A survey," in *2023 International Symposium on Networks, Computers and Communications (ISNCC)*. Doha, Qatar: IEEE, Oct 2023, pp. 1–6.
- [2] I. Kara, "A basic malware analysis method," *Computer Fraud & Security*, vol. 2019, no. 6, pp. 11–19, Jan 2019.
- [3] L. Chen, C. Xia, S. Lei, and T. Wang, "Detection, traceability, and propagation of mobile malware threats," *IEEE Access*, vol. 9, pp. 14 576–14 598, 2021.
- [4] S. R. Gopaldinne, H. Kaur, P. Kaur, G. Kaur, and Madhuri, "Overview of pdf malware classifiers," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*. London, United Kingdom: IEEE, Apr 2021, pp. 337–341.
- [5] D. Maiorca, B. Biggio, and G. Giacinto, "Towards adversarial malware detection: Lessons learned from pdf-based attacks," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–36, July 2020.
- [6] 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, Feb 2021.
- [7] P. Singh, S. Tapaswi, and S. Gupta, "Malware detection in pdf and office documents: A survey," *Information Security Journal: A Global Perspective*, vol. 29, no. 3, pp. 134–153, May 2020.
- [8] C. Carmony, M. Zhang, X. Hu, A. Vasisht Bhaskar, and H. Yin, "Extract me if you can: Abusing pdf parsers in malware detectors," in *Proceedings 2016 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA: Internet Society, 2016.
- [9] A. Lemay and S. P. Leblanc, "Is eval() evil : A study of javascript in pdf malware," in *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*. Nantucket, MA, USA: IEEE, Oct 2018, pp. 1–10.
- [10] A. Jordan, F. Gauthier, B. Hassanshahi, and D. Zhao, "Unacceptable behavior: Robust pdf malware detection using abstract interpretation," in *Proceedings of the 14th ACM SIGSAC Workshop on Programming Languages and Analysis for Security*. London, United Kingdom: ACM, Nov 2019, pp. 19–30.
- [11] D. Maiorca, A. Demontis, B. Biggio, F. Roli, and G. Giacinto, "Adversarial detection of flash malware: Limitations and open issues," *Computers & Security*, vol. 96, p. 101901, Sept 2020.
- [12] S. S. Alshamrani, "Design and analysis of machine learning based technique for malware identification and classification of portable document format files," *Security and Communication Networks*, vol. 2022, pp. 1–10, Sept 2022.
- [13] Y. Chen, S. Wang, D. She, and S. Jana, "On training robust pdf malware classifiers," 2019.
- [14] J. Gu, R. Kong, H. Sun, H. Zhuang, F. Pan, and Z. Lin, "A novel detection technique based on benign samples and one-class algorithm for malicious pdf documents containing javascript," in *International Conference on Computer Application and Information Security (ICCAIS 2021)*, Y. Lu and C. Cheng, Eds. Wuhan, China: SPIE, May 2022, p. 62.
- [15] R. Liu and C. Nicholas, "A feature set of small size for the pdf malware detection," 2023.
- [16] Q. Abu Al-Haija, A. Odeh, and H. Qattous, "Pdf malware detection based on optimizable decision trees," *Electronics*, vol. 11, no. 19, p. 3142, Sept 2022.
- [17] D. Maiorca, "A structural and content-based approach for a precise and robust detection of malicious pdf files," in *Proceedings of the 1st International Conference on Information Systems Security and Privacy (ICISSP)*. Angers, France: SCITEPRESS, 2015, pp. 27–36.
- [18] D. Feng, M. Yu, Y. Wang, C. Liu, and C. Ma, "Detecting malicious pdf files using semi-supervised learning method," 2017, publication venue missing in source.
- [19] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers: A case study on pdf malware classifiers," in *Proceedings 2016 Network and Distributed System Security Symposium (NDSS)*. San Diego, CA: Internet Society, 2016.
- [20] Z. S. Borno, N. Sakib, and S. S. Anwar, "Performance analysis of ensemble machine learning algorithms in pdf malware detection," in *2023 IEEE 9th International Women in Engineering (WIE) Conference*

on *Electrical and Computer Engineering (WIECON-ECE)*. Thiruvananthapuram, India: IEEE, Nov 2023, pp. 195–200.

- [21] 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)*. South Padre Island, TX, USA: IEEE, June 2019, pp. 108–114.
- [22] Y. Wang, Y. Sun, Q. Gao, and S. Sun, “Multi-feature fusion malicious pdf detection based on cbam,” in *2024 5th International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*. Guangzhou, China: IEEE, May 2024, pp. 148–155.
- [23] S. Y. Yerima, A. Bashar, and G. Latif, “Malicious pdf detection based on machine learning with enhanced feature set,” in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*. Al-Khobar, Saudi Arabia: IEEE, Dec 2022, pp. 486–491.
- [24] B. Alabsi, M. Anbar, and S. Rihan, “Cnn-cnn: Dual convolutional neural network approach for feature selection and attack detection on internet of things networks,” *Sensors*, vol. 23, no. 14, p. 6507, July 2023.
- [25] T. Wood, “Convolutional neural network,” DeepAI, accessed: Aug. 11, 2025. [Online]. Available: <https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network>
- [26] “What is a convolutional neural network (cnn)? — definition from techtarget,” Search Enterprise AI, accessed: Aug. 11, 2025. [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>
- [27] L. Alzubaidi *et al.*, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar 2021.
- [28] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, “Vivit: A video vision transformer,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct 2021, pp. 6816–6826.
- [29] D. Shah, “Vision transformer: What it is & how it works [2024 guide],” V7 Labs, accessed: Aug. 11, 2025. [Online]. Available: <https://www.v7labs.com/blog/vision-transformer-guide>
- [30] A. Vaswani *et al.*, “Attention is all you need,” 2017, updated Aug. 02, 2023.
- [31] G. Van Den Broeck, A. Lykov, M. Schleich, and D. Suciu, “On the tractability of shap explanations,” *Journal of Artificial Intelligence Research*, vol. 74, pp. 851–886, June 2022.
- [32] M. Sundararajan and A. Najmi, “The many shapley values for model explanation,” in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, vol. 119. JMLR.org, July 2020, pp. 9269–9278.
- [33] J. Torres and S. De Los Santos, “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)*. Funchal, Madeira, Portugal: SCITEPRESS, 2018, pp. 337–344.
- [34] G. M. S. Hossain, K. Deb, H. Janicke, and I. H. Sarker, “Pdf malware detection: Toward machine learning modeling with explainability analysis,” *IEEE Access*, vol. 12, pp. 13 833–13 859, 2024.



Emmanuel Kwesi Baah is a Ghanaian born in 1992. He is currently a lecturer at Lancaster University, Ghana, and a doctoral candidate at the Department of Computer Science at Kwame Nkrumah University of Science and Technology (KNUST), Ghana. He received his M.Phil. in Computer Science from KNUST. His research interest includes the application of deep learning algorithms for computer vision problems, cybersecurity, and business intelligence.



Processing and Network Security.

Dominic Asamoah is a Ghanaian born in 1960. He is currently a senior lecturer at the Department of Computer Science, KNUST. He received his B.Sc. in 1986, M.Phil. in 2010 and Ph.D. degrees in Computer Science, all from the Kwame Nkrumah University of Science and Technology (KNUST) in 2017. His research interests include Image



Frederick Ledi is a Ghanaian born in 1978. He earned his B.Sc. in Information Technology from the Ghana Institute of Management and Public Administration (GIMPA), Accra, Ghana, in 2008. He has completed his MSc. from KNUST. His research was focused on building transparent and trustworthy deep learning models for secure systems.



Anthony K. Sackey is a Ghanaian born in the year 1987. He received an M.Sc. degree in Information Technology from Kwame Nkrumah University of Science and Technology, Ghana. He is currently a lecturer at the Department of Computer Science, Ho Technical University, Ghana, where he has been working since 2020. He is also currently pursuing a PhD in Information Systems at the University of South Africa. His research interests include Machine Learning, Educational Technology, Human-Computer Interaction and Learning Analytics.



Eric A. N. Boamah is a Ghanaian who was born in 1999. He graduated from Kwame Nkrumah University of Science and Technology (KNUST), Ghana, in 2022, where he pursued a B.Sc. Computer Science. He has been working as a private software developer and is currently working with SmartTransyt as a field technician. His areas of interest are computer software development and practical applications of machine learning.



Richard K. Doh is a Ghanaian born in 1999. He earned his B.Sc. and M.Phil. in Computer Science from the Department of Computer Science, Kwame Nkrumah Uni-

versity of Science and Technology (KNUST), Kumasi, Ghana, in 2022. He is currently serves as a Research Assistant at KNUST. His research focuses on computer vision, sign language translation, and real-time object detection.