National College of Ireland

National College of Ireland

Project Submission Sheet

Student Name:	Ansh Ashwini Jain		
Student ID:	x23308320		
Programme:	MSc in Cybersecurity	**Year:**	2025
Module:	Artificial Intelligence & Machine Learning		
Lecturer:	Mr. Abdul Shahid		
Submission Due Date:	2nd May 2025		
Project Title:	Exploring Malware Detection in PDF's using CNN & Explainable AI Techniques		
Word Count:			

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Ansh Ashwini Jain
Date:	2nd May 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

Your Name/Student Number	Course	Date
Ansh Ashwini Jain/23308320	MSc in Cybersecurity	2nd May 2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA		

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

EXPLORING MALWARE DETECTION IN PDF'S USING CNN & EXPLAINABLE AI TECHNIQUES

Ansh Ashwini Jain
23308320
MSc in Cybersecurity
National College of Ireland

Abstract

This research aims to develop a deep learning model using Convolutional Neural Network (CNN) to detect the presence and effectiveness malware present in Portable Document Format (PDF) Files. By leveraging CNN's capabilities of automatically learning complex patterns from raw input data, the model seeks to identify subtle indicators of malicious behaviour present in the structure and content of PDF's. To build more trust in the detection system, Explainable AI techniques like LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive Explanations) are employed. These tools help discover what features contribute to a PDF being classified as malicious or benign. The integration of deep learning with interpretability not only improves detection accuracy but also provides valuable insights into the characteristics of PDF malware, aiding cybersecurity analysts in threat investigation.

Keywords: PDF, Malware, CNN, XAI

1. Introduction

Nowadays, all of us make use of something we like to call a Portable Document Format (PDF). Due to their complex structure and widespread adoption, PDF's flexibility makes it the most used file format in the world. Malicious attackers on the other hand could embed malicious content in such PDF's while maintaining to evade traditional security measures. There are multiple enhanced techniques like rule-based detection or signature based too, but they often fall short and need a dire change. To address this challenge, this research explores the application of Convolutional Neural Networks (CNN) for the detection of malware. CNN's are widely famous for their capability of extracting features and learning patterns from raw data, which can prove to be an advantage for the metadata in PDF's (Snehal et al., 2024). To make the model more trustworthy and transparent, Explainable AI tools like LIME and SHAP are leveraged with the detection framework. These tools act like a guidance for insights into the model's decision-making process by highlighting the features that most influence classification outcomes. Hence the question that we will be targeting in this study is:

Research Question: How can a Convolutional Neural Network (CNN) be used to detect PDF malware, and how can explainable AI techniques such as SHAP and LIME help in understanding the structural and feature-based indicators that contribute to the maliciousness of a PDF file?

This paper is divided into 3 sections with each section having their own set of subsections. The very first section is the Literature Review which covers the research taken place for the suggested Research Question

till now and how our work differs from it. Next comes the methodology which consists of explanation of all of the machine learning models and the artificial intelligence algorithms used and evaluated for the research question. Followed by that is the findings of the research and summary of the model evaluated signing off with conclusions and future work.

2. Literature Review

Rajagopal, Gaur and P (Rajagopal et al., 2023) are presenting an approach to extract features using the PDFiD tool, and further employ SHAP, LIME and interpret decisions based on the usage of explainable AI. The paper provides a good analysis of the tools, but the explanation lacks a few more support diagrams and there is no usage of a deep neural network. The dataset used is the same dataset presents to be a start of a research paper.

Sowan, Fasha, Matar, Aburub, Al-Khaldy, Nofal and Al-Jaber (Sowan et al., 2024) start off with a very good explanation of the structure of a PDF and have research focused on creating a hybrid Random Forest and KNN models. The research is helpful in making comparisons of the data with work present and work of ours.

For (Phuoc et al., 2024) presented by Phuoc, Tan and Cam, their research involved working on the same dataset leveraging algorithms of XGBoost and AdaBoost and implementing SHAP with it. Their models turned out to be giving out a good score which can be used in comparison with our work, plus the SHAP model they provided can be compared for comparison with machine algorithm outputs and deep neural network outputs.

On the same dataset, (Yudin et al., 2024) where Yudin, Song, Serban and Chadha make use of various kinds of transformations and the Monte Carlo Tree Search (MCTS) algorithm which turns out to be a great technique for evaluation of PDF Malware Detection. This research acts as an alternative to the work I presented and can be used for future scope.

3. Research Method & Evaluation

A. Dataset Collection

This dataset known as EvasiveMal2022 (“CIC-Evasive-PDFMal2022 | Datasets | Canadian Institute for Cybersecurity | UNB,” n.d.) contains of 10,023 records with 4468 being benign records and 5555 being malicious. All of these files are a collection of records from Contagio and Virus Total, which were then further sampled from different clusters and this combination is now what we use to do our work on. Of all the PDF's there are 37 statistical features extracted out of which we work on 33. The features are divided into general and structural, with a greater number of features for the latter. Description of the PDF like its size, content information and more come under the general bracket, while structural includes skeletal information.

B. Cleaning, Pre-Processing and Splitting

Initial exploration of the file revealed most of the features being of type category and float32, with no missing values present in them. The class column was text which I mapped to 0's and 1's for benign and malicious records so as to perform binary classification. To convert the categorical columns into numerical ones, I decided to use the Label Encoder feature to transform those values. Then after creating a list of categorical columns, based on the Class Column I decided to perform target encoding and calculated the mean values for easier calculations. After processing further cleaning, I dropped the FileName and Class Columns from the main dataset and this became my X set, with my Class Column being the Y set. For Feature Normalization, I decided to go ahead with Standard Scaler after splitting on my data in X as it helps speed up training and leads to much stable convergence.

C. Logistic Regression

My reason to use Logistic Regression was just to understand Feature Importance with coefficient values. Used mainly for binary classification tasks, it makes use of the sigmoid function to map predicted values between 0 and 1. The model learns the relationship between features and the target variable by estimating coefficients using maximum likelihood estimation. Logistic Regression is simple, interpretable, and works well when the classes are linearly separable. I applied Regularization to prevent overfitting and see if the model is learning the outliers as well as the noise too but the Train & Test Accuracy scores were very similar to each other (Hossain et al., 2024).

According to the model, the topmost feature which increased malicious probability was Javascript with a score of 1.6448, followed by Images with a score of 1.4721. Whereas on the other hand, which helped decide that the file is benign was Stream with a score of -0.9649 followed by is Encrypted with -0.6965. One thing to notice here was the EmbeddedFile being a part of the benign record decider with a score of -0.2261 which we are going to see change as we go ahead.

D. XGBoost

Extreme Grade Boosting, also known as XGBoost builds an ensemble of decision trees in a sequential manner, where each new tree corrects the errors of the previous ones. XGBoost is known for its speed, scalability, and high predictive performance.

The model performed exceptionally well with high scoring.

E. Random Forest

Random Forest Classifier combines multiple decision trees to improve classification accuracy. It builds a "forest" of decision trees, in this case 100 trees, each trained on a random subset of the data with

bootstrapping. At each split, it selects a random subset of features, enhancing diversity among trees and reducing overfitting. The final prediction is made by aggregating the predictions from all trees (Borno et al., 2023).

F. Convolutional Neural Network

I then proceeded to train a Deep Neural Network model for binary classification using Keras. Using a Sequential model, where each layer passes data to the next in sequence, with input size 31 based on the dataset. An Input layer with 128 neurons using the Rectified Linear Unit activation function which helps in learning more complex patterns. Then there are four hidden layers with neurons 64, 64, 32, and 16 respectively as to create a 'bottleneck' architecture. This is done so that the model can focus on the most important features as it progresses. This helps in avoiding overfitting and ensuring efficient learning. Then two more features which are consistent in all the layers are Dropout and Batch Normalization. The Dropout function randomly sets 30% of the layer's output to zero during training to ensure no overfitting occurs whereas Batch Normalization, normalizes the values of mean and variance to 0 and 1 respectively, which in turn increases the speed of training and enhances stability. The output layer is a single neuron which uses sigmoid activation function that is best used for binary classification. Optimization technique used is Adam with binary cross-entropy for loss function. The model is trained for 50 epochs, with a batch size of 32. Validation split of 20% is used to evaluate performance after each epoch.

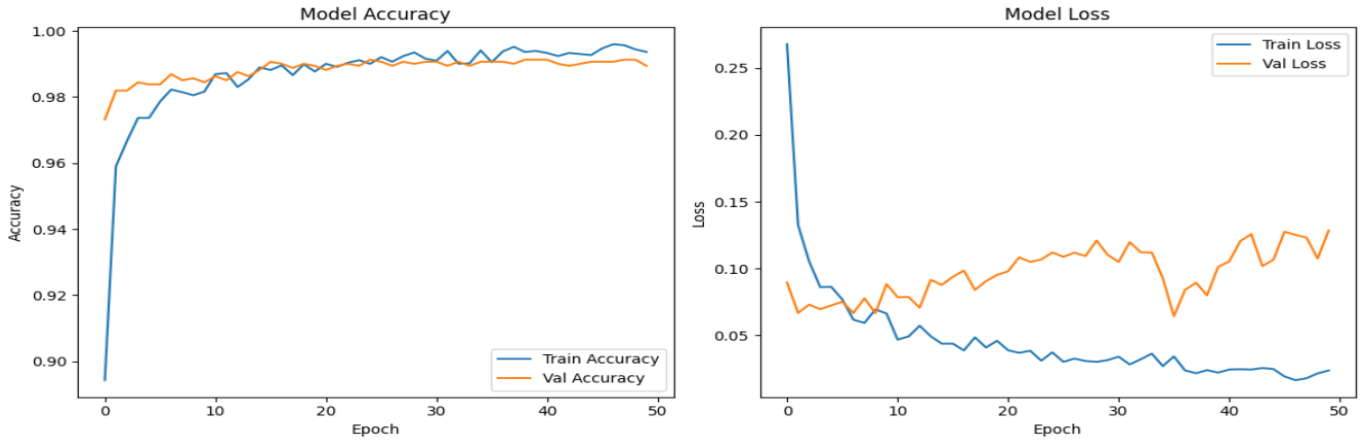

Fig. 1. CNN Model Accuracy and Loss Graphs

As seen in Fig. 1., Both Training and Validation accuracy increases rapidly in the first few epochs, with training accuracy increasing and staying close to 1 while validation accuracy increasing and stabilizing around 0.985.

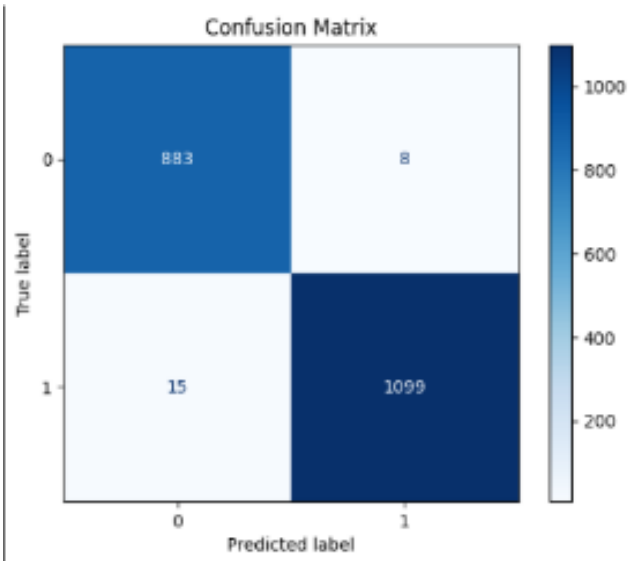

Fig. 2. Confusion Matrix for CNN

From the confusion matrix as shown in Fig. 2., we understand True Positives (TP) being 1099, True Negatives (TN) being 883, False Positives (FP) as 8 and False Negatives (FN) as 15. hence, we derive the following metrics from this data:

- Accuracy = (TP + TN) / Total
  = (1099 + 883) / (1099 + 883 + 15 + 8) ≈ 98.96%
- Precision (for class 1) = TP / (TP + FP)
  = 1099 / (1099 + 8) ≈ 99.28%
- Recall (Sensitivity) = TP / (TP + FN)
  = 1099 / (1099 + 15) ≈ 98.65%
- F1 Score = 2 × (Precision × Recall) / (Precision + Recall)
  ≈ 98.96%
- Matthews Correlation Coefficient (MCC) = = (TP * TN - FP * FN) / sqrt((TP + FP)(TP + FN)(TN + FP)(TN + FN)) = 970397 / 994195.9
  ≈ 0.9761

	Accuracy	F1-Score	Precision	Recall	R2 Score	MAE Score
Logistic Regression	97.6 %	97.3 %	98.4 %	96.4 %		
XGBoost	99.5 %	99.3 %	99.6 %	99.2 %	96.5 %	0.84 %
Random Forest	99.6 %	99.3%	99.5 %	99.3 %	96.9 %	0.75 %
CNN	98.9 %	98.9 %	99.2 %	98.7 %		

G. SHAP

Shapley Additive exPlanations (SHAP) are known to use a game theoretic approach, which helps explain the prediction of the models. It does so by assigning each feature with Shapley value, which quantifies how much of a contribution it has for that specific PDF file. These values are calculated based on a concept called fair contribution derived from cooperative game theory. The main power of this algorithm lies in its global interpretability and local interpretability.

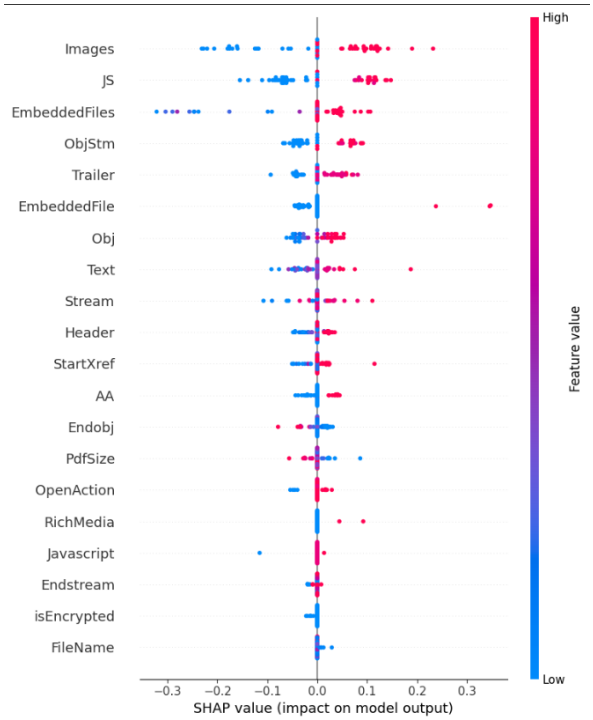

Fig. 3. SHAP features explained

As we can see from Fig. 3., it has 2 axes, X Axis has SHAP values which are positive and negative, representing malicious and benign respectively; Y Axis has Features, ordered by importance with each dot representing a PDF File. We understand that Images, JS and EmbeddedFiles display clear indicators of malicious PDF's. One thing to notice here is PdfSize and FileName having less impact or mixed effects on the prediction (Bragança et al., 2023).

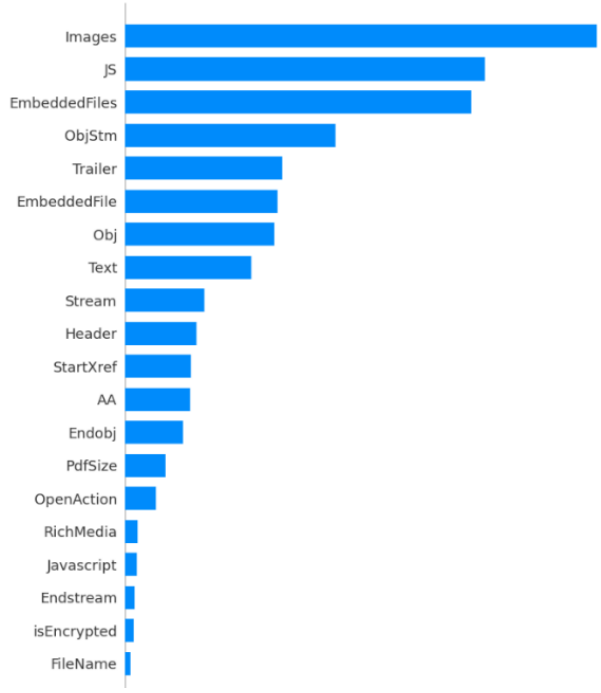

Fig. 4. SHAP Forceplot

Next, we have is a force plot summary as shown in Fig. 4., which displays the distribution of SHAP values for individual features across the entire dataset in terms of model output probability. It visualises how each feature shifts the prediction probability towards one of the two classes. Some of the insights we get from these is that for Text, EmbeddedFile, JS and Images are such features that when they have high values, there is a tendency of the prediction to be pushed towards the malicious side. ObjStm, Trailer and Stream show a similar pattern of contributing positively to the malicious class when present (Rahman et al., 2023).

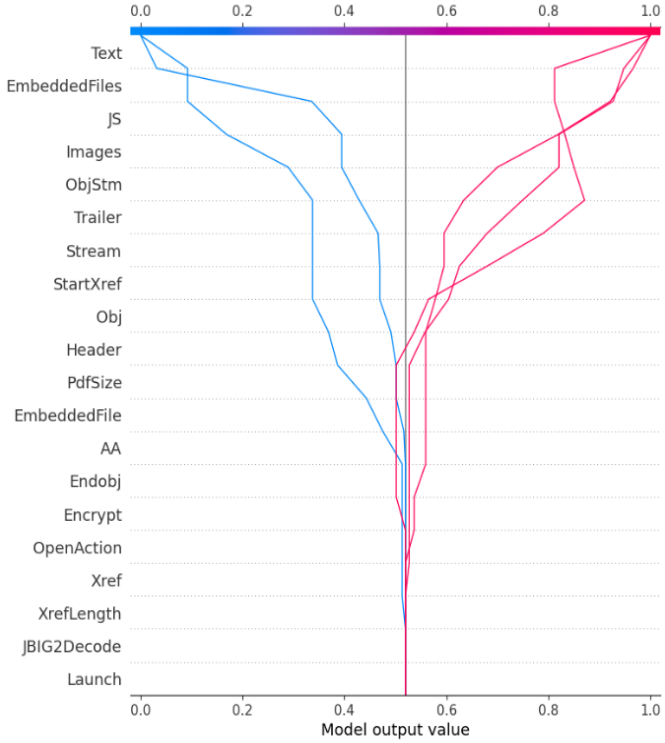

Fig. 5. SHAP Dependence Plot

SHAP Dependence plot as shown in Fig.5., which explains the impact of the features on the model's output based on the predicted probability ranging from 0 to 1, classifying them as benign to malicious respectively. From the image we understand that features like Trailer, StartXref, Stream are pushing predictions towards both sides but what stands out are features like JS, EmbeddedFiles, Launch and JBIG2Decode, which are clearly having strong predictions of class 1. We also have OpenAction feature which means there is a trigger when the PDF is opened, and this is pushing the decision towards maliciousness.

H. LIME

LIME (Local Interpretable Model-Agnostic Explanations) is a technique in explainable AI (XAI) that helps interpret predictions made by black-box models, such as deep learning or ensemble models. It's "model-agnostic," meaning it can be used with any machine learning algorithm. LIME works by approximating the complex model locally with a simpler, interpretable model (like linear regression or decision trees) around the prediction of interest. It perturbs the input data slightly and observes how these changes affect the output. This creates a dataset of modified inputs and corresponding outputs, which is then used to train the simpler surrogate model. The coefficients of this local model indicate which features were most influential in the original prediction. For Malware Detection, we specifically use it to understand as to why a certain file has been tagged as malicious. As LIME provides its insights into each of these predictions, it makes it easier to understand. We used it for our Deep Neural Network, to understand the features that contributed to that decision.

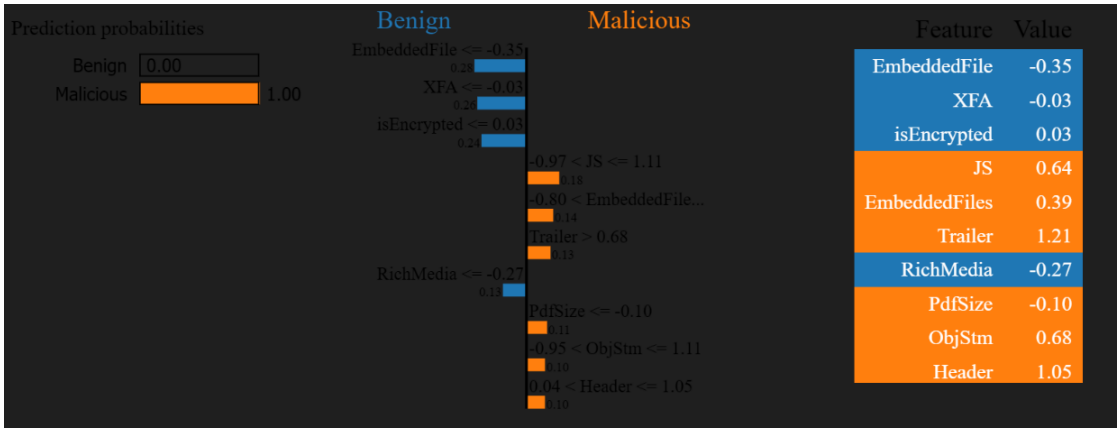

Fig. 6. LIME 10th sample explanation

First exploration involved looking at a random sample, in this case 10th sample in the dataset as shown in Fig. 6., which highlights that JavaScript usage, embedded files, abnormal trailer position, and excessive text content are the key reasons why the model confidently flagged this PDF as malicious. Meanwhile, features like the absence of certain embedded files and additional actions slightly suggested benign behavior but weren't strong enough to change the outcome.

Then for further exploration, I created another model which explained the first 1000 samples in the dataset as shown in Fig. 7., which helped understand which feature had the biggest impact on the neural network prediction made. The features of the PDF which helped explain the prediction went on to the Y-Axis, the X-Axis on the other hand indicated how each feature helped influence the model's decision. The topmost features included Embedded Files, XFA, and the isEncrypted for the model's prediction (Elattar et al., 2024).

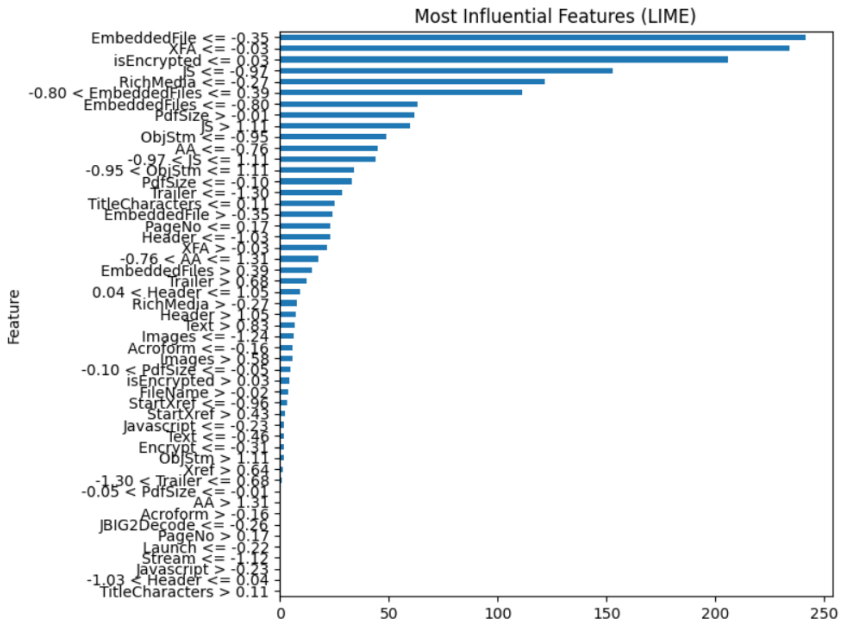

Fig. 7. LIME 1000 samples understanding

4. Findings & Conclusion

Findings

After running our models and performing tests on the 10,023-sample dataset we have, the Random Forest and XGBoost models showed the highest accuracy rates exceeding 99% while the CNN achieved approximately close with 98.9% and a F1 score of the same 98.9%. which displays high reliability in binary classification tasks. A Matthew Correlation Coefficient of 0.976 indicates very strong predictive power. This metric is especially useful because it accounts for true and false positives and negatives, making it a great single-score summary—even for imbalanced datasets. The analysis conducted using SHAP and LIME provide critical insights into the model behavior. Analysis by SHAP revealed features like JavaScript presence, embedded files on the PDF, content of the images and anomalies in the structure of PDF like OpenAction presence and Launch function, are very high contenders of malicious presence. LIME on the other hand further validated these findings on a local level by explaining the individual model predictions hence making the system transparent and more trustworthy for analysts and enthusiasts. Quite interestingly, features that were initially assumed to be benign, for example, EmbeddedFiles, were shown to shift their classification contribution depending on the content and value patterns. Such features call for context-aware detection mechanisms.

Conclusion

The aim of the study was to confirm the integration of CNNs with Explainable AI as a highly effective approach for detecting malware's presence in portable file formats, which can be said is achieved. Combining CNN's behaviour with XAI's techniques has proven to be an enhanced model for transparency and uncovers valuable insights into which structural PDF features contributed most significantly to classification decisions. The findings emphasize the importance of features like the usage of JavaScript and having files embedded with unusual PDF actions being the highest markers consistently for malware activity. Future work can be conducted to explore CNN with tree-based algorithms or further optimization of feature extraction to improve CNN accuracy. This research acts as a solid foundation for building further and advanced explainable malware detection systems which can evolve with emerging threats.

References

Borno, Z.S., Sakib, N., Anwar, S.S., 2023. 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). Presented at the 2023 IEEE 9th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), pp. 195–200. https://doi.org/10.1109/WIECON-ECE60392.2023.10456385

Bragança, H., Rocha, V., Souto, E., Kreutz, D., Feitosa, E., 2023. Explaining the Effectiveness of Machine Learning in Malware Detection: Insights from Explainable AI, in: Anais Do XXIII Simpósio Brasileiro de Segurança Da Informação e de Sistemas Computacionais (SBSeg 2023). Presented at the Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, Sociedade Brasileira de Computação - SBC, Brasil, pp. 181–194. https://doi.org/10.5753/sbseg.2023.233595

CIC-Evasive-PDFMal2022 | Datasets | Canadian Institute for Cybersecurity | UNB [WWW Document], n.d. URL https://www.unb.ca/cic/datasets/pdfmal-2022.html (accessed 4.13.25).

Elattar, M., Younes, A., Gad, I., Elkabani, I., 2024. Explainable AI model for PDFMal detection based on gradient boosting model. Neural Comput & Applic 36, 21607–21622. https://doi.org/10.1007/s00521-024-10314-y

Hossain, G.M.S., Deb, K., Sarker, I.H., 2024. An Enhanced Feature-Based Hybrid Approach for Adversarial PDF Malware Detection, in: 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT). Presented at the 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), pp. 101–106. https://doi.org/10.1109/ICEEICT62016.2024.10534412

Phuoc, N.C.H., Tan, V.N., Cam, N.T., 2024. uitPDFXAI: Malicious PDF files detection with eXplainable Artificial Intelligence, in: 2024 14th International Conference on System Engineering and Technology (ICSET). Presented at the 2024 14th International Conference on System Engineering and Technology (ICSET), pp. 186–191. https://doi.org/10.1109/ICSET63729.2024.10775017

Rahman, T., Ahmed, N., Monjur, S., Haque, F.M., Hossain, M.I., 2023. Interpreting Machine and Deep Learning Models for PDF Malware Detection using XAI and SHAP Framework, in: 2023 2nd International Conference for Innovation in Technology (INOCON). Presented at the 2023 2nd International Conference for Innovation in Technology (INOCON), pp. 1–9. https://doi.org/10.1109/INOCON57975.2023.10101116

Rajagopal, S., Gaur, A., Vinod, P., 2023. Interpretable PDF Malware Detector, in: 2023 16th International Conference on Security of Information and Networks (SIN). Presented at the 2023 16th International Conference on Security of Information and Networks (SIN), pp. 1–6. https://doi.org/10.1109/SIN60469.2023.10474653

Snehal, M.S.C., Nagoor, V., Rohit, S., Raghunandan, S., Thangavel, S.K., Srinivasan, K., Kapoor, P., 2024. Towards Explainability Using ML And Deep Learning Models For Malware Threat Detection, in: 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT). Presented at the 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), pp. 1–6. https://doi.org/10.1109/AIIoT58432.2024.10574689

Sowan, B., Matar, N., Aburub, F., Fasha, M., Al Khaldy, M., Nofal, M.I., Al-Jaber, A., 2024. PDF Malware Detection: A Hybrid Approach Using Random Forest and K-Nearest Neighbors, in: 2024 2nd International Conference on Cyber Resilience (ICCR). Presented at the 2024 2nd International Conference on Cyber Resilience (ICCR), pp. 1–6. https://doi.org/10.1109/ICCR61006.2024.10533046

Yudin, M., Song, Y., Serban, C., Chadha, R., 2024. Exposing Vulnerabilities in PDF Malware Detectors to Evasion Attacks, in: MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM). Presented at the MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM), pp. 807–814. https://doi.org/10.1109/MILCOM61039.2024.10773945