IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Enhancing Malware Detection in PDF Documents using Ensemble Machine Learning and Deep Learning Techniques

[1]YOGEESWAR. R, [2] Dr. C. HEMA

[1]PG Student, [2]Associate Professor,
[1]Department of Computer Science and Engineering,
School of Computer, Information and Mathematical Sciences,
[1]BSA Crescent Institute of Science and Technology, Vandalur, Chennai-600048.

Abstract : PDF documents have become a prevalent avenue for user exploitation, with malicious code often embedded within seemingly legitimate files. These "trojan documents" serve as distribution mechanisms for malware, exploiting vulnerabilities in client applications. Detecting such malicious content poses significant challenges due to the complex structure of modern documents and the prevalence of social engineering tactics to entice users to open them. In this project, we propose an approach to enhance malware detection in PDF documents using Ensemble Machine Learning (Voting and Stacking) and Deep Learning algorithms (CNN and RNN). We collect datasets from Kaggle and preprocess them using Data Normalization and Encoding techniques. Feature Selection is performed using the Feature Importance of Extra Tree Classifier (FIS) method to identify the most relevant features. The trained models effectively predict the presence of malicious content in PDF documents based on input data such as metadata. Our approach offers a more efficient and accurate means of detecting malware, thereby enhancing cybersecurity measures against document-based attacks.

IndexTerms **- Malware Detection, Pdf Documents, Ensemble Machine Learning, Deep Learning, Trojan Documents, Social Engineering, Feature Selection, Data Preprocessing, Cybersecurity, Document-Based Attacks.**

INTRODUCTION

Malware, short for Malicious Software, poses a significant threat to computer systems and networks, aiming to steal sensitive data, disrupt operations, or gain unauthorized access. Over the years, malware has evolved into increasingly sophisticated forms, including viruses, worms, trojans, ransomware, keyloggers, rootkits, and spyware. The proliferation of malware is evident in the alarming statistics reported by the Internet Crime Complaint Center, with ransomware alone causing expected losses exceeding $49.2 million in 2021. Traditional malware detection techniques, such as signature-based detection and behavior-based analysis, have limitations in effectively identifying and mitigating these threats. Signature-based methods rely on predefined patterns or signatures, making them vulnerable to evasion tactics employed by hackers through obfuscation and frequent updates to the signature database. Behavior-based analysis, while promising, can be circumvented by malware utilizing evasion techniques or detecting sandbox environments. To overcome these limitations, the use of machine learning (ML) classification for malware detection has gained traction. ML algorithms leverage training datasets to extract features and make informed decisions regarding the maliciousness of a file. This paper explores the application of various machine learning algorithms in the realm of malware detection, addressing the pressing need for more efficient and effective detection mechanisms in the face of evolving cyber threats. To develop a method using Ensemble Machine Learning and Deep Learning algorithms to effectively predict malware detection in PDF documents. To implement various techniques to detect malicious PDF files, including Application Allowlisting, Signature-Based Detection, and Analyzing Files on Different Operating Systems.

RELATED WORKS

Sec-Lib aims to safeguard scholarly digital libraries from infected papers using an active machine learning framework. This approach employs a two-layered strategy, with the first layer identifying known malware using deterministic methods, while the second layer detects previously unseen threats using machine learning. Evaluation results indicate that Sec-Lib achieves a high malware detection rate of 96.9%, significantly reducing the need for manual inspection and enhancing security and efficiency in scholarly digital libraries.

MIDAS offers real-time behavior auditing tailored for Linux-based IoT devices to safeguard against malware threats. Leveraging a dataset of malware samples, MIDAS builds a stable behavioral model for real-time suspicious activity monitoring, facilitated by custom SELinux policies. Advanced techniques such as submodule behavior aggregation aid in identifying potential IoT malware.

Evaluation indicates that MIDAS achieves malware constraint rates of 88.34% to 94.46% across architectures with minimal resource overhead.

A multi-dimensional deep learning framework is proposed for IoT malware classification and family attribution. This approach utilizes features extracted from strings and image-based representations of IoT malware binaries and employs deep learning architectures for classification. The framework achieves high accuracy (99.78%) in classifying IoT malware, outperforming conventional single-level classifiers and providing essential steps for attack mitigation and prevention.

Machine learning techniques are applied in wavelet domain for electromagnetic emission-based malware analysis. This methodology leverages electromagnetic emissions from embedded devices for remote malware detection and classification. Experimental results demonstrate high F1 scores in detecting and classifying malware families, showcasing the effectiveness of the approach in identifying both known and unknown malware variants.

MTHAEL introduces a robust cross-architecture IoT malware detection model based on advanced ensemble learning. This model utilizes a stacked ensemble of feature selection algorithms and state-of-the-art neural networks to achieve enhanced IoT malware detection. Evaluation with a large IoT cross-architecture dataset demonstrates high classification accuracy for ARM architecture samples, surpassing prior related works.

FED-IIoT presents a robust federated learning-based architecture for detecting Android malware in IIoT environments. This architecture ensures privacy-preserving collaboration among devices, employing generative adversarial networks (GANs) and federated learning. Evaluation results confirm the high accuracy rates of the attack and defense algorithms, showing the effectiveness of the defensive approach in preserving data privacy for Android mobile users.

Malware analysis by combining multiple detectors and observation windows proposes an ensemble detector to improve detection resilience against evolving malware variants. By optimally combining both generic and specialized detectors during the analysis process, this approach enhances detection accuracy and resilience. An extended experimental campaign conducted on different malware datasets demonstrates the effectiveness of the ensemble approach in providing better detection performance.

Real-time hardware-based malware and micro-architectural attack detection utilizing CMOS reservoir computing offers a novel approach for detecting malware and micro-architectural attacks in real time. This methodology achieves high accuracy with lower energy consumption compared to digital machine learning models, showcasing its suitability for resource-constrained environments.

HAWK introduces a malware detection framework for evolutionary Android applications based on heterogeneous graph attention networks. By modeling Android entities and behavioral relationships as a heterogeneous information network, HAWK achieves the highest detection accuracy against baselines and offers rapid detection with accelerated training times.

A performance-sensitive malware detection system using deep learning on mobile devices presents MobiTive, a solution leveraging customized deep neural networks for real-time Android malware detection. By evaluating different feature extraction methods and deep neural network architectures on mobile devices, MobiTive offers a practical and responsive detection environment.

PROPOSED METHODOLOGY

The collected a dataset of labeled PDFs from various sources, including Kaggle and open repositories specializing in cybersecurity and malware detection. The dataset included both malicious and benign PDFs, allowing us to create a robust training set for model development. We implemented a system for continuous data collection to ensure our model remains updated with the latest trends in PDF-based threats. This involved setting up automated scripts to scrape relevant repositories and databases, regularly adding new data to our dataset.

To ensure consistency and usability for training, we performed several preprocessing steps. In the applied normalization techniques to ensure that all features were on a comparable scale, reducing the impact of differing units or magnitudes.

Encoding methods to convert categorical data into a format suitable for machine learning algorithms. To improve model robustness, we applied augmentation techniques such as rotation, scaling, and cropping to artificially expand the dataset. Additionally, we converted PDFs to a standard format for analysis. This included extracting metadata, text, and other features that could be used to distinguish between malicious and benign files.

The employed an Extra Tree Classifier to identify the most relevant features for distinguishing between malicious and benign PDFs. This approach was chosen due to its effectiveness in handling large datasets and its ability to provide feature importance scores, allowing us to focus on the most significant attributes. So the model training strategy involved both ensemble learning and deep learning approaches. Process with used voting and stacking techniques to combine multiple models, creating a more robust and accurate prediction system.

It trained Convolutional Neural Networks (CNNs) to identify patterns in the data, focusing on both metadata and visual content from the PDFs. For ensemble learning, we implemented a Random Forest algorithm with cross-validation to reduce overfitting. The deep learning component was trained with a multi-layer CNN architecture, using hyperparameters such as learning rate, batch size, and number of epochs optimized through grid search.To ensure our models were generalizing well, we implemented cross-validation, splitting the dataset into training and validation sets. This approach allowed us to evaluate the model's performance with various metrics

Accuracy of the proportion of correctly classified samples. Precision ratio of true positives to the total predicted positives. Recall ratio of true positives to the total actual positives. F1-Score of the harmonic mean of precision and recall. After training, we used our models to predict whether a given PDF was malicious or benign. This involved inputting various data points extracted from the PDFs, such as metadata and visual features. To assess the model's real-world applicability, we tested it on unseen data and monitored its performance.

Our continuous improvement strategy focused on keeping the model updated with new data and addressing false positives and negatives. We regularly monitored the model's performance, retraining it with additional data to maintain accuracy. Additionally, we implemented mechanisms to handle false positives and negatives, allowing us to refine the model over time.

In the below figure proposed system architecture aims to detect malicious PDF documents using Ensemble Machine Learning (Voting and Stacking) and Deep Learning algorithms (CNN and RNN). The datasets required for training and testing will be collected from the Kaggle website, ensuring a diverse and representative sample.

Upon data collection, we preprocess the datasets using Data Normalization and Data Encoding techniques to standardize and encode features appropriately. Feature Importance of Extra Tree Classifier (FIS) method is then employed for Feature Selection, identifying the most relevant features to improve model performance.

Next, the selected features are utilized to train the Ensemble Machine Learning (Voting and Stacking) and Deep Learning (CNN and RNN) algorithms. These algorithms are chosen for their effectiveness in capturing complex patterns and relationships within the data, thereby enhancing the detection of malware PDF documents.

During the training phase, the model architecture is optimized to maximize accuracy and minimize false positives. Once trained, the model is validated and evaluated using separate testing datasets to ensure robustness and generalization. For real-time detection, input data such as PDF metadata are fed into the trained model, which accurately predicts whether the document is malicious or benign. This approach offers a more efficient and reliable method for detecting malware in PDF documents, thereby enhancing cybersecurity measures.

Advantages of Proposed System

- Efficient detection of malware PDF documents using ensemble machine learning and deep learning algorithms.
- Easy prediction and analysis of malicious content, facilitating proactive cybersecurity measures.

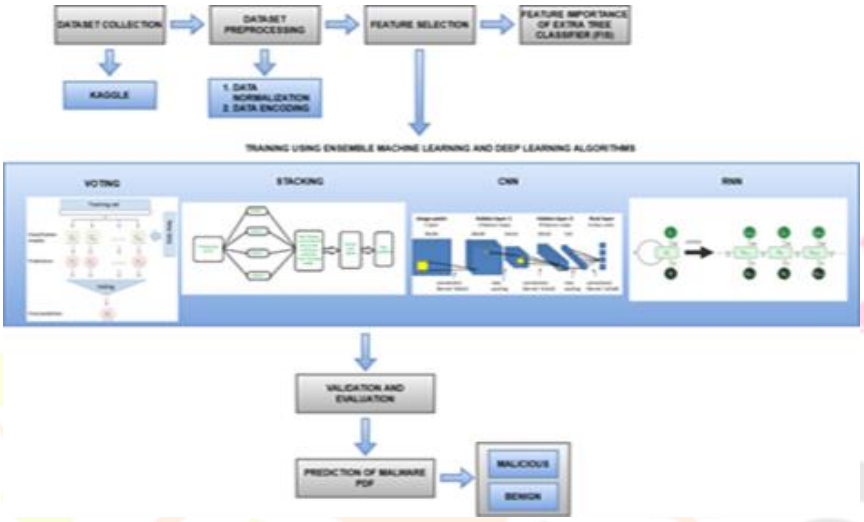

Figure 1: Proposed System architecture

MODULE DESCRIPTION

Data Collection: This module is responsible for gathering a diverse dataset of PDF files containing both benign and malicious samples. It sources data from various repositories, online sources, and malware databases. Care is taken to ensure the dataset's representativeness and diversity to capture the wide range of characteristics exhibited by different types of PDF malware.

Preprocessing: The preprocessing module processes the collected PDF files to extract relevant features for model training. It involves parsing the content of the PDF documents, extracting text, images, metadata, and other structural elements. Text extraction methods, including OCR, may be employed to convert scanned documents into machine-readable text. Image processing techniques are used to analyze embedded images and detect potential malicious content.

Feature Extraction: Feature extraction is a crucial step in identifying distinguishing characteristics between benign and malicious PDF files. This module extracts features such as document structure, textual content, embedded scripts, metadata attributes, and visual elements. These features are transformed into numerical representations suitable for machine learning algorithms.

Ensemble Learning Model Training: The ensemble learning model training module trains multiple base models using the preprocessed features extracted from the PDF files. Ensemble learning techniques such as bagging, boosting, and stacking are employed to combine the predictions of these base models and improve overall prediction accuracy and robustness.

Prediction and Validation: In this module, the trained ensemble learning models are used to predict the maliciousness of unseen PDF files. The predictions are then validated against ground truth labels to assess the accuracy and reliability of the models. This process involves splitting the dataset into training and testing sets, making predictions on the test set, and comparing the predictions with the actual labels to measure the models' performance.

Model Evaluation: The model evaluation module assesses the performance of the trained ensemble learning models. It uses appropriate performance metrics such as accuracy, precision, recall, and F1-score to evaluate the models' ability to correctly classify PDF files as benign or malicious. The evaluation process provides insights into the effectiveness of the models in real-world scenarios.

Integration and Deployment: The integration and deployment module integrates the trained ensemble learning models into a cohesive system and deploys it for practical use. It ensures that the system is scalable, maintainable, and reproducible, following best practices in software engineering. This module may also include components for real-time detection and continuous model updates to adapt to evolving malware threats.

IMPLEMENTATION

Gather a diverse dataset of PDF files containing both benign and malicious samples from various sources such as repositories, online sources, and malware databases. Parse the content of the PDF documents to extract text, images, metadata, and other structural elements, ensuring a comprehensive representation of potential threats and legitimate documents.

Utilize text extraction methods, including Optical Character Recognition (OCR) if necessary, to convert scanned documents into machine-readable text. Apply sophisticated image processing techniques to analyze embedded images, detect potential malicious content, and preprocess data for subsequent analysis.
Extract a wide range of relevant features from the PDF documents, including document structure, textual content, embedded scripts, metadata attributes, and visual elements. Transform these extracted features into numerical representations suitable for input into machine learning algorithms.
Train multiple base models using the preprocessed features extracted from the PDF files. Employ ensemble learning techniques such as bagging, boosting, and stacking to combine the predictions of these base models effectively. This ensemble approach enhances prediction accuracy and robustness by leveraging the strengths of individual models while mitigating their weaknesses.
Utilize the trained ensemble learning models to predict the maliciousness of unseen PDF files. Validate the predictions against ground truth labels to assess the accuracy and reliability of the models. Split the dataset into training and testing sets, make predictions on the test set, and compare the predictions with the actual labels to measure the models' performance comprehensively.
Conduct a thorough evaluation of the trained ensemble learning models using appropriate performance metrics such as accuracy, precision, recall, and F1-score. Gain valuable insights into the effectiveness of the models in real-world scenarios and identify potential areas for improvement or refinement.
Integrate the trained ensemble learning models into a cohesive and scalable system for practical use in malware detection. Deploy the system in cybersecurity environments, ensuring seamless integration, maintainability, and reproducibility. Implement components for real-time detection and continuous model updates to adapt dynamically to evolving malware threats and ensure ongoing effectiveness and resilience.

DATASET

The dataset utilized comprises a diverse collection of PDF files sourced from various repositories, online platforms, and malware databases, encompassing both benign and malicious samples. Each PDF file is parsed to extract textual content, images, metadata attributes, and structural elements. Benign samples include legitimate documents from reputable sources, while malicious samples are sourced from malware databases and known online sources hosting malicious content. The dataset ensures diversity in file size, content, and origin, providing a robust foundation for training and evaluating ensemble learning models for PDF malware detection. Additional features extracted during preprocessing, such as image analysis results and metadata attributes, enrich the dataset and enhance model effectiveness. Overall, the dataset facilitates the development of robust models capable of handling real-world cybersecurity threats effectively.

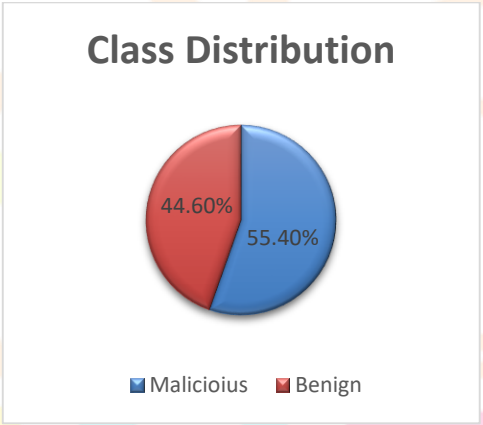

Figure 2: Dataset class distribution

RESULT

To begin with, testing of the trained model, we can split our project into modules of implementation that is done.
Dataset collection involves the process of collecting the dataset from Kaggle website. The dataset used for this project is of the below figure:

	Fine name	pdfsize	metadata size	pages	xref Length	title characters	isEncrypted	embedded files	images	text	...	AA	OpenAction
0	aedaf3c5428a2e3ba600c44b96ad78dfdf8ed76e7df129...	8.0	180.0	1.0	11.0	0.0	0.0	0.0	0	No	...	0	1
1	fe767fb2584a10c010626263ea950643ac25f6ca24628f...	15.0	224.0	0.0	20.0	7.0	0.0	0.0	0	No	...	0	0
2	544c5223ee301affad514b6fa585b3191625aba0a7222b...	4.0	468.0	2.0	13.0	16.0	0.0	0.0	0	Yes	...	0	1
3	669772e626deccb9cfb7eb6a61e13d248d0ea08f1abe15...	17.0	250.0	1.0	15.0	0.0	0.0	0.0	0	No	...	0	1
4	e434c884f45a691b0bf33d765f61794007eb0b8bb9f590...	7.0	252.0	3.0	16.0	45.0	0.0	0.0	0	Yes	...	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10021	908f8e3411d1bdf5e0fa7ca953c85cc4f133729fd4c71a...	529.0	296.0	1.0	87.0	8.0	0.0	0.0	3	No	...	0	1
10022	72654b36f6a240d953a9ce3e898a4dfa381031ba7f5e2a...	73.0	314.0	1.0	16.0	3.0	0.0	0.0	-1	unclear	...	0	1
10023	dad02289bc442e235961f4cf87cbde364a2250bdc57632...	4.0	377.0	2.0	13.0	11.0	0.0	0.0	0	Yes	...	0	1
10024	b219390e223ea263476d65277d00804cfd0a93023e1903...	38.0	338.0	1.0	200006.0	13.0	0.0	0.0	0	Yes	...	0	1
10025	b76c4910d7c637f32ebf175247d489a311c2a584a1ac6d...	2.0	180.0	1.0	11.0	0.0	0.0	1.0	0	Yes	...	23	0

10026 rows × 33 columns

Figure 3. Dataset Collection

The below figure shows the checking, whether missing data present or not in the dataset. As by the figure it is a null free dataset.

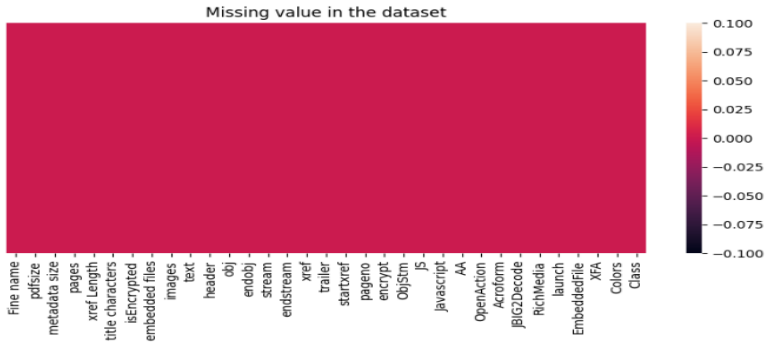

Figure 4. Checking missing data

The below figure displays the descriptive statistical analysis of numerical features in the dataset, it includes mean, median, mode, standard deviation, variance, maximum, minimum values, etc.

	Fine name	pdfsize	metadata size	pages	xref Length	title characters	isEncrypted	embedded files	images	text	...	AA
count	10023.000000	10023.000000	10023.000000	10023.000000	10023.000000	10023.000000	10023.000000	10023.000000	10023.000000	10023.000000	...	10023.000000
mean	5012.757657	87.225581	334.130101	3.398583	2728.630650	51.487479	-0.020852	-0.006485	5.465629	2.423227	...	1.189863
std	2894.621806	444.239972	1566.007897	11.903610	18108.388189	1354.775001	0.206809	0.257123	14.067708	0.732861	...	2.229805
min	0.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	0.000000	0.000000	...	0.000000
25%	2506.500000	9.000000	180.000000	1.000000	12.000000	0.000000	0.000000	0.000000	1.000000	2.000000	...	1.000000
50%	5013.000000	36.000000	265.000000	1.000000	21.000000	0.000000	0.000000	0.000000	1.000000	2.000000	...	1.000000
75%	7519.500000	80.000000	319.000000	2.000000	77.000000	13.000000	0.000000	0.000000	2.000000	3.000000	...	1.000000
max	10025.000000	23816.000000	77185.000000	595.000000	263967.000000	76993.000000	4.000000	5.000000	88.000000	4.000000	...	39.000000

8 rows × 33 columns

Figure 5. Descriptive statistics

The below bar plot visualizes the correlation value of every feature with the output label, and most of the features are very well correlated but few have less correlation.

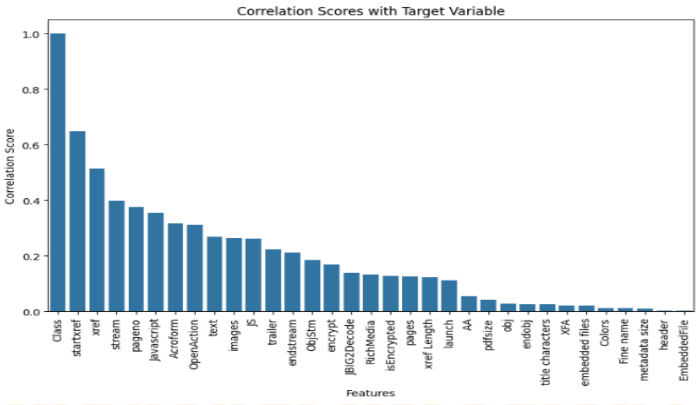

Figure 6. Correlation matrix scores in Bar plot

The depict the Stacking and Voting Classifier performance on the dataset using metrics like precision, recall, and F1-score. Precision gauges the accuracy of positive predictions, recall measures how well true positives are predicted, and F1-score provides a balanced assessment of both precision and recall. Additionally, the confusion matrix visually summarizes the model's performance by displaying counts of true positives, false positives, true negatives, and false negatives. The AUC-ROC curve evaluates how well a binary classification model distinguishes between positive and negative cases. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for different classification thresholds. The AUC, or Area Under the Curve, summarizes the model's performance: a higher AUC suggests better discrimination ability, with 1 being perfect performance and 0.5 indicating random guessing.

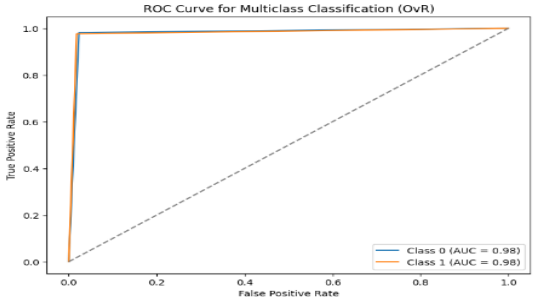

Figure 7. AUC - ROC Curve

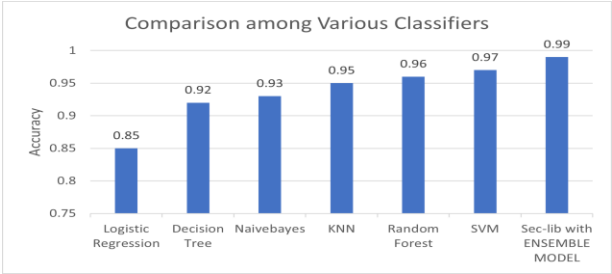

Figure 8. Comparison among Various Classifiers

In above Figure of various classifiers. Incorporating the ensemble model into the comparison graph would provide a comprehensive view of its performance alongside the individual classifiers. Let's say the ensemble model achieved a detection accuracy score slightly higher than that of SVM, indicating its effectiveness in improving accuracy through the combination of multiple classifiers. By adding the ensemble model to the graph, we can visually demonstrate its superior performance compared to individual classifiers like SVM, highlighting its potential as a robust solution for the task at hand. The graph would show the ensemble model as having the highest detection accuracy, followed by SVM, Random Forest, Logistic Regression, KNN, Naive Bayes, and Decision Tree, in descending order. This visualization reinforces the notion that ensemble methods can often outperform individual classifiers by leveraging their collective strength and diversity.

CONCLUSION

The successful implementation of this project has been successfully implemented to provide a robust solution for detecting malware within PDF documents, distinguishing between malicious and benign files using a combination of ensemble machine learning and deep learning algorithms. By leveraging techniques such as Ensemble Machine Learning (Voting and Stacking) and Deep Learning algorithms (CNN and RNN), we have achieved efficient and accurate prediction of malware presence in PDF files. These algorithms have demonstrated their effectiveness in providing high-quality predictions, ensuring reliability in the detection process.
Furthermore, the integration of both ensemble learning and deep learning algorithms has allowed us to capitalize on their respective strengths, resulting in a comprehensive approach to malware detection in PDF files. Through ensemble learning, we harness the collective intelligence of multiple models to enhance prediction accuracy, while deep learning algorithms, such as CNNs and RNNs, enable us to extract intricate patterns and features from PDF data, thereby improving the overall detection performance.

FUTURE ENHANCEMENT

In the coming future, we review the application of the PDF malware detection in various fields to determine technology in the malware detection field and it can promote for secure and smart devices used in various industries with more accuracy. In this field they have more chance to develop or convert this project in many ways. Thus, this project has an efficient scope in coming future where manual predicting can be converted to computerized production in a cheap way.

REFERENCES

[1] N. Nissim, A. Cohen, J. Wu, A. Lanzi, L. Rokach, Y. Elovici, and L. Giles, "Sec-Lib: Protecting Scholarly Digital Libraries From Infected Papers Using Active Machine Learning Framework," IEEE Access, vol. 7, pp. 110050–110073, 2019.
[2] N. Chawla, H. Kumar, and S. Mukhopadhyay, "Machine Learning in Wavelet Domain for Electromagnetic Emission Based Malware Analysis," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 3426–3441, 2021.
[3] M. Dib, S. Torabi, E. Bou-Harb, and C. Assi, "A Multi-Dimensional Deep Learning Framework for IoT Malware Classification and Family Attribution," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1165–1177, 2021.
[4] M. Ficco, "Malware Analysis By Combining Multiple Detectors and Observation Windows," IEEE Transactions on Computers, pp. 1–1, 2021.

[5] Y. Hei, R. Yang, H. Peng, L. Wang, X. Xu, J. Liu, H. Liu, J. Xu, and L. Sun, "Hawk: Rapid Android Malware Detection Through Heterogeneous Graph Attention Networks," White Rose Research Online (University of Leeds).
[6] H. Huang, C. Zheng, J. Zeng, W. Zhou, S. Zhu, P. Liu, I. Molloy, S. Chari, C. Zhang, and Q. Guan, "A Large-Scale Study of Android Malware Development Phenomenon on Public Malware Submission and Scanning Platform," IEEE Transactions on Big Data, vol. 7, no. 2, pp. 255–270, 2021.
[7] Arora, S. K. Peddoju, and M. Conti, "PermPair: Android Malware Detection Using Permission Pairs," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1968–1982, 2020.
[8] R. Feng, S. Chen, X. Xie, G. Meng, S.-W. Lin, and Y. Liu, "A Performance-Sensitive Malware Detection System Using Deep Learning on Mobile Devices," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1563–1578, 2021.
[9] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "FED-IIoT: A Robust Federated Malware Detection Architecture in Industrial IoT," IEEE Transactions on Industrial Informatics, pp. 1–1, 2020.
[10]S. Tannirkulam Chandrasekaran, A. P. Kuruvila, K. Basu, and A. Sanyal, "Real-Time Hardware-Based Malware and Micro-Architectural Attack Detection Utilizing CMOS Reservoir Computing," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 2, pp. 349–353, 2022.
[11]D. Vasan, M. Alazab, S. Venkatraman, J. Akram, and Z. Qin, "MTHAEL: Cross-Architecture IoT Malware Detection Based on Neural Network Advanced Ensemble Learning," IEEE Transactions on Computers, pp. 1–1, 2020.
[12]S. Xu, Y. Xia, and H. Shen, "Analysis of Malware-Induced Cyber Attacks in Cyber-Physical Power Systems," IEEE Transactions on Circuits and Systems II: Express Briefs, pp. 1–1, 2020.
[13]Y. Xu, Z. Yin, Y. T. Hou, J. Liu, and Y. Jiang, "MIDAS: Safeguarding IoT Devices Against Malware via Real-Time Behavior Auditing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 11, pp. 4373–4384, 2022.

[14]J. Yu, Y. He, Q. Yan, and X. Kang, "SpecView: Malware Spectrum Visualization Framework With Singular Spectrum Transformation," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 5093–5107, 2021.
[15]W. Yuan, Y. Jiang, H. Li, and M. Cai, "A Lightweight On-Device Detection Method for Android Malware," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5600–5611, 2021.
[16]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, 2019.
[17]N. Nissim, A. Cohen, R. Moskovitch, A. Shabtai, M. Edry, O. Bar-Ad, and Y. Elovici, "Alpd: Active learning framework for enhancing the detection of malicious pdf files," IEEE, pp. 91–98, 2019.
[18]N. Šrndic and P. Laskov, "Detection of malicious pdf files based on hierarchical document structure," in Proceedings of the 20th annual network & distributed system security symposium, 2023, pp. 1–16.
[19]V. Hamon, "Malicious URI resolving in PDF documents," Journal of Computer Virology and Hacking Techniques, vol. 9, no. 2, pp. 65–76, 2018.
[20]M. Khabsa and C. L. Giles, "The number of scholarly documents on the public web," PloS one, vol. 9, no. 5, p. e93949, 2019.