https://doi.org/10.33472/AFJBS.6.6.2024.7510-7521

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Detection of Malware in Pdfs Using Hybrid Algorithm

Mrs. I. Varalakshmi[1], K. Khalid Mohammed[2], M. ManiKandan[3], M. Mohamed Rizwan[4]

[1]Assistant Professor, Department of Computer Science and Engineering Manakula Vinayagar Institute of Technology, Pondicherry, India.
[2,3,4]B. Tech, Department of Computer Science and Engineering, Manakula Vinayagar Institute of Technology, Pondicherry, India.

Email: [1]varalakshmicse@mvit.edu.in, [2]khalidssmc@gmail.com, [3]yuvarajmani5550@gmail.com, [4]mrizwan1607@gmail.com

Article Info

Volume 6, Issue 6, July 2024

Received: 03 June 2024

Accepted: 31 June 2024

Published: 25 July 2024

doi: 10.33472/AFJBS.6.6.2024.7510-7521

ABSTRACT:

PDF malware refers to malicious software or code that is embedded within PDF (Portable Document Format) files, which are commonly used for sharing and distributing information. In our proposed system for detecting PDF malware, we integrate Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) networks to enhance our detection capabilities. RNNs and LSTMs are adept at capturing temporal dependencies, making them ideal for identifying evolving patterns in PDF malware. By leveraging these networks, our system learns to recognize subtle changes in PDF structures indicative of malicious content. This integration improves predictive accuracy while streamlining the training process, thanks to the recurrent nature of RNNs and LSTMs, which enable effective learning from past steps. The combination of pre-trained models and advanced neural network architectures significantly reduces training times without compromising detection precision. Overall, our hybrid approach represents a powerful advancement in cybersecurity, providing a more adaptable, accurate, and efficient defense against dynamic PDF-based malware threats.

Keywords: Variational Autoencoder (VAE), PDF-based malware threats

1. Introduction

PDF malware refers to malicious software or code embedded within PDF (Portable Document Format) files, a widely used document format for sharing and distributing information. Malicious actors may exploit vulnerabilities in PDF readers or use social engineering techniques to create PDFs that carry hidden malware. This malware can take various forms, such as embedded scripts, links, or executable code, aiming to exploit vulnerabilities in the target system or execute unauthorized actions. The unsuspecting user opens the seemingly harmless PDF file, and the malware is triggered, leading to potential security breaches, data theft, or system compromise. To combat PDF malware, advanced detection techniques, such as hybrid algorithmic approaches and image-based analysis, are increasingly employed to identify and neutralize threats within these documents, enhancing overall cyber security measures.

1.1 Malware:

Malware, short for malicious software, refers to a category of software intentionally designed to cause harm or exploit vulnerabilities in computer systems, networks, and devices. This umbrella term encompasses a wide range of malicious programs created with the intent of disrupting, damaging, or gaining unauthorized access to data and computer systems. Malware can manifest in various forms, including viruses, worms, Trojan horses, ransom ware, spyware, and adware. Cybercriminals deploy malware through deceptive means, often using phishing emails, malicious websites, or infected software downloads to infiltrate and compromise the security of a target system. Once inside, malware can execute a variety of malicious activities, such as stealing sensitive information, disrupting system functions, or providing unauthorized access to the attacker. Combating malware involves employing robust cyber security measures, including antivirus software, firewalls, and regular system updates, to detect, prevent, and mitigate the impact of these malicious threats on digital ecosystems.

2. Related work:

Sobhi Mejjaoul [1] This paper focus on adaptability makes them a prime target for attackers who can quickly insert malware into PDF files. This study proposes a model based on the Fuzzy Unordered Rule Induction Algorithm (FURIA) to detect PDF malware. The model outperforms currently used methods in terms of reducing error rates and increasing accuracy. Other models, such as Naïve Bayes (NB), Decision Tree (J48), Hoeffding Tree (HT), and Quadratic Discriminant Analysis (QDA. The accuracy achieved by the proposed model is 99.81%, with model with the lowest error rate, which is 0.0022, and the worst performance of NB with an error rate of 0.014 P. Pandi Chandra [2] It talk about an Invasive Weed Optimization with Stacked Long Short Term Memory (IWO-S-LSTM) technique for PDF malware detection and classification. The presented IWO-S-LSTM model focuses on the recognition and classification of different kinds of malware that exist in PDF documents. the ridge regression, DT model, and RF model have resulted to lower performance with $accuy$ of 93.50%, 93.47%, and 93.19% respectivel Qasem Abu Al-Haija [3] This presents a new detection system that can analyze PDF documents in order to identify benign PDF files from malware PDF files. The system makes use of the AdaBoost decision tree with optimal hyperparameters, which is trained and evaluated on a modern inclusive dataset, viz. Evasive-PDFMal2022. After that, the MCE sharply decreased toward the minimum MCE hyperparameters only after three learning iterations recording an MCE of 1.3% and classification accuracy of 98.7%

Muhammad Ijaz [4] The paper concentrates on the combinations of different features are used for dynamic malware analysis. Dynamically 800 benign files and 2200 malware files are analyzed in Cuckoo Sandbox and 2300 features are extracted. The dynamic analysis has some limitations due to controlled network behavior and it cannot be analyzed completely due to limited access of network. The AUC (Area under Curve) of static malware analysis is 99.36% dynamic analysis. Static analysis has some limitation due to packed nature of malware Sanjay Sharma [5] The work is on frequency of opcode occurrence to detectunknown malware by using machine learning technique. For the purpose, they have used kaggle Microsoft malwareclassification challenge dataset and also studied multiple classifiers available inWEKA GUI based machine learning tool and found that five of them (Random Forest, LMT, NBT, J48 Graft andREPTree) detect the malware.Then approach Random forest, LMT, J48 Graft, an NBT detect malware with accuracy (96.8%) reported Muhammad Arshad [6] This paper presents a comparative analysis of machine learning (ML) techniques, including Naive Bayes (NB), K-Nearest Neighbor (KNN), Average One Dependency Estimator (A1DE), Random Forest (RF), and Support Vector Machine (SVM) for PDF malware detection. The performance of the techniques is evaluated using F1-score, precision, recall, and accuracy measures. The results indicate that KNN outperforms other models, achieving an accuracy of 99.8599% using 10-fold cross-validation

P. Pandi Chandran [7] This study focuses on the design of mayfly optimization with a deep belief network for PDF malware detection and classification (MFODBN-MDC) technique. The major intention of the MFODBN-MDC technique is for identifying and classifying the presence of malware exist in the PDFs. For demonstrating the improved outcomes of the MFODBN-MDC model, a wide range of simulations are executed , and the results are assessed in various aspects. The comparison study highlighted the enhanced outcomes of the MFODBN-MDC model over the existing techniques with maximum precision, recall, and F1 score of 97.42%, 97.33%, and 97.33%, respectively Hossein Sayadi[8] In this project they have suggested implementing hardware-based malware detection (HMD) countermeasures to address the shortcomings of software-based detection methods. HMD techniques involve applying standard machine learning (ML) algorithms to low-level events of processors that are gathered from hardware performance counter (HPC) registers. The experimental results indicate that our proposed image-based malware detection solution achieves superior performance compared to all other methods, with a 97% detection performance (measured by F-measure and area under the curve) for run-time zero-day malware detection utilizing soley the top four performance counter events. Specifically, our novel approach outperforms the binarized MLP by 16% and the best classical ML algorithm by 18% in F-measure, while maintaining a minimal false positive rate and without incurring any hardware redesign overhead SatishChikkagoudar[9] In this paper, we derive a simple yet effective holistic approach to PDF malware detection that leverages signal and statistical analysis of malware binaries. This includes combining orthogonal feature space models from various static and dynamic malware detection methods to enable generalized robustness when faced with code obfuscations. Using a dataset of nearly 30,000 PDF files containing both malware and benign samples, we show that our holistic approach maintains a high detection rate (99.92%) of PDF malware and even detects new malicious files created by simple methods that remove the obfuscation conducted by malware authors to hide their malware, which are undetected by most antiviruses YaxiaoWang[10] In this paper, we propose an PDF malware evasion method that is using GAN to generate adversarial PDF malware examples and evaluate it against four local machine learning based PDF malware classifiers. The evaluation is conducted on the same dataset which contains 100 malicious PDF files. The experimental results reveal that the proposed evasion attacks are effective, with attacks against three classifiers all attaining

100% evasion rate and attack against the last classifier also attaining 95% evasion rate on the evaluation dataset

3. Proposed Method:

In the proposed system for detecting PDF malware, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) algorithms play a pivotal role in the prediction process. RNNs and LSTMs are specialized neural network architectures designed to effectively capture and analyze sequential data, making them particularly well-suited for tasks involving temporal dependencies and patterns. By incorporating these advanced algorithms into the system, the model gains the ability to learn and adapt to the dynamic patterns characteristic of PDF-based malware over time. RNNs, with their recurrent connections, can retain information from previous time steps, allowing the model to capture long-range dependencies within the sequential data. LSTM networks, on the other hand, excel in capturing and retaining information over long periods, making them effective in recognizing subtle changes in PDF file structures indicative of malicious elements.

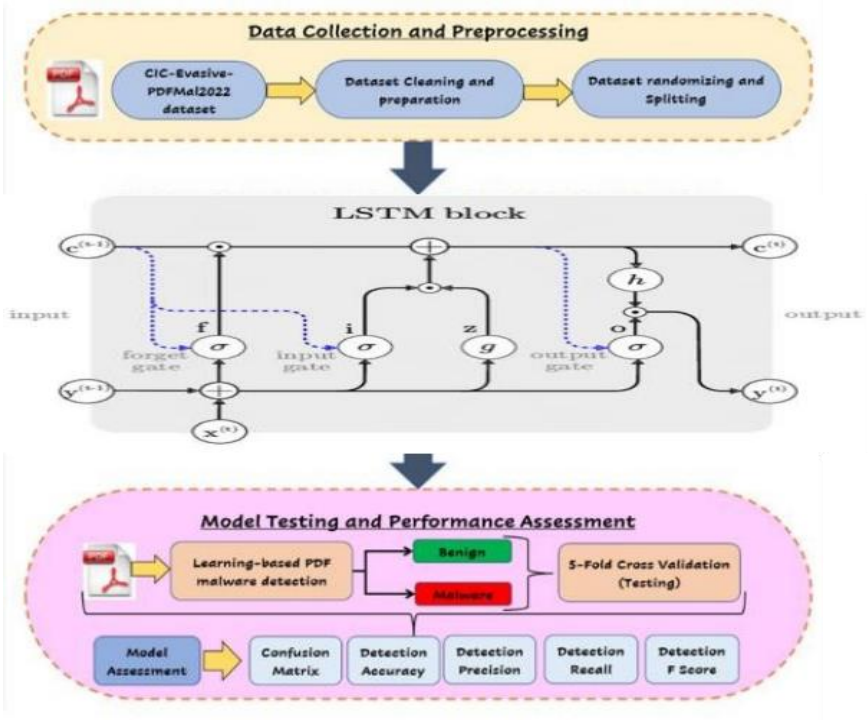

Fig.3.1.Architecture Diagram

The architecture diagram illustrates the structural components and flow of information within a system or model. Typically, it depicts various layers, modules, or components and their interactions or connections. Each element in the diagram represents a specific function or task, and the connections between them show how data or information flows through the system.This visual representation provides a comprehensive overview of the system's design and operation, aiding in understanding its functionality, dependencies, and relationships between different parts.Architecture diagrams are valuable tools for communication, documentation, and analysis, enabling stakeholders to grasp the system's complexity and design principles effectively.

3.2 Data Collection:

The data for this project is sourced from the URL provided by the University of New Brunswick's Canadian Institute for Cyber security (CIC). Specifically, the dataset named "pdfmal-2022" is utilized as the foundation for training, validating, and testing the proposed malware classification system. This dataset, curated by the CIC, is likely to comprise a diverse collection of PDF files with embedded malware instances. The inclusion of real-world, representative samples is crucial for the system to effectively learn and generalize patterns indicative of malicious software. The dataset is expected to encompass variations in the types of malware, encapsulating the evolving nature of cyber threats. Leveraging this dataset is fundamental for evaluating the system's performance under realistic conditions, ensuring that it is robust and adaptable to the dynamic landscape of PDF-based malware. The commitment to using datasets from reputable sources like the CIC underscores the project's dedication to creating a reliable and effective malware classification solution.

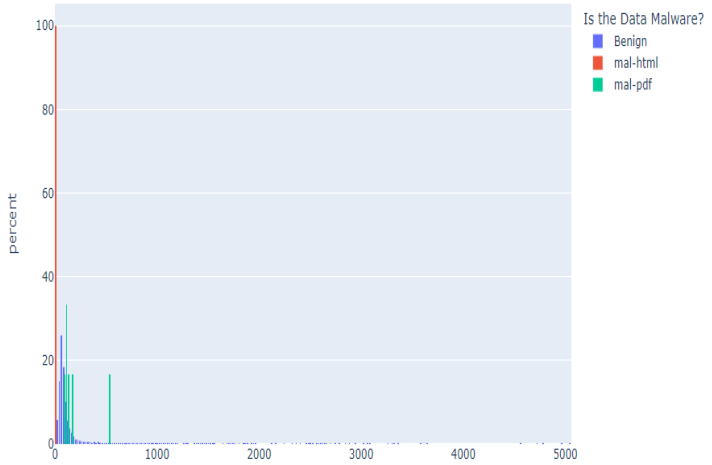

Fig.3.2.Prevalence of malware

The graph visually depicts the prevalence of malware in PDF files compared to both benign PDFs and malware in HTML files. Conversely, while malware in HTML files also poses a notable threat, it appears to be comparatively lower in frequency than malware embedded within PDF documents. Additionally, it highlights the need for comprehensive cybersecurity measures across various file formats to mitigate the risks posed by malware.

3.3 Pre- Processing:

The pre-processing of the dataset is a crucial step in preparing the raw data from the "pdfmal-2022" dataset, obtained from the University of New Brunswick's Canadian Institute for Cyber security, for effective use in the malware classification system. This phase involves several key procedures to enhance the quality and suitability of the data. Initial steps typically include data cleaning, which addresses any inconsistencies, missing values, or irregularities in the dataset. Subsequently, the data may be subjected to normalization or standardization to ensure uniformity in the scale of features.

	filename	obj	end obj	stream	endstream	xref	trailer	startxref	Page	Encrypt	...	AA	OpenAction	AcroForm	JBIG2Decode	RichMedia	Launch	EmbeddedFile	XFA	Colors_2_24	label
0	999875	73	73	25	25	2	2	2	7	0	...	0	0	0	0	0	0	0	0	0	Benign
1	999870	122	122	46	46	2	2	2	22	0	...	0	0	0	0	0	0	0	0	0	Benign
2	999858	66	66	19	19	2	2	2	12	0	...	0	0	0	0	0	0	0	0	0	Benign
3	999854	72	72	22	22	2	2	2	12	0	...	0	0	0	0	0	0	0	0	0	Benign
4	999331	70	70	19	19	2	2	2	11	0	...	0	0	0	0	0	0	0	0	0	Benign

Fig.3.3 Pdfmal -2022 dataset

Feature extraction techniques may also be applied to capture essential characteristics or patterns indicative of malware. Additionally, the dataset is likely partitioned into training, validation, and testing sets to enable the evaluation of the classification system's performance. This pre-processing phase is pivotal for mitigating potential biases, reducing noise, and optimizing the dataset for training the machine learning model. The meticulous handling of data during this stage significantly contributes to the overall effectiveness and generalizability of the malware classification system.

3.4 Feature Extraction:

In the process of feature extraction, duplicate values are identified and removed to enhance the efficiency and accuracy of the analysis. Duplicate values, when present in the dataset, can skew the results by inflating the importance of certain features or introducing redundancy in the information. By eliminating duplicates, the feature extraction stage ensures that each unique characteristic or attribute is appropriately represented and considered during analysis.

```
duplicated_rows = data.loc[np.where(data.duplicated())]
print(duplicated_rows.head())
```

```
      filename   obj   end obj   stream   endstream   xref   trailer   startxref   \
2779    591955    63        63       22          22      2         2           2
2781    591949    68        68       20          20      2         2           2
2783    591942    44        44       12          12      2         2           2
4309    499051    85        85       28          28      1         1           1
4314    499038    51        51       12          12      1         1           1

      Page   Encrypt   ...   AA   OpenAction   AcroForm   JBIG2Decode   RichMedia   \
2779    10         0   ...    0            0          0             0           0
2781    12         0   ...    0            0          0             0           0
2783     8         0   ...    0            0          0             0           0
4309    15         0   ...    0            0          0             0           0
4314    12         0   ...    0            0          0             0           0

      Launch   EmbeddedFile   XFA   Colors_2_24    label
2779       0              0     0             0   Benign
2781       0              0     0             0   Benign
2783       0              0     0             0   Benign
4309       0              0     0             0   Benign
4314       0              0     0             0   Benign
```

Fig.3.4.Analysis Stage

This helps in streamlining the dataset and reducing computational overhead during subsequent stages of the analysis. Additionally, removing duplicate values helps in improving the interpretability of the extracted features, as it ensures that only relevant and distinct information contributes to the final analysis. Overall, the removal of duplicate values in feature extraction plays a crucial role in producing reliable and meaningful insights from the data, facilitating more accurate decision-making processes in various domains.

3.5 Model Creation:

The hybrid model created using a combination of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) algorithms represents a sophisticated approach to modeling sequential data. RNNs excel in processing sequential information by retaining memory of past inputs, while LSTMs enhance this capability by selectively retaining or discarding information through specialized memory cells.

Using Recurrent Neural Network (RNN) unit, the hidden state $\mathbf{h_t}$ at a specific time step ***t*** is computed by incorporating information from the current input $\mathbf{x_t}$ and the previous hidden state. This calculation in$\boldsymbol{h_{t-1}}$volves weight matrices ***W*** and biases ***b***, which serve to adjust the importance of each input and previous hidden state. These parameters are typically followed by an activation function σ, such as the sigmoid or hyperbolic tangent function, which introduces non-linearity into the network's computations.

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b)$$

In an LSTM (Long Short-Term Memory) unit, the calculation of the hidden state $\mathbf{h_t}$ at a given time step (t \) involves several interconnected components, including gates and memory cells, which facilitate the model's ability to capture and retain information over long sequences. These components include the input gate $\boldsymbol{i_t}$, forget gate $\mathbf{f_t}$, output gate $\mathbf{o_t}$, and the candidate cell state $\mathbf{C_t}$Each gate is responsible for controlling the flow of information within the LSTM unit. Additionally, the candidate cell state represents the new candidate values that could be added to the cell state.

a. Input gate i_t, forget gate f_t, and output gate o_t calculations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

b. Candidate cell state $\tilde{C}_t$ calculation:

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

c. Cell state C_t update using input, forget, and candidate gates:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

d. Hidden state h_t calculation using the output gate and the updated cell state:

$$h_t = o_t \odot \tanh(C_t)$$

Here, σ represents the sigmoid activation function, tanh represents the hyperbolic tangent function, ⊙represents element-wise multiplication, and and b represent weight matrices andThese equations govern the flow of information through an LSTM unit, allowing it to learn and retain information over long sequences, thereby addressing the vanishing gradient problem often encountered in traditional RNNs.

4. Performance Analysis:

Performance metrics are essential tools for assessing the effectiveness of machine learning models. Accuracy, the most intuitive metric, provides an overall measure of correctness by quantifying the proportion of correctly classified instances. Precision measures the proportion of true positive predictions among all positive predictions, emphasizing the model's ability to avoid false positives. In contrast, recall, also known as sensitivity, quantifies the proportion of true positive predictions among all actual positive instances, highlighting the model's ability to identify relevant instances of a class. Loss, on the other hand, guides the optimization process during training by quantifying the discrepancy between predicted and true values, aiding in model refinement.

4.1 Accuracy:

Accuracy is a fundamental performance metric used to assess the overall effectiveness of a machine learning model. It measures the proportion of correctly classified instances among the total number of instances evaluated by the model. A high accuracy score indicates that the model's predictions closely match the actual labels, reflecting its ability to make correct decisions across different classes or categories.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Accuracy is particularly valuable for tasks where each class is of equal importance and there is a balanced distribution of instances among classes. However, it's important to note that accuracy alone may not provide a complete picture of a model's performance, especially in scenarios with imbalanced class distributions or when different classes have varying levels of significance.

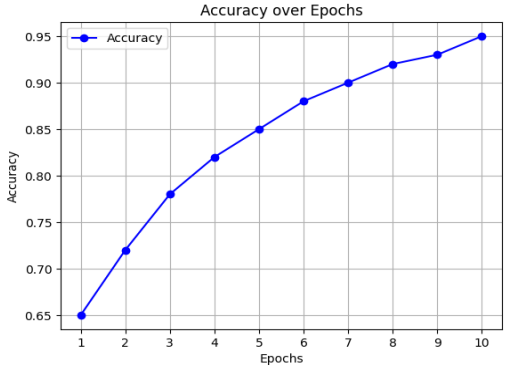

Fig.4.1(a).Accuracy over Epochs

The x-axis typically represents the number of training epochs or iterations. An epoch refers to one complete pass through the entire training dataset.The y-axis represents the training accuracy of the LSTM model at each epoch.. At the beginning of training, the training accuracy may start from a relatively low value. This is because the model's parameters are initialized randomly, and it has not yet learned the underlying patterns in the data. As training progresses through successive epochs, the training accuracy typically increases. The model gradually learns to recognize the features and patterns that distinguish benign PDF files from malicious ones. After reaching a certain point, the training accuracy may plateau or exhibit fluctuations.

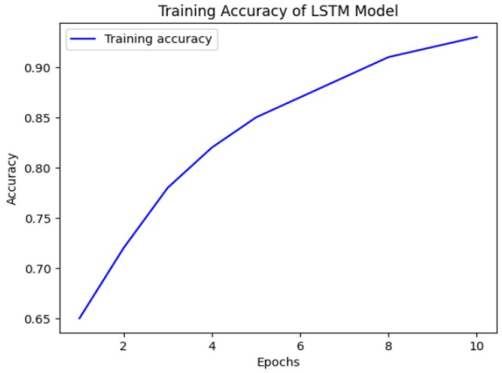

Fig.4.2(b).Training Accuracy

This indicates that the model is approaching its capacity to learn from the training data. Convergence implies that the model has learned the underlying patterns in the training data and further training epochs are unlikely to significantly improve its performance.In some cases, the training accuracy may continue to increase while the model's generalization performance (accuracy on unseen data) decreases. This phenomenon, known as overfitting, occurs when the model memorizes the training data's noise and outliers instead of learning the underlying patterns. Overfitting can be detected by comparing the training accuracy graph with the validation accuracy graph. If the training accuracy continues to improve while the validation accuracy stagnates or decreases, it suggests overfitting.

The x-axis represents the different models being compared, in this case, LSTM and GAN. The y-axis represents the accuracy of each model on the PDF malware detection task. Accuracy measures the proportion of correctly classified samples (both benign and malicious PDF files) out of the total samples. This line or bar in the graph represents the accuracy of the LSTM model on the PDF malware detection task. The accuracy value would indicate how well the LSTM model performs at distinguishing between benign and malicious PDF files. Similarly, another line or bar in the graph represents the accuracy of the GAN model on the same task. The accuracy value would indicate the performance of the GAN in detecting PDF malware.

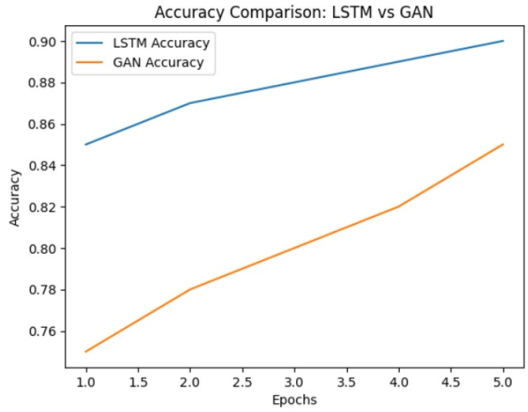

Fig.4.1(c).Accuracy Comparison

By comparing the accuracy values of the LSTM and GAN models, insights can be gained into which approach performs better for PDF malware detection. A higher accuracy for either model would suggest its superiority in accurately identifying malicious PDF files. It's essential to consider whether any observed differences in accuracy between the LSTM and GAN models are statistically significant.In addition to accuracy, other evaluation metrics such as precision, recall, and F1 score can also be compared between the LSTM and GAN models to provide a more comprehensive understanding of their performance. Beyond accuracy.

4.2 Confusion Matrix:

A confusion matrix is a performance measurement tool used in machine learning to evaluate the accuracy of a classification model. It provides a comprehensive summary of the model's predictions by tabulating the actual class labels against the predicted class labels. The matrix is structured as a grid, with rows representing the true classes and columns representing the predicted classes. Each cell in the matrix corresponds to the number of instances that belong to the true class and were predicted as belonging to the predicted class. This arrangement allows for a detailed analysis of the model's behavior, revealing the types of errors it makes.

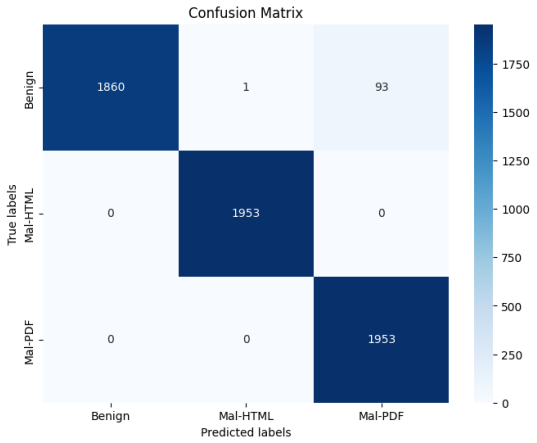

Fig.4.2.Confusion Metrics

From the confusion matrix, various performance metrics such as accuracy, precision, recall, and F1-score can be derived, providing insights into the model's strengths and weaknesses across different classes. Overall, the confusion matrix serves as a powerful diagnostic tool, guiding model refinement and optimization efforts to enhance classification performance.

Classification Report:

	precision	recall	f1-score	support
Benign	1.00	0.95	0.98	1954
Mal-HTML	1.00	1.00	1.00	1953
Mal-PDF	0.95	1.00	0.98	1953
accuracy			0.98	5860
macro avg	0.98	0.98	0.98	5860
weighted avg	0.98	0.98	0.98	5860

Fig.4.3.Classification Report

Precision, recall, and F1 score are key performance metrics used to assess the effectiveness of classification models in machine learning. Precision quantifies the proportion of true positive predictions among all positive predictions made by the model, indicating its ability to avoid false positives. A high precision score signifies that the model makes accurate positive predictions with minimal false alarms. On the other hand, recall measures the proportion of true positive predictions among all actual positive instances in the dataset, illustrating the model's capability to capture all relevant instances of a class. The F1 score, a harmonic mean of precision and recall, provides a balanced assessment of a model's performance, considering both false positives and false negatives. In summary, precision, recall, and F1 score collectively provide valuable insights into the strengths and weaknesses of a classification model, aiding in informed decision-making and model optimization.

5. Conclusion

The integration of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks into our hybrid algorithm significantly enhances PDF malware prediction. RNNs and LSTMs introduce a temporal dimension, enabling the system to capture sequential patterns and dependencies within PDF files effectively. This approach addresses the dynamic nature of PDF-based malware threats, as RNNs and LSTMs excel in handling sequential data. By discerning subtle variations in PDF file structures and content over time, our system can identify evasive malware variants. Combining the strengths of RNNs and LSTMs with other

machine learning techniques ensures a comprehensive analysis beyond the capabilities of individual algorithms. The system's efficiency is further optimized through the strategic use of pre-trained models, minimizing training time while maintaining high accuracy in PDF malware prediction. This integration represents a powerful and efficient solution for combating PDF malware, advancing cybersecurity defenses against evolving threats.

6. Reference

1. Mejjaouli, Sobhi, and Sghaier Guizani. "PDF Malware Detection Based on Fuzzy Unordered Rule Induction Algorithm (FURIA)." Applied Sciences 13.6, 3980,2023.
2. Abu Al-Haija, Qasem, Ammar Odeh, and Hazem Qattous. "PDF malware detection based on optimizable decision trees." Electronics 11.19,3142,2022.
3. P. Pandi Chandran “Invasive weed optimization with stacked long short term memory for PDF malware detection and classification”.2022
4. U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, “Scalable, behavior-based malware clustering,” in Proc. Netw. Distrib. Syst. Secur. Symp, vol. 9, pp. 8–11,2019
5. D. Maiorca, B. Biggio, and G. Giacinto, “Towards adversarial malware detection: Lessons learned from PDF-based attacks,” ACM Comput. Surv., vol. 52, no. 4, pp. 1–36., 2019.
6. W. Hu and Y. Tan, “Generating adversarial malware examples for blackbox attacks based on GAN,” arXiv:1702.05983, 2017.
7. C. Smutz and A. Stavrou, “When a tree falls: Using diversity in ensemble classifiers to identify evasion in malware detectors,” in Proc. Netw. Distrib. Syst. Secur. Conf., 2016.
8. I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” vol. 1050, p. 20, 2015.
9. A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” arXiv:1611.01236., 2016.
10. S. Alfeld, X. Zhu, and P. Barford, “Data poisoning attacks against autoregressive models,” in Proc. 30th AAAI Conf. Artif. Intell., pp. 1452–1458, 2016.
11. J. Saxe and K. Berlin, “Deep neural network based bmalware detection using two dimensionalbinary program features”,in Proc. 10th Int Conf. Malicious Unwanted.Softw, pp. 11–20,2015
12. Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, “Droid-Sec: Deep learning in android malware detection,” ACM SIGCOMM Comput. Commun. Rev., vol. 44, no. 4, pp. 371–372, 2014.
13. M. A. Rajab, L. Ballard, N. Lutz, P. Mavrommatis, and N. Provos, “CAMP: Content-agnostic malware protection,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2013.
14. M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, “COMPA: Detecting compromised accounts on social networks,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2013.
15. G. Stringhini, C. Kruegel, and G. Vigna, “Shady paths: Leveraging surfing crowds to detect malicious web pages,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 133–144, 2013.
16. N. Šrndic and P. Laskov, “Detection of malicious PDF files based on hierarchical document structure,” in Proc. 28th Annu. Netw. Distrib. Syst. Secur. Symp., pp. 1–16, 2013.
17. J. Schlumberger, C. Kruegel, and G. Vigna, “Jarhead analysis and detection of malicious Java applets,” in Proc. 28th Annu. Comput. Secur. Appl. Conf., pp. 249–257, 2012.

18. P. Laskov and N. Šrndi´c, "Static detection of malicious JavaScript-bearing PDF documents," in Proc. 27th Annu. Comput. Secur. Appl. Conf., pp. 373–382, 2011.
19. G. Kakavelakis and J. Young, "Auto-learning of SMTP TCP transportlayer features for spam and abusive message detection," in Proc. 25th Int. Conf. Large Installation Syst. Admin., p. 18, 2011.
20. M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-bydownload attacks and malicious javaScript code," in Proc. 19th Int Conf. World Wide Web, pp. 281–290.,2010.