MALWARE DETECTION USING MACHINE LEARNING

SARANYA.C[1]
Assistant Professor
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore
c.saranyait@gmail.com

KAVINRAJ S.B[2]
712819104717
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore
kavinkavi641@gmail.com

SNEHA.P[3]
712819104712
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore
snehap1892002@gmail.com

YUVABHARATHI.V[4]
712819104724
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore
yuvabharathi001vp@gmail.com

KARTHICK.B[5]
712819104710
Dept of Computer Science & Engineering
RVS College of Engineering & Technology,
Coimbatore
karthickbrs281@gmail.com

Abstract—***Malware detection plays a crucial role in cyber-security with the increase in malware growth and advancements in cyber-attacks. Malicious software applications, malware are the primary source of many security problems. These intentionally manipulate malicious applications intend to perform unauthorized activities on behalf of their originators on the host machines for various reasons such as stealing advanced technologies and intellectual properties, governmental acts of revenge, and tampering sensitive information. Malware detection methods rely on signature databases, including malicious instruction patterns in today's practice. The signature databases are used for matching against a signature generated from a newly encountered executable. Nevertheless, more efficient mitigation methods are needed due to the fast expansion of malicious software on the Internet and their self-modifying abilities like polymorphic and metamorphic malware. We propose stacked bidirectional long short-term memory (Stacked BiLSTM) and generative pre-trained transformer based (GPT-2) models for detecting malicious code online without installing any antivirus software.The proposed algorithms, namely the bidirectional long short-term memory (BiLSTM) model and the generative pre-trained transformer 2 (GPT-2) detect malicious code pieces by examining assembly instructions obtained from static analysis results of Portable Executable (PE) files. Our BiLSTM model processes a sequence of input elements across time to learn and analyse the patterns. In contrast, the transformers-based GPT-2 model enables modelling long dependencies between input sequence elements with parallel sequence processing in which sequential data constituents can connect with others simultaneously.***

Keywords—Malware detection, Cybersecurity, GPT-2, BiLSTM, Malware attacks.

I. INTRODUCTION

The objective of MalFree is to develop a cyber security firmware that can effectively detect and prevent malware attacks on computer systems using advanced machine learning techniques. The firmware will have the following objectives :To develop a large dataset of malware samples and non-malware samples for training and evaluating the Stacked BiLSTM and GPT-2 models. To train the Stacked BiLSTM model to learn the temporal sequence of network traffic data and the GPT-2 model to generate contextualized representations of network traffic data. Tocombine the Stacked BiLSTM and GPT-2 models to develop a hybrid model for detecting and preventing malware attacks on computer systems. To evaluate the performance of the hybrid model in terms of detection accuracy, false positive rate, and speed compared to traditional signature-based detection systems and the Transformer-based system.

To develop a firmware that can run the hybrid model to detect and prevent malware attacks on computer systems. The objective of the proposed firmware is to provide an advanced and effective solution for detecting and preventing malware attacks on computer systems using the state-of-the-art machine learning techniques of Stacked BiLSTM and GPT-2. The hybrid model will leverage the power of both models to learn the temporal sequence and contextualized representations of network traffic data, enabling it to detect and prevent new and unknown malware variants with high accuracy and low false-positive rates. The firmware will provide real-time protection against detected threats, minimizing the impact of a malware attack on computer systems.

II. RELATED WORKS

All the correlated works that have been done that are related to the current problem are follows. [1]Caviglione, L.; Choras, M.; Corona, I.; Janicki, A.; Mazurczyk, W.; Pawlicki, M.; Wasielewska, K. Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection. IEEEAccess 2021, 9, 5371–5369..[3]Cannarile, A.; Dentamaro, V.; Galantucci, S.; Iannacone, A.; Impedovo, D.; Pirlo, G. Comparing Machine learning and Shallow Learning Techniquesfor API Calls Malware Prediction: A Study. Appl. Sci. 2022, 12, 1645.[5]Urooj, U.; Al-Rimy, B.A.S.; Zainal, A.; Ghaleb, F.A.; Rassam, M.A. Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. Appl. Sci. 2022, 12, 172.

III. PROBLEM DEFINITON

Thorough Malware (malicious software) is asignificant threat to computer systems, mobile devices, and networks worldwide. Malware can cause various types of damage, including stealing sensitive data, hijacking systems, and disrupting critical services. Traditional signature-based malware detection systems are not effective in detecting new and advanced malware variants, leading to an increased need for more sophisticated techniques.

Malware mitigation is based on comparing those signatures with the signature of an executable file of newly encountered files for malicious vs. benign detection. The signature-based malware detection is straightforward and fast, yet it may be ineffective against sophisticated malware or overlook relations. Another drawback of such detection methods is that the signatures database grows too quickly to keep up with the growth rate of new malware. Traditional signature-based detection systems rely on a database of known malware signatures to identify threats. However, new and unknown malware can evade detection by these systems, making them ineffective in preventing advanced threats.

The DL is the end-to-end learning approach, which refers to training a possibly complex learning system represented by a single model, DNN. The network represents the complete target system, automating feature extraction nearly without preprocessing. In this project, we extract assembly codes using an open-source disassembler objdump.

This tool creates sequences as documents or sentences. Those data are then used for model development, given that the assembly code provides accurate information for obtaining critical coding patterns. For this, we employ the disassembler output as input data to build a language model assisted with word embedding in a similar way to processing natural language.

IV. PROPOSED SYSTEM

The proposed system, Mal Free, is a cyber security online firmware that uses advanced machine learning techniques, specifically Stacked BiLSTM and GPT-2 based language models, to detect and prevent malware attacks. The proposed algorithms, namely proposes a Stacked BiLSTM and GPT-2 based machine learning language models for detecting malicious code. Developed language models using assembly instructions extracted from .text sections of malicious and benign Portable Executable (PE) files. BiLSTM model processes a sequence of input elements across time to learn and analyze the patterns. In contrast, the transformers-based GPT-2 model enables modeling long dependencies between input sequence elements with parallel sequence processing in which sequential data constituents can connect with others simultaneously. Then use the perspective of NLP modeling by DL to extract similar characteristics, i.e., syntactic and semantic characteristics of assembly instructions. This models were designed to effectively learn and extract the features and characteristics of assembly language and classify the polarity of files.

A. MalFreeWebTool

In order to simulate a real world scenario, a three-tiered web architecture is developed. This architecture consisted of web-servers, application servers, and a database server. A front load balancer is tasked with handling and distributing clients requests to the appropriate web servers. An internal load balancer is used to connect web servers to the application servers and to distribute requests among the application servers, and the application servers are all connected to a single database server. In this module we are going to build the online malware analysis tool using Python and Flask Framework. It is a cloud-based online tool that provides users with a report on system or device security threats. MalFree is a web service that scans registered device from a remote server. MalFree can help in protecting the user's device from getting infected with malware and other web security threats. Online Malfree scanning of system or laptops is widely used to protect system or laptops from viruses, spyware, malware, rootkits, trojans, phishing attacks, spam attacks and many more types of web attacks. It consists of Malware Model Building and Prediction Phase using Machine learning Algorithms. The malware report will provide with a list of all affected files including the possible reasons for detection.

B. Malware Classification

In this module, we developed malware detection approaches using Natural Language Processing(NLP) techniques with ML algorithms. The proposed algorithms, namely the BiLSTM model and GPT-2 detect malicious code pieces by examining assembly instructions obtained from static analysis results of PE Files. Our BiLSTM model processes a sequence of input elements across time to learn and analyze the patterns. In contrast, the transformers-based GPT 2 model enables modeling long dependencies between input sequence elements with parallel sequence processing.

C. Device Registration

In this module users of this system registered with this system. After Login the system with registered username and password. End users configure their system with this module using their system MAC and IP.

D. Attacker Model

In this module the attacker attack is to download and install malware on the victim machine. There are two types of techniques used by attackers to perform malware attacks, namely: (1) Social Engineering: using psychological manipulations and decoys to trick the victims into authorizing the downloading and installation of malware; and (2) Drive-by Download: designing a web page that contains malicious code to trigger the downloading and installation of malware automatically.

E. Malware Scanner

MalFree offers two types of online scans for computer. One is the Antivirus Scan that detects known malware and other malicious programs hiding in your

computer. The other one is the Prevent Scan that detects malware threats that are new and have unknown characteristics. it scans your computer for suspicious files. It is regularly updated to detect real-time threats. The one-click scan feature is present and shows blacklist status.

F. Performance Analysis

In this module the method of evaluating model performance is to calculate False Positive Rates in different runs (FPR = FP/FP+TN, where FP is the number of false positives and TN is the number of true negatives).The FPR shows the probability of a false alarm, i.e., a benign file detected as malware.

V. ALGORITHMS USED

In this work, we propose the use of two models, Stacked BiLSTM and GPT-2. We trained on a large dataset of malware samples to learn how to recognize common malware behaviors such as command-and-control communication, data exfiltration, and network reconnaissance.

When the model detects suspicious behavior in the network traffic, it sends an alert to the GPT-2 model, which generates a natural language alert message that can be displayed to the user or sent to a security operations center (SOC).

A. GPT-2

The GPT-2 component of MalFree is responsible for generating alerts and notifications based on the output of the Stacked BiLSTM model. When the model detects suspicious behavior in the network traffic, it sends an alert to the GPT-2 model, which generates a natural language alert message that can be displayed to the user or sent to a security operations center (SOC).

The system operates in real-time and is designed to be highly adaptable and flexible, working across multiple platforms and environments. It uses a combination of supervised and unsupervised learning approaches to identify both known and unknown malware threats.

The proposed system works as follows:

- **Data Collection**: MalFree collects data from multiple sources, including file systems, network traffic, and system logs.
- **Feature Extraction**: MalFree extracts relevant features from the collected data using techniques such as static and dynamic analysis.
- **Stacked BiLSTM**: MalFree uses a Stacked BiLSTM neural network to analyze the extracted features and detect malware attacks. The Stacked BiLSTM model is trained on a large dataset of known malware samples, enabling it to accurately identify new and previously unknown malware threats.
- **GPT-2:** MalFree also uses a GPT-2 language model to generate natural language descriptions of detected malware threats, making it easier for cyber security analysts to understand and respond to these threats.
- **Prevention and Response**: MalFree takes proactive measures to prevent malware attacks by blocking suspicious files and network traffic. It also generates alerts and notifications to alert cyber security analysts of potential threats, enabling them to take immediate action to prevent further damage.

VI. SOFTWARE REQUIREMENTS

A. Python 3.7.4

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a must for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain..

B. TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and gives developers the ability to easily build and deploy ML-powered applications..

C. Pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

D. NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

E. Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

F. Scikit Learn

scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

VII. OUTPUT SCREENS

Fig 1. User Registration Screen

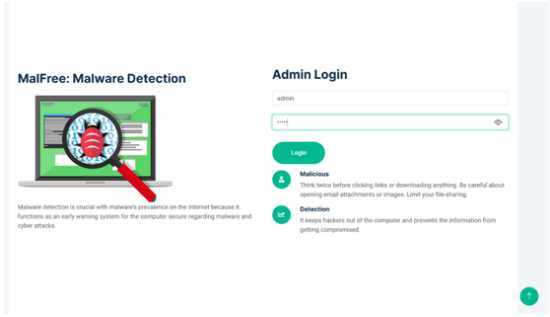

Fig 2. Admin Login Screen

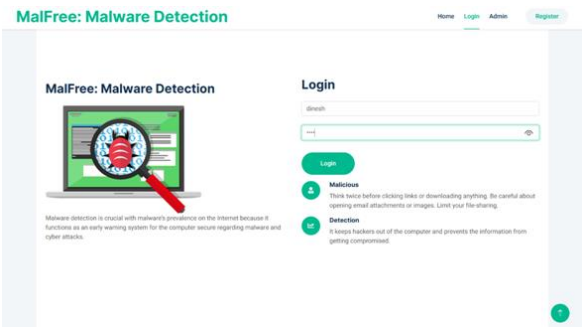

Fig 3.User login Screen

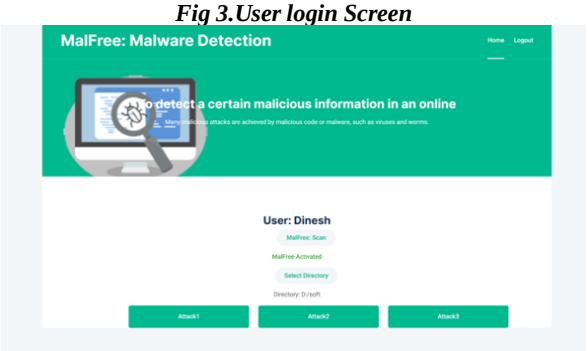

Fig 4 . Home screen

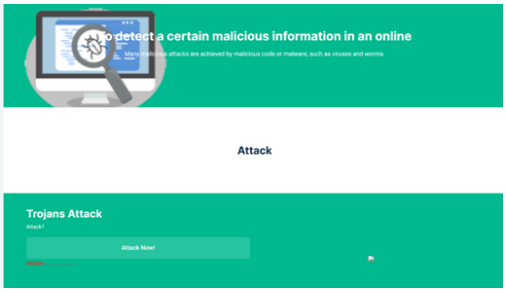

Fig 5 . Trojan attack

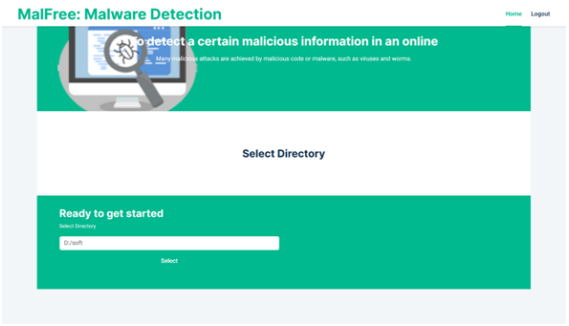

Fig6 . Path select /directory screen

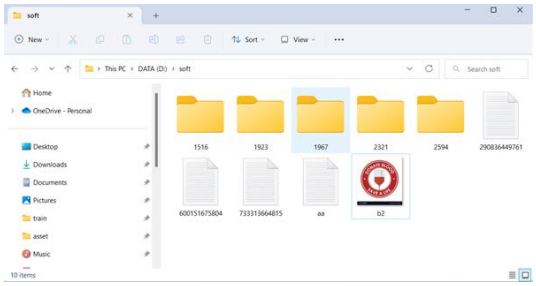

Fig 5. Malware attack

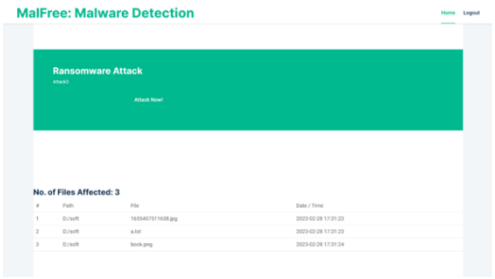

Fig 7. No.of files attack screen

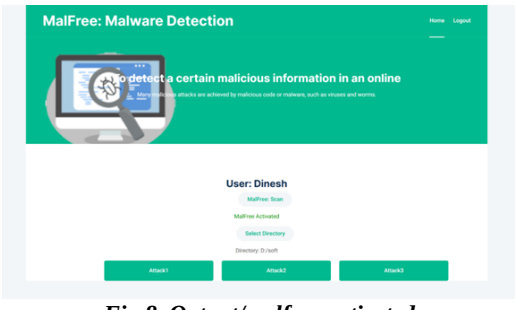

Fig 8. Output/malfree activated

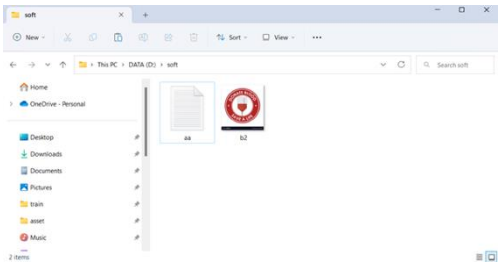

Fig 9. Output / malware removal

VIII. CONCLUSION

Malicious software applications, or malware, are the primary source of many security problems. These intentionally manipulative malicious applications intend to perform unauthorized activities on behalf of their originators on the host machines for various reasons such as stealing advanced technologies and intellectual properties, governmental acts of revenge, and tampering sensitive information, to name a few.

This project introduces MalFree, an interactive visualization platform for hybrid analysis and diagnosis of malware. This approach first represents the behavioral properties of the major malware classes (such as Trojan or backdoor), aiming to capture the common visual signatures of these malicious applications. MalFree implements a web-based prototype for demonstrating our approach to analyzing 60 malware samples from seven different classes .We focused on operation codes and operands, instead of opcodes only, to develop BiLSTM models and the decoder-based transformers GPT-2 models.

The resulting accuracy rate 95.4% shows that it is possible to classify malicious and benign assembly codes by GPT-2 with a custom pre-trained model. By experimental results, we showed that using byte streams of different formats may contribute to performance improvements. This also allowed for faster detection of malware classes, permitting a quicker response in anti-malware cyber security applications.

Overall, the application of this project can help identify malware types faster, prevent from malware attack and more accurately than contemporary approaches which can help save time when defending against malwares.

REFERENCES

[1] Caviglione, L.; Choras, M.; Corona, I.; Janicki, A.; Mazurczyk, W.; Pawlicki, M.; Wasielewska, K. Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection. IEEE Access 2021, 9, 5371–5396

[2] Cannarile, A.; Dentamaro, V.; Galantucci, S.; Iannacone, A.; Impedovo, D.; Pirlo, G. Comparing Machine learning and Shallow Learning Techniques for API CallsMalware Prediction: A Study. Appl. Sci. 2022, 12, 1645.

[3] Urooj, U.; Al-Rimy, B.A.S.; Zainal, A.; Ghaleb, F.A.; Rassam, M.A. Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. Appl. Sci. 2022, 12, 172.

[4] Hansen, S.S.; Larsen, T.M.T.; Stevanovic, M.; Pedersen, J.M. An approach for detection and family classification of malware based on behavioral analysis. In Proceedings of the 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 15–18 February 2016; pp. 1–5

[5] Morgan, S. Cybercrime Damages $6 Trillion by 2021. 2017. Available online: https://cybersecurityventures.com/hackerpocalypsecybercrime-report-2016/ (accessed on 15 July 2021)

[6] Villalba, L.J.G.; Orozco, A.L.S.; Vivar, A.L.; Vega, E.A.A.; Kim, T.-H. Ransomware Automatic Data Acquisition Tool. IEEE Access 2018, 6, 55043–55051

[7] Sahay, S.K.; Sharma, A.; Rathore, H. Evolution of Malware and Its Detection Techniques. In Advances in Intelligent Systems and Computing; Springer: Singapore, 2020; Volume 933, pp. 139–150

[8] Kakisim, A.G.; Nar, M.; Sogukpinar, I. Metamorphic malware identification using engine-specific patterns based on co-opcode graphs. Comput. Stand. Interfaces 2019, 71, 103443.

[9] Vignau, B.; Khoury, R.; Halle, S. 10 Years of IoT Malware: A Feature-Based Taxonomy. In Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion

[10] Asam, M.; Hussain, S.J.; Mohatram, M.; Khan, S.H.; Jamal, T.; Zafar, A.; Khan, A.; Ali, M.U.; Zahoora, U. Detection of exceptional malware variants using deep boosted feature spaces and machine learning. Appl. Sci. 2021, 11, 10464.

[11] I. Santos, Y. K. Penya, J. Devesa, and P. G. Garcia, "N-grams-based filesignaturesformalwaredetection,"2009.

[12] E. Konstantinou, "Metamorphic virus: Analysis and detection," 2008,TechnicalReportRHUL-MA-2008-2,

[13] Y. Ye, D. Wang, T. Li, and D. Ye, "Imds: intelligent malware detectionsystem," in *KDD*, P. Berkhin, R. Caruana, and X. Wu, Eds. ACM,2007,pp.1043–1047.

[14] M. Chandrasekaran, V. Vidyaraman, and S. J. Upadhyaya, "Spycon:Emulating user activities to detect evasive spyware," in *IPCCC*.IEEEComputerSociety,2007,pp.502–509.

[15] S. N. N. Kwang Loong and S. K. K. Mishra, "De novo SVM classi-fication of precursor microRNAs from genomic pseudo hairpins usingglobalandintrinsicfoldingmeasures."*Bioinformatics*,January2007 .

[16] K.Rieck,T.Holz,C.Willems,P.Düssel,andP.Laskov,"Learningand classification of malware behavior," in *DIMVA '08: Proceedings of the5th international conference on Detection of Intrusions and Malware,and Vulnerability Assessment*.Berlin, Heidelberg: Springer-Verlag,2008,pp.108–125.

[17] I. Yoo, "Visualizing Windows executable viruses using self-organizingmaps,"in*VizSEC/DMSEC'04:Proceedings ofthe2004ACMworkshopon Visualization and data mining for computer security*. New York,NY,USA:ACM,2004,pp.82–89.