INTELLIGENT MALWARE DETECTION USING DEEP LEARNING

[1]Mr. G. Gopala Krishna, [2]S. SOUMYA, [3]K. RAVITEJA, [4]T. NITHIN, [5]A. VARUN

[1](Assistant Professor) ,CSE. J.B. Institute of Engineering & Technology, Hyderabad

[2345]B,tech scholar ,CSE. J.B. Institute of Engineering & Technology, Hyderabad

ABSTRACT

Malicious software or malware continues to pose a major security concern in this digital age as computer users, corporations, and governments witness an exponential growth in malware attacks. Current malware detection solutions adopt Static and Dynamic analysis of malware signatures and behavior patterns that are time consuming and ineffective in identifying unknown malwares. Recent malware uses polymorphic, metamorphic and other evasive techniques to change the malware behaviors quickly and to generate large number of malwares. Since new malwares are predominantly variants of existing malwares, machine learning algorithms(MLAs) are being employed recently to conduct an effective malware analysis. This requires extensive feature engineering, feature learning and feature representation. By using advanced MLAs such as deep learning, the feature engineering phase can be completely avoided. Though some recent research studies exist in this direction, the performance of the algorithms is biased with the training data. There is a need to mitigate bias and evaluate these methods independently to arrive at new enhanced methods for effective zero-day malware detection. The train and test splits of public and private datasets used in the experimental analysis are disjoint to each other and collected in different time scales. In addition, we propose a novel image processing technique with optimal parameters for MLA sand deep learning architectures. A comprehensive experimental evaluation of these methods indicates that deep learning architectures outperform classical MLAs. Overall, this work proposes an effective visual detection of malware using a scalable and hybrid deep learning framework for real-time deployments. The visualization and deep learning architectures for static, dynamic and image processing-based hybrid approach in a big data environment is a new enhanced method for effective zero-day malware detection.

I. INTRODUCTION

The comprehensive scope of the project involves not only the development of an advanced malware detection system but also a keen focus on adaptability and resilience. The term "robust" underscores the system's capability to withstand a wide range of challenges and threats. It encompasses not just the detection of prevalent malware but also anticipates and addresses emerging threats. This forward-looking approach ensures that the system remains effective in safeguarding against novel attack vectors and evolving strategies employed by cyber adversaries. The system is designed to be applicable across diverse environments, ranging from individual users concerned about personal cybersecurity to large enterprises and government agencies with complex and interconnected systems. The scalability of the solution ensures that it can seamlessly integrate into existing security infrastructures, offering a layered defense against malware across different scales and complexities of computing environments. The project aims to develop an advanced malware detection system that goes beyond traditional approaches. It focuses not only on detecting existing malware but also on anticipating and addressing emerging threats. The term "robust" emphasizes the system's ability to withstand various challenges and threats effectively. This includes not only known malware but also novel attack vectors and evolving strategies used by cyber adversaries. The system's design considers its applicability across a wide range of environments, from individual users concerned about personal cybersecurity to large enterprises and government agencies with complex systems. Its scalability ensures that it can seamlessly integrate into existing security infrastructures, providing layered defense mechanisms against malware in different computing environments.

1.1 Purpose:

The overarching purpose of developing this malware detection system extends beyond immediate protection to contribute to the larger goal of fortifying the digital landscape. By providing businesses, governments, and individuals with a reliable defense mechanism, the project aims to foster a secure and resilient cyberspace. The integration of deep learning into malware detection aligns with a broader trend in harnessing advanced technologies for societal benefit. As digital threats continue to escalate, the project serves as a proactive initiative to mitigate risks and enhance the overall cybersecurity posture on a global scale. It resonates with the paradigm shift in artificial intelligence, emphasizing 2 the transformative potential of deep learning. By applying deep learning techniques to the complex problem of malware detection, the project not only enhances security measures but also contributes to the broader understanding of how advanced machine learning methodologies can be harnessed for societal well-being and safety. The primary objective of the project is to strengthen the digital landscape by providing reliable malware detection mechanisms to businesses, governments, and individuals. By incorporating deep learning into malware detection, the project aligns with the broader trend of using advanced technologies for societal benefit. It serves as a proactive initiative to mitigate cybersecurity risks on a global scale and contributes to the broader understanding of applying advanced machine learning methodologies for societal well-being and safety..

1.2 Description:

The multifaceted features of the project are designed to provide a holistic solution to the ever- evolving challenges in cybersecurity. Beyond the static and dynamic analysis, the system incorporates heuristic approaches that enable it to proactively identify potential threats based on behavioral patterns. This heuristic component adds an additional layer of intelligence, enabling the system to detect malware variants that might not yet have a known signature or specific behavior. Moreover, the adaptability of the system is enhanced by continuous learning mechanisms. The project incorporates mechanisms for the system to evolve and improve its detection capabilities over time. This ensures that the system remains current and effective in countering emerging threats, aligning with the dynamic nature of the cybersecurity landscape. The project incorporates several key features to comprehensively address the challenges in cybersecurity:

Static and Dynamic Analysis: The system employs both static and dynamic analysis techniques to examine malware samples. Static analysis involves inspecting the code without executing it, while dynamic analysis involves executing the code in a controlled environment to observe its behavior. This dual approach enhances the system's ability to detect known malware variants.

Heuristic Approaches: In addition to static and dynamic analysis, the system utilizes heuristic approaches to proactively identify potential threats based on behavioral patterns. This heuristic component adds an extra layer of intelligence to the

system, enabling it to detect malware variants that may not yet have known signatures or specific behaviors.

Adaptability and Continuous Learning: The system is designed to be adaptable and capable of continuous learning. It incorporates mechanisms to evolve and improve its detection capabilities over time. This ensures that the system remains current and effective in countering emerging threats in the dynamic cybersecurity landscape. The system's adaptability refers to its ability to evolve and adjust its detection capabilities in response to changes in the cybersecurity landscape. This includes updating detection algorithms, rules, or models to address new threats or vulnerabilities. Continuous learning mechanisms enable the system to improve its detection accuracy over time by analyzing new malware samples, collecting feedback from detection outcomes, and refining detection techniques based on the latest threat intelligence. By incorporating adaptability and continuous learning, the system can stay ahead of emerging threats and maintain its effectiveness in detecting and mitigating evolving malware threats.

Scalability: The solution is scalable, allowing it to be deployed across various environments and integrate seamlessly into existing security infrastructures. Whether deployed for individual users or large enterprises, the system offers comprehensive protection against malware. Scalability ensures that the malware detection system can be deployed across various environments, from individual users' devices to large enterprise networks, without sacrificing performance or effectiveness. The system's architecture and design accommodate the needs of different computing environments, allowing it to scale horizontally or vertically to handle increasing volumes of data, users, or network traffic. Seamless integration into existing security infrastructures enables organizations to leverage their existing investments in security tools and technologies while augmenting their defenses with advanced malware detection capabilities. By providing comprehensive protection against malware threats across diverse computing environments, the scalable malware detection system helps organizations safeguard their data, systems, and networks from cyber threats effectively

1.3 PROPOSED SYSTEM

In proposing a methodology to represent binaries in image representation. This can preserve the sequential information of byte codes and it is similar to. The proposed method converts the byte code into byte streams, thereby preserving the sequential order of binary code. Various deep learning architectures such as CNN and bidirectional LSTM and a combination of CNN and bi-directional LSTM architectures are evaluated with sampling and as well as without sampling techniques to handle the samples equally across all the classes. Though some recent research studies exist in this direction, the performance of the algorithms is biased with the training data. Two stage process scalable malware detection framework is proposed.

Advantages Of The Proposed System

Deep learning has some cool benefits when it comes to malware detection. It can detect both known and unknown threats, plus it can pick up on patterns in data that might go unnoticed by us humans. Pretty amazing, huh? It's great at quickly and accurately spotting malware, so it's a useful tool for quickly identifying and responding to any malicious activity.

II. LITERATURE SURVEY

Nataraj et al. (2011) - Malware images: Visualization and automatic classification [1]: This research likely explores the creation of visual representations of malware specimens, possibly utilizing techniques from image processing and computer vision. Visualizations can offer insights into the structural and behavioral attributes of malware, aiding in the identification of patterns and anomalies. Automatic classification, as discussed in the paper, implies the development of machine learning models capable of distinguishing between different types of malware based on their visual features. This could involve feature extraction from images and the application of classification algorithms to categorize malware families. Alazab et al. (2011) - Zero-day malware detection based on supervised learning algorithms of API call signatures [2]: The focus on zero-day malware detection indicates an emphasis on identifying threats that have not been previously encountered, a critical aspect in the rapidly evolving landscape of cybersecurity. The paper likely details the creation of datasets containing API call signatures, the sequences of function calls made by programs. Supervised learning algorithms, mentioned in the paper, suggest the use of labeled data to train models to recognize malicious patterns in API call sequences. The work likely contributes to enhancing the accuracy and timeliness of malware detection systems. Li et al. (2017) - Large-scale identification of malicious singleton files [3]: The challenge of large-scale identification of malicious singleton files underscores the necessity for scalable and efficient methods in handling extensive datasets. The paper may discuss techniques for data preprocessing, feature extraction, and classification algorithms suitable for handling large volumes of diverse files. Addressing the problem of singleton files implies the detection of isolated in- stances of potentially harmful content, highlighting the need for methods that can identify threats even when they exist in small numbers or in isolation. LeCun et al. (2015) - Deep learning [4]: This foundational work on deep learning by LeCun, Bengio, and Hinton introduces the concept of neural networks with multiple layers, enabling the automatic learning of hierarchical representations from data. Although not directly tied to malware, its inclusion suggests an acknowledgment of the transformative power of deep learning. In the context of cybersecurity, this could imply the application of deep neural networks for feature learning and abstraction, potentially improving the ability to detect and classify complex and evolving malware variants. Agarap and Pepito (2017) - Towards building an intelligent antimalware system: A deep learning approach using SVM for malware classification [5]: This paper likely presents a novel approach to antimalware systems by combining the strengths of deep learning and traditional SVM. Deep learning, with its ability to automatically extract hierarchical features, is integrated with SVM, a powerful classifier. The authors may discuss the architecture of the proposed system, the choice of deep learning model, the fusion with SVM, and the advantages gained from this hybrid approach. The work likely contributes insights into achieving a balance be- tween the automatic feature learning capabilities of deep learning and the robust classification abilities of SVMfor malware detection. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer [6]: This seminal work by Christopher Bishop serves as a foundational textbook in the field of pattern recognition and machine learning. It covers essential concepts, algorithms, and techniques, providing a solid theoretical background for understanding the principles underlying intelligent malware detection. The book's comprehensive nature makes it valuable for gaining insights into the mathematical foundations of various machine learning methods. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016) Deep Learning. MIT Press.[7]: Authored by leading experts in the field, this book delves into the world of deep learning, offering both theoretical depth and practical insights. It covers neural network architectures, training method- ologies, and applications across diverse domains. As deep learning is a pivotal component of your project, this resource provides an authoritative guide to the principles and practices of this powerful machine learning paradigm. Skoudis, E., & Zeltser, L. (2004). Malware: Fighting Malicious Code. Prentice Hall.[8]: Skoudis and Zeltser's book offers a detailed exploration of malware, covering its history, types, and countermeasures. Understanding the characteristics and behavior of malware is crucial for designing effective detection systems. This resource provides a comprehensive overview, assisting in contextualizing the threat landscape and informing the development of intelligent malware detection strategies. Landwehr, C., Bull, J., & McDermott, J. P. (Eds.). (2006). Cyber Adversary Characterization: Auditing the Hacker Mind. Digital Press.[9]: This book delves into the psychological and behavioral aspects of cyber adversaries,

offering insights into the motives, strategies, and tactics employed by hackers. Kolter, J. Z., & Maloof, M. A. (2006). Learning to Detect Malicious Executables in the Wild [10]: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. This paper explores the use of machine learning techniques for detecting malicious executable files. It likely discusses feature extraction methods and classification algorithms tailored for this task.

III. SYSTEM DESIGN

3.1 Proposed system architecture:

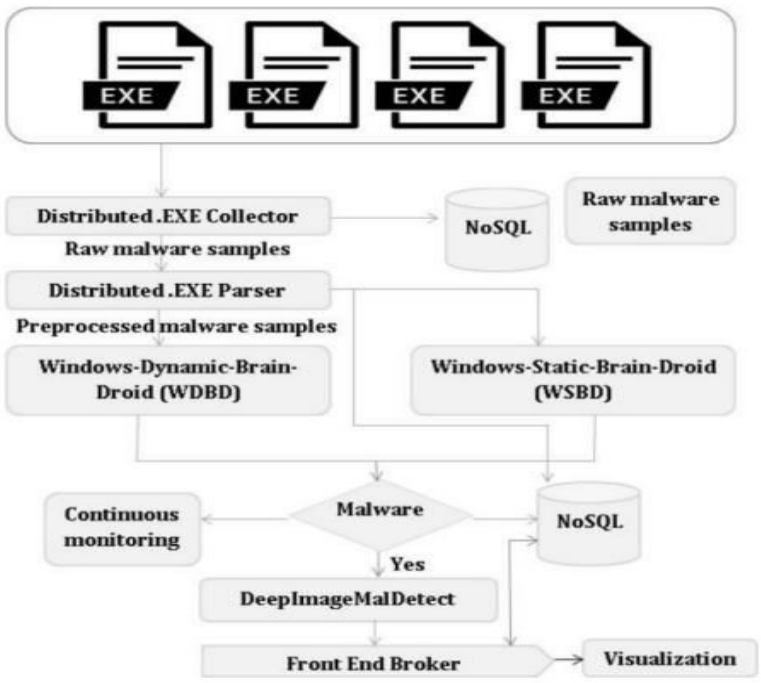

Figure 3.1: Architecture of Robust Intelligent Malware Detection Using Deep Learning.

The system's effectiveness is rigorously assessed through cross-validation, employing a range of performance metrics. Detected malware triggers alerts and reports, while a feedback loop ensures ongoing improvement through retraining and updates. Seamless integration with other security tools, a user-friendly interface, and a strong emphasis on security and privacy measures round out this comprehensive architecture, offering robust protection against evolving malware threats.

1. **Collection**:
   - This initial step involves gathering malware samples from various sources. These sources could include infected files, network traffic, or suspicious URLs.
   - The collection process might be automated through honeypots, which attract and capture malicious activity.
   - The goal is to accumulate a diverse set of malware for further analysis.
2. **Processing**:
   - Once collected, the malware samples undergo preliminary processing.
   - This step includes sorting, categorizing, and cleaning the samples.
   - Some common tasks during processing:
     - **Metadata Extraction**: Extracting information about the file (e.g., file type, creation date, and author).
     - **Decompression**: If the sample is compressed (e.g., in a ZIP file), it needs to be decompressed.
     - **Hashing**: Calculating hash values (e.g., MD5, SHA-256) for identification and comparison.
     - **Signature-Based Scanning**: Checking against known malware signatures.
     - **Behavioural Profiling**: Initial analysis of how the malware behaves.
3. **Analysis**:
   - The heart of the architecture lies in the **malware analysis** phase.
   - Different techniques are employed:
     - **Static Analysis**:
       - Examining the **code** without executing it.
       - Inspecting **file headers**, **strings**, and **API calls**.
       - Identifying **packers** or **obfuscation**.
     - **Dynamic Analysis**:
       - Running the malware in a controlled environment (e.g., a sandbox).
       - Observing its behaviour, interactions with the system, and network traffic.
       - Capturing API calls, registry changes, and file system modifications.
     - **Behavioural Analysis**:
       - Understanding how the malware interacts with the operating system and other software.
       - Detecting malicious activities (e.g., key logging, network communication).
     - **Code Reversing**:
       - Decompiling or disassembling the malware to understand its logic.
       - Identifying vulnerabilities or backdoors.
     - **Machine Learning**:
       - Using ML models to classify and predict malware behaviour.
       - Feature extraction from samples.
     - **Visualization**:
       - Creating visual representations (e.g., graphs, heat maps) to understand patterns.
       - Correlating data points.
4. **Visualization**:
   - The results of the analysis are often complex and need to be visualized.
   - Visualization tools help security analysts:
     - **Identify Patterns**: Visualize relationships between different malware samples.
     - **Track Behaviour**: Observe how malware evolves over time.
     - **Generate Reports**: Create summaries for stakeholders.
   - Examples of visualization techniques:
     - **Heat maps**: Show frequency or intensity of specific behaviours.

- **Graphs**: Represent connections between malware samples.
- **Timeline Charts**: Display events over time.

3.2 Implementation:

Implementing intrusion and malware detection using deep learning involves several steps.

• **Data Collection:** Gather a dataset of network traffic data containing both normal and malicious activities. Datasets like NSL-KDD, UNSW-NB15, or CICIDS2017 are commonly used for this purpose.

• **Data Preprocessing:** Clean the dataset, handle missing values, normalize the data, and perform feature engineering if necessary. This step is crucial for preparing the data for input into the deep learning model.

• **Model Selection:** In this malware detection project, several algorithms are employed to effectively classify and detect malware samples based on their features.

- **K-Nearest Neighbors (KNN):**

  KNN is a simple yet effective classification algorithm used for both supervised and unsupervised learning. In this project, KNN is utilized for supervised learning, where it classifies malware samples based on the similarity of their features to those of known malware families. KNN is chosen for its simplicity and effectiveness, particularly when dealing with a dataset where the decision boundaries between classes are not linear. It's easy to implement and does not require extensive training time.

- **Support Vector Machine (SVM):**

  SVM is a powerful supervised learning algorithm used for classification and regression tasks. In this project, SVM is applied to classify malware samples by finding the optimal hyperplane that best separates them into different classes. SVM is preferred for its ability to handle high-dimensional data and its effectiveness in finding complex decision boundaries. It's suitable for malware detection tasks where the data might not be linearly separable, making it a versatile choice for classification.

- **Artificial Neural Networks (ANN):**

  ANN is a machine learning model inspired by the biological neural networks of the human brain. In this project, ANN is utilized for its capability to learn complex patterns and relationships within the data, making it well-suited for malware detection tasks. ANN is chosen for its ability to handle large amounts of data and learn intricate features that may not be apparent through traditional methods. Its deep learning architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), can capture hierarchical representations of malware features, enhancing detection accuracy.

- **Particle Swarm Optimization (PSO):**

  PSO is a metaheuristic optimization algorithm inspired by the social behavior of bird flocking or fish schooling. In this project, PSO is employed to optimize feature selection in conjunction with ANN, enhancing the model's performance by selecting the most relevant features. PSO helps to overcome the curse of dimensionality by selecting a subset of features that maximize the classification performance while minimizing computational complexity. It aids in improving model efficiency and generalization by focusing on the most discriminative features.

• **Model Training:** Split the dataset into training and testing sets. Train the deep learning model on the training set using techniques like stochastic gradient descent (SGD), Adam, or RMSprop. Monitor the model's performance on the validation set and adjust hyper parameters accordingly.

• **Model Evaluation:** Evaluate the trained model on the testing set using metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). This step helps assess the model's performance in detecting intrusions and malware.

• **Fine-tuning and Optimization:** Fine-tuning the model by adjusting hyper parameters, trying different architectures, or employing techniques like dropout or batch normalization to improve performance.

• **Deployment:** Deploy the trained model in a real-world environment for intrusion and malware detection. This may involve integrating the model into existing security systems or building custom solutions for specific use cases.

IV. OUTPUT SCREENS

Step 1: Run Program and Upload Dataset. After opening click on 'UPLOAD MALWARE MALIMG DATASET' button to upload dataset. In above screen, I am uploading 'X.txt.npy or Y.txt.npy' binary malware dataset and after uploading dataset will get below screen

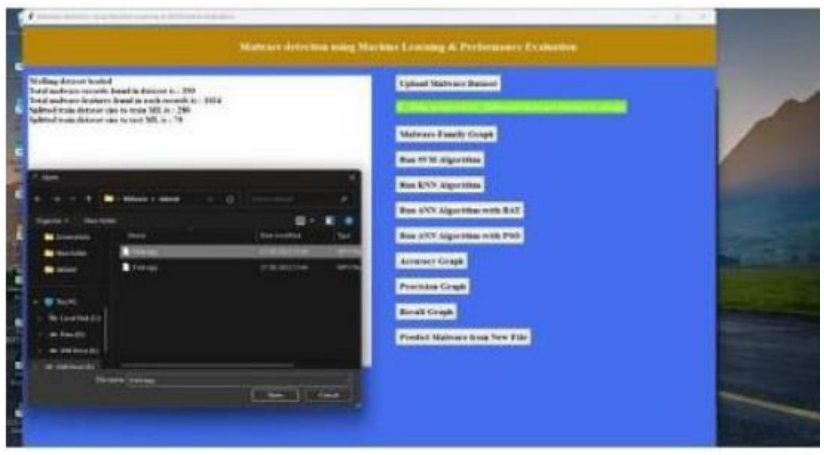

Figure 4.1: Upload Malware MalImg Dataset

Step 2: Click on malware family graphs. 'MALWARE FAMILY GRAPHS' Can Be Used to detect new Malware Samples by Comparing Them to Known Malware Families.

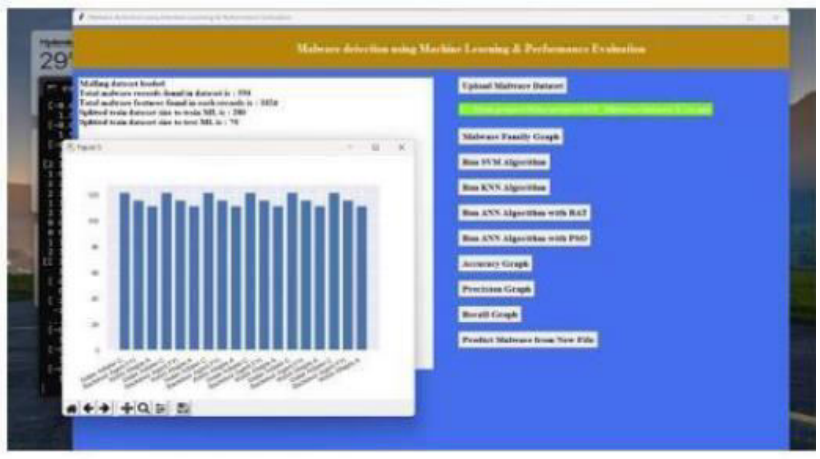

Figure 4.2: Malware Family Graphs

Step 3: Click on Run SVM Algorithm. The 'RUN SVM ALGORITHM' to Read Malware Dataset Generate Train and Test Model and Then Apply SVM Algorithm to Calculate Its Prediction Accuracy, FSCORE, Precision and Recall.

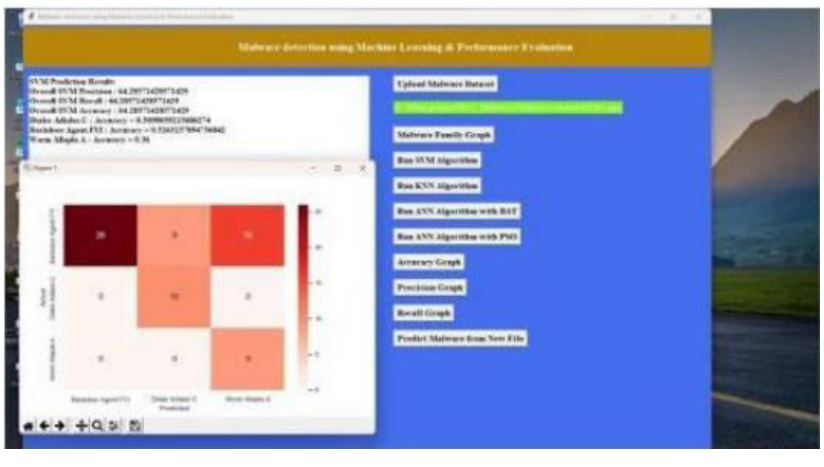

Figure 4.3: Run SVM Algorithm

Step 4: Click on Run KNN Algorithm. The 'RUN KNN ALGORITHM' to Get Its Performance

Figure 4.4: Run KNN Algorithm

Step 5: Click on Accuracy Graph.

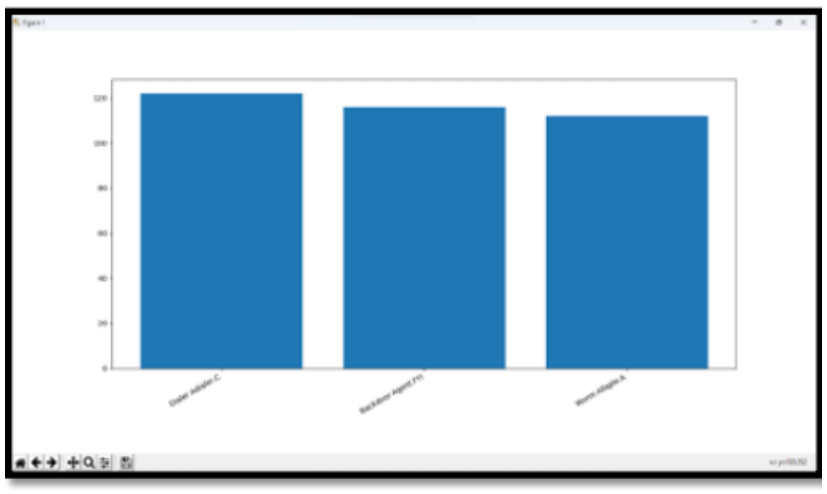

Figure 4.5: Accuracy Graphs

Step 6: Click on Precision Graph

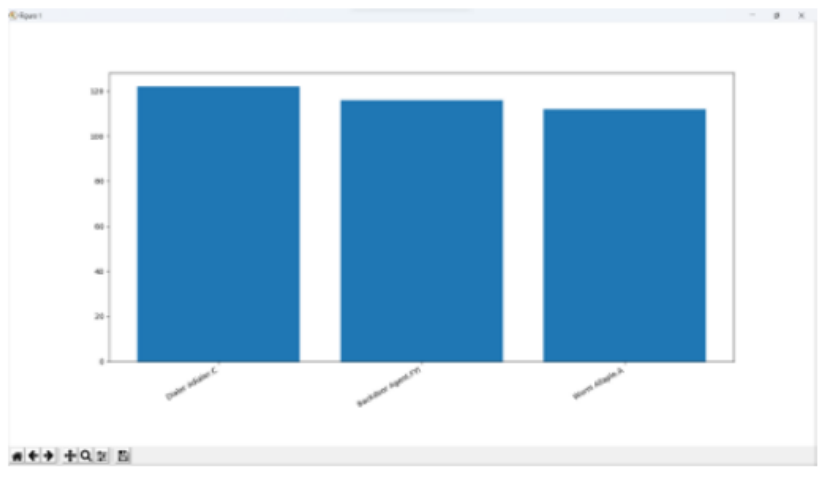

Figure 4.6: Precision Graphs

Step 7: Click on Predict Malware from New File. Upload One Binary File Called 1.Npy and Below Is the Malware Prediction of That File. In Above Screen, We Can See Uploaded Test File Contains 'Dialer Adialer.C' Malware Attack. Similarly, We Can Upload Other Files and Predict Class.

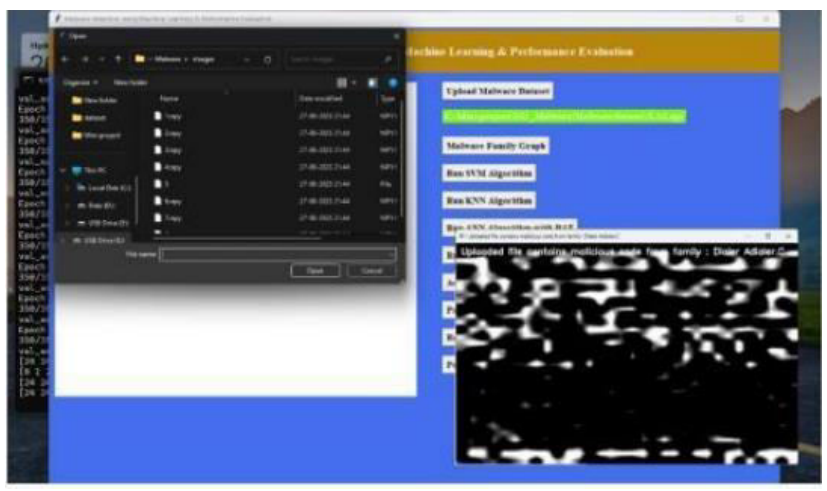

Figure 4.7 Predict Malware from New File

V. CONCLUSION

Deep learning is an awesome way to detect malware with precision. It can detect both familiar and unfamiliar threats, plus it can recognize patterns in the data that we might not catch with a human eye. Deep learning might have some limitations and can give off false positives, but it's still a great choice for detecting malicious activity. Definitely worth considering when creating a security plan.

VI. FUTURE ENHANCEMENT

This project proposed an efficient malware detection and designed a highly scalable framework to detect, classify and categorize zero-day malwares. This framework applies neural network on the collected malware from end user hosts and follows a two-stage process for malware analysis. First stage, a hybrid of static and dynamic analysis was applied for malware classification. In the second stage, malware was grouped into corresponding malware categories using image processing approaches. Various experimental analysis conducted by applying variations in the models on publicly available benchmark dataset and indicated the proposed model outperformed classical MLAs. The developed framework can analyze large number of malwares in real-time and scaled out to analyze even larger number of malwares by stacking a few more layers to the existing architectures. Future research entails exploration of these variations with new features that could be added to the existing data.

VII. REFERENCES

[1] Nataraj, Lakshmanan, et al. "Malware images: Visualization and automatic classification." Proceedings of the 2011 3rd International Conference on Communication Software and Networks. IEEE, 2011.

[2] Alazab, Mamoun, et al. "Zero-day malware detection based on supervised learning algorithms of API call signatures." Computers & Security 30.4 (2011): 256-265.

[3] Li, Shijun, et al. "Large-scale identification of malicious singleton files." Computers & Security 64 (2017): 123-135.

[4] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[5] Agarap, Abien Fred, and Jefferson Ian A. Pepito. "Towards building an intelligent antimalware system: A deep learning approach using SVM for malware classification." 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). IEEE, 2017.

[6] Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.

[7] Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. MIT press, 2016. [8] Skoudis, Ed, and Lenny Zeltser. Malware: Fighting Malicious Code. Prentice Hall, 2004.

[9] Landwehr, Carl, John Bull, and John P. McDermott (Eds.). Cyber Adversary Characterization: Auditing the Hacker Mind. Digital Press, 2006.

[10]Kolter, J. Z., & Maloof, M. A. (2006). Learning to Detect Malicious Executables in theWild. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[11]Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. (2011). Learning and classification of malware behavior. Machine Learning, 81(1), 77-94.

[12]Perdisci, R., Antonakakis, M., Nadji, Y., Dagon, D., & Lee, W. (2013). AVClass: A Tool for Massive Malware Labeling. Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security.

[13]Kolter, J. Z., & Maloof, M. A. (2005). The Learning-Boosting Algorithm: A Framework for Combining Boosting and Learning Automata. Proceedings of the 22nd International Conference on Machine Learning.

[14]Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., ... & Roli, F. (2013). Evasion attacks against machine learning at test time. Machine Learning, 87(2), 173-190.

[15]Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., & Glezer, C. (2009). ANDROIDS: A Novel Anomaly Detection System for Android Applications. Proceedings of the International Conference on Information Systems Security.