2023 International Conference on Data Science and Network Security (ICDSNS)

Performance Comparison of Recurrent Neural Networks for Metamorphic Malware

1st Shubh Mittal
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
shubh.mittal2020@vitstudent.ac.in

2nd Tisha Chawla
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
tisha.chawla2020@vitstudent.ac.in

3rd Jay Jajoo
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
jay.jajoo2020@vitstudent.ac.in

4th Muskan Bansal
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
muskan.bansal2020@vitstudent.ac.in

5th Harsha Parashar
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
harsha.parashar2020@vitstudent.ac.in

6th Ruby D
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
ruby.d@vit.ac.in

2023 International Conference on Data Science and Network Security (ICDSNS) | 979-8-3503-0159-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDSNS58469.2023.10245978

***Abstract*—Metamorphic malware remains a significant challenge in the field of information security as it constantly changes its code and structure, utilizing sophisticated methods to evade detection by antivirus software. This necessitates advanced techniques such as machine learning and deep learning algorithms to effectively detect it. However, the major hurdles lie in the lack of a comprehensive dataset for training algorithms and the continuous evolution of metamorphic malware. To bridge this research gap, a deep learning-based method is proposed that employs bidirectional LSTM (BiLSTM) to detect metamorphic malware based on attack vectors. A comparative analysis is conducted of BiLSTM's performance against other recurrent neural networks (RNNs) such as long short-term memory (LSTM), gated recurrent units (GRU), and a hybrid model of BiLSTM and GRU. The models were trained using a dataset comprising eight categories of metamorphic malware. The results demonstrate that BiLSTM surpasses other RNNs, achieving the highest accuracy of 93.25% compared to 93.20% for LSTM, 91.90% for GRU, and 86.41% for the hybrid model. The use of the outer activation function, Linear, and the inner activation function, SoftSign, in conjunction with the Adam optimizer, significantly contributed to this achievement. The proposed approach has the potential to greatly enhance current malware detection techniques and improve the overall security of computer systems. Furthermore, it can be extended to address other types of malware and cyber threats. By overcoming the challenges of limited training datasets and the ever-evolving nature of metamorphic malware, this research offers practical advantages, including enhanced threat detection capabilities, reduced false positives, and increased system security.**

Keywords*—Metamorphic Malware, API Calls, Bidirectional Long Short Term Memory, Recurrent Neural Network, Gated Recurrent Unit

I. Introduction

The proliferation of malware attacks has led to an increased demand for advanced security measures. Malware is a specific category of software that is created to damage computers, compromise security, or steal sensitive data. The traditional approach to detecting malware is to use signature-based methods, where a database of known malware signatures is used to identify malware. However, this approach is ineffective against metamorphic malware, which constantly changes its code to avoid detection.

Computer systems are severely threatened by metamorphic malware, as it can go undetected for extended periods, allowing it to inflict severe damage. Deep learning applications have demonstrated promise in detecting metamorphic malware. Machine learning is the superset of deep learning, in which training of ANNs is done to discover patterns from data. Deep learning has proven effective in different domains, including but not limited to identifying and classifying images, recognizing speech patterns, and processing natural language. In recent years, the implementation of its techniques has gained attention in the malware detection domain. The paper aims to develop a deep learning-based model that can effectively detect metamorphic malware. The project involves collecting a dataset of known metamorphic malware and benign software. The dataset is trained, and then the model is validated. The model is built using state-of-the-art techniques such as CNNs (convolutional neural networks) and RNNs. The project also involves evaluating the efficiency of the deep learning model on a test set of previously unseen metamorphic malware and benign software. The potential impact of this project is significant, as metamorphic malware is a growing threat that traditional antivirus software is struggling to detect. By developing a model that is capable of accurately detecting metamorphic malware, this project could help improve the security of computer systems and protect against cyberattacks.

II. Related Work

Detecting metamorphic malware is a major challenge because it is designed to constantly modify its code and behavioural, making it difficult for traditional detection methods that rely on signatures to identify it. A recent paper [1] utilizes genetic algorithms like J48 and deep neural networks, a novel approach was implemented to detect malware and the strategy selectively generated generations of malware samples with an optimization technique to achieve 5% better accuracy than other methods. Another study [2] highlights the effectiveness of anti-viruses against existing malware by using bypass detection, making the detection of malware difficult for the anti-virus software by inserting garbage values and substituting the instruction. The technique applied in the study is the Hidden Markov Model (HMM). A scalable deep learning (DL) model is presented in [3] to enable effective visual prediction of malware in real time deployments. The study also checked the efficiency of

979-8-3503-0159-5/23/$31.00 ©2023 IEEE

different ML models and concluded that DL gives the best result.

A study [4] proposes DAEMON, a novel malware classification tool that accurately classifies and explains the behavior of malware families using distinctive features. It is platform and dataset-agnostic and can handle the increasing number of difficult-to-analyze malware variants generated through automatic techniques. Another work [5] proposes a similar tool called SpecView that addresses the challenge of efficiently detecting malware variants using singular spectrum transformation to identify structural changes in challenging malware samples by converting binary code into time series spectrum data with one dimension. It has been shown to achieve high accuracy of over 99% in identifying malware variants that use evasive strategies.

[6] proposes a new malware detection model that combines different behavioural features to improve accuracy and reduce false-negative rates, overcoming the challenge of existing detection models that rely on vulnerable static or API-based features. [7] suggests using LSTM to analyze run trace outputs of Portable Executable files for efficient detection of metamorphic and polymorphic malicious software in dynamic analysis methods. Another study [8] brings attention to the challenge of monitoring the behaviour of sensitive Android applications in IoT devices and cyber-physical systems, given the limitations of the platform's security and privacy model. To tackle this issue, the authors suggest using a deep learning engine that employs a custom neural network and sparse learning to classify system-wide statistics, yielding superior results compared to existing benchmarks.

Detection of malware becomes difficult when it is disguised as a benign app by repackaging. A study [9] proposes an approach that is based on metamorphic testing principles and focuses on identifying changes in the feature vector of the app. By applying this approach, the system can achieve high accuracy in detecting repackaged malware while minimizing false positives and negatives. Study [10] uses a framework to detect malware that are constantly changing, which involves analyzing the behavior of the executable file during runtime to extract malicious features and then comparing the behavior of malware with a sample provided to determine similarity levels. This approach is an improvement over previous methods as it can detect metamorphic malware and eliminates false positives.

A Deep Learning strategy can be a promising answer when it comes to detecting all new and complicated malware types, as it is explained in [11] that conventional Artificial Intelligence algorithms are ineffective. It suggested a revolutionary deep-learning based malware classification architecture that uses a hybrid model to categorize different malware types. The four key phases of this architecture are data collecting, deep neural network architecture design, deep neural network design training, and deep neural network evaluation. An adaptive behavioural-based incremental batch learning malware variants detection model (AIBL-MVD) that uses concept drift detection and sequential deep learning is presented by [12] in order to accept new malware variants. Malware files are run in a sandbox environment and their application programming interface traces are collected to perform dynamic analysis and extract malicious behaviors. The base classifier was trained using a sequential deep learning model based on a subset of previous malware samples after the malware samples had been sorted. The fresh malware samples are gradually included into the learning model in an adaptive batch size incremental learning way, along with a portion of the old data.

DL-FHMC, which uses Control Flow Graph-based behavioural patterns for adversarial harmful software identification in IoT, has been introduced in [13]. It is a hierarchical and fine-grained learning approach utilized for detecting malware in IoT applications. This was accomplished by pulling out a thorough list of behavioural patterns from a dataset of IoT binaries containing malware. Shared execution flows, which are utilized as a modality for the detection of harmful conduct, represent this list.

A strong machine learning-based anti-malware solution has been offered, and [14] has used a visualization technique. In this, malware is portrayed as 2D graphics to overcome the limitations of conventional malware detection technologies. It imitates a layered ensemble strategy, sharing important traits with deep learning methods. [15] developed a brand-new approach to detecting malware on Android devices. It combines the use of static analysis with the collection of maximum helpful features from android applications. Then, it was applied to a dataset of the most recent and categorized Android applications, then supplied to a functional API Deep Learning Model. [16], which also concentrated on malware found in Android OS platforms, presents MalResLSTM, which is based on deep residual long short-term memory that aids in both identifying and categorizing various malware variants. This places a number of restrictions on the deep learning architecture in order to capture dependencies between the features that were taken from an APK file. Using a sequence model based on the residual LSTM network, the input sequence is then processed by translating these feature sets to a vector space. Table. I show the comparative analysis of the base papers of this research and Fig. 2 describes the architecture of the proposed approach.

TABLE. I. ANALYSIS OF BASE PAPERS

Paper Title	Dataset	Approach Used	Results
A Malware Detection Approach Using Autoencoder in Deep Learning [17]	Android side dataset	A deep learning model with an autoencoder network and a gray-scale representation of the infection.	Proposed model performs more accurately than traditional approaches and achieves an accuracy of 96.00%.
Deep Learning Approach for Intelligent Intrusion Detection System [18]	KDDCup 99 and other publicly avaliable datasets	KC-Net utilizes mental state knowledge and attention mechanisms to enhance understanding of speaker's mental states.	A thorough assessment of experiments was conducted to evaluate the performance of Deep Neural Networks (DNNs) and traditional ML classifiers on diverse publicly available benchmark datasets for malware.
Efficient Deep Learning Network With Multi-Streams for Android	AMD and DREBIN malware datasets	CNN's input data was obtained from some of the main files or portions that each malicious software on the Android platform had. The most	The method achieves a superior accuracy of 93.2% by employing a unique strategy that incorporates multi-streams and 1D convolution filters for analyzing the main files and sections of malware samples.

Malware Family Classification [19]		important files or parts for classifying malware families were then intuitively understood.	
Deep Learning and Visualization for Identifying Malware Families [20]	Kaggle Microsoft Malware Classification Challenge	Paper employed a technique for creating malware feature pictures that merged RNN and CNN techniques with static analysis of harmful code	RMVC (Random Matrix Verification Classifier) exhibits a highly accurate performance, evident from the nearly diagonal confusion matrix. It achieves remarkably low false positive rates, with the worst case being 0.0147 and an average of 0.0058 across all malware families.

III. Materials And Methods

A. Dataset

The paper employs a Windows exe API Calls dataset [21], which comprises of attack vectors and the names of the most perilous metamorphic malwares. The dataset is composed of two files, namely calls.csv and types.csv, containing the attack vectors. Each attack vector consists of a sequence of numbers recorded using API calls. The types.csv includes the names of the attack, corresponding to each attack vector in calls.csv. The dataset is comprised of a total of 7107 rows and has 8 unique classes or 8 distinct metamorphic malwares. The utilization of this dataset enables the development and evaluation of deep learning models for metamorphic malware detection, which can provide improved accuracy in identifying and mitigating the risks associated with such malware. The dataset is patriated in 85% for training the machine learning model and 15% for testing the model.

B. Data Visualization

Python libraries, Seaborn and Matplotlib have been utilized to create visualizations of the Windows exe API Calls dataset as shown in Fig. 1. These visualizations have helped in gain insights into the dataset and identify patterns that could aid in the development of the proposed approach for detecting metamorphic malware.

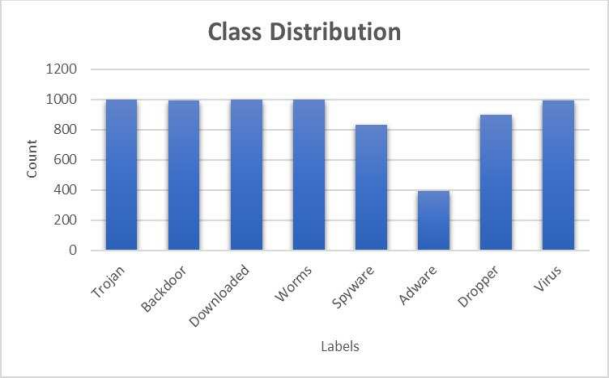

Fig. 1. Distribution of 8 distinct metamorphic malwares

C. Models used

1) Long Short Term Memory (LSTM)

*a) Activation function: Rectified Linear Unit:*This proposed approach for malware classification involves using a sequential neural network with Rectified Linear Unit (ReLU) activation function, which is known to perform well in modeling complex and nonlinear relationships. The model architecture includes three stacked layers of LSTM, each with 32 units of memory, with the first two layers having return sequences set to True for LSTM layer stacking. The model aims to effectively represent and classify malware text by transforming raw text data into fixed-length vectors of features, before feeding them through a stack of recurrent LSTM layers and dense layers with the active activation function ReLU. Ultimately, the model produces a classification result via its output neuron. Paper employed MSE for loss calculation and adam as optimizer , using MSE as loss function and Adam as optimizer might be beneficial in certain situations, such as when working with models that are required to output a probability distribution over multiple classes, or when training deep neural networks with large datasets.

b) Activation function: Softsign: This approach for malware classification involves using a LSTM model with the Softsign activation function to introduce non linearity at the output of each neuron in the network. The Softsign activation function is continuous and differentiable everywhere, making it suitable for optimization techniques such as gradient descent. The Softsign activation function has a range of -1 to 1, which can help normalize the output of each neuron in the network, making it easier for the network to learn and converge to an optimal solution during training. The output layer of the model uses the linear activation function to scale the output of the network to a specific range. The model uses an embedding layer to convert the data inputted into a numerical form, and is then fed into LSTM layers to capture the time-series nature of the data. The LSTM layers produce an output which is further processed through three dense layers for classification of the sequence data. In order to prevent overfitting, three Dropout layers are utilized as regularization layers. The output layer is equipped with linear activation, resulting in a continuous value that can be used for classification after thresholding. The model uses Mean Squared Error (MSE) as the loss function and RMSprop as the optimizer, which may lead to better convergence rates and generalization performance. To obtain the optimal performance for a specific problem, it is crucial to try out various loss functions and optimization algorithms.

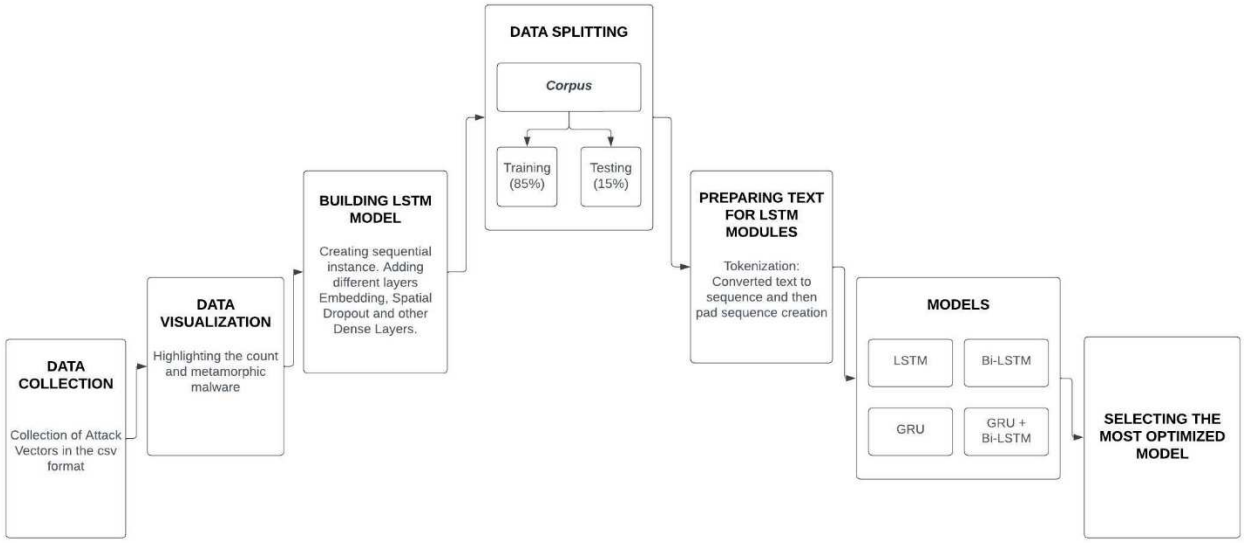

Fig. 2. Architecture of proposed model for metamorphic malware detection

2) Bi – Directional Long Short Term Memory (Bi-LSTM): This approach involved the use of a deep learning model, called malware_model, which is designed to detect metamorphic malware in sequential data. This model utilizes an Embedding layer to map input data into a lower-dimensional space. This architecture involves passing the output of a layer through multiple Bidirectional LSTM layers that process the sequence data in both directions, enabling long-term dependencies in the data to be captured. An additional Bidirectional LSTM layer is included to capture higher-level features in the data. The model also employs several Dense layers with activation functions and dropout for compressing the learned features into a lower-dimensional space. A final Dense layer with a linear activation function is included as the output layer to predict the probability of malware class. This architecture is particularly effective for regression problems and has demonstrated favorable outcomes in identifying metamorphic malware.

3) Gated Recurrent Unit (GRU): In this approach, a GRU neural network architecture is used to detect malware in textual data. The maximum number of words is set to 1000, and the model consists of an embedding layer, two GRU layers, and four dense layers. The embedding layer converts textual data into numerical vector representation. The first GRU layer has 32 units with a Softsign activation function and applies dropout regularization with a recurrent_dropout parameter set to 0.1. The second GRU layer has 32 units with a Softsign activation function and also applies dropout regularization. The four dense layers have 128, 256, and 128 units with the same activation function as the GRU layers, and dropout layers are added to each with a rate of 0.1. Finally, the output layer has a single unit with a linear activation function for regression tasks. This model is effective in handling sequential data, such as text data, with the use of GRU layers. The various dropout layers improve generalization and prevent overfitting. The smooth non-linear activation function used in the GRU and dense layers, Softsign, is suitable for this type of input data. The overall architecture aims to extract features and learn patterns to classify whether a given vector is malware or not.

4) GRU + Bi-LSTM: In this approach, a neural network model is developed using a combination of Bi-LSTM and GRU architectures, implemented in Keras. The malware_model function creates an architecture consisting of an embedding layer, two bidirectional recurrent layers (LSTM and GRU), and four dense layers. The model is designed to learn patterns in the input data to speculate whether a file is malicious or not. The maximum number of words and maximum length of input sequences are defined as max_words and max_len, respectively. The embedding layer has 300 units and is followed by a SpatialDropout1D layer to prevent overfitting. The model architecture includes an initial LSTM layer with 32 units and dropout regularization, with the return_sequences parameter switched to True. The second recurrent layer is a GRU layer with 32 units, also with bidirectional processing and dropout regularization. Four dense layers follow the recurrent layers, each with a dropout layer and an activation function specified by the input parameter. The output layer, named out_layer, has a single unit with a linear activation function. This model is suitable for sequential data analysis with bidirectional recurrent layers, dropout layers prevent overfitting and improve generalization, and the activation function can be customized to meet specific requirements.

IV. Results And Discussions

The proposed approach requires system requirements of a minimum of 8GB RAM, a multicore processor with a clock speed of at least 2.5 GHz, and a storage capacity of 100GB or more. The software dependencies include Python 3.7 or higher as the programming language, TensorFlow 2.0 or later as the deep learning framework, and the scikit-learn library for performing comparative analysis of the models. In this study, four different deep learning models were applied to classify malware data, and their performances were evaluated. To provide a comprehensive evaluation, a comparative analysis with existing state-of-the-art approaches was conducted as shown in Fig. 4. The first model used ReLU and softplus activation functions with the adam optimizer, and achieved an

accuracy of 85.38%. The second model employed SoftSign and linear activation functions with RMSPROP optimizer. It achieved an accuracy of 93.16% for 100 epochs at a batch size of 1000, 92.97% for 200 epochs at batch size of 1000, and 93.72% for 100 epochs at a batch size of 1000. The third model was based on GRU architecture, and it achieved an accuracy of 85.01%.

However, the model showed an absurd plot at some points, which indicated that the model overfitted the training data. The fourth model was based on BiLSTM architecture, and it achieved the best results with an accuracy of 93.25%. The BiLSTM model was not overfitting as it had a close match between the training and testing accuracy curves as shown in Fig.3.

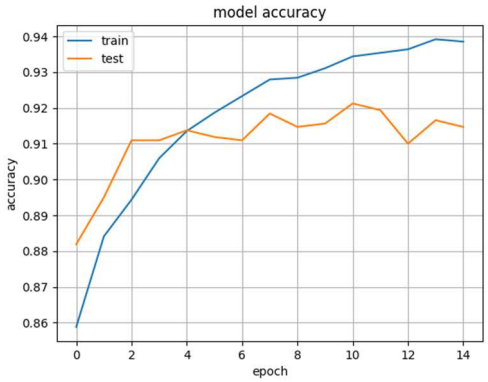

Fig. 3. The graph depicts the test vs train accuracy of model for 14 epochs wherein the accuracy of model hovers around 93.25%

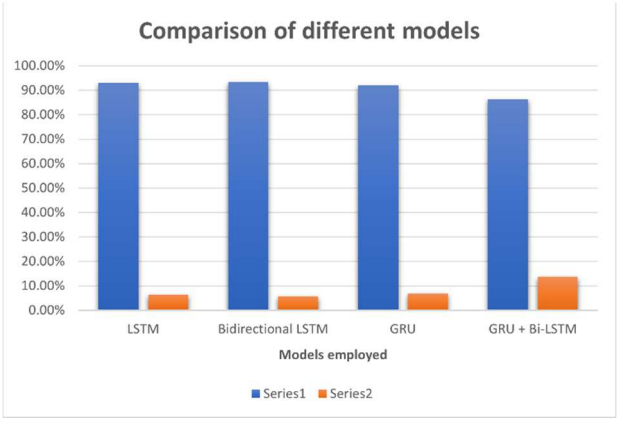

Fig. 4. The above graph represents the accuracy and loss of all 4 approaches used

Thus, this study demonstrated the effectiveness of deep learning models in classifying malware data. The BiLSTM architecture, with its appropriate hyperparameters and activation functions, yielded the best results, achieving an accuracy rate of 93.25% and loss of 5.72%. The findings emphasize the significance of architecture selection, activation functions, and optimization techniques in achieving superior performance. Future studies can further explore alternative deep learning architectures and optimization strategies to enhance the accuracy of malware classification. Evaluation metrics of different models are listed in Table. II.

TABLE. II. METRIC EVALUATION FOR DIFFERENT MODELS

Model	Activation Function		Optimizer	Accuracy	Loss
	Initial Layers	*Output Layer*			
LSTM	Relu	Softplus	Adam	85.38%	13.99%
	Softsign	Linear	Rmsprop	93.06%	6.46%
Bidirectional LSTM	Softsign	Linear	Adam	93.25%	5.72%
GRU	Softsign	Linear	Adam	91.94%	6.80%

V. CONCLUSION

The challenge of detecting metamorphic malware remains a pressing issue in the field of information security. This study proposes a novel deep learning-based approach using Bidirectional LSTM (BiLSTM) to detect metamorphic malware based on attack vectors. The performance of BiLSTM is compared against other recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), as well as a hybrid model of BiLSTM and GRU. The models are trained on a comprehensive dataset containing eight categories of metamorphic malware. The results demonstrate that BiLSTM outperforms other RNNs with the highest accuracy of 93.25%. This approach could significantly enhance current malware detection techniques and improve the overall security of computer systems. By focusing on utilizing attack vectors based on API calls, the research contributes a practical and effective approach to detect metamorphic malware in real-world scenarios. This practicality enhances the applicability of the proposed method and reinforces its potential for implementation in diverse information security systems. While the paper focuses on metamorphic malware, the proposed approach can be extended to detect and classify other types of malware and cyber threats. This flexibility and generalizability expand the impact of this work beyond the specific problem of metamorphic malware, contributing to the broader field of malware detection. Additionally, this method could be extended to other types of malware and cyber threats. One limitation of this study is the reliance on a specific dataset, which may not fully capture the diversity and complexity of metamorphic malware encountered in real-world scenarios but future research can explore more advanced deep learning models to improve detection accuracy and the development of more comprehensive datasets for training and testing.

REFERENCES

[1] Javaheri, D., Lalbakhsh, P., & Hosseinzadeh, M. (2021). A novel method for detecting future generations of targeted and metamorphic malware based on genetic algorithm. IEEE Access, 9, 69951-69970.

[2] Mumtaz, Z., Afzal, M., Iqbal, W., Aman, W., & Iltaf, N. (2021). Enhanced Metamorphic Techniques-A Case Study Against Havex Malware. IEEE Access, 9, 112069-112080.

[3] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust intelligent malware detection using deep learning. IEEE Access, 7, 46717-46738.

[4] Korine, R., & Hendler, D. (2021). DAEMON: dataset/platform-agnostic explainable malware classification using multi-stage feature mining. IEEE Access, 9, 78382-78399.

[5] Yu, J., He, Y., Yan, Q., & Kang, X. (2021). Specview: malware spectrum visualization framework with singular spectrum transformation. IEEE Transactions on Information Forensics and Security, 16, 5093-5107.

[6] Al-Hashmi, A. A., Ghaleb, F. A., Al-Marghilani, A., Yahya, A. E., Ebad, S. A., Saqib, M., & Darem, A. A. (2022). Deep-Ensemble and Multifaceted Behavioral Malware Variant Detection Model. IEEE Access, 10, 42762-42777.

[7] Acarturk, C., Sirlanci, M., Balikcioglu, P. G., Demirci, D., Sahin, N., & Kucuk, O. A. (2021). Malicious code detection: Run trace output analysis by LSTM. IEEE Access, 9, 9625-9635.

[8] Ma, H., Tian, J., Qiu, K., Lo, D., Gao, D., Wu, D., ... & Baker, T. (2020). Deep-learning–based app sensitive behavior surveillance for Android powered cyber–physical systems. IEEE Transactions on Industrial Informatics, 17(8), 5840-5850.

[9] Singh, S., & Kaiser, G. (2021, June). Metamorphic detection of repackaged malware. In 2021 IEEE/ACM 6th International Workshop on Metamorphic Testing (MET) (pp. 9-16). IEEE.

[10] Vahedi, K., Abbaspour, M., Afhamisisi, K., & Rashidnejad, M. (2019, October). Behavioral entropy towards detection of metamorphic malwares. In 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 78-84). IEEE.

[11] Aslan, Ö., & Yilmaz, A. A. (2021). A new malware classification framework based on deep learning algorithms. Ieee Access, 9, 87936-87951.

[12] Darem, A. A., Ghaleb, F. A., Al-Hashmi, A. A., Abawajy, J. H., Alanazi, S. M., & Al-Rezami, A. Y. (2021). An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning. IEEE Access, 9, 97180-97196.

[13] Abusnaina, A., Abuhamad, M., Alasmary, H., Anwar, A., Jang, R., Salem, S., ... & Mohaisen, D. (2021). Dl-fhmc: Deep learning-based fine-grained hierarchical learning approach for robust malware classification. IEEE Transactions on Dependable and Secure Computing, 19(5), 3432-3447.

[14] Roseline, S. A., Geetha, S., Kadry, S., & Nam, Y. (2020). Intelligent vision-based malware detection and classification using deep random forest paradigm. IEEE Access, 8, 206303-206324.

[15] İbrahim, M., Issa, B., & Jasser, M. B. (2022). A Method for Automatic Android Malware Detection Based on Static Analysis and Deep Learning. IEEE Access, 10, 117334-117352.

[16] Alotaibi, A. (2019). Identifying malicious software using deep residual long-short term memory. IEEE Access, 7, 163128-163137.

[17] Xing, X., Jin, X., Elahi, H., Jiang, H., & Wang, G. (2022). A malware detection approach using autoencoder in deep learning. IEEE Access, 10, 25696-25706.

[18] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. Ieee Access, 7, 41525-41550.

[19] Kim, H. I., Kang, M., Cho, S. J., & Choi, S. I. (2021). Efficient deep learning network with multi-streams for android malware family classification. IEEE Access, 10, 5518-5532.

[20] Sun, G., & Qian, Q. (2018). Deep learning and visualization for identifying malware families. IEEE Transactions on Dependable and Secure Computing, 18(1), 283-295.

[21] Ocakat, O. (n.d.). LSTM Malware Detection [GitHub repository]. Retrieved from https://github.com/ocatak/lstm_malware_detection/blob/master/calls.zip