ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

ENHANCED MALWARE DETECTION WITH DEEP LEARNING ALGORITHMS

P.Swapna[1], J.Murali[2], A.Keerthi[3], Ch.Madhan[4], D.Kripal Reddy[5], D.Sai Nikhil[6]

Assistant Professor, Department of Computer Science and Engineering[1]

Student, Department of Computer Science and Engineering[2,3,4,5,6]

Sree Dattha Institute of Engineering and Science, Sheriguda, Telangana. [1,2,3,4,5,6]

ABSTRACT

Malicious software, commonly known as malware, continues to pose a significant security threat to individuals, businesses, and governments in the modern digital age, especially with the exponential increase in malware attacks. Current malware detection systems rely on static and dynamic analysis of malware signatures and behavior patterns to discover unknown infestations, but this process is often inefficient and time-consuming. Due to evasive techniques like metamorphism and polymorphism, modern malware can quickly alter its behavior and generate numerous new variants. Most new viruses are variants of existing malware, which is why machine learning algorithms (MLAs) have recently been utilized to efficiently evaluate malware. However, traditional MLAs require extensive feature engineering, which is time-intensive. Advanced MLAs, such as deep learning, can bypass the feature engineering stage altogether. Despite recent research in this field, the performance of these algorithms can be biased when trained on specific data. To develop more effective methods of detecting zero-day malware, it is crucial to eliminate bias and conduct independent evaluations of these methods.This work addresses a gap in the literature by comparing traditional MLAs with deep learning architectures for malware detection, classification, and categorization using both public and private datasets. Our experimental study involves training and testing with datasets that include timestamps. We propose a novel approach to image processing that leverages optimal parameters in deep learning architectures and MLAs. Extensive experimental investigations have shown that deep learning architectures outperform conventional MLAs. Our research concludes that a hybrid deep learning architecture, which is both scalable and suitable for real-time deployments, can successfully detect malware visually. This hybrid approach, which integrates image processing, visualization, and deep learning within a big data framework, represents an enhanced and novel method for effectively detecting zero-day malware.

Index-Terms: Malware detection, machine learning algorithms, deep learning, metamorphism, polymorphism, zero-day malware, image processing, big data framework, feature engineering.

I. INTRODUCTION

In this digital era of Industry 4.0, rapid technological advancement affects both individuals and businesses in their day-to-day operations. The growth of the IoT and related applications is the foundation of the modern concept of the information society. Unfortunately, security issues make it hard to enjoy the benefits of this industrial revolution. Cybercriminals aim to steal

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

critical information for financial gain or to disrupt system operations by targeting both individual computers and networks. These cybercriminals utilise harmful software, often known as malware, to compromise computers and cause serious harm [1]. The purpose of malicious software is to cause harm to the operating system (OS). Adware, spyware, virus, worm, trojan, rootkit, backdoor, ransomware, and command and control (C&C) bot are just a few examples of the various titles given to malware based on its functions and behaviours. Malware identification and mitigation is an ongoing area of focus for the cyber security sector. Malware authors get more adept at evading detection when new ways are developed by researchers.

II.LITERATURE SURVEY

- Tang, M., Alazab, M., & Luo, Y. proposed that complex Big Data systems in modern organisations are progressively becoming targets for attacks by existing and emerging threat agents. Elaborate and specialised attacks will increasingly be crafted to exploit vulnerabilities and weaknesses. With the ever-increasing trend of cybercrime and incidents due to these vulnerabilities, effective vulnerability management is imperative for modern organisations regardless of their size. However, organisations struggle to manage the sheer volume of vulnerabilities discovered on their networks. Moreover, vulnerability management tends to be more reactive in practice. Rigorous statistical models, simulating anticipated volume and dependence of vulnerability disclosures, will undoubtedly provide important insights to organisations and help them become more proactive in the management of cyber risks. By leveraging the rich yet complex historical vulnerability data, our proposed novel and rigorous framework has enabled this new capability. By utilising this sound framework, we initiated an important study on not only handling persistent volatilities in the data but also further unveiling the multivariate dependence structure amongst different vulnerability risks. In sharp contrast to the existing studies on univariate time series, we consider the more general multivariate case, striving to capture their intriguing relationships. Through our extensive empirical studies using real-world vulnerability data, we have shown that a composite model can effectively capture and preserve long-term dependency between different vulnerability and exploit disclosures. In addition, the paper paves the way for further study on the stochastic perspective of vulnerability proliferation towards building more accurate measures for better cyber risk management as a whole.

- Ross Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Michel J.G. van Eeten, Michael Levi, Tyler Moore, and Stefan Savage present what they believe to be the first systematic study of the costs of cybercrime. It was prepared in response to a request from the UK Ministry of Defence following scepticism that previous studies had hyped the problem. For each of the main categories of cybercrime, we set out what is and is not known of the direct costs, indirect costs, and defence costs – both to the UK and to the world as a whole. We distinguish carefully between traditional crimes that are now ‘cyber’ because they are conducted online (such as tax and welfare fraud); transitional crimes whose modus operandi has changed substantially as a result of the move online (such as credit card fraud); new crimes that owe their existence to the Internet; and what we might call platform crimes, such as the provision of botnets which facilitate other crimes rather than being used to extract money from victims

directly. As far as direct costs are concerned, we find that traditional offences such as tax and welfare fraud cost the typical citizen in the low hundreds of pounds/Euros/dollars a year; transitional frauds cost a few pounds/Euros/dollars; while the new computer crimes cost in the tens of pence/cents. However, the indirect costs and defence costs are much higher for transitional and new crimes. For the former, they may be roughly comparable to what the criminals earn, while for the latter, they may be an order of magnitude more. As a striking example, the botnet behind a third of the spam sent in 2010 earned its owners around US$2.7m, while worldwide expenditures on spam prevention probably exceeded a billion dollars. We are extremely inefficient at fighting cybercrime; or to put it another way, cybercrooks are like terrorists or metal thieves in that their activities impose disproportionate costs on society. Some of the reasons for this are well-known: cybercrimes are global and have strong externalities, while traditional crimes such as burglary and car theft are local, and the associated equilibria have emerged after many years of optimisation. As for the more direct question of what should be done, our figures suggest that we should spend less in anticipation of cybercrime (on antivirus, firewalls, etc.) and more in response – that is, on the prosaic business of hunting down cyber-criminals and throwing them in jail.

- Mamoun Alazab proposed that malware is a major security threat confronting computer systems and networks and has increased in scale and impact from the early days of ICT. Traditional protection mechanisms are largely incapable of dealing with the diversity and volume of malware variants evident today. This paper examines the evolution of malware, including the nature of its activity and variants, and the implication of this for computer security industry practices. As a first step to address this challenge, I propose a framework to extract features statically and dynamically from malware that reflect the behavior of its code, such as the Windows Application Programming Interface (API) calls. Similarity-based mining and machine learning methods have been employed to profile and classify malware behaviours. This method is based on the sequences of API sequence calls and frequency of appearance. Experimental analysis results using large datasets show that the proposed method is effective in identifying known malware variants and also classifies malware with high accuracy and low false alarm rates. This encouraging result indicates that classification is a viable approach for similarity detection to help detect malware. This work advances the detection of zero-day malware and offers researchers another method for understanding impact.

- Islam, R., & Yearwood, J. proposed that malware replicates itself and produces offspring with the same characteristics but different signatures by using code obfuscation techniques. Current generation Anti-Virus (AV) engines employ a signature-template type detection approach where malware can easily evade existing signatures in the database. This reduces the capability of current AV engines in detecting malware. In this paper, we propose a hybrid framework for malware detection by using the hybrids of Support Vector Machines Wrapper, Maximum-Relevance-Minimum-Redundancy Filter heuristics where Application Program Interface (API) call statistics are used as malware features. The novelty of our hybrid framework is that it injects the filter's ranking score in the wrapper selection process and combines the properties of both wrapper and filters and API call statistics, which can

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

detect malware based on the nature of infectious actions instead of signature. To the best of our knowledge, this kind of hybrid approach has not been explored yet in the literature in the context of feature selection and malware detection. Knowledge about the intrinsic characteristics of malicious activities is determined by the API call statistics which is injected as a filter score into the wrapper's backward elimination process in order to find the most significant APIs. While using the most significant APIs in the wrapper classification on both obfuscated and benign types malware datasets, the results show that the proposed hybrid framework clearly surpasses the existing models, including the independent filters and wrappers using only a very compact set of significant APIs. The performances of the proposed and existing models have further been compared using binary logistic regression. Various goodness of fit comparison criteria such as Chi Square, Akaike's Information Criterion (AIC), and Receiver Operating Characteristic Curve (ROC) are deployed to identify the best performing models. Experimental outcomes based on the above criteria also show that the proposed hybrid framework outperforms other existing models of signature types, including independent wrapper and filter approaches to identify malware. A signature-free malware detection approach has been proposed. A hybrid wrapper-Filter based malware feature selection has been proposed. The proposed hybrid approach can take advantage of both filter and wrapper. Models have also been validated by statistical model selection criteria such as Chi Square and Akaike information criterion (AIC).

III.PREVIOUS WORK:

Current malware detection systems rely on static and dynamic analysis of malware signatures and behaviour patterns to discover unknown infestations, but this process is inefficient and takes a lot of time. Thanks to evasive techniques like metamorphism and polymorphism, malicious software today may quickly alter its behaviour and generate a large amount of new malware. Most new viruses are variants of existing malware, hence it is useful that machine learning algorithms (MLAs) have recently been utilised to efficiently evaluate malware. For this, you'll need to put in a lot of time learning about and working with features.

Drawback:

Major security concern in this digital age as computer users, corporations, and governments witness an exponential growth in malware attacks

VI.PROPOSED MODEL:

In order to identify malware, our research suggests ScaleMalNet, a Big Data–ready, scalable deep learning network architecture. Overall, the main takeaways from this research are: ScaleMalNet allows for the distributed application of pre-processing and the distributed collecting of malware samples from different sources. It is a revolutionary hybrid framework method. The framework can process a large number of malware samples either in real-time or as required. 2) The suggested method for malware classification using image processing is novel. Thirdly, ScaleMalNet employs a two-step procedure. Initial step: identifying malicious or legitimate executable files using static and dynamic analysis. The second thing it does is classify harmful executable files into the appropriate malware family. 4) An unbiased evaluation of the efficacy of deep learning architectures and conventional MLAs, as part of a benchmarking study of several malware analysis models.

Advantages:

ScaleMalNet follows two stage approach, in

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

the first stage the executables file is classified into malware or legitimate using Static and Dynamic analysis and in second stage the malware executables file is categorized into corresponding malware family.

SYSTEM ARCHITECTURE:

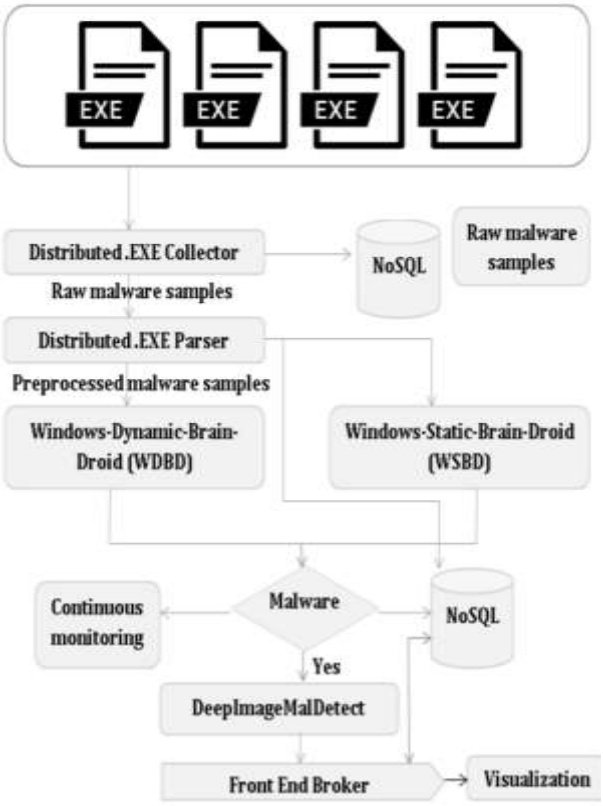

Figure.1 System Architecture

V.MODULES DESCRIPTION:

1.USER

1) A new proposal of a scalable and hybrid framework, namely ScaleMalNet which facilitates to collect malware samples from different sources in a distributed way and to apply pre-processing in a distributed manner. The framework has the capability to process large number of malware samples both in real-time and on demand basis.
2) A proposal of a novel image processing technique for malware classification.
3) ScaleMalNet follows two stage approach, in the first stage the executables file is classified into malware or legitimate using Static and Dynamic analysis and in second stage the malware executables file is categorized into corresponding malware family.
4) An independent performance evaluation of classical MLAs and deep learning architectures, benchmarking various malware analysis models.

2.MALWARE CLASSIFICATION

Several security researchers have applied domain level knowledge of portable executables (PE) for static malware detection. At present, analysis of byte n-grams and strings are the two most commonly used methods for static malware detection without domain level knowledge. However, the ngram approach is computationally expensive and the performance is considerably very low . It is often difficult to apply domain level knowledge to extract the necessary features when building a machine learning model to distinguish between the malware and benign files. This is due to the fact that the windows operating system does not consistently impose its own specifications and standards. Due to constantly changing specifications and standards from time to time, the malware detection system warrants revisions to meet future security requirements. To address this, has applied machine learning algorithms (MLAs) with the features obtained from parsed information of PE file. They adopted formatting of agnostic features such as raw byte histogram, byte entropy histogram which was taken from, and in addition employed string extraction.

3.DEEP NEURAL NETWORK (DNN)

A feed forward neural network (FFN)

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

creates a directed graph in which a graph is composed of nodes and edges [16]. FFN passes information along edges from one node to another without formation of a cycle. Multi-layer perceptron (MLP) is a type of FFN that contains 3 or more layers, specifically one input layer, one or more hidden layer and an output layer in which each layer has many neurons, called as units in mathematical notation. The number of hidden layers is selected by following a hyper parameter tuning approach. The information is transformed from one layer to another layer in forward direction without considering the past values. Moreover, neurons in each layer are fully connected. An MLP with n hidden layers can be mathematically formulated as given below:

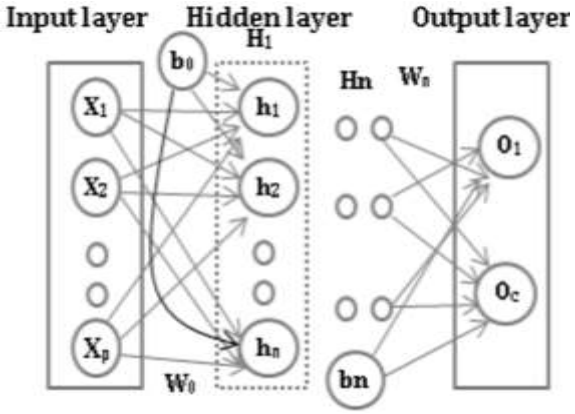

Figure.2 CNN Model

4. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional network or convolutional neural network or CNN is supplement to the classical feed forward network (FFN), primarily used in the field of image processing . It is where all connections and hidden layers and its units are not shown. Here, m denotes number of filters, ln denotes number of input features and p denotes reduced feature dimension, it depends on pooling length. In this work, CNN network composed of convolution 1D layer, pooling 1D layer and fully connected layer. A CNN network can have more than one convolution 1D layer, pooling 1D layer and fully connected layer. In convolutional 1D layer, the filters slide over the 1D sequence data and extracts optimal features. The features that are extracted from each filter are grouped into a new feature set called as feature map. The number of filters and the length are chosen by following a hyper parameter tuning method. This in turn uses non-linear activation function, ReLU on each element. The dimensions of the optimal features are reduced using pooling 1D layer using either max pooling, min pooling or average pooling. Since the maximum output within a selected region is selected in max pooling, we adopt max pooling in this work. Finally, the CNN network contains fully connected layer for classification. In fully connected layer, each neuron contains a connection to every other neuron. Instead of passing the pooling 1D layer features into fully connected layer, it can also be given to recurrent layer, LSTM to capture the sequence related information. Finally, the LSTM features are passed into fully connected layer for classification.

VI.RESULTS

To run this project double click on ‘run.bat’ file to get below screen

Figure.3 Main screen

In above screen click on ‘Upload Malware MalImg dataset’ button to upload dataset

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

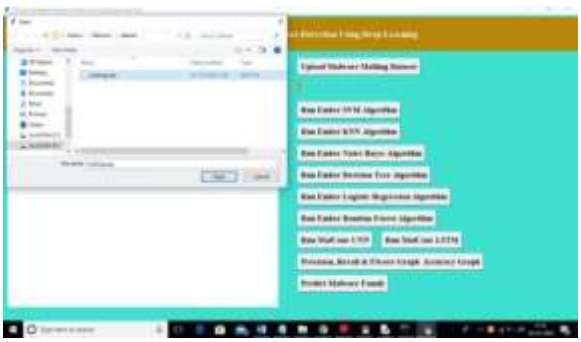

Figure.4

In above screen I am uploading 'malimg.npz' binary malware dataset and after uploading dataset will get below screen

Figure.5

Now click on 'Run Ember SVM algorithm' button to read malware dataset and generate train and test model and then apply SVM algorithm to calculate its prediction accuracy, FSCORE, Precision and Recall. If algorithm performance is good then its accuracy, precision or recall values will be closer to 100.

Figure.6

In above screen we got SVM precision, recall and fSCORE. Now click on 'Run Ember KNN Algorithm' button to get its performance

Figure.7

In above screen we got KNN details and now click on 'Naïve Bayes' button to get its performance details

Figure.8

In above screen we got naïve bayes details and now click on 'Decision Tree' button to get its performance details

Figure.9

In above screen we got decision tree details and now click on 'Logistic Regression' button to get its details

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Figure.10

In above screen we got logistic regression details and now click on 'Run Random Forest' button to get its performance

Figure.11

In above screen we got random forest details and now click on 'Run MalConv CNN' button to get its performance details. CNN may take 10 minutes to complete execution and u can check its ongoing processing in black console

Figure.12

In above console it will take 10 epochs iteration and for each iteration it calculate accuracy for 8395 malware data. So u need to wait till all 10 epochs completed then u will get its performance details

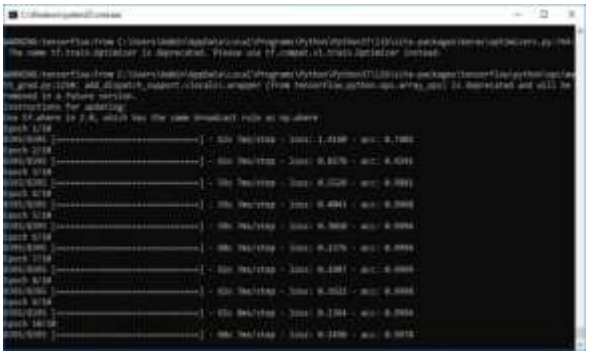

Figure.13

In above screen we can see CNN complete all 10 epochs and after that we will get accuracy details in main screen

Figure.14

In above screen we got CNN performance values and now click on 'Run MalConv LSTM' button to run LSTM algorithm. Similar to CNN LSTM also take 10 minutes and u can see ongoing process in below screen

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 2, 2024

Figure.15

Figure.16

In above screen we can see LSTM details now click on 'Precision, Recall & FScore' button to get comparson graph for all metrics and all algorithms

Figure.17

In above screen we can see precision graph for all algorithms and CNN get better performance. In above graph x-axis represents algorithm name and y-axis represents precision value and now close above graph to get recall graph

Now click on accuracy button to get accuracy graph

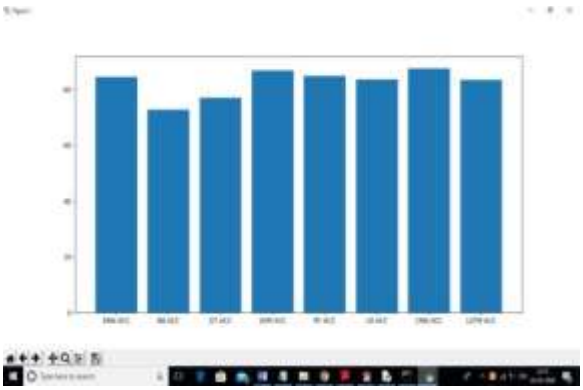

Figure.18

Now click on 'Predict Malware Family' button and upload binary file to get or predict class of malware

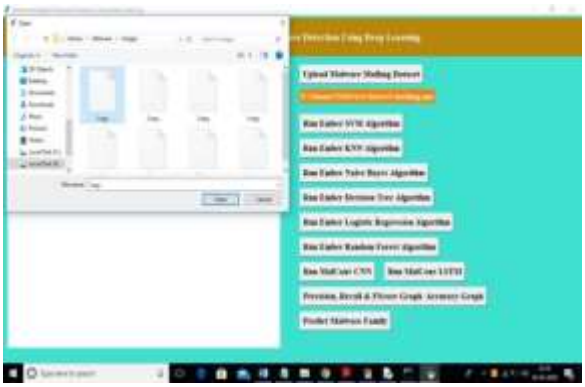

Figure.19

In above graph I am uploading one binary file called 1.npy and below is the malware prediction of that file

Figure.20

In above screen we can see uploaded test file contains 'Dialer Adialer.C' malware attack. Similarly u can upload other files andpredict class

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

VII.CONCLUSION

The article presents ScaleMalNet, a scalable system designed to detect, categorise, and classify zero-day malwares. The study then conducted a comparison of conventional machine learning algorithms (MLAs) and deep learning architectures that use static analysis, dynamic analysis, and image processing approaches to identify malware. This platform employs a two-step process for analysing malware and uses deep learning to examine the malware samples obtained from end user hosts. Initially, malware was classified with a methodology that included both static and dynamic analysis. Next, the malwares were categorised based on their specific attributes using image processing methods. Through several experimental evaluations on both publicly available benchmark datasets and secretly acquired datasets, this study has shown that deep learning approaches outperformed ordinary machine learning algorithms. The suggested system may achieve scalability to assess a larger volume of malwares in real-time by including more layers into the existing structures. It already has the ability to effectively manage a vast quantity of malicious software.

VIII.FUTURE ENHANCEMENT

Future research will examine these variations by including fresh factors into the existingdataset.

The main finding of this research, coupled with its shortcomings and need for improvement, might be summed up as follows:

A two-stage process is used to construct a scalable malware detection system. The suggested solution makes use of state-of-the-art deep learning methods to detect malware at an early stage. The malware is then categorised into its appropriate groups at the second level. There is a bullet point sign in the user's content. Deep learning architectures outperformed conventional machine learning techniques in the domains of image processing-based viral classification and detection as well as static and dynamic virus detection. In the process of investigating malware detection using dynamic analysis, deep learning architectures are used to analyse features gathered from domain expertise. This may be avoided by gathering binary file memory dumps during runtime and turning the memory dump file into a grayscaleimage.

In the study, malware was transformed into a certain size picture and then flattened in order to identify malware using deep learning for image processing. In further studies, the use of the spatial pyramid pooling (SPP) layer could make it possible to employ images with different dimensions as input. To improve the flexibility of our models, this approach may be placed between the completely connected layer and the sub sampling layer. It gathers features at varioussizes.

In the Malimg dataset, there is a significant skew in the distribution of malware families. An economical approach might be used to solve the issue of imbalanced multiclass malware families. This makes it possible for deep learning systems to include cost components in the backpropagation learning method. The cost component essentially represents the importance of categorization by giving classes with more samples a higher value and classes with fewer instances a lower value. Furthermore, in an adversarial context, deep learning systems are vulnerable to assaults [50]. During testing or deployment, samples that are easily able to trick deep learning systems may be generated using the generative adversarial network (GAN) technique. The

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

proposed research does not address the robustness of deep learning architectures. Future study in this field is crucial since virus detection is a critical application in settings where security is paramount. A single misclassification might have a number of negative repercussions for the company.

IX.REFERENCES

[1] Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M. J., Levi, M., ... & Savage, S. (2013). Measuring the cost of cybercrime. In The economics of information security and privacy (pp. 265-300). Springer, Berlin, Heidelberg.

[2] Li, B., Roundy, K., Gates, C., & Vorobeychik, Y. (2017, March). LargeScale Identification of Malicious Singleton Files. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (pp. 227-238). ACM.

[3] Alazab, M., Venkataraman, S., & Watters, P. (2010, July). Towards understanding malware behaviour by the extraction of API calls. In 2010 Second Cybercrime and Trustworthy Computing Workshop (pp. 52-59). IEEE.

[4] Tang, M., Alazab, M., & Luo, Y. (2017). Big data for cybersecurity: vulnerability disclosure trends and dependencies. IEEE Transactions on Big Data.

[5] Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2011, December). Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference-Volume 121 (pp. 171-182). Australian Computer Society, Inc..

[6] Alazab, M., Venkatraman, S., Watters, P., Alazab, M., & Alazab, A. (2011, January). Cybercrime: the case of obfuscated malware. In 7th ICGS3/4th e-Democracy Joint Conferences 2011: Proceedings of the International Conference in Global Security, Safety and Sustainability/International Conference on e-Democracy (pp. 1-8). [Springer].

[7] Alazab, M. (2015). Profiling and classifying the behavior of malicious codes. Journal of Systems and Software, 100, 91-102.

[8] Huda, S., Abawajy, J., Alazab, M., Abdollalihian, M., Islam, R., & Yearwood, J. (2016). Hybrids of support vector machine wrapper and filter based framework for malware detection. Future Generation Computer Systems, 55, 376-390.

[9] Raff, E., Sylvester, J., & Nicholas, C. (2017, November). Learning the PE Header, Malware Detection with Minimal Domain Knowledge. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (pp. 121-132). ACM.

[10] Rossow, C., Dietrich, C. J., Grier, C., Kreibich, C., Paxson, V., Pohlmann, N., ... & Van Steen, M. (2012, May). Prudent practices for designing malware experiments: Status quo and outlook. In Security and Privacy (SP), 2012 IEEE Symposium on (pp. 65-79). IEEE.

[11] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2017). Malware detection by eating a whole exe. arXiv preprint arXiv:1710.09435.

[12] Krcál, M., ˇ Svec, O., Bálek, M., & Ja ˇ sek, O. (2018). Deep Convolutional ˇ Malware Classifiers Can Learn from Raw Executables and Labels Only.

[13] Rhode, M., Burnap, P., & Jones, K.

ISSN2454-9940
www.ijasem.org
Vol 18, Issue 2, 2024

(2018). Early-stage malware prediction using recurrent neural networks. Computers & Security, 77, 578-594.

[14] Anderson, H. S., Kharkar, A., Filar, B., & Roth, P. (2017). Evading machine learning malware detection. Black Hat.

[15] Verma, R. (2018, March). Security Analytics: Adapting Data Science for Security Challenges. In Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics (pp. 40-41). ACM.

[16] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[17] A. F. Agarap and F. J. H. Pepito. (2017). ‘‘Towards building an intelligent anti-malware system: A deep learning approach using support vector machine (SVM) for malware classification.’’ [Online].
Available: https://arxiv.org/abs/1801.00318

[18] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. de Geus, ‘‘Malicious software classification using VGG16 deep neural network’s bottleneck features,’’ in Information Technology-New Generations. Cham, Switzerland: Springer, 2018, pp. 51–59.

[19] J. Saxe and K. Berlin, ‘‘Deep neural network based malware detection using two dimensional binary program features,’’ in Proc. 10th Int. Conf. Malicious Unwanted Softw. (Malware), Oct. 2015, pp. 11–20.

[20] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, ‘‘Malware detection with deep neural network using process behavior,’’ in Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC), vol. 2, Jun. 2016, pp. 577–582.