International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 07 | July 2022 www.irjet.net p-ISSN: 2395-0072

MALWARE DETECTION AND SUPPRESSION USING BLOCKCHAIN TECHNOLOGY

Sweeten Bandodkar[1], Ganesh Manerkar[2]

[1]Student, Dept. of Information Technology and Engineering, Goa College of Engineering, Farmagudi, Goa, India
[2] Assistant Professor, Dept. of Information Technology and Engineering, Goa College of Engineering, Farmagudi, Goa, India

---

Abstract – *Malicious Software, also, known as Malware, can pose serious cybersecurity threats. This can have significant consequences not only for individuals and organisations but, also, for the nation at large. The meteoric surge in malware types and its magnitude dampens the efficacy of traditional techniques, that use feature extraction and comparison, for the detection of cyber threats, thus rendering the identification of malware extremely difficult. Malware detection challenges are intensified by the sizeable increase in the intricacies, in the type and structure, in recent years, including instructions, settings, source code, binary files, and others. The increased intricacies encourage misjudgement. This study adopts Convolutional Neural Networks (CNN) and Blockchain to enhance the security of cyber networks by increasing the efficiency and accuracy of malware detection under large and numerous types of malware. The experiment offers that CNN is extremely effective with source code and binary code detection and, it can recognize malware that is embedded into benign code, making malware impossible to hide. This research suggests a viable solution for network administrators to identify malware at the very start in the complex network environment, so as to enable information technology personnel to engage in timely protective actions and prepare for potential additional cyberattacks.*

Key Words: Malware, Blockchain, CNN, Binary code, Benign, Malign

1.INTRODUCTION

In the world where 'information' is one of the most prized assets, a significant threat is posed to it by the ever-evolving and sophisticated malwares (Sewak, Sahay, & Rathore, 2018). The malware is an industry with ulterior profit motives, although it was initially created for fun (Statista, 2022). The identified malware, as per AV-Test, greatly increased from 182.90 million in the year 2013 to 1332.56 million in 2022 (AV-TEST). The malware file type and structure also grew in complexity with the increase in volume as well as variety of malware. The malicious source files contain source code files, binary files, shell script files, perl script files, make files and read me files, along with several hierarchical directories. There are many types of malwares such as Adware, Rootkit, Virus, Ransomware, Backdoor, Worm and Trojan (Gandotra, Bansal, & Sofat, 2014). Hybrid malwares are embedded in cyber-attacks to compromise the security of the information system. A wide variety of techniques are proposed for detection of malwares (Sewak, Sahay, & Rathore, 2018).

A wide variety of techniques were employed by the researchers and industries for detection of malwares. This includes heuristic-based methods and signature-based methods (Nawroozi & SaravanaGuru, 2020). The signature-based method, considered the simplest method of detection, considers and compares the traffic network or malware for possible attacks with known signature. Antivirus softwares have been used for many years for identification of a particular type of virus.

The authors of malware try to dodge the antivirus with the use of metamorphic and polymorphic malware to bypass virus signatures (Nawroozi & SaravanaGuru, 2020). However, this method is not free of drawbacks. Its inefficiency in detection of unknown malwares and security threats compel the antivirus vendors to rely on heuristic methods, which is based on rules determined by the experts. It relies on static and dynamic methods of analysis. The heuristic method for examining the malware before writing the signature is required to analyse whether it is based on the behaviour or by testing the code in a safe environment. The methods employed in the analysis of the malware is based on two methods, namely, Static analysis and Dynamic analysis. Static analysis refers to analysing the malware without its execution, whereas Dynamic analysis analyses the behaviour of a malware program while running or operating on a controlled environment like a virtual machine or a sandbox. During Static analysis, the pattern will be detected, like opcodes or byte-sequence, string signature, byte-sequence n-gram or opcodes n-grams, etc. Under Dynamic analysis, the behaviour of malware will be observed by utilising tools such as Wireshark, Process Explorer, and Capture BAT, etc. to monitor the function, the network and the flow of data. Although this analysis is more effective, it is both time and resource consuming as compared to the static analysis.

Although these methods are able to detect unknown malwares, it produces more amount of false positive than signature-based method (Nawroozi & SaravanaGuru, 2020). Various vendors of antivirus, therefore, use a hybrid method of analysis which includes both the methods (i.e. heuristic

and signature-based) for improved recognition of unknown malware. Machine learning and Deep learning are the concepts lately being used and implemented for the detection of malware. Therefore, due to the wider applicability of these techniques in this field, much of the success can be attributed to Deep learning methods such as Convolutional Neural Networks (CNN) (Apruzzese, Colajanni, Ferretti, Guido , & Marchetti, 2018).

CNNs consist of layers that are referred to as convolutional layers, pooling and non-linear layers. The first layer which is an input layer moves the input samples to CNN's first block, thus passing the data through the network to the last layer is the key. For the systems to use Artificial Intelligent systems correctly and securely, each block must be flawlessly authenticated. The CNN model will miss the accountability for each block and, hence, blockchain can be beneficial (Gu, et al., 2018). Blockchain technology offers increased security which can be leveraged in securing the local networks (Raje, Vaderia, Wilson, & Panigrahi, 2017).

Blockchain can support the safe implementation of AI systems in the public domain with its unique functions like data privacy, transparency, safety, and authentication. The security element ensures that the network architecture is not manipulated or tampered with. Also, it works on an Artificial Intelligent system. The decision property which is made by a specific block of an AI model would require validation of other blocks connected to the block in concern. This is known as authentication (Ye, Li, Adjeroh, & Iyengar, 2017). Blockchain helps to achieve higher detection accuracy is received in a limited time (Investopedia, 2022).

It is possible to evade the attack in the application method for discovery and classification with the proper deployment of blockchain technology. It is no more possible to manipulate any layer in CNN due to the alarm being raised. It will be supported by feature extraction and matching using cryptography and transitive hash. Any modification in any layer will raise alarms. If any operation executed at a specific block is malicious, the system can be restored at the previous checkpoint. Moreover, the decentralization attribute promises that not all controls are situated with a single entity.

The afore-mentioned characteristics are a pre-requisite to offer a secure model of the Deep Neural Network (DNN) and making blockchain a suitable candidate for the job of detection of malware. In the study, an effort is made to suggest an architecture that derive the benefits of CNN with blockchain technology characteristics. The architecture can detect malware attack that is carried out at the level of the parameter or at the level of feature extraction. The addition of blockchain in CNN can effectively eliminate network level attack on CNN. The tampering vulnerability does not exist if CNN models are used with blockchain.

1.1 Literature Review

Classification of malware can be referred to the classification of files into malware and benign files (Hassen, Carvalho, & Chan, 2017). Based on the execution of malware, the malware detection method is separated into static and dynamic analysis (Cosma & Joy, 2012). Moreover, a Hybrid approach, that use the varied features from both the static and dynamic analysis are combined, also, exists (Ahmadi, Ulyanov, Semenov, Trofimov, & Giacinto, 2016). The static analysis method directly evaluates the extracted feature of binary code or source code (Tahan, Rokach, & Shahar, 2012).

Techniques based on Deep Learning are gaining popularity for the feature extraction. Numerous studies have focused and successfully implemented Deep Learning Techniques viz. Deep Neural Networks (Saxe & Berlin, 2015), Deep Belief Networks (David & Netanyahu, 2015), Recurrent Neural Networks (Pascanu, Stokes, Sanossian, Marinescu, & Thomas, 2015) and combination of Recurrent and Convolution Neural Networks for supervised learning and classification (Kolosnjaji, Zarras, Webster, & Eckert, Deep learning for classification of malware system call sequences, 2016). Likewise, Deep Belief Networks (DBN), Stacked Auto Encoders or Auto Encoders and RNN based Auto-Encoders (RNN-AE) have been used for optimal feature extraction (Wang & Yiu, 2016).

The application of neural network has been in use since 1960 in various regions of Machine Learning, namely computer vision, handwriting recognition and classification of images. In the year 2012, AlexNet was proposed by Krizhevsky, which championed the ImageNet competition (Singh, 2017). It is credited with the fame gained by CNN as the focus of Machine Learning in academic field and attracted the interest of researchers (Nataraj, Karthikeyan, Jacob, & Manjunath, 2011). It was a breakthrough in image and voice process field. The CNN has much better accuracy than humans, especially in the field of image classification (Schmidhuber, 2015).

CNN has a distinct calculus process in convolutional layer, pool layer and full connected layer (Krizhevsky, Sutskever, & Hinton, 2012) (White, Tufano, Vendome, & Poshyvanyk, 2016) (Huang, Al-Dujaili, Hemberg, & O'Reilly, 2018). Using the various stack method, convolutional neural network can be used for the construction of different structures that can be applied in different fields of research (Chen, Wang, Wen, Lai , & Sun, 2019).

(Su, Vasconcellos, Prasad, & Sgandurra, 2018) suggested a light-weight novel method for detection DDos malware in IoT environment. Frist, they extracted one-channel where gray-scale image is converted from binaries then used a light-weight convolution neural network to classify IoT malware families. According to their experiment results they gained it is 94.0% accuracy for the classification of the DDos malware. It showed significant result.

In the study (Prasse, Machlica, Pevný, Havelka, & Scheffer, 2017) created a method that allows them to gather network flows of benign application and known malware as training

data and then they applied a method for detection of malware which it is based on a neural language method and long short-term memory network (LSTM). The approached method can detect new malware.

The authors (De Paola, Gaglio, Re, & Morana, 2018) proposed a novel method based on the deep networks. First, they run malware in the sandbox environment and after this they will convert the log file of sandbox to a long binary bit-string file. It is fed to a deep neural network with eight layered that produced 30 values in the output layer. These values as signature produced by the DBN which is very successful for detecting of malware. And the advantage of this algorithm is that it can be used for the supervised malware classification. In this paper, the proposed method is the cloud-based malware detection system. This method is able to analysis big data which produced on the network. This system provides a fast classification that is based on the static analysis that used deep networks. And when the detection reaches uncertainty exceed a given threshold then it will use dynamic analysis and the result of dynamic analysis is exploited to refine the deep network in the continuous learning loop. And the advantage of this system is that it will be up to date that will detect new malware versions.

In the study by (Kolosnjaji, Zarras, Webster, & Eckert , 2016), a method was featured and was implemented based on the CNN and a current network layer, which came out to be one of the best for malware image detection. A full convolution of n-grams was combined as a sequential model in extraction of the model. The average accuracy was from 85.6% to 89.4%

(Abdelsalam, Krishnan, Huang, & Sandhu, 2018) in their work, they introduced a malware detection approach for virtual machines based on two dimensional convolution neural networks by utilized performance metrics. The result from the testing dataset showed that they gain accuracy of 79% and they also used 3D CNN model to improve the detection performance which used samples over a time-windows, and after applying 3D dimension to the 2D input matrix the achieved result was above 90% that is a significant output result

1.2 Methodology

A. System Description

The process of detection and classification of malware based on SBC algorithm is described in Fig 1. It starts with the input dataset which includes malign and benign datasets. The dataset will preprocess and label as malware or benign. The processed data will be forward as input for the convolution neural network then the algorithm would classify to the mentioned classes. As shown in the above diagram, the vulnerability exists in CNN method during filtering the features and shrinking the input that the attacker can easily modify it and it affects the result of detection. Therefore, the blockchain technology is applied to avoid it. Each layer of CNN stored in each block and each block would have its own hash value and the pervious hash value of block. Using CNN method each gray scale image of malware will analyze one by one based on the train and test model.

B. Dataset

The Malimg dataset [6] used in this work consists of image malware along with the benign image. The binary malware is changed to the grayscale image and there is python script which converts binary malware into the grayscale image using Numpy library. In this work, the gray scale image has used directly as input to CNN model to train the data and to classify it.

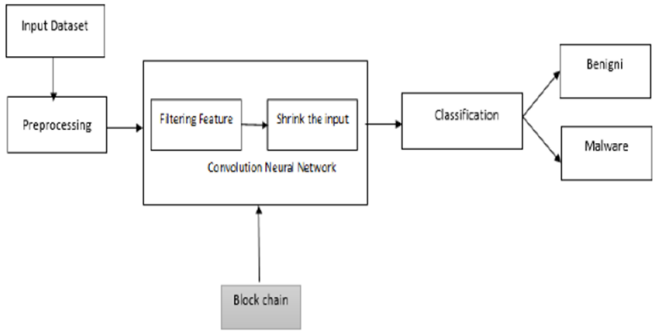

Fig-1: Process of detection and classification of malware

C. Secure Blockchain Convolution Algorithm

As it is shown in the Fig 2, it is the architecture of designed algorithm which the input phase is explained already in Fig 1. And after preprocessing the dataset into train and test data. . The next phase is make up of two fold process that it is Blockchian and Convolution Neural Network. Firstly, the blockchain is created and each layer of the CNN is stored as data in blockchain. Each layer of CNN would have same function with different values and a hash value, then the next step is the validation of the hash value which will check if it is valid. If it is invalid there are some changes in the system that would not apply CNN algorithm and it shows the massage which is invalid. On the other hand if it is valid the CNN method for analyzing and classifying the input is applied. The blockchain technology is explained in the upcoming content with CNN algorithm in detail.

D. Blockchain

Is a distributed model which stores the data between blocks and according to this project each function of CNN stored as data in blocks. The stored data is immutable and permanent that can be easily verified. The blockchain is mostly used in crypto-currencies and basically built from blocks; therefore it is not related directly to CNN in prior. Recently there are many researches in this part of different fields such as health care, smart energy and grids that shows blockchain having good potential and significant output.

The blockchain structure determined in Fig 3 is described as follows [11]:

 Data: The stored data in blockchain depends on the application. In this model, the stored data in each block is the Layer of convolution neural network components such as convolution and max-pooling function.

 Hash: A hash function takes any length of input and generates the output with fixed length and unique. The output would be different if a single value of input is modified and no matter the change is big or small, for instance if someone modified a single character of the convolution neural network layer which is hashed in the block then the modified block would have different hash values completely. This increased and improved the trust of data saved in
blockchain.

Pervious Hash: this is the hash value of the previous block which is stored in the current block to make sure the validation of block is correct. The same process would continue up to the end of blockchain and each of the pervious hash value will combine with the current hash value of block.
Timestamp: It is used to record the time of creating of the block. This is a method to track the modification or creation time of the block in a secure way.

E. Convolution Neural Network
The convolutional layer: This layer is the main part of a CNN. It is extracting the features from the input image and sending it to the next level, then the extracted features values will multiply with the original pixel values in the filter part. Here, we used fifth convolution filters of size 32,64,128,64,32 and the vector size is 5 for each convolution layer.
Rectified linear unit (ReLU) layer: In this layer, all the negative values will be removed from the filtered image and will replace it with zero. It will be done to avoid the values form summing up to zero. Transformed function is activated only when a node if the input is above a certain quantity, while the input is below the zero then the value will be zero.
Pooling layer: Pooling is a non-linear method of down-sampling. For implementing the pooling layer there are many non-linear functions like the minimum, maximum and the average but the maximum is the most common one. In the maximum pooling function the image will partition into a group of non-overlapping rectangles and the maximum value is the output of each sub-region and in this algorithm for reducing the dimensionality of the data, the factor value is 5.
Fully Connected layer: The responsibility of this layer is to classify the image into a label by using the output from the pooling process layer. The filter of shrunk image would put into a single list. To identify the most accurate weights, the fully connected layer goes through its own back propagation process. To prioritize the most appropriate label, it is according to each neuron weights received. The classification will be done accordingly towards the end of the process. The fully connected layer is a classic multi-layer resultant in the output layer with a softmax activation function and after this we used dropout method to prevent overfitting.

2. Design
The process flow diagram and the architecture for the study is given below:

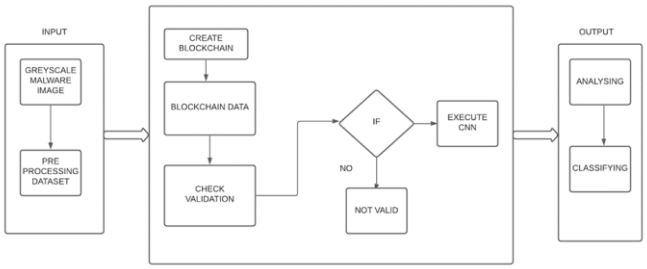

Fig-3: SBC Architecture

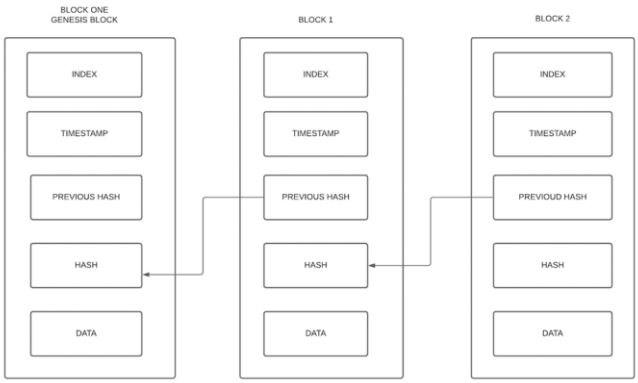

Fig-4: Blockchain Architecture

1.3 IMPLEMENTATION AND RESULT

In this section, the experiment result of the robustness and validity of the purposed algorithm is discussed. Here the experiment of this project is deployed in python3.7. We had performed 30 iterations. In every layer, different filter values are used and the rectified linear unit (ReLU) is used as activation function. This algorithm takes less time for the calculation process of training and the accuracy is higher.

```
Training Step: 25  | total loss: 0.01943 | time: 1.289s
| Adam | epoch: 025 | loss: 0.01943 - acc: 0.9974 | val_loss: 0.00036 - val_acc: 1.0000 -- iter: 46/46
--
Training Step: 26  | total loss: 0.01429 | time: 1.401s
| Adam | epoch: 026 | loss: 0.01429 - acc: 0.9981 | val_loss: 0.00071 - val_acc: 1.0000 -- iter: 46/46
--
Training Step: 27  | total loss: 0.01062 | time: 1.332s
| Adam | epoch: 027 | loss: 0.01062 - acc: 0.9986 | val_loss: 0.00158 - val_acc: 1.0000 -- iter: 46/46
--
Training Step: 28  | total loss: 0.00796 | time: 1.471s
| Adam | epoch: 028 | loss: 0.00796 - acc: 0.9989 | val_loss: 0.00372 - val_acc: 1.0000 -- iter: 46/46
--
Training Step: 29  | total loss: 0.00603 | time: 1.297s
| Adam | epoch: 029 | loss: 0.00603 - acc: 0.9992 | val_loss: 0.00881 - val_acc: 1.0000 -- iter: 46/46
--
Training Step: 30  | total loss: 0.00460 | time: 1.510s
| Adam | epoch: 030 | loss: 0.00460 - acc: 0.9994 | val_loss: 0.01938 - val_acc: 0.9836 -- iter: 46/46
```

Table 1 Accuracy and Loss of SBC Algorithm

Accuracy	0.99
Loss	0.00460

The accuracy is increased in each approach and the final accuracy is 0.99 as shown in the above table and the loss is 0.00460 in the last iteration.

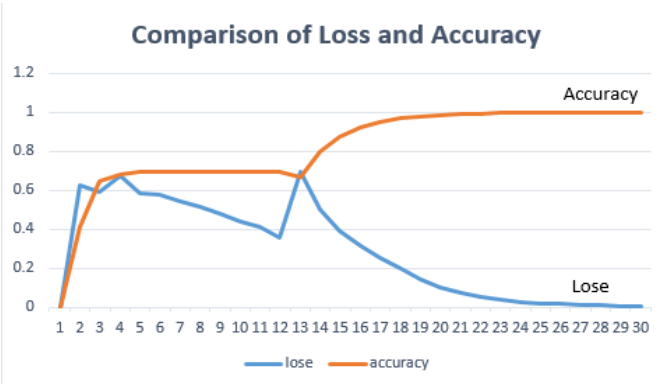

Fig 8 Loss and Accuracy Result Graph

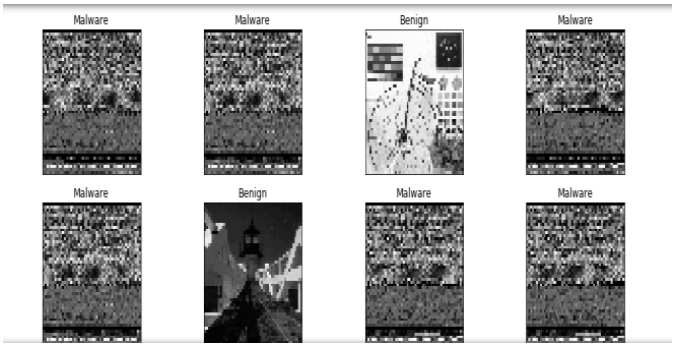

Fig 9. Malware and Benign image Result

The following output was obtained up

Fig-8: Blockchain Output

3. CONCLUSIONS

The dataset plays a very crucial role for the classification, object detection and segmentation for the classification of malware. Preprocessing images is a very crucial task in image processing since accurate malware identification is only attainable by detecting background noise.

The proposed project makes use of CNN and blockchain technology.

REFERENCES

[1] Abdelsalam, M., Krishnan, R., Huang, Y., & Sandhu, R. (2018). Malware Detection in Cloud Infrastructures Using Convolutional Neural Networks. 018 IEEE 11th International Conference on Cloud Computing (CLOUD), (pp. 162-169).

[2] Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M., & Giacinto, G. (2016). Novel feature extraction, selection and fusion for effective malware family classification. Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, CODASPY '16, 183–194.

[3] Apruzzese, G., Colajanni, M., Ferretti, L., Guido , A., & Marchetti, M. (2018). On the effectiveness of machine and deep learning for cyber security. 2018 10th International Conference on Cyber Conflict (CyCon), (pp. 371-390). Tallinn.

[4] AV-TEST. (n.d.). Malware. Retrieved March 14, 2022, from https://www.av-test.org/en/statistics/malware/

[5] Chen, C.-M., Wang, S.-H., Wen, D.-W., Lai , G.-H., & Sun, M.-K. (2019). Applying Convolutional Neural Network for Malware Detection. 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), 1-5.

[6] Cosma, G., & Joy, M. (2012). An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis. IEEE Transactions on Computers, 61(3), 379-394.

[7] David, O. E., & Netanyahu, N. S. (2015). DeepSign: Deep learning for automatic malware signature generation and classification. 2015 International Joint Conference on Neural Networks (IJCNN), 1-8.

[8] De Paola, A., Gaglio, S., Re, G. L., & Morana, M. (2018). A hybrid system for malware detection on big data. IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM, (pp. 45-50). Honolulu, HI.

[9] Gandotra, E., Bansal, D., & Sofat, S. (2014, April). Malware Analysis and Classification: A Survey. Journal of Information Security, 5(2), 56-64.

[10] Gu, J., Sun, B., Du, X., Wang, J., Zhuang, Y., & Wa, Z. (2018). Consortium Blockchain-Based Malware Detection in Mobile Devices. IEEE Access, 6, 12118-12128.

[11] Hassen, M., Carvalho, M. M., & Chan, P. K. (2017). Malware classification using static analysis based features. 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 1-7.

[12] Huang, A., Al-Dujaili, A., Hemberg, E., & O'Reilly, U.-M. (2018). Adversarial Deep Learning for Robust Detection of Binary Encoded Malware. 2018 IEEE Security and Privacy Workshops (SPW), (pp. 76-82).

[13] Investopedia. (2022, March 5). Blockchain Explained. Retrieved from https://www.investopedia.com/terms/b/blockchain.asp

[14] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert , C. (2016). Deep Learning for Classification of Malware System Call Sequences. Australasian Joint Conference on Artificial Intelligence, (pp. 137-149).

[15] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). Deep learning for classification of malware system call sequences. AI 2016: Advances in Artificial Intelligence, 137-149.

[16] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Proceedings of Advances in Neural Information Processing Systems (pp. 1106-1114). Cambridge: MIT Press.

[17] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware images: visualization and automatic classification. International Symposium on Visualization for Cyber Security (VizSec), 1-7.

[18] Nawroozi, S., & SaravanaGuru, R. K. (2020, March). Design of Secure Blockchain Convolution Neural Network Architecture for Detection Malware Attacks. International Journal of Recent Technology and Engineering (IJRTE), 8(6), 3055-3060.

[19] Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A. (2015). Malware classification with recurrent networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1916-1920.

[20] Prasse, P., Machlica, L., Pevný, T., Havelka, J., & Scheffer, T. (2017). Malware Detection by Analysing Network Traffic with Neural Networks. 2017 IEEE Security and Privacy Workshops (SPW), (pp. 205-210). San Jose, CA.

[21] Raje, S., Vaderia, S., Wilson, N., & Panigrahi, R. (2017, November). Decentralised firewall for malware detection. 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), (pp. 1-5).

[22] Saxe, J., & Berlin, K. (2015). Deep neural network based malware detection using two dimensional binary program features. 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), 11-20.

[23] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.

[24] Sewak, M., Sahay, S. K., & Rathore, H. (2018, June). Comparison of Deep Learning and the Classical Machine Learning Algorithm for the Malware Detection. 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 293-296.

[25] Singh, A. (2017). Malware Classification using Image Representation. Department of Computer Science and Engineering,Indian Institute of Technology Kanpur.

[26] Statista. (2022, February 3). Security software - Statistics & Facts. Retrieved March 13, 2022, from https://www.statista.com/topics/2208/security-software/

[27] Su, J., Vasconcellos, D. V., Prasad, S., & Sgandurra, D. (2018). Lightweight Classification of IoT Malware Based on Image Recognition. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), (pp. 664-669). Tokyo.

[28] Tahan, G., Rokach, L., & Shahar, Y. (2012). Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features. Journal of Machine Learning Research 13, 13, 949-979.

[29] Wang, X., & Yiu, S. M. (2016). A multi-task learning model for malware classification with useful file access pattern from API call sequence.

[30] White, M., Tufano, M., Vendome, C., & Poshyvanyk, D. (2016). Deep learning code fragments for code clone detection. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, (pp. 87-98). Singapore.

[31] Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017, June). A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3), 1-40.