Science World Journal Vol. 18(No 2) 2023 DOI: https://dx.doi.org/10.4314/swj.v18i2.6
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Full Length Research Article

MALWARE DETECTION AND CLASSIFICATION USING EMBEDDED CONVOLUTIONAL NEURAL NETWORK AND LONG SHORT-TERM MEMORY TECHNIQUE

*[1]Theophilus Aniemeka Enem, [2]Olalekan J. Awujoola,

[1]Department of Cyber Security, Airforce Institute of Technology, Kaduna, Nigeria
[2]Department of Computer Science, Nigerian Defence Academy, Kaduna, Nigeria

Corresponding Author Email Address: [1] t.enem@afit.edu.ng, [2] ojawujoola@nda.edu.ng

ABSTRACT

The significant growth in the use of the Internet and the rapid development of network technologies are associated with an increased risk of network attacks. As the use of encryption protocols increases, so does the challenge of identifying malware encrypted traffic also increases. Malware is a threat to people in the cyber world, as it steals personal information and harms computer systems. Network attacks refer to all types of unauthorized access to a network, including any attempts to damage and disrupt the network. This often leads to serious consequences. However, various researchers, developers and information security specialists around the globe continuously work on strategies for detecting malware. Recently, deep learning has been successfully applied to network security assessments and intrusion detection systems (IDSs) with various breakthroughs, such as using Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to classify malicious traffic. But, with the diverse nature of malware, it is difficult to extract features from it. Therefore, existing solutions require more computing resources since available resources are not efficient for datasets with large numbers of samples. Also, adopting existing feature extractors for extracting features of images consumes more resources. This paper therefore solved these problems by combining a 1D convolutional neural network (CNN) and long short-term memory (LSTM) to adequately detect and classify malicious encrypted traffic. This work was conducted on the malware Analysis benchmark Datasets with API Call Sequences, which contains 42,797 malwares and 1,079 goodware API call sequences. The experimental results show that our proposed system has achieved 99.2% accuracy and outperformed all other state-of-the-art models.

Keywords: Convolutional neural network, Long-Short-term memory, Malware Detection, Encrypted network traffic.

INTRODUCTION

In recent years, the computer network has become a crucial foundation for the development, and network security issues have received unprecedented attention (Anderson & McGrew, 2016). With the widespread use of encryption technology, the privacy, freedom, anonymity of Internet users have been greatly protected, but also it has allowed attackers to evade the anomaly detection system (Guo et al., 2021). For example, an attacker invades and attacks the system by encrypting malware traffic. Besides, criminals penetrate the darknet through tools such as Tor to trade illegally (Rezaei & Liu, 2019). That is to say, the abuse of encryption technology poses many challenges to network anomaly detection and secure management. Therefore, the identification of malware encrypted traffic has aroused great concern in academia and industry. In this generation of computing, the aspects of artificial intelligence (AI) is one of the most interesting and hot topics for developing intelligent systems, which has the capability to take real time decision without human intervention (Awujoola et al, 2021).

Malicious software (malware) is a type of computer program designed to cause harm to a computer, computing device, or computer user (Tahir, 2018). Malware attacks can disrupt a person's or organizations day-to-day use of their computer systems, steal personal or confidential information, corrupt files or annoy users. Malware can be categorized into different families where the behaviour of malware from one particular family differs from that of another family. Currently, the variants of Mirai malware such as Satori and Miori are still storming to the network with the new records of traffic volume towards the victim, including the systems of the enterprise companies.

In practice, the attacker often handles a large number of the botnets of injected internet of thing (IoT) devices to launch such an attack. As a result, detecting the malware traffic in the early period of distributing the malicious code can significantly help to prevent the malware from becoming widespread, and mitigate the attack magnitude (Hwang et al., 2019).

Furthermore, modern malware attacks are generally facilitated by the Internet. With the rise in the number of devices that are connected to the Internet, it has become more important than ever to keep our devices safe, lest we risk loss of personal or confidential information (Aslan et al, 2021). Malware has been rising at an alarming rate over the past decade and there is no reliable method for detecting all malware. There is a new generation of malware using advanced obfuscation and packing techniques to escape from detection systems. This makes it nearly impossible to detect complex malware with a traditional approach (Aslan et al, 2021).

Consequently, deep learning promises to be a game changer to ignite

new detection approaches. The most benefits of the deep learning approaches are to build a thorough pattern that can highly represent the characteristics of a specific object through auto-learning a large volume of data and species (Ren-Hung et al, 2019). Although applying deep learning (DL) in network security has just emerged recently, the topic has received a lot of attention from the research community due to the robust auto-learning ability of DL (Javaid et al., 2015). In addition, the evolution and the

DOI: https://dx.doi.org/10.4314/swj.v18i2.6

increased availability of graphic processing unit (GPU)-processors significantly help to accelerate matrix computations and massive mathematical calculations, thus directly supporting the feasibility of the DL-based approaches. Deep learning for malicious traffic detection is generally categorized into many classes that are primarily based on the built-in network model, e.g., Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) or unsupervised learning such as an auto-encoder. However, existing literature still suffers several critical drawbacks. First, such systems are built on the flow-based approach (i.e., collecting packets of the same flow for a certain period of time and then classifying packets into flows) and evaluated offline. Furthermore, extracting the features for the flow-based detection system also requires a certain period of time. Apparently, this whole traffic profiling process naturally increases the overall time of the detection. Secondly, the flow-based classification can require a lot of resources, e.g., memory and storage, to store and process the accumulated traffic, including the deep checking on the assembled data.

This work therefore, propose a novel approach that focus on building an hybrid technique of embedding LSTM into CNN by using CNN to extract the malware features then use LSTMs to classify malware. This work is built on the work done by the researchers (Zhang, 2020) by combining various aspects of technologies employed in (Ren-Hung et al, 2019).

Malware detection is one of the key hot points in network security. In order to protect users from the threat of malware, many security companies such as Comodo, 360 securities, Symantec, Kaspersky have developed their own malware defense product (Sun et al,2021). In recent decades, researchers have explored a series of methods and techniques to detect malware after a lot of work. Detection of malware is mainly to detect the characteristic code of malware, which mainly includes anomaly-based detection and signature-based detection. Although traditional methods play a very important role in malicious code detection, they have also made some achievements (Li et al., 2021). However, because malicious code writers often use various means to avoid the traditional detection methods, or study some new types of malicious code or variants of malicious code, the accuracy of the traditional detection model will be greatly reduced in these cases. With the continuous development of machine learning technology, malware detection technology based on machine learning model has also developed and achieved some successful results.

The authors of (Mishra et al., 2019) consider a biLSTM based model to classify malware in a cloud-based system. The model includes a CNN layer and is trained on system call sequences. The authors achieve an overall accuracy of approximately 90%. Interestingly, the authors also show that substituting the biLSTM for a regular LSTM layer resulted in worse accuracies in almost all cases. Similarly, the work in (Lu, 2019) is based on opcodes obtained from disassembled executables. This research also employs word embedding as a feature engineering step. Word embedding techniques are often used in natural language processing (NLP) applications. The result from word embedding is fed into an LSTM layer. For malware detection, this model attains an average area under ROC curve (AUC) of 0.99, while for classification, the model achieves an average AUC of 0.987. Furthermore, LSTM and CNN techniques were used by author Yuan et al., (2018) to build a model to detect insider threats. They applied the model on the CERT insider threat v4.2 dataset, which contained 32M log lines among which 7323 were anomalous activities. The advantage of this version of the CERT dataset was that it contained more samples of insider threats than other versions. The train–test split was 70–30%. The researchers first used LSTM to extract the user behavior, abstracted temporal features, and produced the feature vectors. After that, the researchers transformed the feature vectors into fixed-size matrices. Finally, CNN was used to classify the feature matrices into anomaly or normal. The proposed model resulted in an area under the curve (AUC) of 94.49%

Agarap, (2017) used CNN hybrid networks with support vector machine (SVM) and other SVM hybrid architecture and deep learning models were proposed. Their CNN-SVM model stood at 77.22% accuracy, gated recurrent unit (GRU)-SVM stood at 84.92%, and the MLP-SVM hybrid model stood at 80.46% accuracy. Akarsh et al. (2019) proposed a CNN-LSTM hybrid model with the special transformation of the images. Their two-layer CNN which connected to an LSTM layer with 70 memory blocks and an FCN layer with 25 units with a softmax and categorical cross entropy. The final accuracy on different splits was from 96.64% to 96.68%. In another paper, Akarsh et al. (2019b) used 2 layers of 1D CNN along with an LSTM for feature extraction with 0.1% dropout and 70 memory blocks of LSTM and a cost-sensitive algorithm to their model. They reported the highest accuracy of 95.5%.

In another study, Cui et al., (2019) proposed a method for data equilibrium based on a non-domination based genetic algorithm for multi- objective optimization (NSGA)- genetic algorithm without equalization produced 92.1% accuracy, with a single objective algorithm produced 96.1% accuracy and with the multi-objective algorithm the highest accuracy of 97.1% was achieved. Jain et al. (2021) used extreme learning machines (ELMs) with CNNs and proposed an ensemble model. They produced an accuracy of 96.30% with a single CNN layer and 95.7% with two CNN layers.
The author Naeem et al., (2020) used an IOT based hybrid visualization technique with deep learning. By using different image ratios, they were able to develop models with accuracies up to 98.47% and 98.79% but were dependent on dynamic image features. Venkatraman et al. (2019) proposed a hybrid architecture with a self-learning system. The proposed hybrid CNN BiLSTM and CNN BiGRU models and trained them with both cost-sensitive and cost- insensitive methods. All of their models with different types of parameters and settings range in accuracy from 94.48% to 96.3%. Similarly,
Alzaylaee et al.(2020) proposed a malicious code detection system based on deep learning, DL-Droid, to dynamically detect malicious Android applications, using dynamic features to achieve a detection rate of 97.8%.

LSTM is a network model proposed by Schmidhuber et al. in 1997 (Lu et al., 2020). Similarly, LSTM is a network model designed to solve the longstanding problems of gradient explosion and gradient disappearance in RNN (Zarrad et al., 2019). It has been widely used in speech recognition, emotional analysis, and text analysis, as it has its own memory and can make relatively accurate forecasting (Gupta & Jalal, 2019). In recent years, it has also been adopted in the field of stock market forecasting (Yadav et al., 2020).There is only one repeating module in a standard RNN, and

DOI: https://dx.doi.org/10.4314/swj.v18i2.6

its internal structure is simple. It is usually a tanh layer. However, four of the LSTM modules are similar to the standard RNN modules, and they operate in a special interactive manner (Jin et al., 2020). The LSTM memory cell consists of three parts: the forget gate, the input gate, and the output gate.

CNN is a network model proposed by Lecun et al. in 1998 (Lecun et al.,1998). CNN is a kind of feedforward neural network, which has good performance in image processing and natural language processing (Kim & Kim 2019). It can be effectively applied to the forecasting of time series. The local perception and weight sharing of CNN can greatly reduce the number of parameters, thus improving the efficiency of model learning (Qin et al., 2018). After the convolution operation of the convolution layer, the features of the data are extracted, but the extracted feature dimensions are very high, so in order to solve this problem and reduce the cost of training the network, a pooling layer is added after the convolution layer to reduce the feature dimension.

MATERIALS AND METHODS

This section presents the detailed experiments and evaluation steps undertaken to test the effectiveness of the proposed model. The experiment was based on the classification of malware in an encrypted network. This work used Python open-source deep learning framework with tensor flow backend to build and test the CNN-LSTM model. All experiments were run on a standard PC with a Hp Proliant DL380p Gen8 server with 20Gb Ram.

Proposed Research methodology

The flow of methodology is represented in Figure 1. This defines the steps to be taking to implement how the embedded LSTM shallow Convolutional neural network model. However, Figure 2 shows the model architecture.

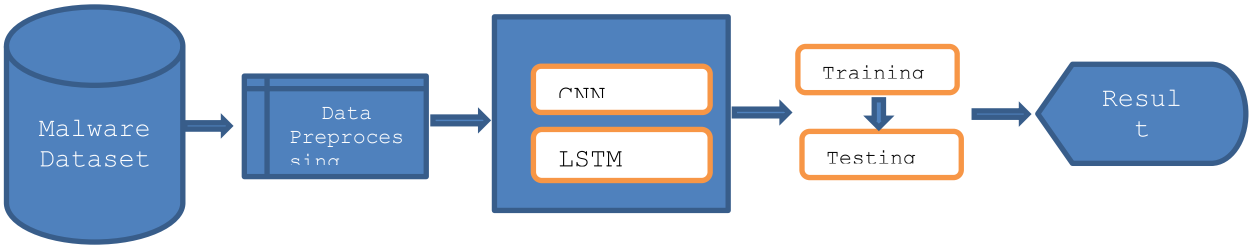

Figure 1: Methodology flow

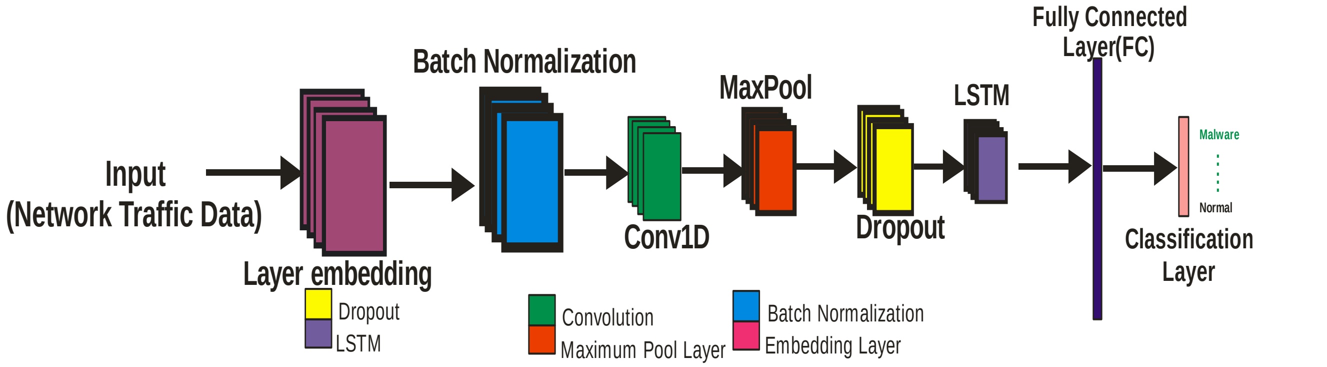

Figure 2: LSTM-CNN architecture

Data Collection

This study uses the publicly available malware. The United States Air Force released the collection to the public for research purposes. The public release of this program has provided a unique opportunity to assess progress in malware detection and classification domain. It contains 42,797 malware API call sequences and 1,079 goodware API call sequences. Each API call sequence is composed of the first 100 non-repeated consecutive API calls associated with the parent process, extracted from the 'call' elements of Cuckoo Sandbox reports. Figure 3 shows the malware dataset class distributions.

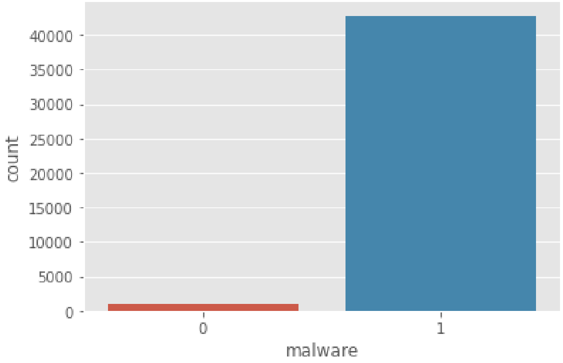

Figure 3: Malware dataset class distribution

Model Evaluation and Performance

The experimental comparison of classification algorithms was done based on the performance measures of classification accuracy, specificity, sensitivity, error rate, ROC and execution time. The model was evaluated based on the following metrics:

Confusion Matrix

The actual and predicted classification done by a classification matrix was generated and represented by a confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

After the confusion matrix was generated for each implemented algorithm the following metric values Accuracy, Sensitivity, Specificity and Error rate were calculated from the confusion matrix using the formulas listed below. The Table 1 shows the confusion matrix for a two-class classifier Strasak, (2017).

Table 1 Shows Confusion Matrix for two class classifiers.

ACTUAL		PREDICTED	
		Positive	Negative
	Positive	A (TP)	B (FP)
	Negative	C (FN)	D (TN)

where TP = True positive, TN = True Negative, FN = False Negative, FP = False Positive

The performance metrics used in evaluating the algorithms are as follows:

Accuracy. This measures the rate of correctly classified applications putting into consideration the true positive, true negative, false positive and false negative (Brownlee, 2020a). This is express in equation 1.1

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \qquad (1.1)$$

False Positive (FP) rate. This measures the rate of wrongly classified as benign. A low FP-rate signifies the classifier is a good one (Brownlee, 2020b). Equation 1.2 shows FP rate

$$\textbf{FPR} = \frac{FP}{FP + TN} \qquad \textbf{(1.2)}$$

True Positive (TP) rate. It is the proportion of positive instances (i.e., feature vectors of malicious applications) classified correctly. This is expressed in equation 1.3

$$\textbf{TP Rate} = \frac{TP}{TP + FN} \qquad \textbf{(1.3)}$$

Precision, Recall and F-measure. Precision is the ratio of positively predicted instances among the retrieved instances. Recall is the ratio of positively predicted instances among all the instances and F-measure is the harmonic mean of recall and precision A high F-measure is required since both precision and recall are desired to be high. Therefore, it is shown in equation 1.4

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (1.4)$$

$$\text{Recall} = \frac{TP}{TP + FP} \qquad (1.5)$$

$$\text{F-Measure} = 2\,x\,\frac{Precision}{Precision + Recall} \qquad (1.6)$$

Receiver Operating Characteristic (ROC) curve. The true positive rate is constructed against the false positive rate A high ROC value signifies that the algorithms is good (Brownlee, 2020b)

ROC Curve is Plot of FPR(x) vs TPR (1.7)

where TPR is True Positive Rate

RESULTS AND DISCUSSION

This work followed the methodology flow in figure 1 and implementation of the embedded CNN - LSTM Model Architecture in figure 2. The experiment was divided into two. The first experiment was performed with only CNN while the second experiment combined both the CNN and LSTM for the classification of the malware.

Table 2 shows the CNN classification model report of the results obtained from the first experiment.

Table2: CNN model classification report

	Precision	Recall	f1-score	support
Benign	0.00	0.00	0.00	283
Malware	0.97	1.00	0.97	10686
Accuracy			0.97	10969
macro avg	0.93	0.82	0.85	10969
weighted avg	0.95	0.97	0.96	10969

Table 2 shows that classification accuracy of the malware to be 97%. Figure 4 and Figure 5 depicts the training versus validation loss and training versus validation accuracy.

DOI: https://dx.doi.org/10.4314/swj.v18i2.6

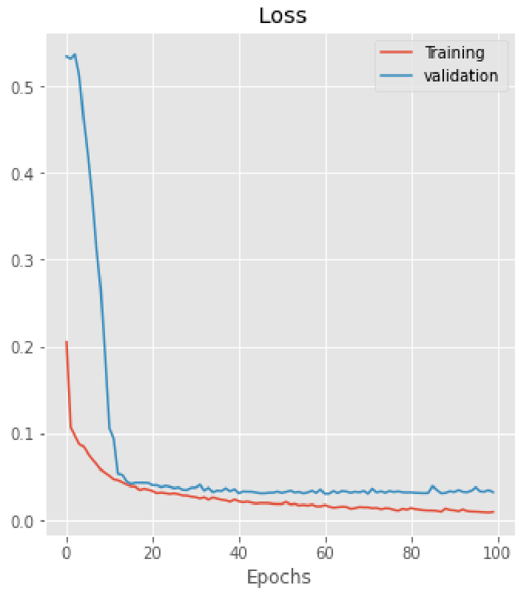

Figure 4: Training versus Validation loss

Figure 4 shows that there is gap between the training and loss at convergence and much wider gap at the training versus validation accuracy in figure 5. This shows that the model suffered overfitting.

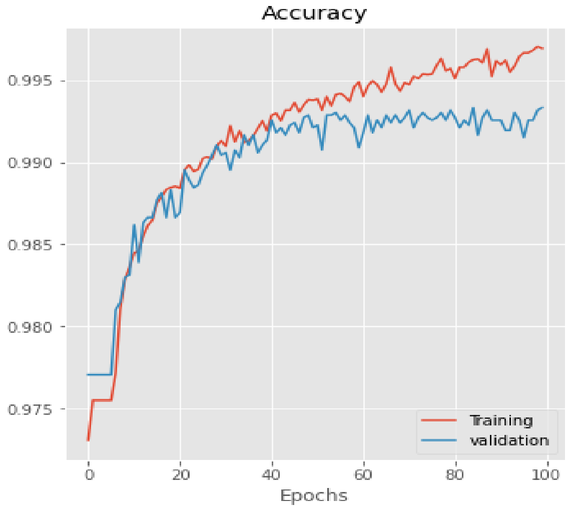

Figure 5: Training versus Validation accuracy

Following the flow of the methodology in figure 1 and implementation of the CNN - LSTM combined Model Architecture in figure 2, the classification model report of the results obtained is displayed in table 3.

Table 3: CNN_LSTM model classification report

	Precision	Recall	f1-score	support
Benign	0.89	0.65	0.75	283
Malware	0.99	1.00	0.99	10686
Accuracy			0.99	10969
macro avg	0.94	0.83	0.87	10969
weighted avg	0.99	0.99	0.99	10969

Table 3 shows that the "Benign" class, the model achieved a precision of 0.89, which means that out of all instances classified as "Benign," 89% were truly benign. The recall, or true positive rate, was 0.65, indicating that the model correctly identified 65% of all actual benign samples. The F1-score, which combines precision and recall, was 0.75. This score provides a balanced evaluation of the model's performance for the "Benign" class. The support indicates that there were 283 instances of the "Benign" class in the dataset. For the "Malware" class, the precision was 0.99, indicating that the model classified 99% of the instances as "Malware" correctly. The recall, or true positive rate, was 1.00, meaning that the model successfully identified all instances of malware. The F1-score for the "Malware" class was 0.99, highlighting the model's excellent performance in correctly classifying malware samples. The support shows that there were 10,686 instances of the "Malware" class in the dataset. Despite the class imbalance, with a significantly larger number of instances in the "Malware" class compared to the "Benign" class, the model achieved high precision, recall, and F1-scores for both classes. This suggests that the model performed well in distinguishing between benign and malware samples. Considering the macro-average F1-score, which calculates the average F1-score across all classes, the model achieved a score of 0.87. This indicates good overall performance in classifying both benign and malware instances. The weighted-average F1-score, which takes into account the class imbalance, was also reported as 0.99, further demonstrating the model's strong performance in this classification task. In summary, despite the class imbalance in the dataset, the model performed exceptionally well in classifying both benign and malware instances. The "Malware" class showed particularly strong results, with high precision, recall, and F1-scores. The model achieved high accuracy and demonstrated good overall performance in this malware classification task. Therefore, combination of one-dimensional Convolutional Neural Network with additional parameter of LSTM assisted in boosting the classification accuracy

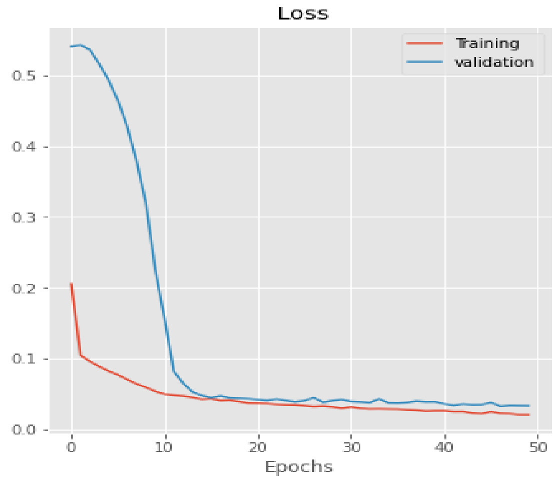

Figure 6: CNN-LSTM model training and validation loss

DOI: https://dx.doi.org/10.4314/swj.v18i2.6

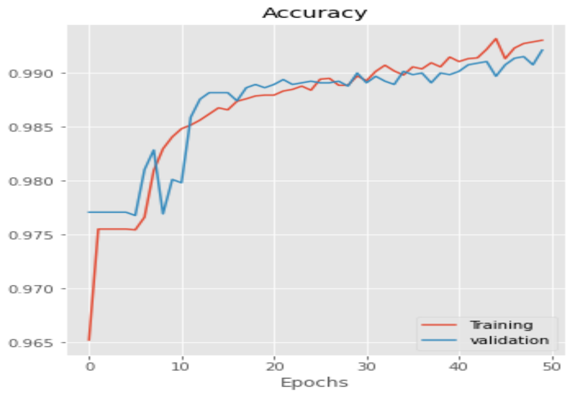

Figure 7 CNN-LSTM model training and validation accuracy

Comparing figure 6 and 7 to figure 3 and 4, one can clearly see that CNN-LSTM model training and validation accuracy and CNN-LSTM model training and validation loss converged quickly with no overfitting.

Figure 8 and 9 revealed the receiver operating character curve (ROC) and precision-recall (P-R) curve obtained from the CNN_LSTM model respectively.

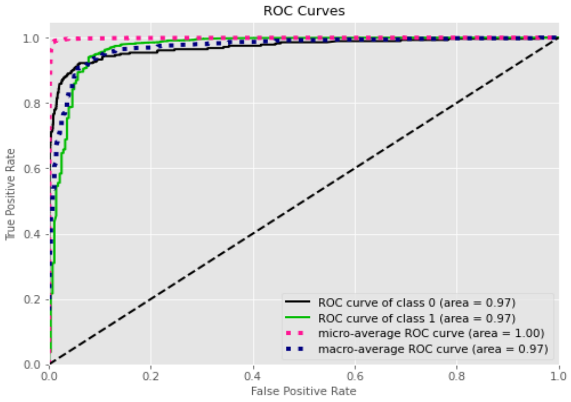

Figure 8 CNN-LSTM ROC curve

The performance of the model is further established as shown in figure 8. This shows that micro average ROC curve of malware is 1.00 while the micro average ROC curve for benign is 0.97. This curve area shows that the percentage accuracy obtained by the model is acceptable.

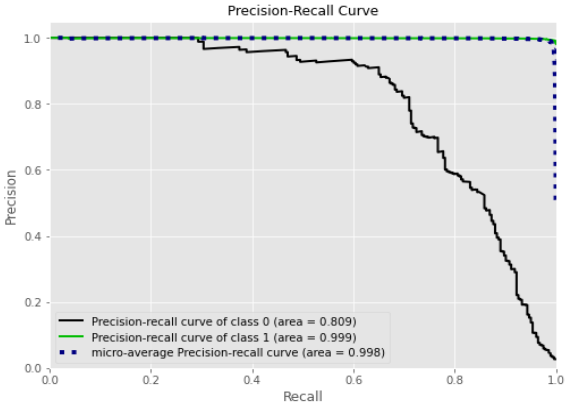

Figure 9: CNN-LSTM P-R curve

Figure 9 is the precision-recall curve of the model. The micro average precision-recall curve area of this model is 0.998, this shows that the model performed greatly. Figure 10 is the confusion matrix obtained from the model.

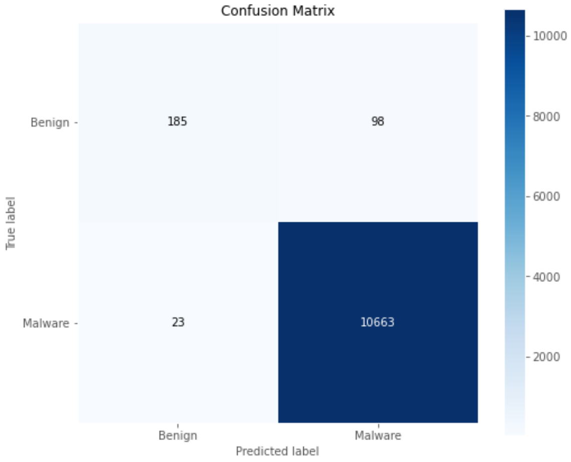

Figure 10: Model Confusion matrix

Figure 10 shows that the model is able to correctly classified 10,663 malwares while it misclassified only 23. Also, its able to classified correctly 185 benign while 98 was misclassified. Therefore, it is very clear that the dataset is highly imbalanced, and the reason for such large numbers of misclassified benign class. This is because benign class is a minority class.

Comparison with existing state-of-the-art algorithms

This work compared other methods with our model to test the performance of the model. Table 4 introduces the comparison results of our model performance and other existing malicious code detection methods based on deep learning or machine learning.

DOI: https://dx.doi.org/10.4314/swj.v18i2.6

Table 4: Performance comparison with the existing state-of-the-art algorithms

SNo	Author	Method	Accuracy
	Agarap,(2017)	CNN-SVM GRU-SVM MLP-SVM	77.22% 84.92% 80.46%
	Mishra et al., 2019	CNN & biLSTM	Approx. 90%
	Lu, 2019	LSTM	98.7%
	Prasse et al (2019)	CNN	73%
	Schultx et al (2019)	Data Mining Methods	97.96%
	Schultz et al, (2019),	Machine learning Naïve Bayes classifier.	97.96%
	Akarsh et al. (2019a)	CNN-LSTM	96.68%.
	Akarsh et al. (2019b)	1D CNN-LSTM	95.5%
	Cui et al.,(2019)	NSGA without equalization NSGA (single obj) NSGA (multi-objective)	92.1% 96.1% 97.1%
	Naeem et al.,(2020)	deep learning	98.79%
	Venkatraman et al. (2019)	CNN BiLSTM CNN BiGRU	94.48% 96.3%
	Jain et al. (2021)	ELMs with CNNs	96.30%
	Nicollo,(2020)	Deep learning	98%
	Lu et al, (2020).	CNN-LSTM	94%
	Fabio, (2021)	LSTM and CNN	81%
	Proposed	CNN LSTM and CNN	97% 99.2%

Conclusion

Malware has a long history, which seriously threatens the security of computer system. With the rapid development of anti-detection technology, the capability of traditional detection methods based on static analysis and dynamic analysis is limited. With neural network having strong prediction performance, the application of AI technology in malware detection has become a research hotspot. However, due to the difference of malware, feature extraction is difficult, which is not conducive to the application of traditional neural network. In other to solve this problem, this work therefore combines CNN and LSTM to form a hybrid and is able to achieve 99.2% accuracy.

REFERENCES

Anderson, B., and McGrew, D. (2016): "Identifying encrypted malware traffic with contextual flow data". In: Proceedings of the 2016 ACM workshop on artificial intelligence and security. pp. 35–46

Awujoola, O. J., Ogwueleka, F. N., Irhebhude, M. E., & Misra, S . (2021). Wrapper based Approach for network intrusion detection model with combination of dual filtering technique of resample and SMOTE. In *Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities* (pp. 139-167). Springer, Cham.

Aslan, Ö., Ozkan-Okay, M., & Gupta, D. (2021). A Review of Cloud-Based Malware Detection System: Opportunities, Advances and Challenges. *European Journal of Engineering and Technology Research*, *6*(3), 1-8.

Agarap, A. F. (2017). Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification.

Akarsh, S.; Simran, K.; Poornachandran, P.; Menon, V.K.; Soman, K. (2019): Deep learning framework and for malware classification. In Proceedings of the 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 1059–1063.

Akarsh, S.; Poornachandran, P.; Menon, V.K.; Soman, K. (2019b): A Detailed investigation and analysis of deep learning architectures and visualization techniques for malware family identification. In Cybersecurity and Secure Information Systems; Springer: New York, NY, USA, 2019; pp. 241–286.

Alzaylaee MK, Yerima SY, Sezer S (2020) DL-Droid: deep learning based android malware detection using real devices. Comput Secur 89:101663.

Cui, Z.; Du, L.; Wang, P.; Cai, X.; Zhang, W. (2019): Malicious code detection based on CNNs and multi-objective algorithm. J. Parallel Distrib. Comput. 2019, 129, 50–58.

Guo, J., Sang, Y., Chang, P., Xu, X., & Zhang, Y. (2021, June). MGEL: A Robust Malware Encrypted Traffic Detection Method Based on Ensemble Learning with Multi-grained Features. In *International Conference on Computational Science* (pp. 195-208). Springer, Cham.

Gupta, N & Jalal, A. (2019 "Integration of textual cues for fine grained image captioning using deep CNN and LSTM, "*Neural Computing and Applications, vol. 12, pp. 1–10, 2019*

Hwang, R. H., Peng, M. C., Nguyen, V. L., & Chang, Y. L. (2019). An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level. *Applied Sciences*, *9*(16), 3414. https://doi.org/10.3390/app9163414

Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies, New York, NY, USA, 3–5 December 2016; pp. 21–26.

Jain, M.; Andreopoulos, W.; Stamp, M.(2021): CNN vs ELM for image-based malware classification. arXiv 2021, arXiv:2103.13820

Jin, Z. Y. Yang, &Liu, Y. (2020) "Stock closing price prediction based on sentiment analysis and LSTM," Neural Computing and Applications, vol. 32, no. 13, pp. 9713–9729, 2020.

Kim, B. S & Kim, T.G, (2019). "Cooperation of simulation & data model for performance analysis of complex systems, "International *Journal of Simulation Modelling, vol. 18, no. 4, pp. 608– 619, 2019.*

Li, S., Zhou, Q., Zhou, R. *et al.* Intelligent malware detection based on graph convolutional network. *J Supercomput* (2021). https://doi.org/10.1007/s11227-021-04020-y

DOI: https://dx.doi.org/10.4314/swj.v18i2.6

Lu, R. (2019). Malware detection with LSTM using opcode language. https://arxiv.org/abs/1906.04593.

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity*, *2020*, 1–10. https://doi.org/10.1155/2020/6622927

Naeem, H.; Ullah, F.; Naeem, M.R.; Khalid, S.; Vasan, D.; Jabbar, S, Saeed, S. (2020): Malware detection in industrial internet of things based on hybrid image visualization and deep learning model. Ad Hoc Networks 2020, 105, 102154.

Qin, L, Yu, N, & Zhao, D. (2018).“Applying the Convolutional neural network deep Learning technology to behavioral recognition in intelligent video,”*TehnickiVjesnik-Technical Gazette,vol. 25, no. 2, pp. 528–535, 2018.*

Rezaei, S. and Liu, X. (2019): Deep learning for encrypted traffic classification: An overview. IEEE Communications magazine 57(5), 76–81.

Ren-Hung Hwang, Min-Chun Peng, Van-Linh Nguyen, Yu-Lun Chang (2019): An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level Appl. Sci. 2019, 9, 3414; doi: 10.3390/app9163414

Sun, Y., Bashir, A. K., Tariq, U., & Xiao, F. (2021). *Effective malware detection scheme based on classified behavior graph in IIoT. Ad Hoc Networks, 120, 102558.* doi:10.1016/j.adhoc.2021.102558.

Tahir, R. (2018). A study on malware and malware detection techniques. International Journal of Education and Management Engineering, 8(2):20–30.

Venkatraman, S.; Alazab, M.; Vinayakumar, R. (2019): A hybrid deep learning image-based analysis for effective malware detection. J. Inf. Secur. Appl. 2019, 47, 377–389.

Yadav A, Jha C. K, and A. Sharan, (2020) “Optimizing LSTM for time series prediction in Indian stockmarket,” *Procedia Computer Science, vol. 167, pp. 2091–2100.*

Yuan, F.; Cao, Y.; Shang, Y.; Liu, Y.; Tan, J.; Fang, B.(2018): Insider Threat Detection with Deep Neural Network. In Computational Science—ICCS 2018; Springer: Cham, Switzerland.

Zarrad, O., Hajjaji, M. A., &Mansouri, M. N., (2019). “Hardware implementation of Hybridwind-solar energy system for pumping water based on artificial neural network controller, ”*Studies in Informatics and Control, vol. 28, no. 1, pp. 35–44, 2019.*

Zhang, J. (2020). Deepmal: A CNN-LSTM model for malware detection based on dynamic semantic behaviours. In 2020 International Conference on Computer Information and Big Data Applications, CIBDA, pages 313–316.