

Classification of Malware using Machine learning and Deep learning Techniques

B.A.S. Dilhara
Dept. of Network and Security
NSBM Green University
Sri Lanka

ABSTRACT

The threats imposed by the cyber-attacks due to malicious software (malware) have been increasing drastically with the evolution of information technology. Since people use web applications on a daily basis these malware attacks have become challenging. There have been various attacks affecting confidentiality, integrity and availability of data which has become a major security concern. Though the manual inspection and classification methods seemed to bring up some light to this facet, these methods are no longer considered effective, since they are time consuming and inefficient. With the high-rate malware spreading, it is a necessity to come up with some novelty approach to classify them as malware or benign software. So, this is where machine learning comes up as a novelty approach in malware classification. In this paper, a malware dataset was used on several machine learning classifiers like Support Vector Machinery (SVM) and Gaussian Naive Bayes classifiers were used and Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN) were used as the deep learning classifiers. Although there are many other methods for malware classification, a machine learning approach could be efficient and effective in detecting malicious software. Thus, the primary objective of this paper is to provide an insight to the machine learning approach in malware classification by depicting, which is the best classifier of the listed, that can effectively classify malware based on their accuracy or precision. In conclusion, based on the results this recognizes Recurrent Neural networks as the best approach that recorded the highest accuracy.

Keywords

Malware Classification, Machine Learning, Deep Learning, Binary Classification

1. INTRODUCTION

With the evolution of technology, the growth of malwares has posed an immense threat to the internet security. A malicious software (malware) can be defined as any code added, altered, or deleted from a software system with the aim of causing detrimental effects or destabilize the intended function of the system [1]. A malware can be used to breach the security policy of a computer system in terms of confidentiality, integrity, and availability. The execution characteristics of the program defines the malware classification. It is also classified based on its payload, method of vulnerability exploitation and the propagation method [2]. Based on this, malwares can be classified as virus, worm, Trojan horse, Adware ransomware and spyware. In the early days of malware evolution, it came into being as a prank or an experiment gone wrong. However, there are several malwares that were created in laboratories like Darwin game, creeper, pervading animal, and Rabbit Virus, but never released out of the labs [2]. It was the Elk Cloner (in 1981), which was the

first virus to be released out and infect a personal computer. Followed by it, Brain; the first Microsoft PC virus came into being as a more harmful virus than Elk Cloner [2]. Followed by them, the malware has systematically conquered the computer world. Zeus in 2007, Koobface in 2008, Stuxnet in 2010, Cryptolocker Trojan in 2013, and Wannacry in 2017 have trembled the world throughout the last decade. Therefore, it is a necessity to develop a method for effectively identify malware before they cause damages to the computer systems.

To keep up with the malware, cyber-defenses are being constantly improved by the researchers and security analysts. Endpoint protection is one such essential element where it provides a suite of security programs like firewall, email protection, URL filtering, anti-spam, and sandboxing [3]. The last layer of defense is mostly provided by the Antivirus software which is responsible for detection and removal of malware installed on the endpoint device. Typically, antivirus software relies on a signature based or heuristic method. A signature can be defined as an algorithm or a hash that identifies a specific malware uniquely while a heuristic method defines a set of rules that determined by analysts after behavior analysis of malware. However, this method is not so effective in detecting malware. The major challenge in such traditional approaches is that the new variant of malware uses bypassing techniques which makes the code obfuscate. This will lead to failures in detecting new types of malwares. A malware analysis is typically composed of 2 parts namely discovery phase and the classification phase. In the discovery phase malware is caught and identified. Feature vector selection methods are being used for malware classification process and this can be classified as static analysis and dynamic analysis. These differ from each other based on the way features are extracted. Static analysis of malware uses a method of capturing information from binary programs without execution and dynamic analysis of malware, monitors the behavior of malware at run time in an isolated system [4]. However, malware detection methods based on dynamic analysis is more robust to obfuscation methods in comparison to static analysis methods. A hybrid approach can be reached by the combination of these two methodologies. So, it is a necessity to implement a novelty approach in detecting malicious software and that is where the machine learning techniques comes in. This research aims at analyzing the usage of machine learning techniques to examine Malicious software. This paper further exemplifies the categorization of feature representation used to generate training data, while comprehending algorithms used to understand a perfect prediction model. As the primary objective, this paper intends to conceptualize the most effective and efficient machine learning approach in classifying malware, which basically divides them into malware or benign that involves a binary classification approach.

2. BACKGROUND

Bugra Cakir et al. [1], has used a shallow deep learning-based method for feature extraction. Here for the classification a Gradient Boosting Algorithm has been used and the performance of the model has been evaluated using k-fold cross validation split without sacrificing a validation split, which reached an accuracy of 96%. However, Matilda Rhode et al. [10], has used a recurrent neural network to predict whether an executable is malicious or benign which produced an accuracy of 94%. The tested dataset was obtained from VirusTotal which consists of 1000 malicious and 600 trusted windows 7 executables along with along with 800 trusted samples from the system files of a fresh Windows7 64-bit installation. Huan Zhou in his paper proposes a deep learning methodology for malware detection using static and dynamic combined features to classify a portable execute file (PE) is malicious or not [11]. In the research by Ivan Firdausi et al. [12], uses several machine learning algorithms like k-Nearest Neighbors (kNN), J48 Decision Tree, Support Vector Machine (SVM), Naïve Bayes and Multiplayer Perceptron Neural Network (MLP). However, the best accuracy of 96.8%, precision of 95.9% along with a false positive rate of 2.4% was obtained for J48 decision tree and the dataset used consists of malware and benign data in the format of Windows PE [12]. Furthermore, Muhammad Furqan Rafique et al. [13] propose a deep learning malware detection technique. The features are extracted using two types of convolutional neural networks and finally most important and distinctive features are selected using a SVM and a wrapper-based mechanism. The dataset BIG 2015 published by Microsoft on the Kaggle platform, is finally trained using a Multilayer Perceptron.

3. MACHINE LEARNING APPROACH

The usage of Ai powered antimalware tools has been immensely increased in the past decade, due to the general belief among many cybersecurity experts that these tools will help in efficient malware diagnosis. Due to the ability to learn from massive datasets and effectively predict the outcome, the machine learning methods are widely used over the traditional trends. Conventional machine learning approaches include supervised machine learning models, and most novelty approaches include deep learning techniques.

3.1 Machine Learning Algorithms

In the supervise machine learning approach models are built from malware input data and its labeled class. Support Vector Machinery (SVM) and Naive Bayes have been used in this context as the machine learning algorithms.

Support Vector Machinery (SVM): To solve of problems related to data classification, learning and prediction SVM is widely used. When compared with classifiers such as decision trees, SVM provides a higher accuracy rate. Using a hyperplane, the SVM classifier separate the data points by the largest margin [5]. The core idea of SVM to find a maximum marginal hyperplane that divides the dataset into best classes.

Naive Bayes: This is another commonly used classifier which is simple and effective in many real-world problems. Naive Bayes is a probabilistic classifier, that uses a very simple Bayesian Network, and it uses Maximum A posteriori decision rule in a Bayesian setting [6].

3.2 Deep learning Algorithms

Within the recent years deep learning, which is a subset of machine learning has been an emerging research area. Since it possesses the ability to learn from mistakes and uses a deep

neural network to simulate human brain's learning process it has outsmarted the traditional machine learning algorithms at many instances [4]. Due to the powerful ability of deep learning algorithms, many researchers have applied deep learning for malware detection [7].

Convolutional Neural Network (CNN): A CNN consists with one or more convolutional layers with fully connected layers on top and it has pooling layers, and weights [8]. Being able to train easily than feed forward neural networks and their ability to train with standard back propagation makes it a highly attractive architecture to be used [8]. Fig 1 depicts an image of a typical CNN

Recurrent Neural Network (RNN): RNN is commonly used for sequential attribute recognition of a dataset and predicting the next likely scenario using patterns [9]. Long Short-Term Memory (LSTM) is and type of RNN that uses backpropagation, but it uses memory blocks connected to layers to learn sequence data. A typical structure of a LSTM is denoted in Fig2.

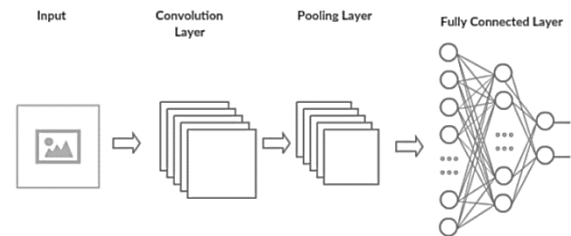


Fig 1: Convolutional Neural Network Structure

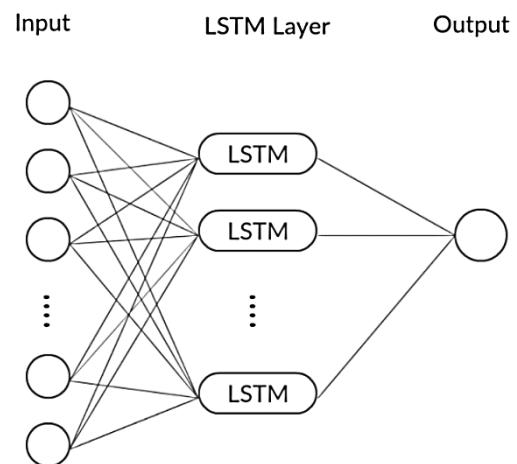


Fig 2: LSTM Network Structure

4. METHODOLOGY

The machine learning approach for malware classification that is described in this context involves the following steps. That is dataset selection, data preprocessing and feature selection, training and finally the testing

4.1 Dataset Selection

The machine learning models must be trained before it is being tested on a real-life scenario. For the training purpose a recognized data set should be used. However, the problem face by most researchers is that finding out a high-quality dataset. The scarcity of good datasets has set a drawback in this domain. The data set used here is taken from the Kaggle dataset repository [14]. This dataset consists of 35

columns and consists of 50000 unique samples of benign and 50000 unique samples of malware which altogether makes a dataset of 100000 samples.

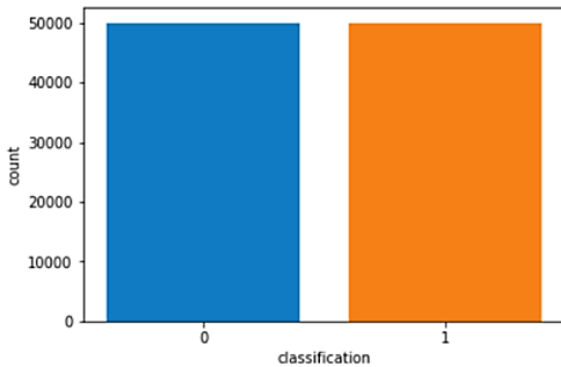


Fig 3: Dataset Distribution

Data preprocessing and feature selection: Before training the machine learning models the dataset should be preprocessed and features should be selected. The feature selection was carried out only for the machine learning models and it was done using Extra Tree Classifier. The deep learning models were trained without feature selection. The dataset was imported in the format of data frames, and it was then shuffled. Since the first 500000 data samples were malware and the next 500000 samples were benign, the dataset was shuffled to increase the randomness and prevent the overfitting during training. Then the features and targets were selected from the shuffled dataset. At the basic level 33 columns were selected as data (features) dropping the ‘hash’ column and the column with the name ‘classification’ was selected as the target (labels). Since data consist of values from different ranges, it had to be normalized. As the classification is addressed in a binary form, the strings ‘malware’ and ‘benign’ in the target were converted to ‘0’ and ‘1’ respectively.

Training: The pre-processed dataset with selected features is then used to train the desired machine learning algorithm. Selection of the machine learning algorithm should be done, based on the scenario it is being applied for. Several, machine learning and deep learning models are discussed earlier in the literature. The dataset with data and targets is then feed into the respective machine learning model and the model is allowed to train. If deep learning methods like Neural Networks are being used the model can be trained for several epochs to increase the level of training. Since there is no separate dataset for testing the dataset should be split into train data, test data, train target and test target respectively. In this context 2 machine algorithms namely SVM and Naive Bayes were used. As the deep learning algorithms Convolutional Neural Networks and Recurrent Neural Networks were used

Testing: This is the final phase of a machine learning approach. This is where the testing dataset is feed into the machine learning model to predict the accuracy of the model which provides an idea of how well the model reacts for unseen data. When feeding the data into the trained machine learning model, the data of the testing data set should be in the same format of the training dataset. Since there is no separated testing dataset, 10% of the training dataset is used for the testing purpose

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 31, 128)	512
activation_1 (Activation)	(None, 31, 128)	0
max_pooling1d_1 (MaxPooling1D)	(None, 31, 16)	0
conv1d_2 (Conv1D)	(None, 27, 64)	5184
activation_2 (Activation)	(None, 27, 64)	0
max_pooling1d_2 (MaxPooling1D)	(None, 27, 8)	0
dropout_1 (Dropout)	(None, 27, 8)	0
flatten_1 (Flatten)	(None, 216)	0
dense_1 (Dense)	(None, 1)	217
Total params: 5,913		
Trainable params: 5,913		
Non-trainable params: 0		

Fig 4: Proposed CNN Architecture

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 33, 64)	16896
dropout_1 (Dropout)	(None, 33, 64)	0
lstm_2 (LSTM)	(None, 33, 128)	98816
lstm_3 (LSTM)	(None, 33, 64)	49408
dropout_2 (Dropout)	(None, 33, 64)	0
lstm_4 (LSTM)	(None, 128)	98816
dense_1 (Dense)	(None, 1)	129
Total params: 264,065		
Trainable params: 264,065		
Non-trainable params: 0		

Fig 5: Proposed RNN Architecture

4.2 Performance Evaluation parameters

Evaluating the performance of a machine learning model is a fundamental aspect of machine learning. This helps in refining the parameters and selecting the best and most appropriate model from which are being tested. The most commonly used methods for evaluation of machine learning models, is called matrix [15]. Furthermore, metrics like Coefficient of determination (R2 score), accuracy and precision are also being used in determining the classifiers performance

Confusion Matrix: A confusion matrix consists of 2 rows and 2 columns which gives the number of false positive (FP), false negative (FN), true positive (TP) and true negative (TN). The Fig 6 depicts an image of a confusion matrix. In a confusion matrix an observed class is represented by each row while each column represents a predicted class [15]. Integer numbers are the entries of a confusion matrix, and the total of TP, TN, FP, and FN becomes equal to the number of test data [15].

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Fig 6: Confusion Matrix

Accuracy: Accuracy can be considered as a proportion of total number of predictions that were correctly identified. The following equation can be used to determine the accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision: The precision is defined as a proportion of predicted positive cases that were correctly identified, and the equation below can be used to determine the precision [15].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Coefficient of determination (R2 score): This is a statistic used in statistical models where primary objective is to predict and provide a measurement of the future outcome of the tested model [15].

$$r = \frac{n\sum(xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (3)$$

Here, r=The Correlation coefficient, n = number in the given dataset, x = first variable in the context, y = second variable

5. RESULTS

The Table 1 and Table 2 depicts the results obtained after the testing phase. The Table 1 consists of results without feature selection and Table 2 provides a comparison of results for machine learning classifiers with and without feature selection. In this study the main performance evaluation parameters that were included was accuracy, precision, R2 score, and parameters related to the confusion matrix. Based on these parameters several factors can be highlighted. According to the table 1, CNN and RNN which are deep learning algorithms depicts a prominent accuracy and precision over the traditional machine learning algorithms. However, it was obvious that when using the SVM, it had a clear advantage over the Gaussian Naïve Bayes algorithm. Moreover, the false positive and negative rates of the Gaussian Naïve Bayes algorithm were at a higher level which indicates that it is not that suitable for malware classification. This also gives a clear indication that proper usage of deep learning algorithms can result in great results.

Table 1. Results of the classifiers without feature selection

	Accuracy	Precision	FP	FN	R2
Gaussian NB	69.56%	62.22%	53.54%	7.06%	-0.21
SVM	98.61%	97.41%	2.55%	0.24%	0.94
CNN	99.67%	99.99%	0.28%	0.38%	0.96
RNN	99.96%	99.93%	0.06%	0.02%	0.99

The table 2 depicts the impact of feature selection and not selecting features on machine learning algorithms. This tabular depiction provides a clear idea that, through proper feature selection accuracy and precision. This was very much obvious in Gaussian Naïve Bayes algorithm, which initially had an accuracy of 69.56% and precision of 62.22% resulted in an accuracy of 72.51% and precision of 65.08%. It also resulted in reducing the false positive rate from 53.54% to 40.13%. In addition to this, even the SVM algorithm too showed some promising results. According to the table 2, the SVM algorithm has obtained an accuracy of 98.61% and precision of 97.41% when it was trained without feature selection. After the feature selection SVM resulted in an accuracy of 99.94% and precision of 99.23%. This also confirms the fact that feature selection has a clear advantage when it is used with machine learning models. Furthermore, it has also reduced the false positive rate of SVM from 2.55% to 0.39%. The R2 score of SVM also depicted a clear progression from 0.9443 to 0.9775, which is once again a very remarkable improvement.

Table 2. Results of classifiers with and without feature selection

	Accuracy	Precision	FP	FN	R2
Gaussian NB without feature selection	69.56%	62.22%	53.54%	7.06%	-0.217
Gaussian NB with feature selection	72.51%	65.08%	40.13%	14.69%	-0.099
SVM without feature selection	98.61%	97.41%	2.55%	0.24%	0.9443
SVM with feature selection	99.94%	99.23%	0.39%	0.72%	0.9775

However, the feature selection was done only for the machine learning classifiers while deep learning classifiers were trained without feature selection. Based on the results it was the RNN that had the better accuracy and R2 score and lowest percentage of False Positive and False Negative rates. The Fig. 9 provides a graphical representation of the results depicted in the Table 1. Fig.7 and Fig.8 presents how the validation accuracy and Accuracy varied during CNN and RNN training and testing phase.

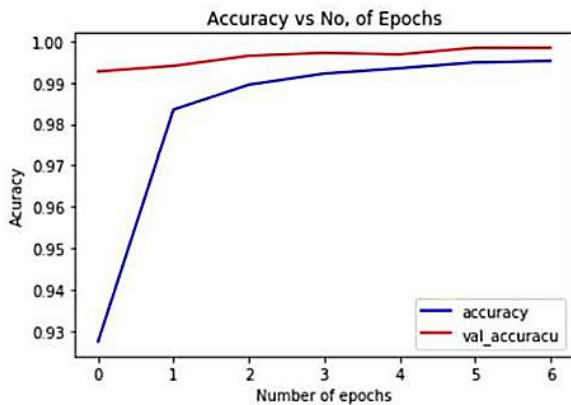


Fig 7: Variation of Accuracy and Validation Accuracy for CNN Model

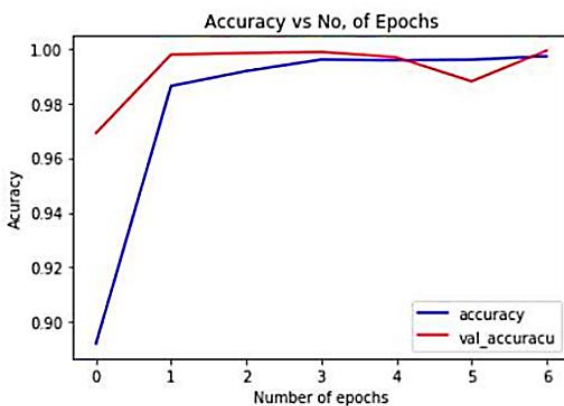


Fig 8: Variation of Accuracy and Validation Accuracy for RNN Model

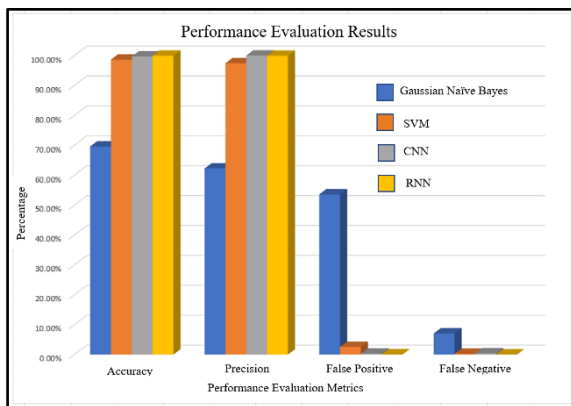


Fig 9: Graphical representation of the results

Fig 9 depicts a clear graphical representation of the variation of accuracy, precision, false positive rates and false negative rates of Gaussian Naïve Bayes, SVM, CNN and RNN. Based on this graph too, it is clear that the RNN is having the highest accuracy as well as the highest precision while Gaussian Naïve Bayes is having the lowest accuracy and the lowest precision level. Apart from that it is clearly visible that, the Gaussian Naïve Bayes is having the highest FP and FN rate. Another factor that can be highlighted here is that, RNN is having very low FP and FN values compared to the Gaussian Naïve Bayes Algorithm. So, based on the graph it is obvious that RNN is having a clear advantage against the other models.

6. LIMITATIONS AND FUTUREWORK

When compared to the other detection methods, several research areas in this domain are yet to be discovered. However, for the existing research work modifications can be done to obtain greater results. The analysis of the literature discussed above outsmarted the fact that the feature selection technique needs paramount improvements in detecting malware at a best accuracy rate. This can prevent confusions in malware identification. Modifications of the machine learning models is a necessity because over time attackers find mechanisms to overrule the prevailing defense measures. The fine tuning of the machine learning algorithms can produce better results utilizing the features to function at an optimum level. The limitations in datasets create drawbacks in most studies. A small data set can generate incorrect predictions. Since the same dataset was used for both training and testing the results of this research is bit higher than using separate datasets for training and testing. Generation of proper datasets of a considerable size can greatly help researchers to generate machine learning models with better results. To avoid the overfitting problems in classifiers more attention should be given for the data preprocessing stage and in processing, clustering can be used to sort out outliers. By preventing the inclusion of such noisy samples in training, can also generate good results.

7. CONCLUSION

Despite the extensive studies and staggering progress that the machine learning approach on malware classification have gained in the recent years; yet it remains a very challenging domain. However, in this domain, it is a necessity to upgrade the existing strategies since the attackers too develop counter measures to overrule the defense strategies. This paper focused on how to overtake such attackers in this race for data and privacy protection with a promising inclusion of machine learning in this domain. Based on the results it was the Recurrent Neural Network (RNN), that recorded the highest accuracy rate of 99.96% and R2 score of 0.9989. It also had the lowest False Negative Rate of 0.02% and False Positive Rate of 0.06%. However, Convolution Neural Network (CNN), too had promising results when compared with the machine learning classifiers. The results of this research have also provided an insight to the machine learning approach in malware classification by highlighting the fact that, feature selection can be effectively used for machine learning classifiers to produce outstanding results. In conclusion, it can be said that Recurrent Neural Network (RNN) is a promising machine learning approach in malware classification.

8. REFERENCES

- [1] B. Cakir and E. Dogdu, "Malware classification using deep learning methods," Proc. ACMSE 2018 Conf., vol. 2018-January, no. April 2018.
- [2] A. P. Namanya, A. Cullen, I. U. Awan, and J. P. Disso, "The World of Malware: An Overview," Proc. - 2018 IEEE 6th Int. Conf. Futur. Internet Things Cloud, FiCloud 2018, no. September, pp. 420–427, 2018.
- [3] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," J. Netw. Comput. Appl., vol. 153, no. July 2019, p. 102526, 2020. [Online]. Available: <https://doi.org/10.1016/j.jnca.2019.10.2526>
- [4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust

- Intelligent Malware Detection Using Deep Learning,” *IEEE Access*, vol. 7, pp. 46 717–46 738, 2019.
- [5] V. Menger, F. Scheepers, and M. Spruit, “Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text,” *Applied Sciences*, vol. 8, no. 6, p. 981, Jun. 2018. [Online]. Available: <https://doi.org/10.3390/app8060981>
- [6] C. Liu, L. Wang, B. Lang, and Y. Zhou, “Finding effective classifier for malicious URL detection,” in *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences - ICMSS 2018*. ACM Press, 2018. [Online]. Available: <https://doi.org/10.1145/3180374.3181352>
- [7] Z. Cui, F. Xue, X. Cai, Y. Cao, G. G. Wang, and J. Chen, “Detection of Malicious Code Variants Based on Deep Learning,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.
- [8] B. B. Benuwa, Y. Zhan, B. Ghansah, D. K. Wornyo, and F. B. Kataka, “A review of deep machine learning,” *Int. J. Eng. Res. Africa*, vol. 24, no. February 2017, pp. 124–136, 2016.
- [9] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-Aware neural language models,” 30th AAAI Conf. Artif. Intell. AAAI 2016, pp. 2741– 2749, 2016.
- [10] M. Rhode, P. Burnap, and K. Jones, “Early-stage malware prediction using recurrent neural networks,” *Comput. Secur.*, vol. 77, no. December 2017, pp. 578–594, 2018. [Online]. Available: <https://doi.org/10.1016/j.cose.2018.05.010>
- [11] R. Olson, G. James, and R. Howard, *Cyber Security Cyber Security*. Springer Singapore, 2011. [Online]. Available: <http://dx.doi.org/10.1007/978-981-10-8536-98>
- [12] I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, “Analysis of machine learning techniques used in behavior-based malware detection,” *Proc. - 2010 2nd Int. Conf. Adv. Comput. Control Telecommun. Technol. ACT 2010*, pp. 201–203, 2010.
- [13] A. M. M. Muhammad Furqan Rafique, Aqsa Saeed Qureshi, Asifullah Khan, Jin Young Kim, “Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique Muhammad,” pp. 1–20.
- [14] Saravana, “Malware detection,” Apr 2018. [Online]. Available: <https://www.kaggle.com/nsaravana/malware-detection>.
- [15] L. Zhang, Y. Zhu, P. Shi, and Q. Lu, “Performance analysis,” *Stud. Syst. Decis. Control*, vol. 53, pp. 59–85, 2016.