An Enhanced Classification Model for Likelihood of Zero-Day Attack Detection and Estimation

Victor T. Emmah, Chidiebere Ugwu, and Laeticia N. Onyejebu

Abstract — The growing threat to sensitive information stored in computer systems and devices is becoming alarming. This is as a result of the proliferation of different malware created on a daily basis to cause zero-day attacks. Most of the malware whose signatures are known can easily be detected and blocked, however, the unknown malwares are the most dangerous. In this paper a zero-day vulnerability model based on deep-reinforcement learning is presented. The technique employs a Monte Carlo Based Pareto Rule (Deep-RL-MCB-PR) approach that exploits a reward learning and training feature with sparse feature generation and adaptive multi-layered recurrent prediction for the detection and subsequent mitigation of zero-day threats. The new model has been applied to the Kyoto benchmark datasets for intrusion detection systems, and compared to an existing system, that uses a multi-layer protection and a rule-based ranking (RBK) approach to detect a zero-day attack likelihood. Experiments were performed using the dataset, and simulation results show that the Deep-RL-MCB-PR technique when measured with the classification accuracy metrics, produced about 67.77%. The dataset was further magnified, and the result of classification accuracy showed about 75.84%. These results account for a better error response when compared to the RBK technique.

Keywords — Deep Learning, Dimensionality Reduction, Malware, Sparse features, Zero-day Attack.

I. INTRODUCTION

A zero-day vulnerability represents serious and potential threat to many organizations, particularly in our modern day society. In zero-day vulnerability a hole in software that is unknown to the vendor is left undetected for quite a period as it extends beyond the expected instantaneous or roughly daily instantaneous detection. This security hole is typically exploited by hackers before the vendor even becomes aware or hurries to fix it. Thus, this exploit is typically referred to as a zero-day attack. Since the major concern for network administrators is to prevent attacks before they occur, zero-day attacks can pass undetected through conventional defences for a long period. This further exposes the networks to harmful agents and makes the administrator's job very difficult [1].

Some adverse impact of zero-day attacks can come in the form of data theft, unauthorised control/account takeover, reputation damage, loss of production and productivity and financial loss. Hackers can exploit the vulnerability to take unauthorised control and access to a network, website, server,

program or any other system [2]. In most cases, if the attack goes public, whether there is a patch for the vulnerability or not, it can harm the organisation's brand reputation in a big way. It sends out a public message that the cybersecurity measures are not in place and that the organisation's data and systems are highly susceptible to breaches.

Most existing approaches use datasets captured from honeypot traffic to generate an attack graph. This allows the use of very large datasets of malware samples, including all the dataset features, thereby making the possible distinct set of vulnerabilities to increase exponentially. The analysis of malware samples with its full dataset including all the dataset features poses a problem known as "curse of dimensionality" when analysing and organising data in high dimensional space [3], [4]. This means that as the size of the dataset increases, the possible set of expected outcome increases exponentially. Some other limitations present within the existing models include the use a single-pass detection phase which will not unravel the hidden patterns and exposed the network to zero-day vulnerability distribution (this is probably due to the nature of the data feed). Also, malware is only analysed to discover if they can cause a zero-day exploit, but without prioritizing the level of exploit for which they are capable of causing. This is what this paper seeks to do.

A typical enterprise use firewalls, antivirus and intrusion detection systems to secure its IT infrastructure which offer good first-level protection, but despite their best efforts, they are unable to protect enterprise infrastructure against zero-day attacks and are not very effective in keeping up with the various types of malwares which is rapidly increasing. In this paper, a technique based on deep reinforcement learning and sparse feature generation is proposed. The system is applied to the modelling of exploit using readily available network intrusion dataset with the aim of accurately determining the possibility that a new feature instance of network access in the dataset is zero-day vulnerable. In addition to identifying unknown vulnerabilities, the unknown malwares files will be ranked based on the priority for which they can cause harm, so as to easily trace and contain the threat.

II. RELATED WORKS

Different literatures have been considered which have previously studied how malware are analysed as a means of discovering a zero-day attack. Such works are:

Submitted on July 29, 2021.

Published on August 19, 2021.

Victor T. Emmah, Department of Computer Science, Rivers State University, Port Harcourt, Nigeria.
(e-mail: victor.emmah@ust.edu.ng).

Chidiebere Ugwu, Department of Computer Science, University of Port Harcourt, Rivers state, Nigeria.

Laeticia N. Onyejebu, Department of Computer Science, University of Port Harcourt, Rivers state, Nigeria.

[5] proposed a supervised learning approach by employing several data mining techniques to detect and classify zero-day malware based on the frequency of Windows API calls. Eight classifiers were used to develop a machine learning framework. These include Naïve Bayes (NB), k-Nearest Neighbour (KNN), and Sequential Minimal Optimization (SMO) algorithm with 4 different kernels (SMO-Normalized Polykernel, SMO-Polykernel, SMO-Puk, and SMO-Radial Basis Function (RBF)), Backpropagation Neural Networks, and J48 decision tree algorithms. The system proves to be better, compared to similar signature-free techniques that detect polymorphic malware and unknown malware based on analysis of Windows APIs.

[6] designed Honey farm, a hybrid scheme that combines anomaly and signature detection with honeypots. The advantages provided by existing detection techniques is applied to develop an effective defence against Internet worms. The system works on three levels. At first level, signature-based detection is used to filter known worm attacks. At second level, an anomaly detector is set up to detect any deviation from the normal behaviour. In the last level, honeypots are deployed to detect zero-day attacks. Low interaction honeypots track attacker activities while high interaction honeypots analyse new attacks and vulnerabilities.

[7] also proposed a hybrid technique, Suspicious Traffic Filter (STF), for detecting zero-day polymorphic worms. The two techniques combined were signature-based and anomaly-based, which fell into the category of behaviour-based. Their technique first tries to detect zero-day polymorphic worms and then tries to quarantine them. "STF observes all network traffic at an edge network and the internet. The traffic is passed simultaneously to both HoneyNet and IDS/IPS (Intrusion Detection System/Intrusion Prevention System) sensors through a port mirroring switch.

[8] proposed a hybrid real-time zero-day attack detection and analysis system which combines anomaly-based detection, behaviour-based detection, and signature-based detection techniques to analyse zero-day attacks in real-time. It is a layered architecture where all layers work together in parallel to provide solution to the system. The system is implemented and evaluated against various standard metrics like True Positive Rate (TPR), False Positive Rate (FPR), F-Measure, Total Accuracy (ACC) and Receiver Operating Characteristic (ROC) curve. The overall results were very promising, a detection rate of nearly 98% with 0.02 false positive rate and in the worst case, detection rate was 89% with 0.03 false positive rate.

[6] in their paper "Zero-day attacks defence technique for protecting against unknown vulnerabilities" proposed ZDAR (Zero Day Attack Remedy). Their approach, which senses the organization's network and monitors the behavioural activity of zero-day exploit at each and every stage of their life cycle can detect zero-day attacks using feature extraction and transformation by sensing suspicious network connections which do not match known attack signatures at run-time. The feature transformation module discovers the suspicious connections which differentiate between the behaviour of known attacks and anomalous activities. The anomaly detection technique is used to discover anomalies and thus to

identify types of zero-day attack using an assigned anomaly score.

[9] in their work "Malware Detection Using Machine Learning and Deep Learning", used supervised and unsupervised learning for malware classification. The purpose of their work was to detect and classify various malware that can cause zero-day attacks using different machine learning algorithms and deep learning models. Classification models were built using two approaches namely, Random forests and Deep learning. The results from their experiments show that Random forests outperformed the deep learning models and achieved the highest accuracy of 99.7%. Random forest also produced the second highest accuracy with no feature reduction. Between different deep learning models, DNN-3L and DNN-7L both combined with AE-1L attained an accuracy of 98.99%.

[10] proposed an approach for file signature generation for differentiating between malware of known and unknown families. In their work, "DeepOrigin: End-to-End Deep Learning for Detection of New Malware Families", they introduced a method for visualizing the signatures in a low dimensional space for improved malware analysis. Using an extensive dataset that consists of thousands of variants of malicious files, their approach achieved 97.7% accuracy when classifying between known and unknown malware families.

[11] in their work "Transfer Learning for Image-Based Malware Classification" used image analysis to detect and classify malware. To do this, they converted executable files to images and apply image recognition using deep learning models. To train these deep learning models, they employ transfer learning, relying on models that have been pre-trained on large image datasets. The performance of this technique was compared with k-Nearest neighbours machine learning technique which indicated that Deep Learning models are better able to generalise data as they outperformed the k-NN in simulated zero-day experiments.

[12] designed a scalable approach towards discovery of unknown vulnerabilities. They designed a hybrid architecture framework for zero-day attack detection based on a Rule-Based Ranking (RBK). Their proposed solution consists of three phases namely, zero-day attack path analyser, Risk analyser and physical layer.

The different literatures reviewed all presented malware as a threat involved in a zero-day attack without indicating which malware samples provides a higher or lower level of risk. It is however important to identify the malware with higher risks so as to be able to focus more attention on combating it. Our approach therefore uses a Deep Reinforcement learning technique to assess the risk level and also prioritize the zero-day binaries with respect to the likelihood of an attack.

III. MATERIALS AND METHODS

The model presented in this paper is similar to the work of [13] but with some key modifications. The existing system is primarily based on the Rule based Ranking (RBK) which follows from simple decision trees or logic. In the existing technique, a decision rule is used to filter out unknown malicious packet and update the database. In a data mining

paradigm, this resorts to using a trained model to locate a malicious class based on novel test data. This is, of course achieved with some initial training datasets and predefined malicious rule checks.

The concept of attack rank is derived from the Page Rank principle, an idea earlier proposed in [14]. In this paper, this concept is adapted to dataset processing using the diagram as shown in Fig. 1.

Fig. 1. Conceptual Model of the Node-Host-to-Attacker AttackRank Algorithm.

In the concept model, the attack paths are captured as anomalies or vulnerabilities while nodes are the corresponding exploited Protocol feature units. By default, all nodes (Attacker and System Host Protocol nodes) are assigned probabilities equal to $1/N$ where N is total number of nodes. Then a particular node-Host-Attacker probability can be identified by taking the ratio of the total number of all interconnections of Attacker-Nodes to respective Host-nodes to the total number of possible instances of these interconnections. For example, to determine the probability of Node Attacker-TCP (Node-A: TCP), we need to compute the ratio of Node1-4 and Node 3-4 interconnections and add it up as follows:

$PR(\text{Node-A: TCP}) = 5/7 \rightarrow (\text{no. of links in NodeA to Node TCP}) / \text{total no. of links scanned which is a high chance of vulnerable node.}$

This model comes after the input states have been filtered and the User Nodes (based on their behavioural responses) are identified as anomalies.

The dataset used in the experimentation was obtained from the Kyoto University benchmark data for real-time Intrusion

Detection Systems (IDS) and is available at [15]. It contained 1000 anomalous instances and was adapted for anomaly detection by separating a sample of known (existing) and unknown (new) exploits from the population of labelled exploits. A sample of the anomalous dataset is shown in Fig. 2.

```

|00f538c3d410822e241486ca061a57ee|Win32/ASuspect.HHDYL|genus|VIRUS
|0955924ebc1876f0b849b3b9e45ed49d|W32.EloradoKQ.Worm|WORM
|01b656578e9f00f289d4c560fb8f2ff8|Win32/Zuten.JZ|VIRUS
|0615864d027b6f46731ee3bdcbe24edd|W32/Zbot.TH.gen|Eldorado|VIRUS
|058c705488b0f173fb782d422f4092b|Win32.Packed.5EC0|VIRUS
|01bc2c90c9b5bffe79a0dd0f08dfe6f6|Win32/StartPage.UNH|VIRUS
|0c8078e676a7ac7cac5c927db4d45db3|Win32.Packed.1530|VIRUS
|0d0e1475b0e143d4c6a6b925507f94e7|Win32/SillyD1.YFM|VIRUS
|0645142dd01109963441a96025eecd67|W32.HfsAutoB.6A0E|VIRUS
|06e76cf96c7c7a3a138324516af9fce8|Win32.Packed.A9B9|VIRUS
|0f9829e53f83e2c61513b291f7bb7348|W32.382|VIRUS
|04df65889035a471f8346565600841af|Trojan.Genome.Win32.230770|TROJAN
|069506a02c4562260c971c8244bef301|UnclassifiedMalware|MALWARE
|0d273f3cf24f8f59f1a4ed7996e97507|Gen:Variant.Graftor.27475|VIRUS
|069cec530ac649d49d6f8f79b779368b|Win32/Gamepass.PPO|VIRUS
|0bc699a7117b3b07eb4d6d1583b13924|Win32/Llac.AA|VIRUS
|6ce6f415d8475545be5ba114f208b0ff|ETCART Test Virus|TEST_VIRUS
|07c22b7ee5b22247c4a63df6d50f5c|Backdoor.Death.25.H|BACKDOOR
|0d596bd439c7177a88cbfee0f5737f36|Win32.Packed.D69F|VIRUS
|0fe78378ed3d77612ab139a3aaf30842|Trojan.Generic.6538009|TROJAN
|0648e3483ed760aa63e34307b8696080|W32.OLGJA1DLL.Trojan|TROJAN
|063d2c2d9ca83cbe055b5f548b7ebcc|Gen:Dos.MbrInfecton.aal|aaaaa|VIRUS

```

Fig. 2. Sample of Anomalous Data for Known Exploits.

A. Proposed Model

Fig. 3 shows the architecture of the system. It is composed of three layers namely: **Zero-day Attack Path Generator** which emulates acts of intrusion and its consequent sparse detection using snort rule and PB-MCFR - it creates the zero-day attack paths (new exploit indexes) scenario for subsequent analysis by Risk Analyzer. The **Risk Analyzer Layer** creates a host-centric graph of the zero-day attack paths generated in the previous layer. Based on the in-coming and out-going exploit indexes (host-centric attack paths) an exploit-centric graph is generated and passed over to a modified attack rank algorithm via a Deep-RL block. The function of Deep-RL is to reinforce the scores of the paths whose exploits re-occur and then predict the score in the next time step. The modified attack rank algorithm uses an ordered (ascending-order) quick-sort algorithm to rank the scores computed by the Deep-RL stage and this is followed with a search for the matching indexes. The matching indexes are then preserved in a cellular matrix from which the actual matching exploit features may be decoded. **Dataset Layer** simply serves as a container from which the intrusion (zero-day exploits) features are obtained.

Fig. 3. System Architecture.

The key components of the new system architecture are the inclusion of the *Pareto-Based Monte Carlo Filtering Rule* (PB-MCFR) and the *Deep Reinforcement Learning* which are modifications done within the zero-day attack path generator layer and Risk Analyser Layer respectively. The PB-MCFR is used to reduce the dimensionality of the dataset for accurate analysis; while the Deep-RL is used to predict the index of the new exploit feature and reinforce the corresponding detected signal based on the frequency of occurrence.

The key processes in the model include, highly sparse filtering, which is done optimally to extract unknown anomalous paths in the network by using a Pareto-Based Monte Carlo Filtering Rule (PB-MCFR). This allows only 20% of the network traffic data to be randomly analysed at a time; and for a finite number of trial runs (called the Monte Carlo observation space), a filtering of unknown anomalous paths is also performed. Also, a ranking is automatically determined via deep reinforcement predictive learning with numeric encoding. If an attack is known and is detected, its corresponding numeric encoding label is incremented. For a set of such instances, a deep reinforcement learning with multiple layers of representations is then adaptively used to estimate in advance which attack path is most likely to be featured in the network graph; then sorting is performed to build an ordered representation of the mined graph. The Deep reinforcement predictive learning for zero-day detection is shown in algorithm 1.

Algorithm 1: Deep Reinforced Learning based Zero-day detection algorithm

```

Routine begins
initialize all detected malicious buffer state to 1.
for network_scenario in Network do
  if (equals(packet_content,snort_rules)) then
    drop current_packet;
  else
    preserve filtered_packet := current_packet;
  end if
  tag:= hash (preserve filtered_packet (arvl_time,
source_ip, destination_ip, source_port, destination_port,
protocol));
  update_database(tag);
  tagged_packet:= preserve filtered_packet;
  if (NOT (isMalicious (tagged_packet))) then
    Capture good traffic network scenario from safe
systems
    Extract features and update Snort NIDS database
  else
    unknown:=tagged_packet;
    insert unknown; set the buffer state vector,  $s_l$ 
    if unknown (isexist), then
      Reinforce unknown by incrementing its corresponding
buffer state by 1
       $s_l = s_l + 1$ 
      Call LSTMLearn routine
      Call Rank unknown routine with Sorting
    end if
    update zero_day_database(unknown);
  end if
end for
end routine

```

IV. EXPERIMENTS AND RESULTS

Table I shows the specification indexed representation for the simulation input data. For the task development, seventeen (17) instances of the input data were obtained from the dataset collected, and the expected anomaly state for each instance were recorded. The following indices ids 6, 8, 9, 10, 11, and 12 were seen to contain instances whose anomaly state is not known. These unknown anomalies are responsible for the zero-day attacks. The remaining instances contain known anomalous data, which is expected to be filtered by the snort filtering.

In this experiment, the conditioning threshold is varied in steps of 10 from a value of 50 to 100, with the number of simulations runs kept at 20. The initial threshold value of 50 was chosen after some prior experiments, which found 50 to be a more realistic threshold setting for the zero-day analysis routine. The variation in the threshold value is an incremental approach that will be used to identify when the threshold value may no longer have any impact on the prediction performance of the unknown anomaly. For each case scenario, the detected new exploit reinforced indices and their corresponding anomaly correctness score (1 for correct or 0 for incorrect) are as shown in Table II.

As shown in Table II, when the threshold value stands at 50 units, the model can predict accurate estimates of the anomalous pattern instance. That is, the malware instances that are capable of causing an attack can accurately to be identified. Also, as the threshold value is increased beyond 70 units, it becomes clear from the table that there is no significant change in the prediction pattern; thus, the range for expected performance is deemed to be narrow because no substantial changes are noticeable.

Fig. 4 shows the simulation interface of the zero-day detection application, which displays the comparison of the simulation procedure and output for both existing and the new models. This is done by launching the application programme developed. It is then followed by uploading the training and testing datasets using the buttons shown in the application. The Deep-RL application was initiated with the “DEEP-RL ZDAY” button which displays the output on its accompanying result window (RESULT SUMMARY). The existing model which is Attack-Rank algorithm is executed by clicking the “STANDARD Z-DAY” button. The outputs are also displayed on its accompanying “RESULTS SUMMARY” window. This procedure (1-3) is repeated for different 20 trial runs taking particular note of the reported metric, i.e., classification accuracy.

In the experiment, the Deep-RL model is compared to the existing system based on the influence of the Monte Carlo Based Pareto Rule (MCB-PR) for the base dataset and a magnified version of the dataset. To build the magnified dataset, the original dataset was duplicated 5 times.

The interfaces shown in Fig. 5-7 are the screenshots showing the results of launching the application program, which shows the comparison interfaces of both the existing and new models. Fig. 5 displays the results of classification accuracy and runtime values of both the proposed and existing models using the base datasets for the first iteration. Fig. 6 and 7 shows the screenshots of the same values (CA and Runtime) of both models for second and third iterations respectively. The values indicated in the screenshots are

extracted into a table and they correspond to the values represented in Table III.

TABLE I: INDEXED REPRESENTATION AND EXPECTED ANOMALY STATE

Index id	1	2	3	4	5	6	7	8-12	13-17
Expected Anomaly State	Known	known	Known	known	known	Unknown	known	Unknown	known

TABLE II: INDEXED REPRESENTATION FOR THRESHOLDS AND EXPECTED ANOMALY STATE.

Thresholds	Predicted Indices												Correctness Score			
50	1	6	8	9	10	11	12	0	1	1	1	1	1	1	1	1
70	NA	6	8	9	10	11	12	NA	1	1	1	1	1	1	1	1
90	NA	6	8	9	10	11	12	NA	1	1	1	1	1	1	1	1
100	NA	6	8	9	10	11	12	NA	1	1	1	1	1	1	1	1

Fig. 4. Zero-day detection interface.

Fig. 7. Result Summary Window 3.

Fig. 5. Result Summary Window 1.

Fig. 6. Result Summary Window 2.

TABLE III: SUMMARY OF RESULTS FOR BASE DATASET

Simulation Run	CA _{proposed} (%)	Run-time _{proposed} (s)	CA _{existing} (%)	Run-time _{existing} (s)
1	65.22	5	53.10	0
2	71.74	6	53.10	1
3	65.22	6	53.10	0
4	64.13	7	53.10	0
5	70.65	9	53.10	0
6	59.78	9	53.10	0
7	69.59	9	53.10	0
8	65.22	9	53.10	6
9	67.39	8	53.10	1
10	73.91	9	53.10	0
11	65.22	9	53.10	0
12	63.04	4	53.10	0
13	75.00	8	53.10	0
14	69.57	8	53.10	0
15	78.26	9	53.10	0
16	77.17	8	53.10	1
17	63.04	8	53.10	1
18	63.04	8	53.10	0
19	68.48	5	53.10	0
20	59.78	5	53.10	0
Mean	67.77	7.45	53.10	0.5

Table III shows the classification accuracy and runtime values for the new (Deep-RL-MCB-PR) and existing models. The values shown represent values extracted from the application interface of the summary windows in Fig. 5-7. The first column is the number of simulations runs which, for this experiment, is 20. The second and third columns represent the classification accuracy (CA) and run-time values for the new model while the fourth and fifth columns represent the classification accuracy (CA) and Run-time values for the existing model. The values for the first to third rows in the table can be traced to the Fig. 5-7 as indicated.

Other values shown are as the experiments were conducted. Table IV shows the CA and Run-time values of the new and existing systems for the magnified dataset.

TABLE IV: SUMMARY OF RESULTS FOR MAGNIFIED DATASET

Simulation Run	CA _{proposed}	Run-time _{proposed} (s)	CA _{existing}	Run-time _{existing} (s)
1	82.14	33	53.10	12
2	75.47	33	53.10	11
3	79.19	6	53.10	23
4	74.84	7	53.10	28
5	79.97	19	53.10	0
6	76.86	10	53.10	34
7	73.45	49	53.10	12
8	75.31	46	53.10	10
9	68.17	45	53.10	10
10	80.12	34	53.10	11
11	78.26	34	53.10	9
12	76.86	34	53.10	9
13	68.63	55	53.10	10
14	72.98	33	53.10	10
15	80.12	34	53.10	8
16	79.34	35	53.10	8
17	79.65	35	53.10	8
18	76.39	33	53.10	11
19	69.25	34	53.10	0
20	69.87	35	53.10	9
Mean	75.84	32.2	53.1	11.65

Fig. 8. A Graph of Classification Accuracy on X-axis and Simulation Trial Runs on Y-axis for both Proposed and Existing Model.

V. DISCUSSION OF RESULTS

Fig. 5–7 represent the graphical snapshot of the results of the simulation procedure which displays the normal traffic and abnormal ones capable of causing zero-day exploit, ranking them in the order of higher priority through the influence of the deep reinforcement learning technique. The reported metric is based on the classification accuracy (CA) and runtime values of both the new and existing system using the base datasets as displayed in the ‘RESULT SUMMARY’ window of each figure and the values are presented in Table III. Also, the results based on the classification (CA) and runtime values of both the proposed and existing systems for magnified datasets are shown in Table IV.

The results represented in Table III show that the mean classification accuracy for the proposed DEEP-RL-MCB-PR model, as can be traced from Fig. 5–7 for the first three simulation runs. Using the base dataset, the DEEP-RL-MCB-PR performed better by producing a mean classification accuracy of 67.77% surpassing that of the existing model which produced 53.10%. For the magnified dataset, the mean classification accuracy for the new model as represented in table IV is 75.84% while the existing model produced 53.10%. This implies that the new model performed better to accurately predict a malware-based zero-day attack. These results make it clear that the prediction performance of the new model is about 3 times better than the existing model while the run-time of existing model is roughly 2 times better.

Fig. 7 shows the graph of classification accuracy for both the new and existing models. It clearly shows that the Deep-RL model performs better as all the bars have higher peaks compared to the existing model. This can be concluded that using a Deep-learning approach for zero-day detection and prioritization is better compared to using a rule-based approach.

VI. CONCLUSION

Zero-day vulnerability refers to the ‘hole’ in the software or network system that is not yet known to the user, or the manufacturers have not yet released a patch for it. Malware play very important roles in leveraging these vulnerabilities to cause zero-day attacks; and as such they cannot be underestimated. In this paper, we have analysed the operations of malware as regards to how they target hardware and software systems in order to cause an attack. From the results obtained, it has been shown that the Deep-RL zero-day exploit technique based on the LSTM is dependent on the threshold setting and the depth of the LSTM learning network. The experimental results show that a correct prediction pattern sequence can be obtained when the threshold setting is about 50. Conversely, as the threshold value grows beyond 70, the result of the prediction tends to narrow because no significant changes were noticed. The depth of the learning network does not necessarily improve the log-likelihood error response rather too much depth may degrade the performance.

From experiments performed, it was observed that the prediction performance improves as the number of monte carlo trial runs is increased but there is a limit to desired pattern change.

REFERENCES

- [1] Dalziel, H. (2014). How to defeat advanced malware: new tools for protection and forensics: *Syngress*.
- [2] Parrend, P., Navarro, J., Guigou, F., Deruyver, A., & Collet, P. (2018). Foundations and applications of artificial intelligence for zero-day and multi-step attack detection. *Journal on Information Security EURASIP*, 2018(1), 4.
- [3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. In: MIT Press.
- [4] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Journal of Science*, 313(5786), 504-507.
- [5] Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2011). Zero-day malware detection based on supervised learning algorithms of API call signatures. *Paper presented at the Proceedings of the Ninth Australasian Data Mining Conference*-Volume 121. 171-181.

- [6] Jain, P., & Sardana, A. (2012). Defending against internet worms using honeypot. *Paper presented at the Proceedings of the CUBE International Information Technology Conference.*
- [7] Kaur, R., & Singh, M. (2014). Efficient hybrid technique for detecting zero-day polymorphic worms. *Paper presented at the 2014 IEEE International Advance Computing Conference (IACC).*
- [8] Kaur, R., & Singh, M. (2015). A Hybrid Real-Time Zero-day Attack Detection and Analysis System. *International Journal of Computer Network and Information Security*, 9, 19-31. doi: 10.5815/ijcnis.2015.09.03.
- [9] Rathore, H., Agarwal, S., Sahay, S. K., & Sewak, M. (2018). Malware Detection Using Machine Learning and Deep Learning. *Paper presented at the International Conference on Big Data Analytics.*
- [10] Cordonsky, I., Rosenberg, I., Sicard, G., & David, E. O. (2018). DeepOrigin: End-to-end deep learning for detection of new malware families. *Paper presented at the 2018 International Joint Conference on Neural Networks (IJCNN).* 1-7.
- [11] Bhodia, N., Prajapati, P., Di Troia, F., & Stamp, M. (2019). Transfer Learning for Image-Based Malware Classification. *arXiv preprint arXiv:1903.11551.*
- [12] Singh, U. K., & Joshi, C. (2018). Scalable approach towards discovery of unknown vulnerabilities. *International Journal of Network Security*. 20(5), 827-835.
- [13] Singh, U. K., Joshi, C., & Singh, S. K. (2017). Zero day attacks defense technique for protecting system against unknown vulnerabilities. *International Journal of Scientific Research in Computer Science and Engineering*. 5(1), 13-18.
- [14] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web.* Stanford InfoLab.
- [15] Song, J., Takakura, H., & Okabe, Y. (2008). Cooperation of intelligent honeypots to detect unknown malicious codes. *Paper presented at the 2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing.*

Mr. V. T. Emmah, obtained his Bachelor of Science degree and Master of Science degree from Department of Computer Science, Rivers State University and University of Port Harcourt respectively. He is currently pursuing a Ph. D in Computer Science at the University of Port Harcourt. He is a member of the Computer Professional (Registration Council) of Nigeria. His main research work focuses on Machine Learning, Data Science, Deep Learning and Artificial Intelligence.