Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security

Rahul Vigneswaran K*, Vinayakumar R†, Soman KP† and Prabaharan Poornachandran‡
*Department of Mechanical Engineering, Amrita School of Engineering, Amritapuri
†Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore
‡Center for Cyber Security Systems and Networks, Amrita School of Engineering, Amritapuri
Amrita Vishwa Vidyapeetham, India
Email: rahulvigneswaran@gmail.com, vinayakumarr77@gmail.com

***Abstract*—Intrusion detection system (IDS) has become an essential layer in all the latest ICT system due to an urge towards cyber safety in the day-to-day world. Reasons including uncertainty in finding the types of attacks and increased the complexity of advanced cyber attacks, IDS calls for the need of integration of Deep Neural Networks (DNNs). In this paper, DNNs have been utilized to predict the attacks on Network Intrusion Detection System (N-IDS). A DNN with 0.1 rate of learning is applied and is run for 1000 number of epochs and KDDCup-'99' dataset has been used for training and benchmarking the network. For comparison purposes, the training is done on the same dataset with several other classical machine learning algorithms and DNN of layers ranging from 1 to 5. The results were compared and concluded that a DNN of 3 layers has superior performance over all the other classical machine learning algorithms.**

***Index Terms*—Intrusion detection, deep neural networks, machine learning, deep learning**

I. Introduction

In the modern world, the fast-paced technological advancements have encouraged every organization to adopt the integration of information and communication technology (ICT). Hence creating an environment where every action is routed through that system making the organization vulnerable if the security of the ICT system is compromised. Therefore, this call for a multilayered detection and protection scheme that can handle truly novel attacks on the system as well as able autonomously adapt to the new data.

There are multiple systems that can be used for shielding such ICT systems from vulnerabilities, namely anomaly detection and IDSs. A demerit of anomaly-detection systems is the complexity involved in the process of defining rules. Each protocols being analyzed must be defined, implemented and tested for accuracy. Another pitfall relating to anomaly detection is that harmful activity that falls within usual usage pattern is not recognized. Therefore the need for an IDS that can adapt itself to the recent novel attacks and can be trained as well as deployed by using datasets of irregular distribution becomes indispensable.

Intrusion Detect Systems (IDSs) are a range of cybersecurity based technology initially developed to detect vulnerabilities and exploits against a target host. The sole use of the IDS is to detect threats. Therefore it is located out-of-band on the infrastructure of the network and is not in the actual real-time communication passage between the sender and receiver of data. Instead, they solutions will often make use of a TAP or SPAN ports to analyze the inline traffic stream's copy and will try to predict the attack based on a previously trained algorithm, hence making the need of a human intervention trivial [56].

In the field of cybersecurity, algorithms of machine learning have played an essential part. Especially, due to the incredible performance and potential of deep learning networks in recent days in various problems from a wide variety of fields which were considered unsolvable in past, the reliability of applying it for Artificial Intelligence (AI) and unsupervised challenges have increased [39]. Deep-learning is nothing but a partition of machine-learning that mimics the functions of the human brain and hence the name artificial neural network. The concept of deep learning consists of creating hierarchical representations that are complex that involve the creation of simple building blocks to solving of high-level problems. In recent days the application of deep learning methods are leveraged towards various use cases of cyber security such as [40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55].

Therefore it becomes obvious that Deep neural networks and IDSs, when combined together, can work at a superhuman level. Also, since the IDSs are out-of-band on the infrastructure, common attacks like DoS which primarily aims at choking the network band to gain access of the host, cannot bottleneck the performance of it, hence this security layer cannot tamper with ease. Towards the end, the sections are organized as followe: Section II reviews the work related to IDS, different deep neural networks and some discussions about KDDCup-'99' dataset that was published. Section III takes an in-depth look at Deep Neural Networks (DNN) and the applications of ReLU activation function. Section IV analyses the dataset used in this paper, explains the shortcoming of it and evaluates the final results. Section V concludes and states a plausible workflow into the future of this research work.

II. Related work

The research on ID in network security has existed since the birth of the computer architectures. The use of ML techniques and solutions to holistic IDS has become common in recent days, but training data at hand is limited and are mostly used only for bench-marking purposes. DARPA datasets [1], are one of the most comprehensive datasets available publicly. The data of tepdump offered by the 1998 DARPA ID Evaluation network of 1998 was cleaned and utilized for the KDDCup-contest of 1999 at the 5th International Conference on Knowledge Discovery and Data Mining. The job was to organize the records of the connections that are already preprocessed into either traffic which is normal, or one of the following categories of attack: 'DoS', 'Probing', 'R2L' and 'U2R'.

The preprocessing of the KDDCup-'99' competition's data was done using the MADAMID framework [2]. The entries that used variants of decision trees showed only marginal differences in performance occupied the first three places [3, 4, 5]. The first 17 submissions of the competition were all bench-marked to perform well and are summarized [6]. The majority of published results were tested and trained with only 10% training set observing the feature reduction on the KDDCup-'99' datasets [7, 8, 9]. Few researchers used custom built datasets, with extracted from the 10% KDDCup-'99' training set [10, 11, 12].

There are a number of interesting publications where the results are indirectly compared due to the use of different training and test datasets. In a paper [13], genetic algorithm and decision trees were used for automatic rule generation for an intelligent system for enhancing the capability of an existing IDS. The integrated utilization of neural networks in IDS was suggested by [14] and [15]. [16] proposed an application of recurrent neural networks (RNNs) and [17] compared the neural network architectures' performance for statistical anomaly detection to datasets from four different scenarios.

Although the datasets of KDDCup-'99' has various issues [18], [19] argues that they are still an effective bench-marking dataset which is publicly available to compare different intrusion detection methods.

The fundamental reason for the popularity of ML-based approaches is because of its capability to attack the constantly evolving complex and diverse threats to achieve an acceptable false positive rate of ID with the reasonable computational cost. In early stages, [36] used PNrule method which is derived from P-rules and N-rules to figure out the existence and non-existence of the class respectively. This has an advantage due to the enhancement of the detection rate in the other types of attacks except for the U2R category.

An extrapolation to traditional Feed Forward Networks (FFN) in the plane of taking inspiration from biological elements, is a network named Convolutional Neural Network (CNN). In early stages, CNN was used for processing of images by making use of normal 2D layers, pooling 2D layers and completely connected layers. [37] studied the applications of CNN for IDS with the KDDCup of '99' dataset and compared the results with several other bleeding-edge algorithms. After a broad analysis, they have concluded the superiority of CNN over the other algorithms. The study of the utilization of the Long Short-Term Memory (LSTM) classifier was conducted by [38] with the same dataset. It has been stated that because of the capability of LSTM to see into the past and relate the successive records of connections demonstrates usefulness towards intrusion detection systems.

The ultimate motive of this paper is to exploit the possibility of randomness of the inbound cyber attack which is unsuspicious to human sight but can be filtered by adding an artificial intelligence layer to the network. Hence, by training the neural network with the existing cyber attacks data, it can learn to predict an inbound attack easily and can either alert the system or initiate a pre-programmed response which may abstain the attack from proceeding further. As a result, millions worth, aftershock collateral damage and expensive data leaks can be prevented just by simply adding an extra layer to the security system. The benchmarking dataset used for training the networks are bygone and for a better real-time robustness of the algorithm, more recent data must be used for retraining before deploying in the field. The obligatory of this paper is to introduce the essence of artificial neural networks into the much rapidly evolving field of cybersecurity.

III. Background

Deep neural networks (DNNs) are Artificial Neural Network (ANN) with a multi-layered structure comprised within the input-output layers. They can model complex non-linear relationships and can generate computational models where the object is expressed in terms of the layered composition of primitives.

Below we roughly cover simple DNNs and applications of ReLU and why it is preferred over other activation functions.

A. Deep Neural Network (DNN)

While traditional machine learning algorithms are linear, deep neural networks are stacked in increasing hierarchy of complexity as well as abstraction. Each layer applies a nonlinear transformation onto its input and creates a statistical model as output from what it learns. In simple terms, the input layer is received by the input layer and passed onto the first hidden layer. These hidden layers perform mathematical computations on our inputs. One of the challenges in creating neural networks is deciding the hidden layers' count and the count of the neurons for each layer. Each neuron has an activation function which is used to standardize the output from the neuron. The "Deep" in Deep learning refers to having more than one layer which is hidden. The output layer returns the output data. Until the output has reached an acceptable level of accuracy, epochs are continued.

B. Application of rectified linear units (ReLU)

ReLu has turned out to be more efficient and have the capacity to accelerate the entire training process altogether

[20]. Usually, Neural networks use a sigmoidal activation function or tanh (hyperbolic tangent) activation functions. But these functions are prone to vanishing gradient problem [21]. Vanishing gradient occurs when lower layers of a DNN have gradients of nearly null because units of higher layers are nearly saturated at the asymptotes of the tanh function. ReLU offers an alternative to sigmoidal non-linearity which addresses the issues mentioned so far [22].

IV. Experiments

We consider Keras [23] as a wrapper on top of TensorFlow[24] as software framework. For exponentially increasing the agility of processing of data in deep-learning architectures, a GPU enabled tensorflow in a single Nvidia-GK110BGL-Tesla-k40 has been used.

A. Datasets description

The DARPA's program for ID evaluation of 1998 was managed and prepared by Lincoln Labs of MIT. The main objective of this is to analyze and conduct research in ID. A standardized dataset was prepared, which included various types of intrusions which imitated a military environment and was made publicly available. The KDD intrusion detection contest's dataset of 1999 was a well-refined version of this [25].

B. Shortcomings of KDDCup-'99' dataset

ReLu has turned out to be more efficient and have the A detailed report and major shortcomings of the provided synthetic data set such as KDDCup-'98' and KDDCup-'99" were discussed by [26]. The main condemnation was that they failed to validate their data set a simulation of real-world network traffic profile. Irrespective of all these criticisms, the dataset of KDDCup-'99' has been used as an effective dataset by many researchers for bench-marking the IDS algorithms over the years. In contrast to the critiques about the creation of the dataset, [27] has revealed a detailed analysis of the contents, identified the non-uniformity and simulated the artifacts in the simulated network traffic data.

The reasons behind why the machine learning classifiers have limited capability in identifying the attacks that belong to the content categories R2L, U2R in KDDCup-'99' datasets have been discussed by [28]. They have concluded that it is not possible to get acceptable detection rate using classical ML algorithms. It is also stated the possibility of getting high detection rate in most of the cases by producing a refined and augmented data set by combining the train and test sets. However, a significant approach has not been discussed.

The DARPA / KDDCup-'88 failed to evaluate the traditional IDS resulting in many major criticisms. To eradicate this [29] used Snort ID system on DARPA / KDDCup-'98' tcpdump traces. The system performed poorly resulting in low accuracy and the impermissible false positive rates. It failed in detecting dos and probing category but contrasting performing better than the detection of R2L and U2R.

Despite the harsh criticisms [30], still KDDCup-'99' set is one of the most widely used publicly available bench-marking datasets reliable for studies related to IDS evaluation and other security-related tasks [31]. In the effort of mitigating the underlying problems existing with KDDCup-'99' set, a refined version of dataset named NSL-KDD was proposed by [31]. It removed the connection redundancy records in the entire train and test data. In addition to that, the invalid records were also removed from the test data. These measures prevent the classifier from being biased in the direction of the more frequent records. Even after the refinement, this failed to solve the issues reported by [32, 33], and a new dataset named UNSW-NB15 was proposed.

C. DARPA / KDDCup-'99' dataset

The DARPA's ID evaluation group, accumulated network based data of IDS by simulation of an air force base LAN by over 1000s of UNIX nodes and for continuously 9 weeks, 100s of users at a given time in Lincoln Labs which was then divided into 7 and 2 weeks of training and testing respectively to extract the raw dump data TCP. MIT's lab with extensive financial support from DARPA and AFRL, used Windows and UNIX nodes for almost all of the inbound intrusions from an alienated LAN unlike other OS nodes. For the purpose of dataset, 7 distinct scenarios and 32 distinct attacks which totals up to 300 attacks were simulated.

Since the year of release of KDD-'99' dataset [34], it is the most vastly utilized data for evaluating several IDSs. This dataset is grouped together by almost 4,900,000 individual connections which includes a feature count of 41. The simulated attacks were categorized broadly as given below :

- **Denial-of-Service-Attack (DoS):** Intrusion where a person aims to make a host inaccessible to its actual purpose by briefly or sometimes permanently disrupting services by flooding the target machine with enormous amounts of requests and hence overloading the host [35].
- **User-to-Root-Attack (U2R):** A category of commonly used maneuver by the perpetrator start by trying to gain access to a user's pre-existing access and exploiting the holes to obtain root control.
- **Remote-to-Local-Attack (R2L):** The intrusion in which the attacker can send data packets to the target but has no user account on that machine itself, tries to exploit one vulnerability to obtain local access cloaking themselves as the existing user of the target machine.
- **Probing-Attack:** The type in which the perpetrator tries to gather information about the computers of the network and the ultimate aim for doing so is to get past the firewall and gaining root access.

KDDCup-'99' set is classified into the following three groups: Basic features: Attributes obtained from a connection of TCP/IP comes from this group. Majority of these features results in implicitly delaying the detection. Traffic features: Features computed w.r.t. a window of time is categorized under this group. This can be further subdivided into 2 groups:

- **"Same host" features:** The connections that has identical end host as the connection under consideration for the continuously 2 seconds fall into this category and serves the purpose of calculating the statistics of protocol behaviour, etc.
- **"Same service" features:** The connections that are only having identical services to the present connection for the last two seconds fall under this category.
- **Content features:** Generally probing attacks and DoS attacks have at least some kind of frequent sequential intrusion patterns unlike R2L and U2R attacks. This is due to the reason that they involve multiple connections to a single set of a host(s) under short span of time while the other 2 intrusions are integrated into the packets of data partitions in which generally only one connection is involved. For the detection of these types of attacks, we need some unique features by which we will be able to search for some irregular behaviour. These are called content features.

D. Identifying network parameters

Hyper-tuning of parameters to figure out the optimum set of parameters to achieve the desired result is all by itself a separate field with plenty of future scope for research. In this paper, the learning is kept constant at 0.01 while the other parameters where optimized. The count of the neurons in a layer was experimented by changing it over the range of 2 to 1024. After that, the count was further increased to 1280 but didnt yield any appreciable increase in accuracy. Therefore the neuron count was tuned to 1024.

E. Identifying network structures

Conventionally, increasing the count of the layers results in better results compared to increasing the neuron count in a layer. Therefore, the following network topologies were used in order to scrutinize and conclude the optimum network structure for our input data.

- DNN with 1,2,3,4,5 layers.

For all the above network topologies, 100 epochs were run and the results were observed. Finally, the best performance was showed by DNN 3 layer compared to all the others. To broaden the search for better results, all the common classical machine learning algorithms were used and the results were compared to the DNN 3 layer, which still outperformed every single classical algorithm. The detailed statistical results for different network structures are reported in the table I.

F. Proposed Architecture

An overview of proposed DNNs architecture for all use cases is shown in Fig. 1. This comprises of a hidden-layer count of 5 and an output-layer. The input-layer consists of 41 neurons. The neurons in input-layer to hidden-layer and hidden to output-layer are connected completely. Back-propagation mechanism is used to train the DNN networks. The proposed network is composed of fully connected layers, bias layers and dropout layers to make the network more robust.

TABLE I
NETWORK STRUCTURE INFORMATION

Layer (type)	Output Shape	Param
Dense-1 (Dense)	(NIL, 1024)	43008
Dropout-1 (Dropout)	(NIL, 1024)	0
Dense-2 (Dense)	(NIL, 768)	787200
Dropout-2 (Dropout)	(NIL, 768)	0
Dense-3 (Dense)	(NIL, 512)	393728
Dropout-3 (Dropout)	(NIL, 512)	0
Dense-4 (Dense)	(NIL, 256)	131328
Dropout-4 (Dropout)	(NIL, 256)	0
Dense-5 (Dense)	(NIL, 128)	32896
Dropout-5 (Dropout)	(NIL, 128)	0
Dense-6 (Dense)	(NIL, 1)	129
Activation-1 (Activation)	(NIL, 1)	0

Input and hidden layers: This layer consists of 41 neurons. These are then fed into the hidden layers. Hidden layers use ReLU as the non-linear activation function. Then weights are added to feed them forward to the next hidden layer. The neuron count in each hidden layer is decreased steadily from the first to the output to make the outputs more accurate and at the same time reducing the computational cost.

Regularization: To make the whole process efficient and time-saving, Dropout (0.01). The function of the dropout is to unplug the neurons randomly, making the model more robust and hence preventing it from over-fitting the training set.

Output layer and classification: The out layer consists only of two neurons Attack and Benign. Since the 1024 neurons from the previous layer must be converted into just 2 neurons, a sigmoid activation function is used. Due to the nature of the sigmoid function, it returns only two outputs, hence favouring the binary classification that was intended in this paper.

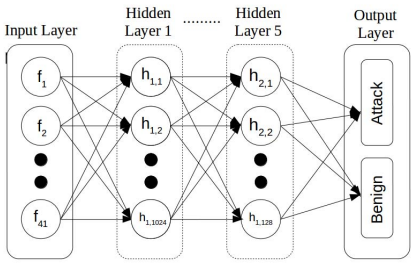

Fig. 1. Proposed architecture

V. RESULTS

For the scope of this paper, the KDDCup-'99' dataset was fed into classical ML algorithms as well as DNNs of varying

TABLE II
RESULTS

Algorithm	Accuracy	Precision	Recall	f1-score
DNN-1	0.929	0.998	0.915	0.954
DNN-2	0.929	0.998	0.914	0.954
DNN-3	0.930	0.997	0.915	0.955
DNN-4	0.929	0.999	0.913	0.954
DNN-5	0.927	0.998	0.911	0.953
Ada Boost	0.925	0.995	0.911	0.951
Decision Tree	0.928	0.999	0.912	0.953
K-Nearest Neighbour	0.929	0.998	0.913	0.954
Linear Regression	0.848	0.989	0.821	0.897
Navie Bayes	0.929	0.988	0.923	0.955
Random Forest	0.927	0.999	0.910	0.953
SVM*-Linear	0.811	0.994	0.770	0.868
SVM*-rbf	0.811	0.992	0.772	0.868

*Support Vector Machine

hidden layers. After the training is completed, all models were compared for f1-score, accuracy, recall and precision with the test dataset. The scores for the same has been compared in detail in Table II.

DNN 3 layer network has outperformed all the other classical machine learning algorithms. It is so because of the ability of DNNs to extract data and features with higher abstraction and the non-linearity of the networks adds up to the advantage when compared with the other algorithms.

VI. CONCLUSION

This paper has elaborately recapitulated the usefulness of DNNs in IDS comprehensively. For the purpose of reference, other classical ML algorithms have been accounted and compared against the results of DNN. The publicly available KDDCup-'99' dataset has been primarily used as the benchmarking tool for the study, through which the superiority of the DNN over the other compared algorithms have been documented clearly. For further refinement of the algorithm, this paper takes into account of DNNs with different counts of hidden layers and it was concluded that a DNN with 3 layers has been proven to be effective and accurate of all.

Since the neurons are trained with a bygone benchmarking dataset, as discussed several times in this paper, this comes as a pitfall for this methodology. Fortunately, it can be vanquished by using a fresh dataset with the essences of the latest attack strategies before the actual deployment of this artificial intelligence layer to the existing network systems to ensure the agility of the algorithms real-world capabilities.

From the empirical results of this paper, we may claim that deep learning methods are a promising direction towards cyber security tasks, but even though the performance on artificial dataset is exceptional, application of the same on network traffic in the real-time which contains more complex and recent attack types is necessary. Additionally, studies regarding the flexibility of these DNNs in adversarial environments are required. The increase in vast variants of deep learning algorithms calls for an overall evaluation of these algorithms in regard to its effectiveness towards IDSs. This will be one of the direction towards IDS research can travel and hence will remain as a work of future.

REFERENCES

[1] R. Lippmann, J. Haines, D. Fried, J. Korba and K. Das. "The 1999 DARPA off-line intrusion detection evaluation". Computer networks, vol. 34, no. 4, pp. 579 595, 2000. DOI http://dx.doi.org/10.1016/S1389-1286(00)00139-0.

[2] W. Lee and S. Stolfo. "A framework for constructing features and models for intrusion detectionsystems". ACM transactions on information and system security, vol. 3, no. 4, pp. 227261, 2000. DOI http://dx.doi.Org/10.1145/382912.382914.

[3] B. Pfahringer. "Winning the KDD99 classification cup: Bagged boosting". SIGKDD explorations newsletter, vol. 1, pp. 6566, 2000. DOI http://dx.doi.org/10. 1145/846183.846200.

[4] M. Vladimir, V. Alexei and S. Ivan. "The MP13 approach to the KDD'99 classifier learning contest". SIGKDD explorations newsletter, vol. 1, pp. 76 77, 2000. DOI http://dx.doi.org/10.1145/846183. 846202.

[5] R. Agarwal and M. Joshi. "PNrule: A new framework for learning classier models in data mining". Tech. Rep. 00-015, Department of Computer Science, University of Minnesota, 2000.

[6] C. Elkan. "Results of the KDD'99 classifier learning". SIGKDD explorations newsletter, vol. 1, pp. 63 64, 2000. DOI http://dx.doi.org/10.1145/846183. 846199.

[7] S. Sung, A.H. Mukkamala. "Identifying important features for intrusion detection using support vector machines and neural networks". In Proceedings of the symposium on applications and the Internet (SAINT), pp. 209216. IEEE Computer Society, 2003. DOI http://dx.doi.org/10.1109/saint.2003.1183050.

[8] H. Kayacik, A. Zincir-Heywood and M. Heywood. "Selecting features for intrusion detection: A feature relevance analysison KDD 99 intrusion detection datasets". In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

[9] C. Lee, S. Shin and J. Chung. "Network intrusion detection through genetic feature selection". In Seventh ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD), pp. 109114. IEEE Computer Society, 2006

[10] S. Chavan, K. Shah, N. Dave, S. Mukherjee, A. Abraham and S. Sanyal. "Adaptive neuro-fuzzy intrusion detection systems". In Proceedings of the international conference on information technology: Coding and computing (ITCC), vol. 1, pp. 7074. IEEE Computer Society, 2004. DOI http://dx.doi.org/ 10.1109/itcc.2004.1286428.

[11] S. Chebrolu, A. Abraham and J. Thomas. "Feature deduction and ensemble design of intrusion detection systems". Computers & security, vol. 24, no. 4, pp. 295307, 2005. DOI http://dx.doi.org/10.1016/j.Cose.2004.09.008.

[12] Y. Chen, A. Abraham and J. Yang. "Feature selection and intrusion detection using hybrid flexible neural tree". In Advances in neural networks (ISNN), vol. 3498 of Lecture notes in computer science, pp. 439 444. Springer Berlin / Heidelberg, 2005. DOI http://dx.doi.org/10.1007/11427469_71.

[13] C. Sinclair, L. Pierce and S. Matzner. "An application of machine learning to network intrusion detection". In Proceedings of the 15th annual computer security applications conference (ACSAC), pp. 371377. IEEE Computer Society, 1999. DOI http://dx.doi.org/10.1109/csac.1999.816048.

[14] H. Debar, M. Becker and D. Siboni. "A neural network component for an intrusion detection system". In Proceedings of the IEEE computer society symposium on research in security and privacy, pp. 240250. IEEE Computer Society, 1992. DOI http://dx.doi.org/10.1109/risp.1992.213257.

[15] J. Cannady. "Artificial neural networks for misuse detection". In Proceedings of the 1998 national information systems security conference (NISSC), pp. 443456. Citeseer, 1998.

[16] H. Debar and B. Dorizzi. "An application of a recurrent network to an intrusion detection system". In International joint conference on neural networks, 1992. IJCNN., vol. 2, pp. 478 483 vol.2. jun 1992. DOI http://dx.doi.org/10.1109/ijcnn. 1992.226942.

[17] Z. Zhang, J. Lee, C. Manikopoulos, J. Jorgenson and J. Ucles. "Neural networks in statistical anomaly intrusion detection". Neural network world, vol. 11, no. 3, pp. 305316, 2001.

[18] J. McHugh. "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". ACM transactions on information and system security, vol. 3, no. 4, pp. 262294, 2000. DOI http://dx.doi.org/10.1145/382912.382923.

[19] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set". In IEEE symposium on computational intelligence for security and defense applications, Cisda, pp. 16. IEEE, Jul. 2009. DOI http://dx.doi.org/10.1109/cisda. 2009.5356528.

[20] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315323.

[21] Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.

[22] Maas, A.L., Hannun, A.Y. and Ng, A.Y., 2013, June. Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, No. 1, p. 3).

[23] F. Chollet, "Keras (2015)," URL http://keras. Io, 2017.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., "Tensorflow: A system for large-scale machine learning." in OSDI, vol. 16, 2016, pp. 265283.

[25] Stolfo, S., Fan, W. and Lee, W., KDD-CUP-99 Task Description. 1999-10-28)[2009-05-08]. http://KDD. ics. uci. edu/databases/kddcup99/task, html.

[26] J. McHugh. "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". ACM transactions on information and system security, vol. 3, no. 4, pp. 262294, 2000. DOI http://dx.doi.org/10.1145/382912.382923.

[27] M. Mahoney and P. Chan. "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection". In Recent advances in intrusion detection, vol. 2820 of Lecture notes in computer science, pp. 220237. Springer Berlin / Heidelberg, 2003.

[28] Sabhnani, Maheshkumar, and Gursel Serpen. "Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set." Intelligent Data Analysis 8, no. 4 (2004): 403-415.

[29] S. Brugger and J. Chow. "An assessment of the DARPA IDS evaluation dataset using snort". Tech. Rep. CSE-2007-1, Department of Computer Science, University of California, Davis (UCDAVIS), 2005.

[30] J. McHugh. "Testing intrusion detection systems:A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory". ACM transactions on information and system security, vol. 3, no. 4, pp. 262294, 2000. DOI http://dx.doi.org/10.1145/382912.382923.

[31] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali-A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009. 2009.

[32] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the U[8] H. Kayacik, A. Zincir-Heywood and M. Heywood. "Selecting features for intrusion detection: A feature relevance analysison KDD 99 intrusion detection datasets". In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

[33] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data)."Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[34] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/database[8] H. Kayacik, A. Zincir-Heywood and M. Heywood. "Selecting features for intrusion detection: A feature relevance analysison KDD 99 intrusion detection datasets". In Proceedings of the third annual conference on privacy, security and trust (PST). 2005.

[35] McDowell, M. (2013). Understanding Denial-of-Service Attacks US-CERT. United States Computer Emergency Readiness Team.

[36] R. Agarwal and M. V. Joshi, Pnrule: A new framework for learning classifier models in data mining (a case-study in network intrusion detection), in Proceedings of the 2001 SIAM International Conference on Data Mining. SIAM, 2001, pp. 117.

[37] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1222-1228). IEEE.

[38] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal, 56(1), 136-154.

[39] Sommer, R. and Paxson, V., 2010, May. Outside the closed world: On using machine learning for network intrusion detection. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 305-316). IEEE.

[40] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Evaluating deep learning approaches to characterize and classify malicious URLs. Journal of Intelligent & Fuzzy Systems, 34(3), 1333-1343.

[41] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Detecting malicious domain names using deep learning approaches at scale. Journal of Intelligent & Fuzzy Systems, 34(3), 1355-1367.

[42] Vinayakumar, R., Soman, K. P., Poornachandran, P., & Sachin Kumar, S. (2018). Evaluating deep learning approaches to characterize and classify the DGAs at scale. Journal of Intelligent & Fuzzy Systems, 34(3), 1265-1276.

[43] Vinayakumar, R., Poornachandran, P., & Soman, K. P. (2018). Scalable Framework for Cyber Threat Situational Awareness Based on Domain Name Systems Data Analysis. In Big Data in Engineering Applications (pp. 113-142). Springer, Singapore.

[44] Vinayakumar, R., Soman, K. P., Poornachandran, P., & Sachin Kumar, S. (2018). Detecting Android malware using long short-term memory (LSTM). Journal of Intelligent & Fuzzy Systems, 34(3), 1277-1288.

[45] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Deep encrypted text categorization. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 364-370). IEEE.

[46] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Secure shell (ssh) traffic analysis with flow based features using shallow and deep networks. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 2026-2032). IEEE.

[47] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Evaluating shallow and deep networks for secure shell (ssh) traffic analysis. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 266-274). IEEE.

[48] Vinayakumar, R., Soman, K. P., Velan, K. S., & Ganorkar, S. (2017, September). Evaluating shallow and deep networks for ransomware detection and classification. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 259-265). IEEE.

[49] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Applying deep learning approaches for network traffic prediction. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 2353-2358). IEEE.

[50] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Long short-term memory based operation log anomaly detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 236-242). IEEE.

[51] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Deep android malware detection and classification. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1677-1683). IEEE.

[52] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1222-1228). IEEE.

[53] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Evaluating effectiveness of shallow and deep networks to intrusion detection system. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1282-1289). IEEE.

[54] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). International Journal of Information System Modeling and Design (IJISMD), 8(3), 43-63.

[55] Vysakh S Mohan, Vinayakumar R, Soman Kp and Prabaharan Poornachandran. S.P.O.O.F Net: Syntactic Patterns for identification of Ominous Online Factors In Security and Privacy Workshops (SPW), 2018 IEEE [InPress].

[56] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.