ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/320335100

Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System

Conference Paper · October 2017

DOI: 10.1109/ICTCS.2017.29

CITATIONS
11

READS
634

3 authors:

Malek Al-Zewairi
Jordan Information Security and Digital Forensics Research Group
17 PUBLICATIONS **100** CITATIONS

SEE PROFILE

Sufyan Almajali
Princess Sumaya University for Technology
38 PUBLICATIONS **168** CITATIONS

SEE PROFILE

Arafat Awajan
Princess Sumaya University for Technology
91 PUBLICATIONS **294** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Developing Email Spam Detection Systems based on Evolutionary Algorithms for Academic Networking Environments View project

Project

Secure RFID Access Control System View project

All content following this page was uploaded by Malek Al-Zewairi on 11 October 2017.

The user has requested enhancement of the downloaded file.

Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System

Malek Al-Zewairi[†]*, Sufyan Almajali[†], and Arafat Awajan[†]
[†] Computer Science Department,
King Hussein Faculty of Computing Sciences,
Princess Sumaya University for Technology,
Amman 11941, Jordan
[†]m.alzewairi@psut.edu.jo, [†]s.almajali@psut.edu.jo, [†]awajan@psut.edu.jo

***Abstract*—Deep Learning has been proven more effective than conventional machine-learning algorithms in solving classification problem with high dimensionality and complex features, especially when trained with big data. In this paper, a deep learning binomial classifier for Network Intrusion Detection System is proposed and experimentally evaluated using the UNSW-NB15 dataset. Three different experiments were executed in order to determine the optimal activation function, then to select the most important features and finally to test the proposed model on unseen data. The evaluation results demonstrate that the proposed classifier outperforms other models in the literature with 98.99% accuracy and 0.56% false alarm rate on unseen data.**

***Index Terms*—Deep Learning; Network Security; Intrusion Detection System; UNSW-NB15; H2O;**

I. Introduction

With the current advancement in network technology, the proliferation of Internet of Things (IoT) and the rapid growth of Software Defined Network (SDN), the number of network-connected devices is expected to exceed the global population by more than three times in the year 2021 [1]. Moreover, according to a recent study by IHS Markit Ltd, the number of connected IoT devices will reach 20 Billion in 2017 [2]. Evidently, this rapid increase had an enormous impact on the overall security posture of the Internet and all connected services. For example, the Mirai worm has scanned the Internet for vulnerable IoT devices and turned them into botnets, eventually participating in the two largest and record-breaking Distributed Denial of Service (DDoS) attacks in September and October 2016 with 620 Gbps and 1 Tbps estimated attack size respectively [3], [4].

Network Intrusion Detection System (NIDS) constitutes an essential security tool for organizations to monitor network traffic and identify network attacks. NIDSs can be categorized into three main categories based on the detection method they use in identifying potential attacks as signature-based, anomaly-based or specification-based NIDS [5].

Due to the continuous change of attacks signature and the emerging of new threats and normal traffic alike, the research community is always in need for modern datasets that accurately reflect the current status of security threats as well as benign applications traffic. UNSW-NB15 [6] is a new dataset for evaluating NIDS created by the Cyber Security Research Group at the Australian Centre for Cyber Security (ACCS), which contains over 2 million labeled modern normal and abnormal network traffic.

In this study, the authors propose employing the Deep Learning (DL) machine-learning algorithm as a binomial classifier for NIDS and experimentally evaluating the proposed model using the UNSW-NB15 dataset.

The rest of the paper is organized as follows: Section II explores the recent work on NIDS in the past two years. Section III describes how the UNSW-NB15 dataset is prepared for the use in this study. In Section IV, the proposed DL model is defined. Thereafter, the proposed model evaluation results are presented and discussed in Section V. Finally, the study is concluded in section VI.

II. Literature Review

In this section, the current literature on NIDS is presented and discussed in chronological order.

The UNSW-NB15 dataset was first introduced as a modern dataset aims to replace the ancient datasets for evaluating NIDSs in [6] in November 2015. The study presented the dataset features, explained the collection and processing tools and methodology, then compared it with the KDD-CUP99 dataset [7].

In [8], authors of UNSW-NB15 dataset presented a subset of the dataset divided into training and testing sets with 60%, 40% ratio respectively. Then, the two new sets were statistically analyzed using the Z-Test and the KS-Test to determine its distribution. The results indicated that both sets have the same distribution. Moreover, the multivariate skewness and kurtosis functions were used to prove that the training and the testing sets are statistically similar. Furthermore, the dependency between features was measured using two methods (i.e. Pearsons correlation coefficient and Gain Ratio). Finally, five algorithms (namely; artificial neural network, decision

978-1-5386-0527-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ICTCS.2017.29

tree, expectation-maximization, logistic regression and nave Bayes) were used to compare the dataset with the KDD-CUP99 dataset using accuracy as a performance measure; however, the target class label is not balanced in the dataset, which has led to poor results using accuracy as a metric.

In [9], an Adaptive IDS for industrial internet of things that uses One Class Support Vector Machine (OCSVM) algorithm was proposed. The use of OCSVM would allow the model to classify unknown anomalies. To accommodate the changes in the network architecture, the proposed system utilizes Spearman's rank correlation coefficient to match the unknown traffic with known traffic. Six datasets with different configurations were used to evaluate the proposed system. The datasets were created using the hybrid environment for design and validation testbed.

In [10], an IDS for cloud service providers named the Program Semantic-Aware Intrusion Detection at Network and Hypervisor Layer (PSI-NetVisor) was proposed. PSI-NetVisor employs 7 detection models in a two-layer detection system; the first layer operates at the Cloud Network Server level, which is responsible for maintaining network configuration and forwarding traffic. While, the second layer operates at the Cloud Compute Server level, which is responsible for hosting the VMs. The proposed system was evaluated using the UNSW-NB15 dataset, and the evaluation results showed that it was capable of detecting anomalies in traffic with high accuracy (i.e. 94.54).

A cascade of boosting-based artificial neural network multiclass classifier for intrusion detection system has been proposed in [11]. The performance of the proposed method was evaluated using two datasets (i.e. KDD-CUP99 and UNSW-NB15). The results showed that the proposed method has performed better on the KDD-CUP99 dataset when approaching the problem as both a binomial and a multiclass problem.

In [12], the authors proposed an evaluation metric for NIDS datasets to measure the degree of which the dataset accurately represents realistic network traffic. The proposed metric uses a fuzzy logic system based on the Sugeno-Type fuzzy inference method with two linguistic variables (i.e. good and average). Then, the authors generated a new NIDS dataset using the same platform that was used to generate UNSW-NB15 dataset (see Section III). However, the generated dataset contains fewer attack categories than UNSW-NB15.

In [13], the authors employed the Ramp Loss K-Support Vector Classification-Regression algorithm to develop a multiclass classification model. The results show that the proposed model out-performs other models when compared with the results in [8].

A multi-layer perceptron feed-forward artificial neural network with a single hidden layer was proposed in [14] to detect DoS attacks. The model was evaluated on the NSL-KDD [15] dataset and the UNSW-NB15 dataset. The authors also experimented with the number of hidden layers and reported their effect on the model accuracy, loss, training and testing time.

III. Dataset Description and Preparation

The UNSW-NB15 [6] is a modern labeled dataset for evaluating NIDSs created by the cyber security research group at ACCS and it is publicly available for researchers since late 2015. The dataset contains over 2.5 million records in CSV format with 49 features including both packet-based and flow-based features. The features are further categorized into four categories: flow, basic, content, time and additional general and connection features. The Ixia PerfectStorm testing platform was used to set up the testbed and generate about 100GB of both normal and malicious network traffic. While, Argus and Bro-IDS were utilized to extract the features from the raw "pcap" files. The dataset has two labeled features, that is, the traffic label (i.e. normal or attack) and the attack category label, which classifies the attack traffic into 9 different classes based on the attack type as follows: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

Unlike the well-known NIDS dataset such as KDD98 [16], KDD-CUP99 [7] and NSL-KDD [15] have been extensively used by researchers for evaluating their proposed NIDS. The UNSW-NB15 is a fairly new dataset containing modern network traffic both normal and abnormal including newer low footprint attack types, which makes it a more suitable choice for testing the proposed classifier. Besides, several studies such as [6], [8] have identified multiple anomalies in the aforementioned datasets. Examples of these anomalies are the incredible amount of redundant records, unbalanced distribution among traffic classes and many missing values in the KDD-CUP99. In addition, the datasets are incredibly old (i.e. 1998, 1999 and 2009 respectively); thus, do not reflect the current posture of security threats and do not include network traffic for modern benign applications.

A. Dataset Preprocessing

In this study, the full UNSW-NB15 dataset presented in [6] has been utilized to test the proposed deep learning intrusion detection classifier instead of the prepared training/testing partial datasets presented in [8]. The full dataset is provided as four separated CSV files. While exploring the dataset, the following observations were realized:

- The number of features is different between the full dataset in [6] and the prepared dataset in [8], where the full dataset contains 49 features, the prepared dataset has only 45 features. Six features from the full dataset were eliminated from the prepared dataset (i.e. "srcip", "sport", "dstip", "dsport", "stime" and "ltime"), while 2 new features were introduced (i.e. id and rate).
- Although, it was stated in [6] that the dataset contains 2,540,044 records, in reality, the dataset contains additional 3 records; one in each of the first 3 CSV files. Nonetheless, it also contains a tremendous amount of redundant records (i.e. 480,633 records), which constitutes around 19% of the dataset. Interestingly, a single record with local-host IP address was found.

- Minor issues such as missing the header row in the full dataset and the several typos in the columns names of the prepared dataset.

  Prior using the dataset, the following steps were conducted to prepare it:
- Merge the 4 CSV files.
- Remove whitespaces.
- Remove the duplicate records.
- Remove the record with localhost IP address.
- Replace the "-" with an empty value.
- Replace the values (0x000b and 0x000c) in the "sport" column with an empty value.
- Replace the values (0x20205321, 0xc0a8 and 0xcc09) in "dport" column with empty value.
- Add header row as the first row in the dataset.
- Add "Normal" value to the "attack_cat" column wherever appropriate.
- Ignore source IP, destination IP and attack category features when training and testing the model.

IV. Deep Learning Intrusion Classifier

In this study, the authors propose using DL algorithm as a binomial classifier for NIDS and experimentally evaluated this proposal. The proposed DL model is built using the H2O platform native implementation of DL algorithm that is a multilayer feedforward artificial neural network using backpropagation and stochastic gradient descent method.

A. Model Definition

The proposed DL model consists of five hidden layers with fifty neurons in total, separated evenly across the hidden layers each having ten neurons. The number of hidden layers corresponds to the number of features categories, while the number of neurons is equal to the number of features in the dataset. The model is trained on the full UNSW-NB15 dataset in ten epochs maximum using 10-folds cross-validation with a stratified cross-validation fold assignment method, where the squared sum of the incoming weights per unit is constrained by a maximum value of ten. The dataset records are shuffled randomly prior training the model and the importance of features are calculated using the Gedeon method [17].

B. Methodology

In order to evaluate how well the DL algorithm can classify network traffic as normal or abnormal network traffic, the authors have set up three different experiments. Furthermore, each experiment were executed ten times consecutively and the average values are reported to ensure that the reported results are accurate.

The goal of the first experiment is to find out the optimal activation function to be used. Therefore, three different activation functions with two different configurations; namely, Tanh, Tanh with dropout, Rectifier, Rectifier with dropout, Maxout and Maxout with dropout functions, are experimentally evaluated on the entire dataset using 10-folds cross-validation.

In the second experiment, the model is trained using the best activation function from the first experience and the most important features are identified using the Gedeon method [17] and their effects on the model generalization error metrics are also calculated. Interestingly, the H2O platform reports the important variables at the class level for categorical features. Therefore, in the second experiment, the output was normalized to the feature level.

Finally, in the third experiment the findings from the first two experiments are tested on unseen data by splitting the dataset into three sets (i.e. training: 60%, validation: 10%, and testing: 30%) and the best and worst models are reported.

C. Activation Function

An activation function is a non-linear function used in an artificial neural network to decide the output value of each neuron in the network [18]. The activation function plays an important role in the overall performance of the model. In this subsection, the 3 activation functions supported by the H2O platform are briefly described in addition to one technique known as the Dropout method.

The hyperbolic tangent function, or simply Tanh, converts input values in the range of $(-\infty, +\infty)$ to output values in the range of $(-1, +1)$. The value is mapped using equation 1 [19].

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{1}$$

The Rectifier activation function (i.e. Rectified Linear Unit) calculates the sum of all the weighted inputs after setting the negative values to zero and output that sum in the range of $[0, +\infty)$. The rectifier equation is presented in 2 [20].

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

Finally, the Maxout function divides the input values into groups of size k and outputs the maximum input value after weighting but without calculating the sum of the inputs [20]. On the other hand, the dropout technique is a generalization technique used to address the overfitting problem in deep neural networks by randomly dropping units from the neural network during training [21]. The H2O platform supports the dropout technique in all of the three aforementioned activation functions.

V. Evaluation Results

In order to evaluate the proposed model, three experiments were conducted as explained in section (IV, B) and their results are presented and discussed in this section.

A. Evaluation Metrics

The following evaluation metrics have been used to compare the performance of the different models: accuracy, Area Under Curve (AUC), F1-score, precision, recall, specificity, and training time normalized to the rage $[0, 1]$.

B. Evaluation Environment

All experiments were executed on a standalone H2O (v3.10.5.1) cluster running under Windows 10 with 24GB RAM and 3.4GHz Intel Core i7 quad processor.

C. Results and Discussion

In the first experiment, the performance of the 3 activation functions (i.e. Tanh, Rectifier and Maxout) in two configurations (i.e. with and without dropout) was compared. Figure 1 shows the results of the evaluation process. In terms of accuracy, AUC, F1-score, precision, recall, and specificity all of the 3 functions without dropout scored convergent results; however, when considering training time, the rectifier function scored the lowest time among them, while the maxout function scored the highest time among all the functions both with and without the dropout method. Conversely, when the dropout method was used, almost all of the evaluation metrics were negatively affected with the exception of training time, which significantly dropped for all functions. Therefore, the rectifier function without the dropout method was selected as the activation function.

In the second experiment, the most important features were investigated as follows: (1) the DL model was trained on the entire dataset using the rectifier function without dropout and 10-folds cross-validation. The important features for the 10 runs were identified using the Gedeon method [17], and then normalized to the feature level instead of the class level. Moreover, the top 5, 10, 15, 20 and 25 percent important features were selected for further evaluation in addition to the full features. Table I lists the important features in their respective percentage. Figure 2 compares the cross-validation results for training the model using the top 5%, 10%, 15%, 20%, 25% and 100% features. With the exception of training time, all models have scored convergent results.

TABLE I
TOP IMPORTANT FEATURES.

	Count	Important Features
Top 5%	19	service, proto, state, swin, sttl, dttl, dmeansz, ct_srv_dst, dwin, ct_state_ttl, trans_depth, djit, spkts, sjit, ct_dst_sport_ltm, sloss, dsport, sload, ct_dst_src_ltm
Top 10%	25	Top 5%, ct_srv_src, dload, dloss, synack, ackdat, dtcpb
Top 15%	31	Top 10%, ct_src_ltm, tcprtt, ltime, stcpb, smeansz, dpkts
Top 20%	33	Top 15%, stime, dur
Top 25%	35	Top 20%, sport, ct_src_dport_ltm
Full	45	Top 25%, dbytes, ct_dst_ltm, sbytes, sintpkt, ct_flw_http_mthd, res_bdy_len, is_sm_ips_ports, dintpkt, ct_ftp_cmd, is_ftp_login

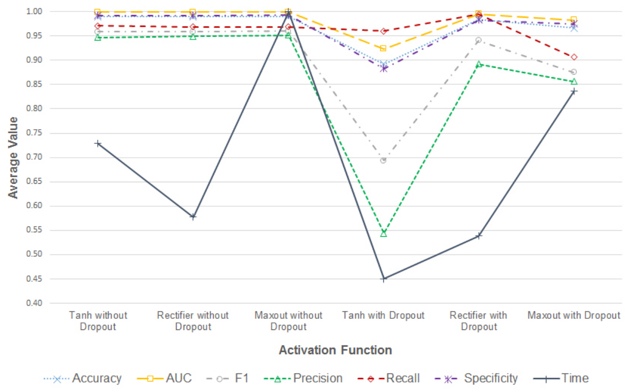

Fig. 1. Compare model performance using different activation functions.

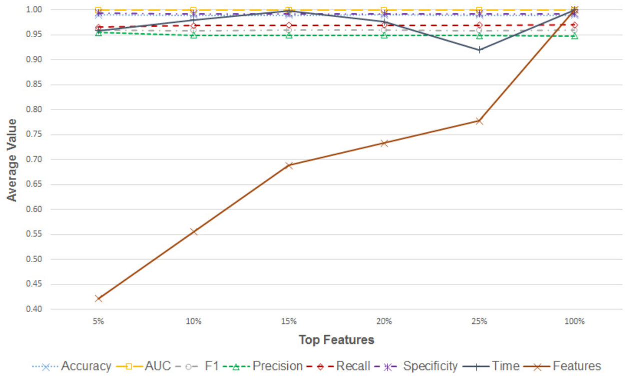

Fig. 2. Compare model performance using different percentage of the top important features.

To evaluate the proposed DL model on unseen data, the dataset was randomly divided into 3 different sets (i.e. training: 60%, validation: 10%, and testing: 30%) in the third experiment. The model was trained also using 10-folds cross-validation but on the training and the validation sets with the top 5%, 10%, 15%, 20%, 25% and 100% features resulting in 6 different models. Thereafter, the testing set was used to test the 6 models on unseen data. The results of the best and worst model are shown in Figures 3 and 4 respectively using AUC as a metric to define the best and worst model among 60 different models (6 models x 10 runs). As seen in figure 3, the DL model has achieved very high accuracy, AUC and specificity values (i.e. 0.9898, 0.99929, and 0.9944 respectively) when trained using the top 20% features with threshold 0.4482. Additionally, the model has achieved high F1-score, precision and recall values (i.e. 0.9599, 0.9614, and 0.9584 respectively). Conversely, the models with minimum AUC values are show in figure 4. In this case, the DL model, when trained using the top 20% features, scored the best among all the other models because the difference between the max and min AUC values was on average less than 0.001.

Finally, we compared the performance of the best model (i.e. the DL model when trained using the top 20% features with threshold 0.4482) with the other proposed models in the literature. Table II shows the results of the comparison with the other models in the literature [8], [10], [11], [13]. The results show that using DL algorithm has significantly higher accuracy value and lower False Alarm Rate (FAR) than the

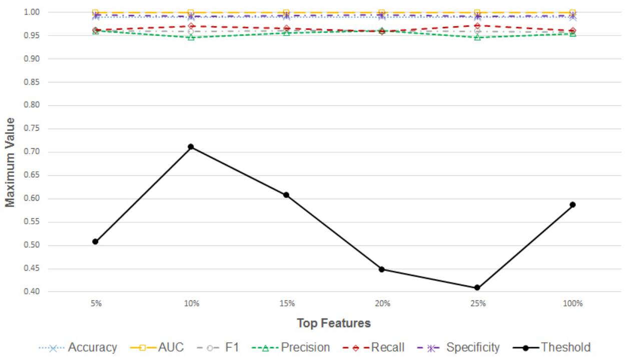

Fig. 3. Compare the performance of the final models with maximum AUC value on unseen data.

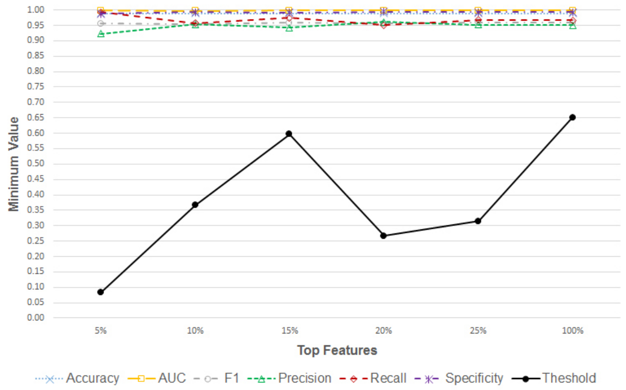

Fig. 4. Compare the performance of the final models with minimum AUC value on unseen data.

other proposed models in the literature. Nonetheless, when comping with the model proposed in [11] our model achieves converged results when trained using the top 5% features (i.e. accuracy: 99.01%, F1-score: 96.10%, precision: 96.01%, and recall: 96.20%) with a 50.78% threshold.

TABLE II
COMPARE THE PERFORMANCE OF THE PROPOSED DL MODEL WITH OTHER PROPOSED MODELS IN THE LITERATURE.

Algorithm	Accuracy (%)	FAR (%)
Decision Tree [8]	85.56	15.78
Logistic Regression [8]	83.15	18.48
Nave Bayes [8]	82.07	18.56
Artificial Neural Network [8]	81.34	21.13
EM Clustering [8]	78.47	23.79
Ramp-KSVCR [13]	93.52	02.46
PSI-NetVisor [10]	94.54	02.81
CANID [11]	99.36	-
Deep Learning (Proposed)	**98.99**	**00.56**

VI. CONCLUSION

In this paper, a deep learning model based on a multilayer feedforward artificial neural network using backpropagation and stochastic gradient descent method has been evaluated as a binomial classifier for Network Intrusion Detection System. The evaluation results show outstanding performance with extremely high accuracy (i.e. 98.99%) and very low false alarm rate (i.e. 00.56%).

REFERENCES

[1] The zettabyte era: Trends and analysis. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[2] P. Brown. 20 billion connected internet of things devices in 2017, IHS markit says. [Online]. Available: http://electronics360.globalspec.com/article/8032/20-billion-connected-internet-of-things-devices-in-2017-ihs-markit-says

[3] B. Krebs. KrebsOnSecurity hit with record DDoS. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[4] D. Bonderud. Leaked mirai malware boosts IoT insecurity threat level. [Online]. Available: https://securityintelligence.com/news/leaked-mirai-malware-boosts-iot-insecurity-threat-level/

[5] B. B. Zarpelo, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," vol. 84, pp. 25–37. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300802

[6] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6.

[7] 1999 DARPA intrusion detection evaluation data set. [Online]. Available: https://www.ll.mit.edu/ideval/data/1999data.html

[8] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," vol. 25, no. 1, pp. 18–31. [Online]. Available: http://dx.doi.org/10.1080/19393555.2015.1125974

[9] B. Stewart, L. Rosa, L. A. Maglaras, T. J. Cruz, M. A. Ferrag, P. Simoes, and H. Janicke, "A novel intrusion detection mechanism for SCADA systems which automatically adapts to network topology changes," vol. "4", no. 10. [Online]. Available: http://eudl.eu/doi/10.4108/eai.1-2-2017.152155

[10] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "PSI-NetVisor: Program semantic aware intrusion detection at network and hypervisor layer in cloud," vol. 32, no. 4, pp. 2909–2921. [Online]. Available: http://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs169234

[11] M. M. Baig, M. M. Awais, and E.-S. M. El-Alfy, "A multiclass cascade of artificial neural network for network intrusion detection," vol. 32, no. 4, pp. 2875–2883. [Online]. Available: http://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs169230

[12] W. Haider, J. Hu, J. Slay, B. P. Turnbull, and Y. Xie, "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling," vol. 87, pp. 185–192. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517301273

[13] Seyed Mojtaba Hosseini Bamakan, H. Wang, and Y. Shi, "Ramp loss k-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem," vol. 126, pp. 113–126. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705117301314

[14] M. Idhammad, K. Afdel, and M. Belouch, "DoS detection method based on artificial neural networks," vol. 8, no. 4. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=8&Issue=4&Code=IJACSA&SerialNo=61

[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6.

[16] 1998 DARPA intrusion detection evaluation data set. [Online]. Available: https://www.ll.mit.edu/ideval/data/1998data.html

[17] T. D. Gedeon, "Data mining of inputs: analysing magnitude and functional measures," vol. 8, no. 2, pp. 209–218.

[18] D. Cook, "Chapter (8): Deep learning (neural nets)," in *Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI*. O'Reilly Media, Incorporated, pp. 251–252.

[19] A. Graves, “Chapter (3): Neural networks,” in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Science & Business Media, p. 17, google-Books-ID: 4UauNDGQWN4C.

[20] D. J. F. Wiley, “Chapter (5): Tarining deep prediction models,” in *R Deep Learning Essentials*. Packt Publishing Ltd, p. 98, google-Books-ID: U5njCwAAQBAJ.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” vol. 15, pp. 1929–1958. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

View publication stats