Computers, Materials & Continua
DOI:10.32604/cmc.2020.012432
Article

Tech Science Press

A Stacking-Based Deep Neural Network Approach for Effective Network Anomaly Detection

Lewis Nkenyereye[1], Bayu Adhi Tama[2] and Sunghoon Lim[3,*]

[1]Department of Computer and Information Security, Sejong University, Seoul, 05006, Korea
[2]Data Science Group, Institute for Basic Science (IBS), Daejeon, 34126, Korea
[3]Department of Industrial Engineering, Ulsan National Institute of Science and Technology, Ulsan, 44919, Korea
[*]Corresponding Author: Sunghoon Lim. Email: sunghoonlim@unist.ac.kr
Received: 30 June 2020; Accepted: 26 August 2020

Abstract: An anomaly-based intrusion detection system (A-IDS) provides a critical aspect in a modern computing infrastructure since new types of attacks can be discovered. It prevalently utilizes several machine learning algorithms (ML) for detecting and classifying network traffic. To date, lots of algorithms have been proposed to improve the detection performance of A-IDS, either using individual or ensemble learners. In particular, ensemble learners have shown remarkable performance over individual learners in many applications, including in cybersecurity domain. However, most existing works still suffer from unsatisfactory results due to improper ensemble design. The aim of this study is to emphasize the effectiveness of stacking ensemble-based model for A-IDS, where deep learning (e.g., deep neural network [DNN]) is used as base learner model. The effectiveness of the proposed model and base DNN model are benchmarked empirically in terms of several performance metrics, i.e., Matthew's correlation coefficient, accuracy, and false alarm rate. The results indicate that the proposed model is superior to the base DNN model as well as other existing ML algorithms found in the literature.

Keywords: Anomaly detection; deep neural network; intrusion detection system; stacking ensemble

1 Introduction

Intrusion detection system (IDS) has been an active research in the cybersecurity domain recently. It contributes a critical role to a modern computing infrastructure in repealing any malicious activities in the network. In addition, as a protection mechanism, an IDS is accountable for taking preventive action to overcome any malignant acts in the computer network. By examining network access logs, audit trails, and other security-relevant information within an organization, an IDS detects and blocks attack without human intervention [1].

An IDS is typically split into two main techniques, i.e., anomaly and misuse. The differences lie in the number of attack classes to be predicted. An anomaly-based IDS (A-IDS) attempts to solve a binary classification problem, where the classifier is trained so that it is able to distinguish anomaly traffic from normal traffic. Since the trained model is only capable in handling two classes, a new type of attack can

be discovered by A-IDS. Apart from this merit, this technique always suffers from high false alarm rate (FAR), thus bringing the network into vulnerable state. In contrast to A-IDS, a misuse-based IDS (M-IDS) attempts to solve multiclass classification problem, where a future attack could be detected by comparing it with some known attacks signatures stored in knowledge-based system. It results shows a lower FAR, however, unknown attacks cannot be easily detected [2].

Owing to the fact that A-IDS are powerful to find new types of attacks, it is more adopted in IDS research. Even though it offers a small improvement in the performance, such A-IDS would be a significant asset for an organization. For instance, it could help an organization to get rid of successful attack, e.g., service inaccessibility and performance breakdown, that might result into huge financial loss. However, maintaining a lower FAR while increasing the detection accuracy is also a challenging task. This trade-off is prevalently solved using the combination of feature selection and classification algorithms. Feature selection or feature importance methods are crucial as some irrelevant features might contribute to degrading classifier's performance.

To develop an A-IDS that is able to learn anomaly or normal pattern within the network, a classification algorithm is trained using publicly available network traffic log datasets such as NSL-KDD [3], UNSW-NB-15 [4], and more recently, CICIDS-2017 [5]. These datasets are commonly used in the current literature for benchmarking the proposed A-IDS model. To improve an A-IDS, a considerable number of classification algorithms have been carried out, ranging from shallow machine learning models to deep neural network (DNN) models [6,7]. Besides, some ensemble learners are also taken into account due to their performance advantages over individual classification algorithms [8,9].

In an ensemble learner, multiple classification algorithms are trained to predict the same problem. Over the last few decades, ensemble learners have shown remarkable performance in various applications, including cybersecurity field. However, there still exist several research challenges while utilizing ensemble learners. For instance, the selection of the mixture technique for combining the base learner's predictions and the multifariousness of classifiers in the wild. Thus, this study focuses on the development of an A-IDS technique using stacking-based deep neural network (DNN). Stacking is chosen due to its flexibility in combining multiple classifiers in heterogeneous way. The contributions of this paper lie in two different angles: (i) An ensemble approach of DNN is proposed, instead of just using DNN as an individual classifier; and (ii) A two-step significance test is employed to prove the effectiveness of the proposed model over individual model.

2 Related Work

In this section, a brief review of existing A-IDS techniques is discussed. Since A-IDS is an active research field, we only provide the proposed techniques published in the last two years, e.g., 2018 and 2019 and studies that employed at least one classifier ensemble in their experiment. This is also to show the position of this paper in comparison with other state- of-the-art techniques. We summarize and classify the trend of A-IDS research in Tab. 1. Interested readers might refer to recently survey papers [10–14].

Table 1: Classification of A-IDS w.r.t detection approaches and other important categories

Year	Ensemble approaches	Feature selection	Validation technique	Dataset	Author(s)
2018	Bagging	Gain ratio	Hold-out	NSL-KDD	Pham et al. [15]
2018	Random forest	–	10cv	NSL-KDD, Kyoto+	Al-Jarrah et al. [16]
2018	Majority voting	Information gain	Hold-out	NSL-KDD	Aljawarneh et al. [17]

Table 1 (continued).

Year	Ensemble approaches	Feature selection	Validation technique	Dataset	Author(s)
2018	Random forest	–	Hold-out	KDD Cup99	Vigneswaran et al. [18]
2018	Random forest	–	Hold-out	ISCX 2012	Injadat et al. [19]
2018	Random forest	–	Hold-out	UNSW-NB15	Belouch et al. [20]
2018	Random forest	–	Hold-out	ISCX 2012	Ahmad et al. [21]
2018	Gradient boosting tree	–	Hold-out	NSL-KDD, UNSW-NB15	Zhou et al. [22]
2018	Majority voting	Information entropy	Hold-out	Kyoto 2006+	Zaman et al. [23]
2019	Weighted majority voting	Chi-square	Hold-out, cv	NSL-KDD	Thaseen et al. [24]
2019	Two-stage ensemble	Correlation-based	Hold-out	NSL-KDD, UNSW-NB15	Tama et al. [6]
2019	Boosted tree	–	5 cv, 10 cv, hold-out	Private	Verma et al. [25]
2019	Bagging, boosting, stacking	–	10 cv	Synthetic	Subudhi et al. [26]
2019	Adaboost	Artificial bee colony	Hold-out	NSL-KDD, ISCX 2012	Mazini et al. [27]
2019	Gradient boosting machine	–	Hold-out, 10cv	NSL-KDD, UNSW-NB15, Wi-Fi intrusion	Tama et al. [9]

3 Material and Methods

This section describes several publicly available datasets used in the experiment. The remaining part of this section details the proposed A-IDS model.

3.1 Intrusion Datasets

The following datasets are very common in IDS community. NSL-KDD and UNSW-NB15 are considered for network packets-based analysis, while CICIDS 2017 is used for Web traffic-based analysis. The datasets are described chronologically as follows.

NSL-KDD [3]:

It is an improved version of long-standing intrusion dataset, called KDD Cup 99. Unlike its predecessor, NSL-KDD possesses no redundant samples, providing more realistic and reliable dataset while applying machine learning algorithm to develop an IDS model. A number of training samples (e.g., 125,973 instances) are used for creating the classification model, where the number of samples representing anomaly and normal class is 67,343 and 58,630 samples, respectively. In addition, for the sake of the evaluation procedure, an independent testing set (e.g., KDDTest+) is taking into consideration. The testing set consists of 22,544 instances.

UNSW-NB15 [4]:

It was built by generating real-life normal network packets as well as synthetic attacks using IXIA PerfectStorm tool. A training set consisting of 37,000 normal and 45,332 attack samples is used in our experiment. In addition, an independent test set, called UNSW-NB15 test (e.g., 175,341 samples) is also used for evaluating the proposed classification model. The number of input feature is 42 with 1 class label attribute.

CICIDS 2017 [5]:

B-profile system was used to generate realistic benign background traffic. Moreover, several network protocols such as HTTP, HTTPS, FTP, SSH, etc. were also taken into consideration, providing a complete network traffic dataset with a diverse attack profiles. There are 78 input features, while the number of benign and malicious samples is 168,186 and 2,180 samples, respectively. Since an independent dataset is not provided, we simply apply a train-test split with a ratio of 80% and 20% for training and testing set, respectively.

3.2 Proposed Method

The idea of our proposed model is briefly presented in the following subsections:

3.2.1 Deep Neural Network

Since the advent of artificial neural networks (ANNs) that mimic human thought, deep neural networks (DNNs) (e.g., deep learning) is one of the most effective tools in comparison with other machine learning algorithms in the wild. DNN is built based on the initial ANN architecture that has a multilayer structure, activation and optimization functions. It is highly recognized due to the advancement of computing hardware. Fig. 1 denotes a base DNN model. The base DNN architecture consists of one input layer, three hidden layer, and one output layer. All features are fed into input layer, in which some nonlinear operations are then performed to provide the final class prediction in the output layer.

3.2.2 Stacking Ensemble

Stacking was firstly introduced by the researcher in [28]. Despite the fact that it was originally invented by Wolpert, the present-day stacking that uses internal k-fold cross-validation was Breiman's contribution. Our proposed stacking-based deep learning model is detailed in Algorithm 1. In this study, five different DNN base models are taken into account. The goal of using such different models is to maximize the diversity of the ensemble. This is quite essential since without diversity, an ensemble is deemed to be unsuccessful as it is [29]. Diversity can be achieved in several ways: By using different base learners for constructing the ensemble (e.g., heterogeneous) and by using different training set. This paper is emphasized on the first strategy, specifically, different learning parameters of each base DNN are used. Moreover, a gradient boosting machine learning (GBM) [30] is considered as meta-learning classifier.

4 Results and Discussion

In this section, the experimental results of staking-based deep neural network for an A-IDS is described. First of all, learning parameters of each base DNN model are specified in Tab. 2. As mentioned previously, by specifying different learning parameters, our objective is to maximize the diversity and we expect that an improved final ensemble prediction could be obtained. To evaluate the proposed model and baseline models, a Matthews correlation coefficient (MCC) is considered. The metric is found to be meaningful to measure the performance of classifier applied to imbalance datasets. Furthermore, two other metrics, i.e., accuracy and false alarm rate (FPR) that are commonly used in IDS research are also taken into consideration. Those three performance measures can be obtained as follows Fig. 2:

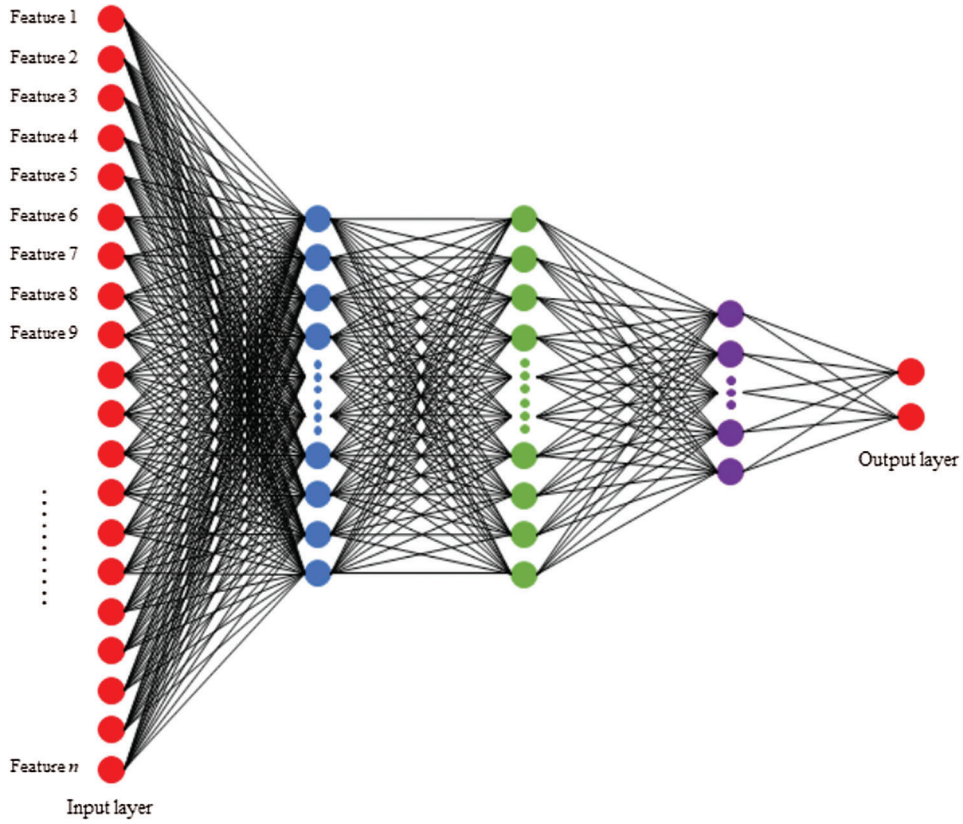

Figure 1: Architecture of a base DNN model

Algorithm 1: Proposed stacking-based deep neural network for A-IDS

Setup:
Intrusion dataset D with m instances and n features, which is denoted as input matrix X and response matrix Y

$$m\left\{\overbrace{\left[\ X\ \right]}^{n}\left[\ Y\ \right]\right.$$

Determine L DNN base models, along with their optimal hyperparameters. Determine the level-1 classifier, e.g., gradient boosting machine (GBM).

Train the ensemble:
Train each of the L base model on the training set.
Implement *stratified* 5-fold cross-validation on each DNN base model.
Gather the prediction results, S_1, S_2, ..., S_L
Gather M prediction values from L base models and generate a matrix M x L, which is later called as matrix W
Along with original response vector Y, train level-1 classifier: $Y = f(W)$

(Continued)

Algorithm 1 (continued).

$$m\left\{\left[S_1\right]\ldots\left[S_L\right]\left[Y\right]\right. \rightarrow m\left\{\overbrace{\left[W\right]}^{L}\left[Y\right]\right.$$

Prediction of new test

Get the prediction results from base models and feed into level-1 classifiers.

Get the final ensemble prediction, O_f

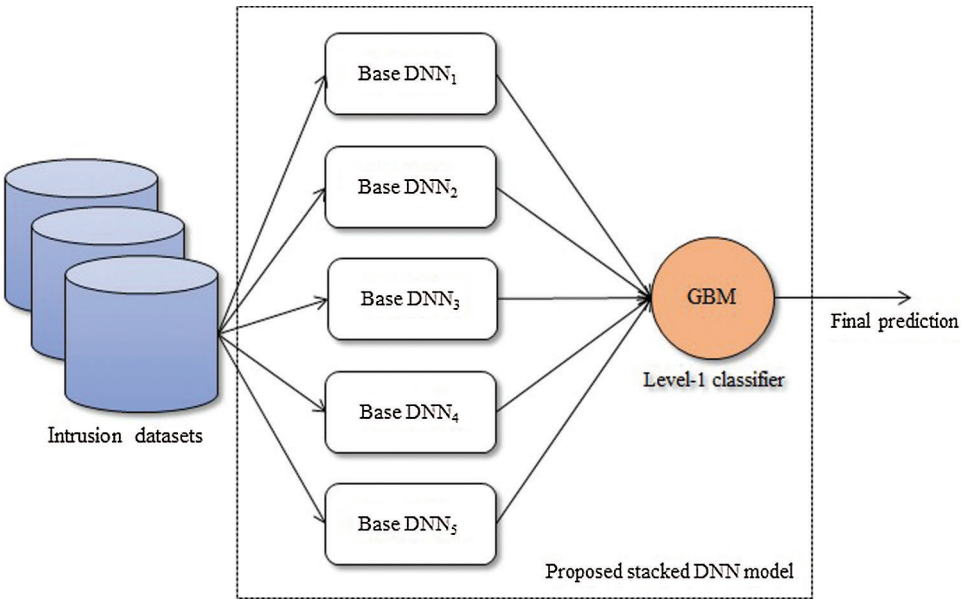

Figure 2: Proposed stacking-based DNN for anomaly-based IDS

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \in \{-1, 1\} \tag{1}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \in \{0, 1\} \tag{2}$$

$$FPR = \frac{FP}{FP+TN} \in \{0, 1\} \tag{3}$$

A deep learning framework, i.e., H_2O was utilized for running classification task. All codes were implemented in R on a machine with Linux operating system, 32 GB memory, and Intel Xeon processor. First of all, the performance of all classifiers with respect to MCC metric are presented in Fig. 3. It is clear that for all IDS datasets, the proposed stacking-based DNN outperforms all baseline models, except for UNSW-NB15. Using NSL-KDD, the proposed model (MCC = 0.7994) has achieved better than

DNN1 (MCC = 0.7189), DNN2 (MCC = 0.7737), DNN3 (MCC = 0.6893), DNN4 (MCC = 0.6724), and DNN5 (MCC = 0.6675). Similarly, the proposed model has a significant improvement over the baseline models when it is applied to CICIDS 2017. Tab. 3 compares relative performance between the proposed model and baseline models.

Table 2: Learning parameters for each DNN base model

Learning parameter	DNN_1	DNN_2	DNN_3	DNN_4	DNN_5
Activation	Rectifier with Dropout	Rectifier with Dropout	Maxout with Dropout	Maxout with Dropout	Tanh with Dropout
Hidden	(50,50,50)	(100,100,100)	(200,200,200)	(300,300,300)	(500,500,500)
Epochs	100	100	100	100	100
L1	0.00001	0.0001	0.00001	0.0001	0.00001
L2	0.0001	0.00001	0.0001	0.00001	0.0001
Rate	0.005	0.005	0.005	0.005	0.005
Rate annealing	1E-8	1E-6	1E-8	1E-6	1E-8
Rho	0.99	0.99	0.99	0.99	0.99
Epsilon	1e-10	1e-10	1e-10	1e-10	1e-10
Adaptive rate	Yes	Yes	Yes	Yes	Yes

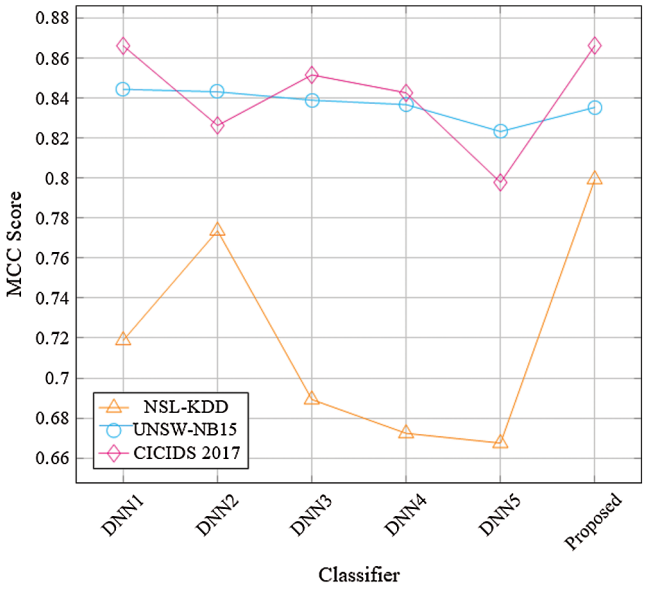

Figure 3: Performance of stacking-based DNN and baseline models w.r.t MCC score

Table 3: Relative performance differences (%) between the proposed model and the base- lines. For example, the proposed model performance on NSL-KDD is 11.20% higher than DNN_1

Classifier I	Classifier II	NSL-KDD	UNSW-NB15	CICIDS 2017
Proposed	DNN_1	11.20	−1.07	0.02
	DNN_2	3.32	−0.93	4.84
	DNN_3	15.97	−0.43	1.73
	DNN_4	18.89	−0.17	2.81
	DNN_5	19.76	1.47	8.59

Table 4: Results of all pair-wise comparisons using Quade *post hoc* test (bold indicates significance)

	DNN_1	DNN_2	DNN_3	DNN_4	DNN_5	Proposed
DNN_1	n/a	0.798	0.655	**0.406**	**0.160**	0.798
DNN_2	0.798	n/a	0.848	0.565	**0.250**	0.609
DNN_3	0.655	0.848	n/a	0.701	**0.338**	**0.482**
DNN_4	**0.406**	0.565	0.701	n/a	0.565	**0.277**
DNN_5	**0.160**	**0.250**	**0.338**	0.565	n/a	**0.097**
Proposed	0.798	0.609	**0.482**	**0.277**	**0.097**	n/a

Table 5: Performance comparison between the proposed model and some state-of-the-art techniques (bold indicates best value)

Study	Year	Accuracy (%)	FPR (%)
Performance comparison on testing set, i.e., KDDTest+			
Proposed	**2020**	**89.97**	**1.32**
Two-stage ensemble [6]	2019	85.80	11.7
SVM [31]	2019	81.58	n/a
Bagging (C4.5) [15]	2018	84.25	2.79
Two-tier classifier [32]	2017	83.24	4.80
Performance comparison on testing set, i.e., UNSW-NB15$_{test}$			
Proposed	**2020**	**92.83**	8.91
Stacked ensemble [8]	2020	92.45	11.3
GBM [9]	2019	91.31	8.60
Two-stage ensemble [6]	2019	91.27	8.90
Two-stage classifier [33]	2018	85.78	15.64
Decision tree [34]	2017	81.42	**6.39**

Table 5 (continued).

Study	Year	Accuracy (%)	FPR (%)
Performance comparison on CICIDS 2017			
Proposed	**2020**	**99.65**	1.67
Ensemble learning [35]	2020	96.80	0.03
Deep learning [36]	2019	96.30	n/a
Deep RNN [37]	2019	89.00	n/a
k-NN [38]	2019	99.46	n/a
Random forest [8]	2018	99.40	**0.01**

For the sake of completeness, an empirical comparison using statistical significance tests is also provided in this section. For this purpose, a two-fold Quade-Quade *post hoc* test [36] is employed. Quade test is deemed to be more powerful than other tests when comparing five or less different classifiers. The two or more classifiers are significantly different if p-value is less than a threshold (0.5 in our case). First of all, an omnibus test using Quade test yields p-value = 0.067, with degree of freedom, d f = 5 is conducted. Therefore, it can be inferred that at least one classifier has performed differently than others. Since the test demonstrates its contribution, Quade *post hoc* test is carried out. Tab. 4 exhibits the p-values of all pair-wise comparisons using Quade *post hoc* test. It conveys an information that the proposed model is statistically significant than DNN3, DNN4, and DNN5. Finally, in order to ensure the comprehensiveness of this study, it is compulsory to benchmark the proposed model and other existing approaches. Tab. 5 depicts such a fairer comparison with the state-of-the-arts in terms of accuracy and FPR. It proves that the proposed model is obviously superior to every other approach published in some major outlets.

5 Conclusion

Anomaly detection in computer network has always been an active research in cybersecurity domain. Many studies have been implemented to address network traffic logs as a binary classification problem. In the current literature, there is no available stacking-based deep neural network approach applied to anomaly-based IDS thus far. In this study, a stacking-based deep neural network is designed for anomaly detection, coping with a two-class detection problem, i.e., normal and malicious. To evaluate the effectiveness of the proposed model, the experiments were performed on three different intrusion datasets such as NSL-KDD, UNSW- NB15, and CICIDS 2017. Experimental results demonstrate that the proposed model is a first-rate method for anomaly detection with a detection accuracy of 89.97%, 92/83%, and 99.65% when dealing with specified training sets of KDDTest+, UNSW-NB15test, and CICIDS 2017, respectively.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1F1A1059346). This work was supported by the 2020 Research Fund (Project No. 1.180090.01) of UNIST (Ulsan National Institute of Science and Technology).

Conflict of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Gupta, S. Tanwar, S. Tyagi and N. Kumar, "Machine learning models for secure data analytics: A taxonomy and threat model," *Computer Communications*, vol. 153, pp. 406–440, 2020.

[2] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," in *Proc. IEEE ICoDSE*, Palembang, Indonesia, pp. 1–6, 2017.

[3] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE CISDA*, Ottawa, Canada, pp. 1–6, 2009.

[4] 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," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[5] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, Funchal, Portugal, pp. 108–116, 2018.

[6] B. A. Tama, M. Comuzzi and K. H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[7] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[8] B. A. Tama, L. Nkenyereye, S. R. Islam and K. S. Kwak, "An enhanced anomaly detection in web traffic using a stack of classifier ensemble," *IEEE Access*, vol. 8, pp. 24120–24134, 2020.

[9] B. A. Tama and K. H. Rhee, "An in-depth experimental study of anomaly detection using gradient boosted machine," *Neural Computing and Applications*, vol. 31, no. 4, pp. 955–965, 2019.

[10] A. Khraisat, I. Gondal, P. Vamplew and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 384, 2019.

[11] R. Chapaneri and S. Shah, "A comprehensive survey of machine learning-based network intrusion detection," in *Smart Intelligent Computing and Applications, Smart Innovation, Systems and Technologies*, Singapore: Springer, pp. 345–356, 2019.

[12] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.

[13] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2019.

[14] N. Moustafa, J. Hu and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33–55, 2019.

[15] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proc. ACSW*, Brisbane, Australia, pp. 1–6, 2018.

[16] O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat and M. Al-Qutayri, "Semi-supervised multi-layered clustering model for intrusion detection," *Digital Communications and Networks*, vol. 4, no. 4, pp. 277–286, 2018.

[17] S. Aljawarneh, M. Aldwairi and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[18] K. R. Vigneswaran, R. Vinayakumar, K. Soman and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *Proc. IEEE ICCCNT*, Bengaluru, India, pp. 1–6, 2018.

[19] M. Injadat, F. Salo, A. B. Nassif, A. Essex and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection," in *Proc. IEEE GLOBECOM*, Abu Dhabi, UAE, pp. 1–6, 2018.

[20] M. Belouch, S. El Hadaj and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using Apache Spark," *Procedia Computer Science*, vol. 127, pp. 1–6, 2018.

[21] I. Ahmad, M. Basheri, M. J. Iqbal and A. Rahim, "Performance comparison of sup- port vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.

[22] Y. Zhou, M. Han, L. Liu, J. S. He and Y. Wang, "Deep learning approach for cyber- attack detection," in *IEEE INFOCOM*, Honolulu, USA, pp. 262–267, 2018.

[23] M. Zaman and C.H. Lung, "Evaluation of machine learning techniques for network intrusion detection," in *Proc. IEEE IFIP*, Taipei, Taiwan, pp. 1–5, 2018.

[24] I. S. Thaseen, C. A. Kumar and A. Ahmad, "Integrated intrusion detection model using chisquare feature selection and ensemble of classifiers," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3357–3368, 2019.

[25] A. Verma and V. Ranga, "Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things," in *Proc. IEEE IoT-SIU*, Ghaziabad, India, pp. 1–6, 2019.

[26] S. Subudhi and S. Panigrahi, "Application of optics and ensemble learning for database intrusion detection," *Journal of King Saud University—Computer and Information Sciences*, pp. 1–10, 2019.

[27] M. Mazini, B. Shirazi and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and adaboost algorithms," *Journal of King Saud University—Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, 2019.

[28] L. Breiman, "Stacked regressions," *Machine Learning*, vol. 24, no. 1, pp. 49–64, 1996.

[29] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[30] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[31] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimization," *Neural Computing and Applications*, vol. 32, no. 10, pp. 6125–6137, 2020.

[32] H. H. Pajouh, G. Dastghaibyfard and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *Journal of Intelligent Information Systems*, vol. 48, no. 1, pp. 61–74, 2017.

[33] W. Zong, Y. W. Chow and W. Susilo, "A two-stage classifier approach for network intrusion detection," in *Proc. ISPEC*, Tokyo, Japan, pp. 329–340, 2018.

[34] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[35] Y. Zhou, G. Cheng, S. Jiang and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, pp. 107247, 2020.

[36] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat *et al.,* "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[37] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi and M. Ghogho, "Intrusion detection in SDN-based networks: Deep recurrent neural network approach," in *Deep Learning Applications for Cyber Security*, Cham, Switzerland: Springer, pp. 175–195, 2019.

[38] M. Alrowaily, F. Alenezi and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," in *Proc. SpaCCS*, Georgia, USA, pp. 277–288, 2019.