The ISC Int'l Journal of Information Security
ISeCure

November 2020, Volume 12, Number 3 (pp. 37–44)
http://www.isecure-journal.org

Selected Paper at the ICCMIT'20 in Athens, Greece

Anomaly-Based Network Intrusion Detection Using Bidirectional Long Short Term Memory and Convolutional Neural Network**

Isra Al-Turaiki *, Najwa Altwaijry, Abeer Agil, Haya Aljodhi, Sara Alharbi, and Lina Alqassem
Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

ARTICLE INFO.

Keywords:
Network Intrusion Detection Systems, Deep Learning, Long Short Term Memory, LSTM, Convolutional Neural Network, CNN

Type: Research Article

doi: 10.22042/isecure.2021.271076.624

ABSTRACT

With present-day technological advancements, the number of devices connected to the Internet has increased dramatically. Cyber-security attacks are increasingly becoming a threat to individuals and organizations. Contemporary security frameworks incorporate network intrusion detection systems(NIDS). These systems are an essential component for ensuring the security of computer networks against attacks. In this paper, two deep learning architectures are proposed for both binary and multi-class classification of network attacks. The models, CNN-IDS and LSTM-IDS, are based on convolutional neural network and long short term memory architectures, respectively. The models are evaluated using the well-known NSL-KDD dataset. The performance is measured in terms of accuracy, precision, recall, and F-measure. Experimental results show that the models achieve good performance in terms of accuracy and recall.

1 Introduction

Network intrusion detection systems(NIDS) play an important role in the security of information systems. The main purpose of a NIDS is to identify abnormal behavior and attacks in a network. Detection and prevention of these attacks is an important topic, as the attacks have major effects on a large number of computer systems in various corporations. In order to protect a system from threats, NIDS monitor and analyze network traffic. When threats are found, based on their severity, the system may notify administrators, or restrict the source IP address from accessing the network.

For the past three decades, NIDS has been a very active field of research. NIDS are either *signature-based* (SNIDS) *or anomaly detection-based* (ADNIDS). In signature-based NIDS, network traffic is matched against predefined patterns that are stored in a data file. In anomaly-based NIDS, attacks are identified as patterns that deviate from normal behavior within the network [1]. Research has demonstrated the effectiveness of ADNIDS, when compared to SNIDS, in detecting attacks that have not previously been observed.

* Corresponding author.
**The ICCMIT'20 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: ialturaiki@ksu.edu.sa, ntwaijry@ksu.edu.sa, 437203947@student.ksu.edu.sa, 436925138@student.ksu.edu.sa, 436925159@student.ksu.edu.sa, 436204540@student.ksu.edu.sa

ISSN: 2008-2045 © 2020 ISC. All rights reserved.

The problem of network intrusion detection can be formulated as a two-class or multi-class classification problem. Many ADNIDS have been proposed in the literature using different machine learning algorithms, such as: *random forests* (RF), *self-organized maps* (SOM), *support vector machines* (SVM), and *artificial neural networks* (ANN). Traditional machine learning algorithms require the availability of engineered features beforehand.

Deep learning is a growing field of machine learning that has been applied successfully to solve various real world problems [2]. Deep learning algorithms overcome limitations of traditional machine learning algorithms by automatically capturing features using unsupervised or semi-supervised feature learning algorithms [3].

In this research, two ADNIDS models are proposed for the classification of network attacks. The models are based on *bidirectional long short term memory* (B-LSTM) and *convolutional neural network* (CNN) architectures, and are used to classify binary and multi-class network attacks. The NSL-KDD dataset is used to evaluate the performance of the two models on both classification tasks. The obtained results are compared with similar deep-learning approaches and state-of-the-art classification models.

The rest of this paper is organized as follows. Section 2 discusses previous studies in the area of anomaly detection using machine learning techniques. Section 3 describes the dataset. Section 4 presents the proposed ADNIDS models. Section 5 outlines experimental settings and performance measures, then presents and discusses experimental results. Conclusions and planning for future work are presented in Section 6. Finally, Table 1 shows a list of acronyms used throughout this manuscript.

2 Related Work

Many intrusion detection models have been introduced in the literature. This section discusses several intrusion detection models using data mining, machine learning, and deep learning.

Lin *et al.* [4] proposed an anomaly-based intrusion detection system combining an SVM, feature selection, and decision rules. SVM and simulated annealing (SA) were used for feature selection with the goal of improving accuracy values. SA is also used for obtaining decision rules for new attacks in addition to the decision tree. They tested the system on the KDD-Cup 1999 dataset and achieved an accuracy of 99.96%.

Koc *et al.* [5] investigated the potential of the *Hidden Nave Bayes* (HNB) model as a solution to the intrusion detection problem. The authors trained HNB models to detect four basic attack classes: probe, DoS, U2R and R2L. Experimental results using the KDD'99 dataset show that the HNB model with proportional k-Interval discretization and INTERACT feature selection methods outperforms the compared methods in all three performance categories. In addition, the HNB model has better overall performance in detecting DoS attacks than the traditional and improved Nave Bayes methods.

Table 1. Nomenclature of abbreviations

Acronyms	Definition
ADNIDS	Anomaly detection-based Intrusion Detection System
ANN	Artificial neural networks
CNN	Convolutional neural network
DNN	Deep Neural Network
LSTM	Long short-term memory
NIDS	Network Intrusion Detection System
NSL-KDD	Network Security Laboratory
ReLU	Rectified Linear Unit
RF	Random forests
RNN	Recurrent Neural Networks
SNIDS	Signature-based Network Intrusion Detection System
SOM	Self-organized maps
STL	Self-taught learning
SVM	Support vector machines

Thaseen and Kumar [6] proposed a multi-class intrusion detection model using *support vector machine* (SVM) and chi-square feature selection. Experiments were conducted on the NSL-KDD dataset. The proposed model shows higher accuracy than KNN and CANN and also outperforms binary class SVM techniques. Al-Yaseen *et al.* [7] used k-means clustering in order to reduce the disadvantages of the KDD'99 dataset. The authors proposed a hybrid intrusion detection model of SVM and extreme learning machine to improve the accuracy of attack detection. Using KDD'99, the proposed model achieved an accuracy of 95.75%.

Deep learning algorithms have also been used to solve the intrusion detection classification problem. Yin *et al.* [8] employed a Recurrent Neural Network (RNN). Experimental results on the NSL-KDD dataset showed the accuracy values were 83.28% and 81.29% for binary and multi-class classification, respectively. Compared with other machine learning algorithms, such as Nave Bayes and RF,the proposed RNN-IDSper forms better in both binary and multi-class classification of network attacks.

A study was conducted by Zhipeng *et al.* [9] to use the two CNN models ResNet 50 and GoogLeNet for the binary classification problem of network attacks. The features of the NSL-KDD dataset were converted into binary vectors and then the data was converted into an image form. Several experiments were carried out using the NSL-KDD dataset. ResNet 50 obtained an accuracy value of 79.14%, whereas GoogLeNet achieved 77.04% accuracy. From the ADNIDS literature, it is observed that the accuracy of the proposed models is usually better in the binary classification problem using KDD'99 [10] dataset. The results presented in the literature on KDD'99 may be regarded as biased, because KDD'99 has an inherent bias in its training and testing sets [10, 11].

Altwaijry *et al.* [12] proposed a DNN-based intrusion detection system. The DNN was built using 4 hidden fully connected layers and tested on the NSL-KDD dataset. Experimental results showed that the model achieved an accuracy of 84.70% and 77.55% in the binary and multi-class classification problems, respectively. They also showed that the proposed DNN-based IDS performs better than shallow learning algorithms, such as Nave Bayes, J48, Bagging, and Adaboost.

Recently, a survey on deep learning approaches for anomaly-based intrusion detection was published by Aldaweesh *et al.* [13]. The authors presented a taxonomy of deep learning based IDSs in terms of: input data, detection, deployment, and evaluation methods. According to the survey, deep learning in intrusion detection is mostly used for the task of feature learning. They concluded that more efforts are needed to improve the accuracy of the current state-of-the-art IDSs.

3 Dataset Description

It is known that the KDD'99 dataset contains a large number of redundant records in both training and test datasets [10]. Thus, the classification results are biased towards the more frequent records. In addition, testing accuracy increases whenever these same records appear in the test set. For this reason, in this work, the NSL-KDD benchmark dataset is used. It is regarded as an improved and reduced version of the original KDD'99 dataset [10], and is used to assess the performance of intrusion detection systems. NSL-KDD solves the issues in KDD'99 by removing redundant records [10, 14]. There are 125,973 records in KDDTrain^{+} and 22,544 records in KDDTest^{+} [10]. KDDTest^{-21} contains 11,850 records, where these records were not classified correctly by all 21 classifiers [10]. The dataset contains the following types of network attacks:

- Denial of Service (DoS): where an attacker floods the server with requests such that it becomes unable to handle valid requests or denies legal users access to a machine.
- Probing Attack: an attacker probes the network by searching it for open ports, to discover which ports are up.
- User to Root Attack (U2R): an attacker gains access as a normal user account on the system, then attempts to exploit a vulnerability to gain root access.
- Remote to Local Attack (R2L): an attacker tries to gain access to a computer in order to gain access to the network.

The records in the dataset come in five classes, which are: normal, DoS, Probe, R2L, and U2R. The last four labels correspond to four types of attacks. Table 2 shows the number of records and attribute names for each of the different attack classes in the NSL-KDD dataset.

Records in the NSL-KDD dataset are described using 41 features. The features fall into three groups: basic features which are derived from TCP-IP connections, traffic features which are collected from window intervals or the number of connections, and content features which are taken from the application layer data of connections.

Table 2. The number of records and the name of attributes in the NSL-KDD dataset

Attack class	Training set	Training attributes	Testing set	Additional testing attributes
DOS	45927	back, land, teardrop, neptune, pod, smurf	7458	udpstorm, apache2, processtable, worm, mailbomb
Probing	11656	ipsweep, nmap, portsweep, satan	2421	mscan, saint mscan, saint
User to Root Attack (U2R)	52	loadmodule, buffer-overflow, perl, rootkit	200	sqlattack, xterm, ps
Remote to Local Attack (R2L)	995	fpt-write, guess-passwd, imap, multihop, phf, warezclient, warezmaster	2754	xlock, xsnoop, snmpguess, snmpgetattack, httptunnel, sendmail, named

4 Methodology

This paper presents two models to solve the problem of Anomaly Detection in NIDS, as follows: [15] Bidirectional Long Short Term Memory (B-LSTM), and [1] Convolutional Neural Network (CNN). The next sections present each model and the specifics of the data processing performed for each model.

4.1 Long Short Term Memory (LSTM-IDS) Model

4.1.1 Data Preprocessing

NSL-KDD contains symbolic features that are inappropriate for training, and so these features are converted into numeric ones using 1-to-N encoding, then features are normalized using min-max normalization.

4.1.2 B-LSTM Architecture

The bidirectional LSTM model (LSTM-IDS) architecture has a number of layers. The input layer is comprised of a tensor that contains the number of time-steps and the number of features. LSTM-IDS has 9 time-steps, and so the tensor is of shape 9×41. The hidden layers in the model are LSTM and fully connected layers. The first and second hidden layers are bidirectional LSTM layers, with 64 and 32 units, respectively, with the tanh activation function, as defined in Equation Equation 1.

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

Both these layers employ a dropout of 0.2, in addition to two types of regularization. The recurrent regularizer is Ridge Regularization (L2) with $\lambda = 0.001$, and the activity regularizer is Lasso (L1) with $\lambda = 0.0001$. Lasso regression adds the absolute value of magnitude of coefficients, which works well for feature selection, while ridge regressions adds the squared magnitude of coefficients, forcing all weights to be small and heavily penalizing outliers, and learning more complex data patterns.The regularizers were selected experimentally. These two layers are followed by two fully connected layers with 16 and 8 units, respectively, with Leaky ReLU as the activation function (see Equation 2) and $\alpha = 0.3$. Batch normalization is applied in both these layers.

$$Leaky - ReLU(x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases} \tag{2}$$

The final output layer is the classification layer, where for binary classification the Sigmoid activation function (Equation 3) is used, and for multi-class classification the Softmax function (Equation 4) is employed, which outputs the probability of each class, and the largest class probability is selected as the output class.

$$Sigmoid = \frac{1}{1+e^{-x}}, \tag{3}$$

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{4}$$

where x_i defines an input.
The loss function used is binary CrossEntropy for binary classification, and categorical CrossEntropy for multiclass classification. Both the binary LSTM-IDS and the multi-class LSTM-IDS use RMSprop [16] as the optimizer with default learning rate, decay and momentum.
The number of epochs used for training is 500, with a batch size of 64. However, if after 30 epochs, there is no improvement in the loss function, then training is halted. The training data is split into training and validation sets with a ratio of 80%-20%. All parameters are set experimentally on the validation set.

4.2 Convolutional Neural Network (CNN-IDS) Model

4.2.1 Data Preprocessing

For the CNN-IDS model, the non-numeric features protocol type, service and flag are converted to numeric features using one hot encoding. For example, the protocol type column has three symbolic values: tcp, udp, and icmp. These are converted into three columns, with the value [1,0,0] representing tcp, [0,1,0] representing udp, and [0,0,1] representing icmp. The 41 features are converted into 121 numeric features. These are represented as an 11 × 11 × 1 matrix.
Features are normalized using min-max scaling. A sample of the 11 × 11 input matrix is shown in Figure 1.

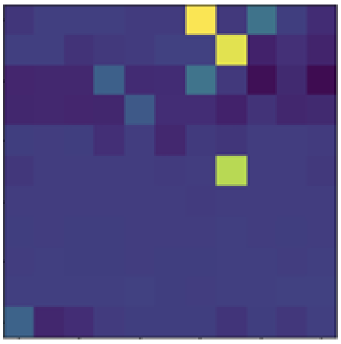

Figure 1. An example of an 11×11 input matrix

4.2.2 CNN Architecture

The CNN-IDS model input layer is an 11 × 11 × 1 matrix, as described above. For the hidden layers, the CNN model has 3 convolutional layers, with 20, 30, and 64 feature maps, respectively. The kernel size for all layers is 3 × 3, and all layers implement Ridge (L2) regularization with $\lambda = 0.001$, and the ReLU activation function (Equation 5). In order to prevent the image from shrinking through the convolution operations, zero padding is used. These convolutional layers are followed by a max-pooling layer, with a pool size of (2,2). This is fed into two fully-connected layers with 50 and 20 units, respectively. Both fully-connected layers use the ReLU activation function (Equation 5).

$$ReLU(x) = \max(0, x) \tag{5}$$

Overfitting is reduced by employing dropout after the fully connected layers, with a value of 0.4. The output layer depends on the classification task. For binary classification, the Sigmoid activation function (Equation 3) is used, and for multi-class classification, the Softmax function (Equation 4) is used, which outputs the probabilities of each class, with the largest class probability being selected as the output class. The loss function used is binary CrossEntropy for binary classification, and categorical CrossEntropy for multiclass classification. Both the binary CNN-IDS and the multi-class CNN-IDS use Adam [17] as the optimizer with the default learning rate of 0.001. The number of epochs used for training is 50, with a batch size of 32. The training data is split into training and validation sets with a ratio of 80%-20%. All parameters are set experimentally on the validation set.

5 Expperimental Results

5.1 Experimental Settings

In this research, the proposed models are implemented using Tensorflow[15] 1.4.1, an open source machine learning library. All experiments were run on an Intel Core i73.4 GHz machine, with 64GB RAM, and NVIDIA TESLA K40. Two experiments are designed to study the performance of the proposed models, one for binary classification and the other for multi-class classification of network attacks.

5.2 Evaluation Metrics

The proposed models are evaluated using the following measures: accuracy, precision, detection rate, and F-measure.
Each measure is defined as follows:

- Accuracy is the percentage of records classified correctly.
- Precision is the percentage of records correctly classified as anomaly out of the total number of records classified as anomaly.
- Recall: also True Positive Rate or detection rate, the percentage of records correctly classified as anomaly out of the total number of anomaly records.
- F-measure is a measure that combines both precision and detection rate.

5.3 Results and Discussion

Two experiments are conducted in order to evaluate the performance of the proposed models for binary and multi-class classification of network attacks. Results are compared to similar approaches in the literature, as well as comparing the results with some state-of-the-art machine learning algorithms, in particular: nave Bayes, J48, Random Forest, Bagging, and Adaboost. Table 3 shows the performance measures for the binary classification models. In terms of accuracy, CNN-IDS outperforms LSTM-IDS. It also performs better than nave Bayes, J48, Random Forest, Bagging, Adaboost, ANN [18], and CNN [19]. It is comparable to RNN-IDS [8]. In DNN [2], the testing set is not explicitly stated, rendering a direct comparison difficult. STL-IDS [20] performs better than both the proposed models in terms of accuracy, most likely due to its use of a sparse autoencoder to capture the significant features in the dataset. BDNN [12] also outperforms the proposed models, as a result of its deeper architecture. Both models proposed in this paper will likely show improved performance if the number of fully-connected layers are increased. In terms of recall, CNN-IDS outperforms all models except DNN [2] and RNN-IDS [8]. DNN [2] does not report the test set, and the disparity between the accuracy and recall of RNN-IDS indicates it is identifying traffic as attacks when traffic is normal, inflating its recall metric.
Finally, in terms of F-measure, CNN-IDS outperforms all models except for STL-IDS [20]. CNN-IDS performs better than LSTM-IDS because the dataset does not represent time-series data, allowing the CNN architecture to better classify attacks.
For multi-class classification, as shown in Table 4, both models achieve the same accuracy and detection rates. They are better than nave Bayes, J48, Bagging, and Adaboost, and comparable to the other models. As in binary classification, STL-IDS [20] outperforms the proposed models. However, a look at both Tables 3 and 4 shows that STL-IDS [20] has high precision and low recall, indicating that it is identifying few attacks, but when it does identify an attack, then it is usually correct. Generally, a good system should have high precision and high recall. Both proposed LSTM-IDS 79% 84% 79% 79% KDDTest$^+$ models are more consistent in terms of precision and recall. They are able to detect more attacks than STL-IDS [20], which can be crucial to inform a network administrator in case of unauthorized network access. Presented next are the evaluation measures for each attack type. Figures 2 and 3 show the evaluation measures per attack type for CNN-IDS and LSTM-IDS, respectively. The

Table 3. The performance of the proposed models compared with other approaches for binary classification using the NSL-KDD dataset

Model	Accuracy	Precision	Recall	F-measure	Testing set
STL-IDS [20]	84.96%	96.23%	76.57%	85.28%	KDDTest$^+$
ANN [18]	81.20%	N/A	N/A	N/A	KDDTest$^+$
DNN [2]	88.39%	85.44%	95.95%	90.40%	Not reported
CNN (6 features) [19]	75.75%	83.00%	76.00%	75.00%	KDDTest$^+$
RNN-IDS [8]	83.28%	N/A	97.09%	N/A	KDDTest$^+$
BDNN [12]	84.70%	79.45%	87.00%	83.05%	KDDTest$^+$
nave Bayes	76.12%	92.38%	63.27%	75.10%	KDDTest$^+$
J48	81.53%	97.14%	69.61%	81.10%	KDDTest$^+$
Random Forest	80.45%	97.05%	67.72%	79.77%	KDDTest$^+$
Bagging	82.63%	91.87%	76.23%	83.32%	KDDTest$^+$
Adaboost	78.44%	95.28%	65.37%	77.54%	KDDTest$^+$
CNN-IDS	83%	85%	83.00%	83%	KDDTest$^+$
LSTM-IDS	79%	84%	79%	79%	KDDTest$^+$

Table 4. The performance of the proposed models compared with other approaches for multi-class classification using the NSL-KDD dataset

Model	Accuracy	Precision	Recall	F-measure	Testing set
STL-IDS [20]	80.48%	93.92%	68.28 %	79.078 %	KDDTest$^+$
ANN [18]	79.9%	N/A	N/A	N/A	KDDTest$^+$
DNN [2]	79.10%	83%	68%	75.76%	Not reported
RNN-IDS [8]	81.29%	N/A	97.09%	N/A	KDDTest$^+$
MDNN [12]	77.55%	81.23%	77.55%	75.43%	KDDTest$^+$
nave Bayes	72.73%	76.1%	72.7%	72.6%	KDDTest$^+$
J48	74.99%	79.6%	75.0%	71.1%	KDDTest$^+$
Random Forest	76.45%	82.1%	76.4%	72.5%	KDDTest$^+$
Bagging	74.83%	78.3%	74.8%	71.6%	KDDTest$^+$
Adaboost	66.43%	N/A	66.0%	N/A	KDDTest$^+$
CNN-IDS	76%	80%	76%	73%	KDDTest$^+$
LSTM-IDS	76%	81%	76%	73%	KDDTest$^+$

recall of DoS attacks in both models is better than the other three attack types. However, the best recall is achieved for the *normal traffic*, because *normal* is the highest class represented in the multi-class case, with just over half the total number of records. Achieving good results in the multi-class problem is difficult on the NSL-KDD dataset. From the confusion matrices in Table 5 and Table 6, it is observed that the models do not perform well in detecting the U2R attack.

This is because of the severe class imbalance problem, with the U2R class representing 0.04% of the dataset and R2L class representing less than 8% of the dataset. Using dataset balancing techniques before model training will surely improve the performance of the proposed models.

The performance of the proposed models on the binary classification task is superior to the performance on the multi-class classification task. This is consistent with results in the literature. The results of the present study suggest that these models may be generalized to classify traffic in different types of networks. It has been demonstrated by Lopez-Martin [21] that deep learning architectures such as CNN and RNN can be used in classifying IoT networks traffic.

Table 5. The confusion matrix for the CNN model

		Predicted				
		Dos	Normal	Prob	R2L	U2R
True	Dos	5904	1482	72	2	0
	Normal	63	9025	616	7	0
	Probe	159	633	1589	40	0
	R2L	3	2454	101	327	0
	U2R	1	58	1	7	0

Table 6. The confusion matrix for LSTM model

		Predicted				
		Dos	Normal	Prob	R2L	U2R
True	Dos	5740	1641	75	0	0
	Normal	52	9433	216	5	2
	Probe	161	633	1625	1	0
	R2L	0	2684	44	155	2
	U2R	0	50	0	1	16

In addition, Stacked autoencoder (SAE) and convolutional neural network (CNN) architectures have been used for the tasks of traffic characterization and application identification [22]. Deep learning architectures have also been studied for mobile traffic classification which can deal with encrypted traffic [23, 24].

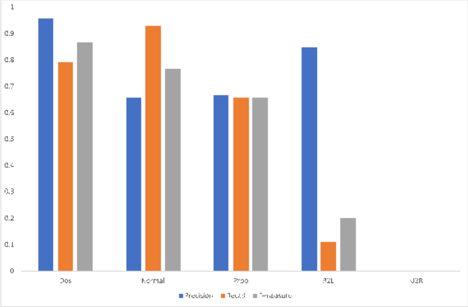

Figure 2. Evaluation measures of CNN-IDS per attack type

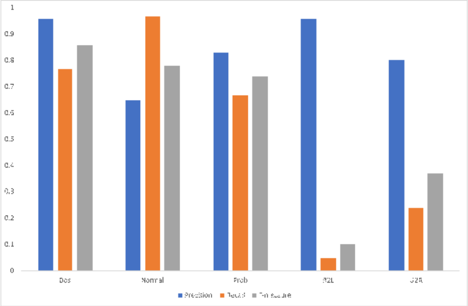

Figure 3. Evaluation measures of LSTM-IDS per attack type

6 Conclusion

Network intrusion detection systems are an integral part of contemporary networks. They provide administrators with an early warning for known and unknown attacks. In this paper, two deep learning architectures to aid administrators in detecting network attacks are outlined. The proposed models detect anomalous behavior, and classify the type of the attack.

In particular, two B-LSTM models are presented, for binary and multi-class classification, as well as two CNN models, also for binary and multi-class classification. The proposed models show good performance when compared to other models in the literature. Both models show superior performance on the binary classification task as compared to the multi-class classification task. In the future, the proposed models can be improved by increasing the number of hidden layers and neurons, or adding some other specialized layers. In addition, using different optimizers and trying new values for the learning rate are possible.

Acknowledgements

This research project was supported by a grant from the Research Center of the Female Scientific and Medical Colleges, the Deanship of Scientific Research, King Saud University.

References

[1] Ajith Abraham, Crina Grosan, and Yuehui Chen. Cyber security and the evolution in intrusion detection systems. *Journal of Engineering and Technology, ISSN*, pages 0973–2632, 2005.

[2] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 21–26, 2016.

[3] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

[4] Shih-Wei Lin, Kuo-Ching Ying, Chou-Yuan Lee, and Zne-Jung Lee. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Applied Soft Computing*, 12(10):3285–3290, 2012.

[5] Levent Koc, Thomas A Mazzuchi, and Shahram Sarkani. A network intrusion detection system based on a hidden naïve bayes multiclass classifier. *Expert Systems with Applications*, 39(18):13492–13500, 2012.

[6] Ikram Sumaiya Thaseen and Cherukuri Aswani Kumar. Intrusion detection model using fusion of chi-square feature selection and multi class svm. *Journal of King Saud University-Computer and Information Sciences*, 29(4):462–472, 2017.

[7] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri. Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. *Expert Systems with Applications*, 67:296–303, 2017.

[8] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, 5:21954–21961, 2017.

[9] Zhipeng Li, Zheng Qin, Kai Huang, Xiao Yang, and Shuxiong Ye. Intrusion detection using convolutional neural networks for representation learning. In *International conference on neural information processing*, pages 858–866. Springer, 2017.

[10] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. IEEE, 2009.

[11] Brian Lee, Sandhya Amaresh, Clifford Green, and Daniel Engels. Comparative study of deep learning models for network intrusion detection. *SMU Data Science Review*, 1(1):8, 2018.

[12] Najwa Altwaijry, Ameerah ALQahtani, and Isra AlTuraiki. A deep learning approach for anomaly-based network intrusion detection. In *International Conference on Big Data and Security*, pages 603–615. Springer, 2019.

[13] Arwa Aldweesh, Abdelouahid Derhab, and Ahmed Z Emam. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189:105124, 2020.

[14] Yasir Hamid, Veeran Ranganathan Balasaraswathi, Ludovic Journaux, and Muthukumarasamy Sugumaran. Benchmark datasets for network intrusion detection: A review. *IJ Network Security*, 20(4):645–654, 2018.

[15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[16] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[17] Kingma Da. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Bhupendra Ingre and Anamika Yadav. Performance analysis of nsl-kdd dataset using ann. In *2015 international conference on signal processing and communication engineering systems*, pages 92–96. IEEE, 2015.

[19] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. Deep learning approach for network intrusion detection in software defined networking. In *2016 international conference on wireless networks and mobile communications (WINCOM)*, pages 258–263. IEEE, 2016.

[20] Majjed Al-Qatf, Yu Lasheng, Mohammed Al-Habib, and Kamal Al-Sabahi. Deep learning

approach combining sparse autoencoder with svm for network intrusion detection. *IEEE Access*, 6: 52843–52856, 2018.

[21] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. Network traffic classifier with convolutional and recurrent neural networks for internet of things. *IEEE Access*, 5:18042–18050, 2017.

[22] Mohammad Lotfollahi, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, and Mohammdsadegh Saberian. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012, 2020.

[23] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescapè. Mimetic: Mobile encrypted traffic classification using multimodal deep learning. *Computer Networks*, 165:106944, 2019.

[24] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescapé. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Transactions on Network and Service Management*, 16(2):445–458, 2019.

Isra AL-Turaiki is an associate professor of computer science at King Saud University. She received her Ph.D. degree in 2014 from the college of computer sciences at King Saud University. Her research interests include data mining, machine learning, and bioinformatics.

Najwa Altwaijry is an assistant professor of computer science at King Saud University. She received her Ph.D. degree in 2014 from the college of computer sciences at King Saud University. Her research interests include machine learning, swarm intelligence, evolutionary computation, cybersecurity and bioinformatics.

Abeer Agil/Haya Aljodhi/Sara Alharbi/Lina Alqassem received her Bachelor's degree from the Computer Science Department at the College of Computer and Information Sciences at King Saud University in January 2020. Her interests are in the fields of machine learning, artificial intelligence and cyber security.