Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 19 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i19.49907

PAPER

Combined Deep Learning Approaches for Intrusion Detection Systems

Sawsan Alshattnawi[1](✉), Hadeel Rida Alshboul[2]

[1]Yarmouk University, Irbid, Jordan

[2]The World Islamic Sciences and Education University, Amman, Jordan

Sawsan_kh@yu.edu.jo

ABSTRACT

Cybersecurity has become increasingly important because of the widespread use of data and its enormous global storage. Hackers and other invaders always want to breach data security by interfering with network traffic. The breaches must be stopped by several tools, such as firewalls. Other solutions, such as intrusion detection systems (IDSs), may detect network intrusions effectively. In this paper, we introduce a hybrid technique (CNN-LSTM) that combines the convolutional neural network (CNN) with long short-term memory (LSTM), a modified version of the recurrent neural network (RNN). The model is tested using the CSE-CIC-IDS2018 dataset. Both CNN and LSTM were individually applied to the datasets, and the results are compared with our hybrid CNN-LSTM model. The hybrid CNN-LSTM model demonstrated higher accuracy (99%) during both training and validation processes compared to individual models; the accuracy of the CNN model is 92% and the accuracy of the LSTM is 93.5%. The outcomes validate the usefulness and effectiveness of the hybridizing model.

KEYWORDS

cyber-security, intrusion detection system (IDS), convolutional neural network (CNN), long short-term memory (LSTM)

1 INTRODUCTION

Information technology (IT) has recently converged, and many IT devices have grown incredibly complicated [1]. Digital data creation and storage are crucial, and as more digital devices are connected and spread all over the world, the systems become more vulnerable [2] [3]. Many behaviors, such as unauthorized access attempts, malware infections, or any other unusual activity, can point to a possible security breach [4].

To detect and handle the undesired activities taking place within a computer system or network, a specialized security tool such as firewalls or IDSs must be designed and operated effectively [5] [6]. An attack detection method known as a network-based intrusion detection system (NIDS) monitors the network traffic to spot unusual activities in order to offer the necessary protection [7].

Alshattnawi, S., Alshboul, H.R. (2024). Combined Deep Learning Approaches for Intrusion Detection Systems. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(19), pp. 144–155. https://doi.org/10.3991/ijim.v18i19.49907

Article submitted 2024-05-13. Revision uploaded 2024-07-26. Final acceptance 2024-07-26.

Intrusion detection systems come in two primary varieties: network-based NIDS and host-based HIDS [8] [9]. NIDS is installed throughout a network to keep track of all network activity, while HIDS is installed to scan and keep track of every host traffic, process, and device connected to the network [7]. IDSs come in two primary categories [10]: 1) signature-based techniques that monitor the network packets and connections with prior known patterns called signatures; and 2) anomaly-based techniques that create a baseline of typical behavior, sounds an alarm when abnormalities are found, and notify network administrators of potential threats [11].

Intrusion detection systems rely heavily on signature-based methods, but the limitations of these approaches in detecting novel and sophisticated attacks have driven the adoption of machine learning and deep learning approaches. Traditional methods often struggle with polymorphic malware, prompting studied to explore more dynamic solutions [12].

Several machine learning methods have been successfully applied to build IDSs. The application of artificial neural networks (ANNs), especially in the form of deep learning models, is one extensively studied method [13] [14] [15] [16]. ANNs demonstrate the ability to learn complex patterns and relationships within network data, improving detection accuracy. The importance of security today requires analysis of the vast amount of data; both deep learning and artificial intelligence are required for this to be feasible.

Effective feature selection is crucial for the success of machine-learning-based IDS [17]. Several methods, such as genetic algorithms and principal component analysis [18], have been investigated by studied to extract the most pertinent aspects from the massive amount of data produced by network traffic.

The work in this paper aims to assess the system's performance of deep learning approaches by hybridizing CNN and RNN-LSTM. The model is compared to CNN and LSTM separately for the CSE-CIC-IDS2018 datasets and examines the system's efficacy in identifying intrusions on various network traffic types.

The objective of developing the CNN-LSTM is to extract the features of the input dataset. The outputs of the CNN layers were then passed to the LSTM layers, and a dense layer was added at the output to support sequence prediction.

This paper is organized to explore the state of art in Section 2, the proposed approach is presented in Section 3, then the results are shown in Section 4, and finally, the work conclusion is presented in Section 5.

2 LITERATURE REVIEW

Studies have demonstrated that deep learning surpasses conventional methods. It, a recently trendy topic in machine learning, has been used to build IDS. This section provides an overview of earlier research that generated IDS using deep learning techniques. Since its introduction, anomaly detection has been the subject of continuous research within the context of intrusion detection and computer security. An IDS shows normal system or network traffic behavior in addition to any anomalies that occur within a specified time frame. IDS are taught to identify the essential characteristics of a system or network environment in order to depict its normal behavior. [19], [13]. Even though payload-based attacks are becoming more common, most IDSs concentrate on packet header data and ignore the important information in payloads. In this study [20], the authors presented a new IDS called TR-IDS that makes use of payload attributes in addition to statistical features. Word embedding and text-convolutional neural networks are efficiently utilized for obtaining data from the payload. The combination of statistical data and payload features is then subjected to

the complex random forest method. Comprehensive experimental analyses show that the suggested approaches are effective. The researchers in [21] suggested an integrated machine learning algorithm (KMC + NBC) that combined the strengths of K-Means and Naïve Bayes classifiers (NBC). When labeling and classifying entire sets of data into corresponding clusters based on data behavior (attack and normal), erroneously classified data is reorganized into legitimate classes using the NBC and K-Means clusters. To test KMC's efficacy with NBC and NBC against the ISCX 2012 assessment dataset, experiments were conducted. The results show that although false alarms have decreased to 2.2%, accuracy and detection rate have significantly increased to 99% and 98.8%, respectively, by KMC and NBC. The researchers in [22] deployed an IDS using genetic algorithms and K-Centroid clustering. The K-Centroid grouping was used to divide the training group into various groups. To confirm each interaction with the test group, GA was also carried out. Ultimately, the outcome of each contact was determined. After evaluating the data from the Kdd99Cup and NSLKDD datasets, they were able to determine that the applied detection system had a respectable detection rate.

The authors in [23] employed a variety of autoencoder, recurrent, and convolutional deep neural network architectures and created anomaly detection models. The two test data sets—NSLKDDTest+ and NSLKDDTest—provided by NSLKDD were used to evaluate these deep models after they had been trained on the NSLKDD training dataset. Well-known classification approaches, such as extreme learning machines, support vector machines, closest neighbors, decision trees, random forests, naive bays, and quadratic discriminate analysis, have been used to create traditional machine learning-based intrusion detection models. High scores in anomaly detection systems were observed in the experimental findings of deep IDS models.

Based on neural network learning, the researchers [16] provided a group model for detecting anomalies in a single class. Before producing a forecast for each time step, the LSTM RNN was only trained on typical time-series data. Rather than examining every time step independently, the model was assessed using the 1999 KDD data set's time-series version. The suggested model may effectively identify collective anomalies, according to experiments.

In order to identify anomalies that transpired within a minute interval, the researchers in [24] created a time series model utilizing samples from process logs for both normal and anomalous events. This allowed them to identify and categorize the occurrences as either normal or abnormal. Experiments are conducted for the different network architectures and parameters in order to choose an appropriate LSTM network. Using real-world test data from CDMC 2016, the S-LSTM network's design showed its resilience by attaining a maximum accuracy of 0.996 with a false positive rate of 0.02.

A deep learning neural network was made available by the researchers [25] to categorize data about network traffic. An activation function-corrected linear CNN was used. Using the leakage mechanism, the degree of separation in the completely linked final layer was computed. The model was validated using k-fold cross-validation, using a k value of 10. The experiment classified four distinct attacks against the normal state using the 1999 NSL KDD Cup datasets. So to demonstrate the usefulness of the given model, its accuracy for the NSL-KDD 199 cup was demonstrated along with other cutting-edge technologies.

By reviewing the previous literature, we conclude that deep neural network methods are an effective way to detect new attacks with high efficiency. In addition, a major drawback in a real-time data set is completely unbalanced data, so learning from unbalanced data presents low accuracy. Therefore, it must be taken into account in our study how to choose a data set that is balanced, regular, and strong to give high accuracy and better results than previous works. To resolve the anomaly-based IDS problem, a hybrid system utilizing CNN and LSTM in deep learning is presented in this study.

3 OVERALL APPROACH DESIGN

The proposed solution to the problem of anomaly-based IDSs consists of several stages: the pre-processing stage, the feature engineering stage, the training stage of multiple deep learning models as binary mode classifiers, and the final stage of evaluating the models' performance. Figure 1 shows the main phases of the presentation.

1. Data preprocessing and feature engineering: in this phase, the data is prepared for classification and prediction. The preprocessing may include data cleaning, such as converting string values into numeric values; Normalization, scaling, and breaking the data into testing and training sets.
2. Building the predictive models: the prediction will be made by the hybrid CNN-LSTM model, and the CNN and LSTM models will be built separately.
3. Training phase: Using two different kinds of deep learning models, we will suggest an IDS based on deep learning techniques (CNN and LSTM). To compare the performance of the proposed hybrid system with the performance of each model alone, we will construct three models: CNN, LSTM, and CNN-LSTM.
4. The final step involves determining if the system is under attack or not by applying the predictive models. Deep learning techniques will be utilized to assess how well the models in the suggested IDS operate.

In the following subsection, the phases will be explored in detail.

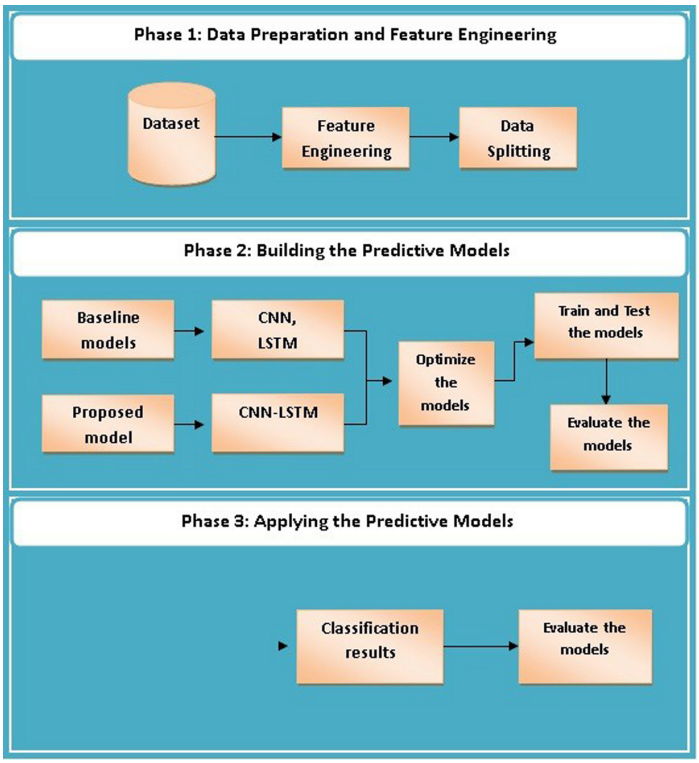

Fig. 1. Phases of the proposed solution

3.1 Dataset

The Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) collaborated on this initiative to create cybersecurity statistics

systematically using the idea of profiles. It includes abstract models of distribution models for protocols, lower-level network components, and applications. as well as a thorough analysis of intrusions. Seven distinct attack scenarios are included in the dataset. There are 50 machines in the attacking infrastructure, 420 personal computers, and 30 servers distributed among five divisions inside the victim organization. This dataset [26] contains 80 network traffic features that were extracted from the extracted traffic, as well as log files for each victim framework.

3.2 Data preparation and feature engineering

The most crucial phase of the pre-processing study is getting the data ready for forecasting and classification, engineering its features, and pre-processing. In order to prevent issues that might arise during the training phase of the system process, we will use the CSE-CIC-IDS2018 data sets in this assessment, which rely on deep learning to detect and classify attacks. We first balance the data because it is unbalanced after data analysis. We then convert values containing text in the data into numerical values by encoding using the labelEncoder function. The features in the data were also found to have fixed values, requiring the removal of columns with fixed values and reducing the number of features to 66 entries, as done in the feature engineering step. Because there is a variance between values (high values and low values), we need to enhance the geometry of features that measure and unite data using the standardscaler function. In the final step, we will split the dataset into a training set, a validation set, and a test set. The dataset has a very large size of 400 GB. We focused on one type of attack present in this data, which is brute force. It consists of two tools, SSH and FTP, which are classified as attack, and the other type is bengin, which is considered normal traffic.

The data was unbalanced; as we mentioned previously, the left part of Figure 2 shows the data distribution. It is represented in binary classification 0 for beginning, 1 for FTP-brute force, and 2 for SSH-brute force, and this was done by the labelencoder so that it is easy to identify in deep learning models. The data is balanced by dividing the beginning category by 2.5 so that it becomes balanced with the rest of the categories in this data, as shown in the right of Figure 2. It shows that benign or normal traffic has a percentage of 41.2% and the attack or abnormal traffic has a percentage of 58.7%, the abnormal attack has an equal balance (28.9% and 29.8%). After preparing the data, we performed the partitioning for training (80%) and validation (20%).

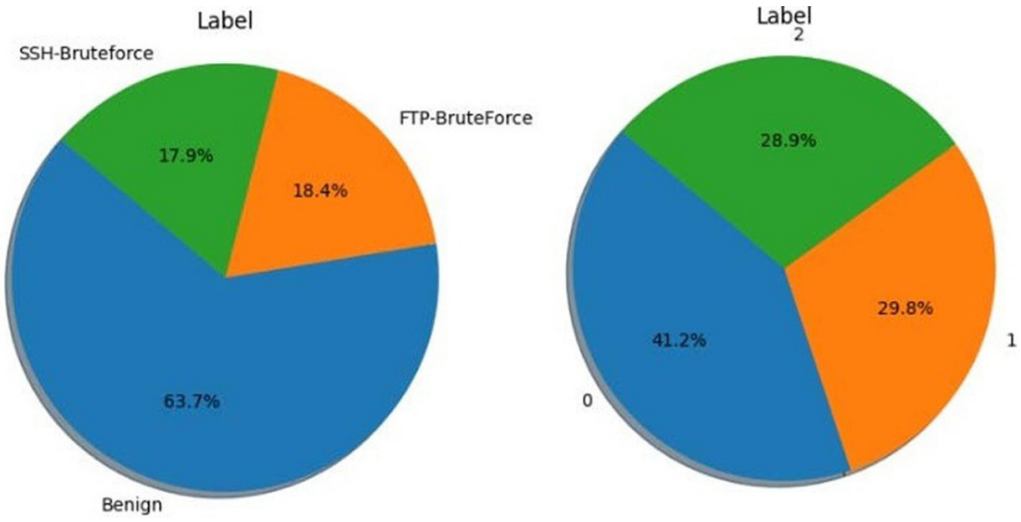

Fig. 2. Unbalanced and balanced dataset

3.3 Building the predictive models

The prediction model construction process involves training, validating, evaluating, and refining the models. To identify the deep network intrusion detection methods based on the architecture and specific technologies applied, we focus on deep networks with classification training in the field of deep learning. This section describes the models that were utilized in the analysis. The first model is the CNN model, the second model is the LSTM model, and the third is the CNN-LSTM model. By training as well as assessing these models using common performance criteria such as the accuracy and error loss of each deep learning model.

Convolutional neural network model. A CNN is composed of three layers: the first is the input layer, the second is the hidden layer, and the third is the output layer. Since the convolution and final activation functions hide the inputs and outputs of the intermediate layers in a direct automated neural network, these layers are collectively referred to as hidden. CNN is a well-such as deep learning method that does not require human feature extraction procedures; instead, it learns directly from a dataset. We employ a categorical classification (benign, FTP-brute force, or SSH-brute force). Our model consists of a linear stack of layers, where each layer processes data in turn before using the output layer to forecast the type of traffic input.

Long short-term memory model. Long short-term memory is an ongoing artificial neural network design used in deep learning. LSTM has feedback relationships, in contrast to conventional feed-forward neural networks. The complete data stream can be processed by it. A typical LSTM unit consists of three layers. While learning momentary characteristics, LSTM is utilized to learn the temporal aspects of several traffic vectors. The temporal correlations between the packet vectors are further learned by the LSTM. The outcome is a single flow vector that captures the network flow's spatiotemporal properties. Effective handling of mixing and gradient burst problems will enhance the capacity to identify temporal and spatial relationships and acquire knowledge from sequences with differing intervals. After CNN processes the input for the first time, the output is sent to the feature selection stage, and sequences are generated at each time step by the LSTM, which helps with the modeling of the temporal and spatial features. After that, the sequence vector passes via a fully linked layer in the last stage of ranking before being incorporated into the layers to allocate the possibilities to the categories.

The hybrid model: convolutional neural network–long short-term memory. The traffic classifier proposed learns and classifies traffic packets from the dataset utilized in both time and space by utilizing a CNN and LSTM combination. The CNN-LSTM classifier is composed of the following layers: input, embedding, convolution, pooling, and completely connected. The CNN partition transmits a high-level, dimensional vector packet to the LSTM partition following its receipt and analysis of the dataset. The LSTM section consists of a pair of LSTM layers, a fully connected layer, and an output layer. After processing a number of high-dimensional packet vectors, it can produce a vector that shows the likelihood of an attack on any class, whether it be normal or an attack. For CNN-LSTM testing, we randomly separated the data set into training (80%) and test (20 percent) sets. Furthermore, 20% of the training set sample was used for validation. First-order gradient-based optimization methods with variable learning rates, including Adam, AdaGrad, RMSprop, and AdaMax, were applied to optimize the binary loss during the training stage step by connecting nodes and features, which aids in the creation of LSTM features. After

going through a fully connected layer, the sequence vector is sent into a sigmoid layer, which determines the probability distribution among the classes. At this point, the test set was one of the inputs used by the trained model to judge if the training traffic's behavior was harmful or normal. Lastly, the learned classifiers are assessed using the test set.

3.4 Applying the predictive models

The predictions for each model will be calculated and compared with the rest of the models, with the models' performance calculated based on loss and accuracy [27].

$$\text{Precision} = TP / (TP + FP), \text{Recall} = TP / (TP + FN)$$

Where: TP stands for true positive, which indicates the number of positive examples classified accurately. FP shows a false positive value, i.e., the number of actual negative examples classified as positive, TN stands for true negative, which shows the number of negative examples classified accurately. FN means a false negative value, which is the number of actual positive examples classified as negative.

The harmonic means of recall and precision is called the F-measure, or F1-score, and it is used to determine accuracy [28].

$$\text{F-score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

4 EXPERIMENTAL RESULTS

To prove the effectiveness of the hybrid model CNN-LSTM, we compared it with CNN and LSTM separately. The three models were built with TensorFlow, Keras, and scikit-learn packages in Python on Google Colab. The evaluation is based on the CSE-CIC IDS2018 dataset, which includes seven categories of records: brute force (FTP, SSH), heartbleed, botnet, DoS, DDoS, web attacks, and intrusion. We are dealing with a categorical classification problem, and one of three possible outputs will be made: Benign, FTP, or SSH.

4.1 Experiment the convolutional neural network model

Seven layers are included in the CNN model: Two dropout layers and three dense layers Each dense layer has 64 features, with two activation layers and 4288, 4160, and 195 parameters in total. To extract the feature vector from the input shape, these layers must be active. The Relu activation function was applied to activate these dense layers. Following every dense layer, a 64-feature dropout layer was employed. Connecting the set of nodes in the output layer and the set we acquire after flattening, we must build a completely connected layer once the dense and dropout procedures are completed.

Situated between the output layer and the final pooling layer, the hidden layer is a fully connected layer. Throughout the experiment, the number of nodes in this layer must always fall between the number of input and output nodes. To activate the completely connected layer, Relu was utilized. The three models' experiment parameters are compiled in Table 1. The outcomes are displayed in Table 2.

Table 1. Parameters of three model

Parameters	CNN Model	LSTM Model	Hybrid Model
Number of Dense layers	3	1	2
Activation (Dropout)	Relu	Sigmoid	Relu
Number of Dropout layers	2	3	2
Number of activations	2	3	3
Number of nodes in the output layer	3	3	3
Activation (output layer)	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adam	Adam	Adam

Table 2. The results of CNN

Measures	Benign	FTP-Bruteforce	SSH-Bruteforce
Precisions	.88	.93	.90
Recall	.94	.89	.94
F1-score	.92	.92	.92

4.2 Experiment the long short-term memory model

The LSTM model used only two layers: the LSTM layer for the input and the dense layer for the output with the same activation layer (sigmoid) and the same optimizer used previously, "Adam." Both work in the same way to solve similar classification problems. It is evident that the model's output matches and approximates that of the prior model, and that the validation and training accuracy are consistent with one another. This indicates that the model performs similarly when predicting accuracy and validation data. The results are shown in Table 3.

Table 3. The results of LSTM

Measures	Benign	FTP-Bruteforce	SSH-Bruteforce
Precisions	.90	.94	.90
Recall	.95	.90	.94
F1-score	.92	.92	.92

4.3 Experiment the hybrid model (convolutional neural network, long short-term memory)

The training accuracy, validation accuracy, training loss, and validation loss for the hybrid model are more powerful for the whole training procedure as shown in Figure 3, indicating that the unit performs identically on both the training and validation sets of data. The identical outcomes for all three models are displayed in Table 4, suggesting a decreased likelihood of equipping. Table 4 displays identical outcomes across the three models.

The CNN and the LSTM models give similar results to a large extent when they are compared. Both of which give the same results: CNN accuracy is 93%, while the LSTM model is 92% according to measures of recall, precision, and f1-score. The CNN-LSTM has a higher accuracy of 99.98%. As a result, we trained some alternative classifiers using the same dataset; the results are shown in Table 5. The Python machine-learning library Scikit-Learn was used to create these classifiers. In contrast to CNNs, which need preprocessed data for input, Scikit-Learn classifiers are susceptible to overfitting issues. Deep learning models are completely free from problems. These classifiers can give high results.

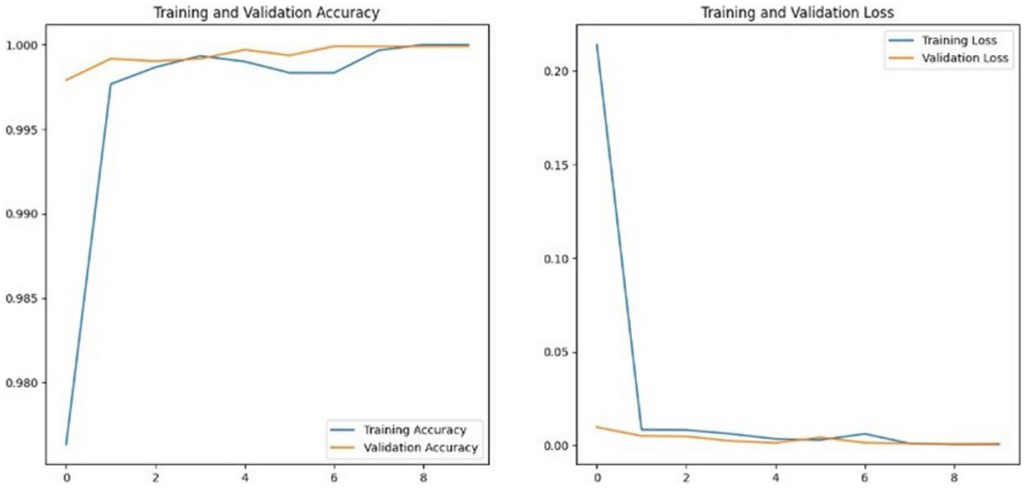

Fig. 3. Accuracy and loss in the hybrid model

Table 4. Results of the hybrid model

Measures	Benign	FTP-Bruteforce	SSH-Bruteforce
Precisions	.98	.98	.99
Recall	.99	.99	.99
F1-score	.98	.99	.98

Table 5. Comparison with other machine learning classifiers (based on CSE-CIC IDS2018)

Classifier	Accuracy	Classifier	Accuracy
Random Forest	0.98	AdaBoost	0.92
XGBoost	.93	GaussianNB	0.82
Decision Tree	0.94	BernoulliNB	0.77

5 CONCLUSION

In this study, we integrated two deep learning approaches (CNN and LSTM) to develop an anomaly-based intrusion detection system. In addition, each method was separately applied to the same data set. The CNN-LSTM model gave higher accuracy than the remaining models separately during training and validation. The outcomes demonstrated the effectiveness and high efficiency of the CNN-LSTM model.

In addition, we trained some alternative classifiers using the same dataset, including Random Forest, XGBoost, Decision Tree, AdaBoost, LAD, GaussianNB, and BernoulliNB. The results show that deep learning models outperform these seven classifiers, and they are certainly free from any problems that can arise with machine learning classifiers, such as overfitting. In addition, for future work, it can be implemented in real network traffic.

6 REFERENCES

[1] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. I. AlZubi, "Intensive pre-processing of KDD Cup 99 for network intrusion classification using machine learning techniques," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 13, no. 1, pp. 70–84, 2019. https://doi.org/10.3991/ijim.v13i01.9679

[2] I. D. Wahyono, D. Saryono, K. Asfani, M. Ashar, and S. Sunarti, "Smart online courses using computational intelligence," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 12, pp. 29–40, 2020. https://doi.org/10.3991/ijim.v14i12.15601

[3] A. Abozeid, A. A. AlHabshy, and K. ElDahshan, "A software security optimization architecture (SoSOA) and its adaptation for mobile applications," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 15, no. 11, pp. 148–165, 2021. https://doi.org/10.3991/ijim.v15i11.20133

[4] A. R. Muhsen, G. Jumaa, N. F. AL Bakri, and A. T. Sadiq, "Feature selection strategy for network intrusion detection system (NIDS) using Meerkat Clan algorithm," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 15, no. 16, pp. 158–171, 2021. https://doi.org/10.3991/ijim.v15i16.24173

[5] D. Musleh, M. Alotaibi, F. Alhaidari, Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," *J. Sens. Actuator Netw.*, vol. 12, no. 2, p. 29, 2023. https://doi.org/10.3390/jsan12020029

[6] M. Gottlieb and M. C. Utesch, "Publish or perish: A scientific blueprint for a journal article," *International Journal of Engineering Pedagogy (iJEP)*, vol. 12, no. 3, pp. 171–177, 2022. https://doi.org/10.3991/ijep.v12i3.28253

[7] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 313–316, 2017.

[8] V. Hnamte and J. Hussain, "DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, p. 100053, 2023. https://doi.org/10.1016/j.teler.2023.100053

[9] M. Maabreh *et al.*, "Towards data-driven network intrusion detection systems: Features dimensionality reduction and machine learning," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 14, pp. 123–135, 2022. https://doi.org/10.3991/ijim.v16i14.30197

[10] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Computer Communications*, vol. 199, pp. 113–125, 2023. https://doi.org/10.1016/j.comcom.2022.12.010

[11] S. M. Othman, N. T. Alsohybe, F. M. Ba-Alwi, and A. T. Zahary, "Survey on intrusion detection system types," *International Journal of Cyber-Security and Digital Forensics*, vol. 7, no. 4, pp. 444–463, 2018.

[12] S. S. Anshu Gangwar, "A survey on anomaly and signature based intrusion detection," *Journal of Engineering Research and Applications*, vol. 4, no. 4, pp. 67–72, 2014.

[13] B. S. Harish and S. V. A. Kumar, "Anomaly based intrusion detection using modified fuzzy clustering," *International Journal of Interactive Multimedia and Artificial Intelligence*, 2017.

[14] Y. Vinoth and K. Kamatchi, "Anomaly based network intrusion detection using ensemble machine learning technique," *International Journal of Research in Engineering, Science and Management (IJRESM)*, pp. 290–296, 2020.

[15] B. M. Serinelli, A. Collen, and N. A. Nijdam, "Training guidance with KDD Cup 1999 and NSL-KDD data sets of anidinr: Anomaly-based network intrusion detection system," *Procedia Computer Science*, vol. 175, pp. 560–565, 2020. https://doi.org/10.1016/j.procs.2020.07.080

[16] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2018, pp. 1394–1401.

[17] G. M. Hassan, A. Gumaei, A. Alanazi, and S. M. Alzanin, "A network intrusion detection approach using extreme gradient boosting with max-depth optimization and feature selection," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 5, pp. 120–134, 2023. https://doi.org/10.3991/ijim.v17i15.37969

[18] N. Nguyen Thi, V. L. Cao, and NA. Le-Khac, "One-class collective anomaly detection based on LSTM-RNNs," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXVI.* in Lecture Notes in Computer Science, A. Hameurlain, J. Küng, R. Wagner, T. Dang, and N. Thoai, Eds., Springer, Berlin, Heidelberg, vol. 10720, pp. 73–85, 2017. https://doi.org/10.1007/978-3-662-56266-6_4

[19] S. Alshattnawi, "Evaluation of deep learning and machine learning algorithms in intrusion detection systems," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 5, pp. 1944–1953, 2023.

[20] S. Kishor Wagh, V. K. Pachghare, and S. R. Kolhe, "Survey on intrusion detection system using machine learning techniques," *International Journal of Computer Applications*, vol. 78, no. 16, pp. 30–37, 2013. https://doi.org/10.5120/13608-1412

[21] J. L.ong, Q. Liu, J. Cui, and W. Chen, "Tr-ids: Anomaly-based intrusion detection through text-convolutional neural network and random forest," *Security and Communication Networks*, vol. 2018, no. 1, pp. 1–9, 2018. https://doi.org/10.1155/2018/4943509

[22] N. I. U. Z. M and M. N. S. W. Yassin, "Anomaly-based intrusion detection through k-means clustering and naives bayes classification," 2013.

[23] B. Chakrabarty, O. Chanda, and Md. Saiful Islam, "Anomaly based intrusion detection system using genetic algorithm and k-centroid clustering," *International Journal of Computer Applications*, vol. 163, no. 11, pp. 13–17, 2017. https://doi.org/10.5120/ijca2017913762

[24] S. Naseer *et al.*, "Enhanced inetwork anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018. https://doi.org/10.1109/ACCESS.2018.2863036

[25] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Long short-term memory based operation log anomaly detection," *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 236–242. https://doi.org/10.1109/ICACCI.2017.8125846

[26] S. Behera, A. Pradhan, and R. Dash, "Deep neural network architecture for anomaly based intrusion detection system," in *5th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2018.

[27] M. C. Nadia and N. Ibrahim Nife, "Improved detection and tracking of objects based on a modified deep learning model (YOLOv5)," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 21, pp. 145–160, 2023. https://doi.org/10.3991/ijim.v17i21.45201

[28] N. K. M. A. Alheeti, N. A. A. A. Lateef, N. A. Alzahrani, N. A. Imran, and N. D. Al_Dosary, "Cloud intrusion detection system based on SVM," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 11, pp. 101–114, 2023. https://doi.org/10.3991/ijim.v17i11.39063

7 AUTHORS

Sawsan Alshattnawi is a Full Professor at the Department of Computer Science at Yarmouk University (Jordan) since April 2021. She joined Yarmouk University academic staff as an Assistant Professor in 2009. Her research interests include cloud computing, mobile computing, and cyber-Security (E-mail: Sawsan_kh@yu.edu.jo).

Hadeel Rida Alshboul is a full-time Lecturer in the Department of Computer Science at World Islamic Science and Education University (Jordan). Her research interests include cyber security, machine learning, artificial intelligence, and e-learning systems (E-mail: hadeel.alshboul@wise.edu.jo).