

# Deep self-taught learning framework for intrusion detection in cloud computing environment

Thavavel Vaiyapuri, Adel Binbusayyis

Department of Computer Sciences, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University,  
Al Kharj, Saudi Arabia

## Article Info

### Article history:

Received Apr 5, 2021

Revised Oct 5, 2023

Accepted Oct 31, 2023

### Keywords:

Cloud computing

Cybersecurity

Deep learning

Intrusion detection

Long short-term memory

Self-taught learning

Sparse autoencoder

## ABSTRACT

Cloud has become a target-rich environment for malicious attacks by cyber intruders. Security is a major concern and remains an obstacle to the adoption of cloud computing. The intrusion detection system (IDS) is regarded as defense-in-depth. Unfortunately, most machine learning approaches designed for cloud intrusion detection require large amounts of labeled attack samples, but in real practice, they are limited. Therefore, the key impetus of this work is to introduce self-taught learning (STL) combining stacked sparse autoencoder (SSAE) with long short-term memory (LSTM) as a candidate solution to learn the robust feature representation and efficiently improve the performance of IDS with respect to false alarm rate (FAR) and detection rate (DR). Accordingly, the proposed approach as a first step employs SSAE to achieve dimensional reduction by learning the discriminative features from network traffic. The approach adopts LSTM to recognize the intrusion with the features encoded by SSAE. To evaluate the detective performance of our model, a comprehensive set of experiments are conducted on NSL-KDD. Also, ablation experiments are conducted to show the contribution of each component of our approach. Further, the comparative analysis shows the efficacy of our approach against the existing approaches with an accuracy of 86.31%.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Thavavel Vaiyapuri

Department of Computer Sciences, College of Computer Engineering and Sciences

Prince Sattam bin Abdulaziz University

Al Kharj, Saudi Arabia

Email: t.thangam@psau.edu.sa

## 1. INTRODUCTION

Cloud computing is an information technology (IT) model enabling to provide on-demand access with rapid elasticity to a shared pool of computing resources. It offers several benefits to individual users and organizations in terms of reducing capital expenditure required to build IT infrastructure and avoiding the operational expenditure spent in maintaining the IT infrastructure [1]. These benefits have opened business avenue for new budding entrepreneur as well have encouraged many organizations across the globe to migrate their business activities to cloud and witness remarkable growth with essential characteristics of cloud such as higher availability, geographic reach, and business continuity [2], [3]. For example, 2018 Gartner study states “public cloud market will grow by 21.4% in 2018 alone, accounting for more than \$186 billion in revenues for the IT industry. In three years, the total revenue for cloud computing services is expected to exceed \$300 billion” [4]. Despite evidence demonstrate that cloud computing has become the influencing IT landscape and is seen as a major business avenue, security still exist as one of the main

obstacles hampering companies and businesses from migrating towards Cloud. Removing this obstacle has become key requirement to realize the potential of cloud computing.

According to National Institute of Standards and Technology (NIST) definition, “cloud computing services are provisioned over internet using threefold service models viz, platform as a service (PAAS), infrastructure as a service (IAAS), and software as a service (SAAS)” [5]. This openness and the distributed structure of cloud computing has made it an attractive target for malicious attacks by cyber intruders [6]. Traditionally, techniques such as authentication, encryption and firewalls were used as first line of defense in computer and network security [7]. But it is formally stated that it is easy for intruders to go around these techniques. Also, it is stated that they are not potential to prevent various new and sophisticated modern attacks. In this context, the antivirus software is also treated as essential security tool and many businesses employ it as an alternative defensive mechanism. Though, very powerful antivirus software is constrained in their capability to thwart only the attacks for which signature are available. To circumvent this context, intrusion detection system (IDS) is regarded as a promising alternative. Recent reports on cloud computing also implies that Intrusion detection system is the very crucial and powerful second line of defense to safeguard cloud infrastructure from intrusions analyzing the user and network traffic behavior [8]. Therefore, intrusion detection is receiving more attraction among the researchers in the field of security community.

Recently, several IDSs are published involving machine learning techniques [9], [10]. Notwithstanding, the breakthrough results of these IDS solely rely on the data quality utilized for developing the machine learning models to solve the problem under study. In general, IDS analyze the network traffic data to detect the attacks. These traffic data are noisy. Hence, feature representation learning is crucial not only in improving the performance of IDS but also in diminishing the computation complexity by pruning off redundant and irrelevant information [11]. It also prevents the learning model used in building IDS from overfitting. Whilst many researches have been proposed in the literature in this direction, accuracy of IDS still remains an issue [13]. This issue was impetus for this research to investigate the application of self-taught learning incorporating stacked sparse autoencoder (SSAE) to encode the strongest feature representation that can enhance the performance of long short-term memory (LSTM) with the learnt feature representation for intrusion detection in cloud network. The key contribution of the research work is summarized in three points: i) Investigate the advantage of applying self-taught learning in enhancing the performance of LSTM for intrusion detection; ii) Conduct ablation experiments to evaluate the design decision of our approach and select optimal hyperparameter values for developing an effective IDS for cloud environment; and iii) Compare and evaluate the effectiveness of our approach for gain in detection accuracy through comprehensive set of evaluation metrics

## 2. RELATED WORKS

In this section, we describe the required background knowledge and fundamental concepts that are utilized in the previous literature and required to better understand the building blocks of the proposed work. This is mainly to enhance the understanding of the proposed approach. In this direction, the section following discusses the working principles of two key components namely, autoencoder and deep neural network.

### 2.1. Sparse autoencoder (SAE)

SAE is a special type of autoencoder introduced in 2007 by Ranzato *et al.* [13] with an intuition of imposing sparsity constraint over the basic notion of autoencoder and learn the sparse feature representation from the given input sequence as shown in Figure 1. In this principle, SAE evaluates the activation function associated with each hidden neuron and then based on the activation value the neuron is treated as active if the value equates to 1 else the neuron is treated as inactive. As a result, if the most of the neuron become inactive then hidden layer may become sparse [14]. Thus, imposing sparsity constraint enables to limit the activation of undesired neurons and stimulates SAE to encode the most robust sparse features at its hidden layer [3]. Now, accounting to the sparsity constraint, the cost function of SAE as (1):

$$J_{SSAE}(\theta) = J_{AE}(\theta) + \alpha \sum_{j=1}^s KL(\rho || \hat{\rho}) \quad (1)$$

In the above equation,  $J_{AE}(\theta)$  is the cost function of basic AE [15]. As well the divergence of kullback–leibler (KL) across  $\hat{\rho}$  and actual  $\rho$  is minimized to determine the optimal value for sparsity constraint  $\rho$  given in the equation. Here, the parameter  $\alpha$  defines the proportional contribution of sparse constraint in the above given cost equation [14].

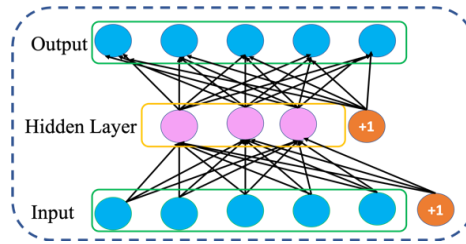


Figure 1. SSAE structure

## 2.2. LSTM

LSTM network is an enhanced recurrent neural network (RNN) variant with memory cells to learn long-term dependence information by storing the previous knowledge for long period. It was proposed in 1997 to overcome the gradients loss while learning long temporal sequence [16]. Technically, LSTM network is developed with series of three gates as shown in Figure 2. The three gates which are input and forget to control the information flow and ensure that the past information is stored with stable gradient in the memory cells [17]. Here input gate is responsible to decide the capacity of information to be stored in the memory as given in (2) and (4). The forget gate is responsible to decide the degree to which the previously stored information is forgotten, and the output gate takes the responsibility in deciding the stored information to be used as output. Finally, the state and output LSTM memory cell are determined by (5) and (6) [18], [19] respectively:

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, X_t] + b_i) \quad (2)$$

$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, X_t] + b_f) \quad (3)$$

$$O_t = \sigma(W_o \cdot [C_{t-1}, h_{t-1}, X_t] + b_o) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \quad (5)$$

$$h_t = O_t * \tanh(C_t) \quad (6)$$

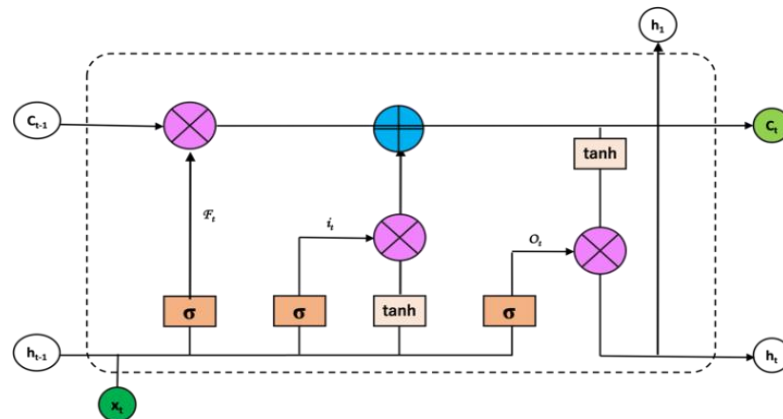


Figure 2. LSTM structure

## 3. PROPOSED APPROACH

This section introduces our model and describes how self-taught learning enables in improving the detection performance of IDS in cloud computing environment. The core focus of our approach is twofold: First to introduce self-taught learning framework combining SSAE and LSTM with an objective to learn the most robust features from the provided network traffic dataset. Second, to investigate the performance of the identified subset of informative features in intrusion detection. The key tasks of the proposed model are data preprocessing, feature representation learning and intrusion detection. Figure 3 demonstrates the architecture diagram of the proposed model for effective detection of intrusion in cloud environment.

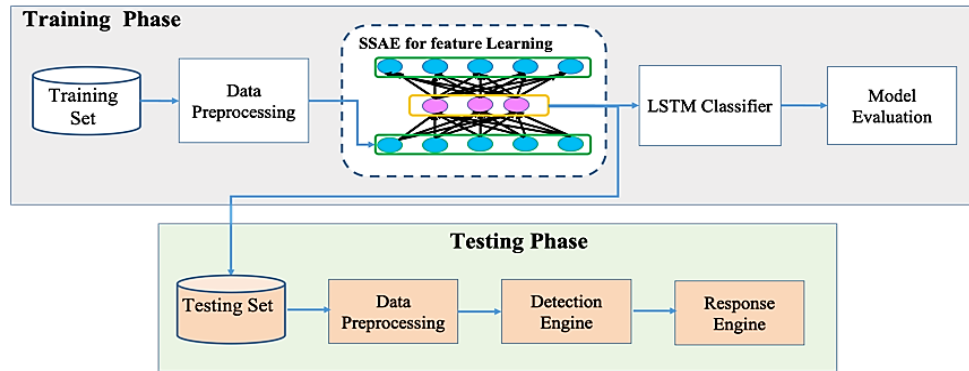


Figure 3. Architecture of proposed self-taught learning (STL)-based IDS

### 3.1. Data preprocessing

This stage performs two-key operations to prepare the network data for subsequent. Firstly, each extracted feature is converted to format that are consistent for anomaly detection process. For this, all non-numeric features are replaced with numeric ones. This enables the proposed model to work efficiently with numeric features. Secondly, all features in numeric form are normalized so that values of all features are within the range [0,1]. This enables to remove the bias that can be induced by features with different range of values [11], [20].

### 3.2. Feature learning

One of the cores aims of our approach is to effectively identify and eliminate irrelevant features from network traffic to achieve dimensionality reduction which reduces the computational overload and enhances the detection performance with respect to detection rate (DR) and false alarm rate (FAR) [20]. In achieving this, the proposed model employs SSAE to analyze the network traffic and encode the essential features with minimum mean square error to detect the malicious activities more accurately. To deep encode the most robust features of normal network traffic, the work here introduces stacked SAE (SSAE) by feeding the hidden layer output as input to succeeding SAE. The SSAE employs greedy approach in bottom-up for layerwise learning. The training process starts from first SAE. Once the training is completed, hidden layer of first SAE is utilized as input to train the subsequent SAE. In our work, three SAE is stacked to obtain SSAE with three hidden layers. Also, after the training process of SSAE network, the hidden layers represent the encoded network traffic features. The network structure of SSAE is given in Table 1.

Table 1. Hyperparameters of SSAE network

Hyperparameters	Value
1 <sup>st</sup> Hidden Layer Size	32
2 <sup>nd</sup> Hidden Layer Size	24
3 <sup>rd</sup> Hidden Layer Size	16
Activation Function	Tanh
Sparsity term	1e-5

### 3.3. Intrusion detection

The proposed model employs LSTM network to recognize the intrusion detection in cloud environment based on features that deep encoded by SSAE. As illustrated in Table 2, the structure of LSTM network developed in this study contains 5 layers. The LSTM layer was designed with block size of 32 to interface with SSAE network and receive the encoded deep features for detection process. Next, a flatten layer was placed next to interface the LSTM layer and following dense layer with 128 units. To prevent the network from overfitting, a dropout layer with 0.1 rates was added on the top of dense. Finally, as dense layer with two unit added as the last layer to detect whether the incoming network traffic is intrusion or not. After pretraining SSAE, the proposed model is trained for fine tuning using the stochastic gradient optimization method, Adam to learn the optimal network parameters in 20 epochs with learning rate of 0.001 and batch size of 128.

### 3.4. Evaluation metric

The effectiveness of an IDS is measured by its capacity to accurately identify the provided network traffic data packet as malicious or legitimate. A good IDS should have a low FAR and a high detection rate and accuracy. In order to calculate these three measures, the current work uses the confusion matrix as:

- a. Detection rate: It is also termed as Sensitivity or Recall. It measures the ratio of correctly classified malicious network traffic against the sum of network traffic records in the given dataset.

$$DR = \frac{TP}{TP+FN} \quad (7)$$

- b. FAR: This metric evaluates the ratio of misclassified normal network traffic against the sum of normal network traffic records in the given dataset. The consistent increases in this metric may enable the cloud network administrator to intentionally overlook the warning alerts from IDS and entire cloud network may enter into unsafe state. Therefore, it is advisable to keep this metric at lower value.

$$DR = \frac{FP}{FP+TN} \quad (8)$$

- c. Accuracy (ACC): This metric evaluates the ratio of correctly classified network traffic to the sum of network traffic records in the given dataset.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Table 2. Network parameters of LSTM

Layers	Units	Parameters
LSTM	16	return_sequence=true
Flatten	-	-
Dense	32	Activation = Tanh
Dropout	-	Rate=0.1
Dense	2	Activation = softmax

#### 4. RESULTS AND DISCUSSION

In this section, we have presented the experimental results to demonstrate the effectiveness of our model for intrusion detection. First, the experimental dataset is discussed. Later, the results of ablation and comparative analyses are presented to confirm its effectiveness for intrusion detection.

##### 4.1. Intrusion dataset

The Massachusetts Institute of Technology (MIT) Lincoln Lab published NSL-KDD dataset and was made publicly available for the purpose of research in the domain of cybersecurity. The dataset consists of a training and testing datasets [21]. The training dataset has 125,973 samples and 22,544 records in testing set. Each traffic record in these datasets contains 41 attributes and a target label which categorizes the record as normal or intrusion network traffic. The dataset consists of malicious traffic samples with 24 intrusion types that can be categorized into four main intrusion types such as unauthorized access to local supervisor privileges (U2R), unauthorized access from a remote machine (R2L), Denial-of-Service (DoS), and scanning network to find known vulnerabilities (Probing). Table 3 tabulates the statistics of the dataset with normal and intrusion records.

Table 3. Statistics of NSL-KDD dataset

Labels	Training set	Testing set
Normal	67,343	9,710
Abnormal	58,630	12,833
Total	125,973	22,543

##### 4.2. Ablation Analysis

This section elaborates the three different ablations designed to validate the essential components of the proposed system. The ablation experiments are conducted to investigate the significance and impact of each component in our model towards the gain in intrusion detection performance. To accomplish this, the following ablations are created by removing the certain components from the proposed model as: SSAE: created by replacing the LSTM network with softmax layer as shown below in Figure 4 and LSTM: created by removing the SSAE network as shown below in Figure 5.

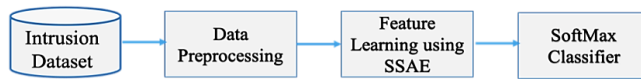


Figure 4. Designed system architecture for SSAE based ablation

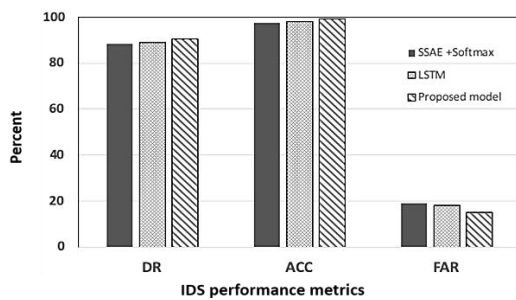


Figure 5. Designed system architecture for LSTM based ablation

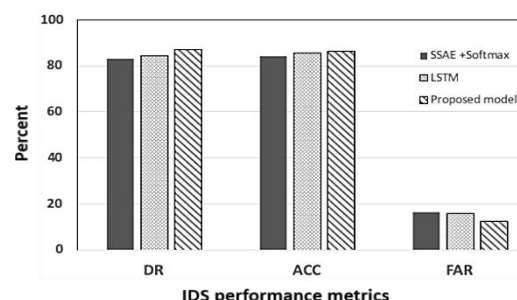
To perform fair comparison, the ablations and our model was trained under the same parameter settings and the experiments were conducted with same configuration setup. Table 4 reports the findings of the ablation analysis. The observation of the findings indicate that the detection rate of our model is better in comparison to its ablations. This confirms the significant contribution of both SSAE and LSTM network for intrusion detection performance. For example, it is evident from the results that the incorporation of SSAE component in the proposed model has boosted the detection accuracy of LSTM by 1.15%. The FAR rate and rate also show a similar trend. The visual analysis of the ablation performance on the training and testing set is also depicted in Figure 6(a) and Figure 6(b) respectively.

Table 4. Results of ablation experiment

AE Variants	Training set			Testing set		
	DR	ACC	FAR	DR	ACC	FAR
SSAE + Softmax	88.21	97.65	19.16	83.11	84.37	16.4
LSTM	89.1	98.03	18.21	84.52	85.64	15.76
Proposed Model	90.50	99.18	15.13	86.92	86.31	12.25



(a)



(b)

Figure 6. Performance analysis of ablation results of (a) training set and (b) testing set

#### 4.3. Receiver operating characteristic (ROC) analysis

The second step analysis employs ROC curve, an acronym for receiver operating characteristic to evaluate the effectiveness of our approach for intrusion detection performance on NSL-KDD dataset. In literature, it is stated that ROC curve is one of the most important metrics to analyze and compare IDS performance based on binary classification. This might be due to two reasons, First, the intrusion detection datasets are highly imbalanced. Second, ROC depicts the performance of a model as 2D plot between DR and FAR. These two metrics are considered as very vital requirement for an effective IDS.

For binary classification, a perfect ROC curve tends towards upper-left corner to demonstrate best performance. The visual inspection of Figure 7(a) and Figure 7(b) shows that on training set, ROC curves of three methods including proposed approach are closer. But on testing set, the proposed method displays better performance with ROC curve closer compared to the ablation methods. This clearly indicates that our approach performs more proficient in detecting unseen new attacks compared to the ablations.

Furthermore, to quantitatively express the detection performance of the proposed method over the designed ablations, the area under ROC curve (AUC) is calculated and displayed in the legend section of Figure 7 on NSL-KDD training and testing dataset. The AUC values vividly exhibit the best performance of the proposed method with values of 99.1% and 83.3% on training and testing set. Thus, the ROC curve and AUC value are consistent with the results presented in Table 4 revealing the potential of the proposed method over the designed ablation methods.

#### 4.4. Precision-recall (PR) analysis

This subsection applies PR curve, an acronym for PR to compare and analyze the effectiveness of the proposed method more intuitively for detection performance. The recent literature recommends PR curve under

two grounds, it helps to compare the performance of several classifiers for binary classification and find the classifier that can maximize detection precision with reasonable DR value. Considering the benefits of PR curve, this work presents the PR analysis for the proposed and its ablation methods on NSL-KDD dataset. An idle PR curve tends towards upper-right corner demonstrating best performance for binary classification.

Observing the Figure 8(a) and Figure 8(b), it is clear that all the three methods including proposed method display comparably equal detection performance on training set. However, the proposed method reveals better detection performance on testing set when compared to other ablation methods. Thus, PR analysis results are also in accordance with ROC analysis confirming that the proposed method is effective in detecting unseen new attacks compared to the designed ablation methods.

Similarly, the area under PR curve (PRAUC) presented in the legend section of Figure 8 on training and testing set are in agreement with AUC values presented in Figure 7. The obtained PRAUC values confirm that our approach is more efficient in gaining better detection performance with respect to DR and precision. Overall, the promising success of the proposed method over the designed ablations demonstrates the significant contribution of each component in achieving best detection performance even against unseen new attacks in testing set.

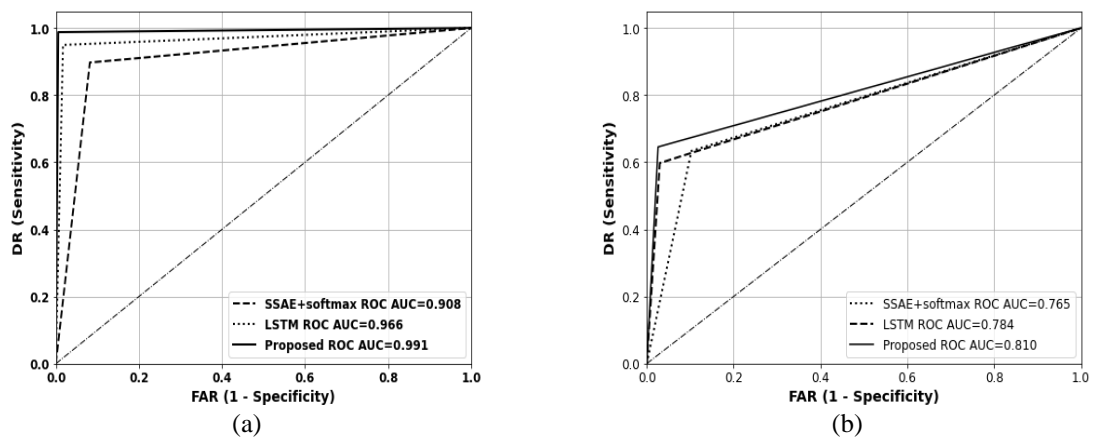


Figure 7. Comparison of ROC curve for proposed method against the designed ablations:  
(a) training set of NSL-KDD and (b) testing set of NSL-KDD

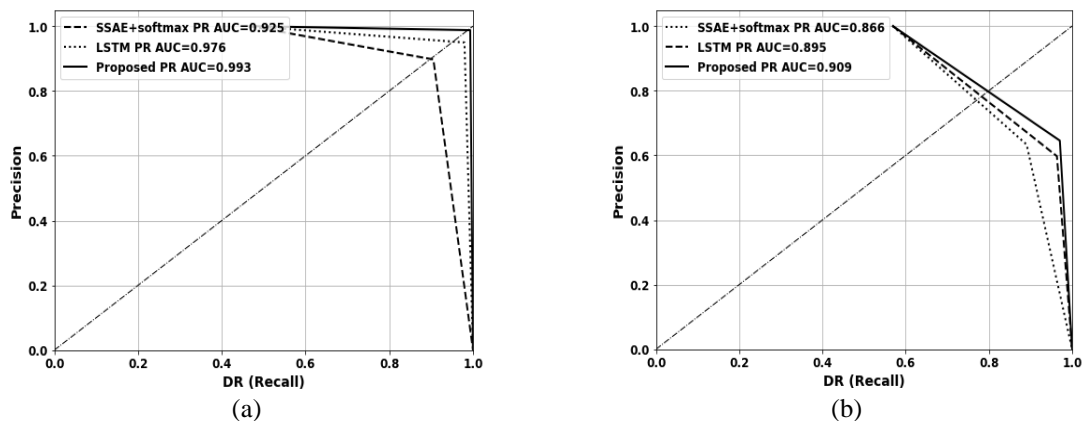


Figure 8. Comparison of PR curve for proposed method against the designed ablations:  
(a) training set of NSL-KDD and (b) testing set of NSL-KDD

#### 4.5. Comparative analysis

To further evaluate the effectiveness of our model, comparative analysis was conducted comparing the proposed model against the recent related deep learning-based IDS models. As it is very challenge to consider all recent deep learning approaches, only those IDS model that leverages the feature representation learning for intrusion detection on NSL-KDD dataset are considered for comparison. Also, to establish a reasonable comparison across all models, the results published by their respective authors are used.



Table 5 presents the comparative analysis results on NSL-KDD testing set. The visual analysis of the comparison performance of the proposed method against related works given in Table 5 is depicted in Figure 9. The result analysis indicates no single approach is best in all metrics. Also, it can be seen that our model displays comparably better performance than all other existing approaches except STL with SAE+SMR. Notwithstanding, it is clear that though STL with SAE + SMR outperforms the proposed model in terms DR and ACC. It is not evaluated in terms of FAR which is key metric with regard to intrusion detection. Hence, we confirm that STL with SSAE and LSTM can be recommended to boost the performance of the intrusion detection.

Table 5. Comparative analysis with related works

Related Works	DR	ACC	FAR	Related Works	DR	ACC	FAR
NADE [22]	85.42	85.42	14.58	STL SAE + SVM [24]	76.56	84.96	-
SAE [21]	85.36	86.02	15.50	STL SAE [25]	84.60	85.05	14.05
STL SAE+SMR [23]	90.50	88.39	-	Proposed model	86.92	86.31	9.25

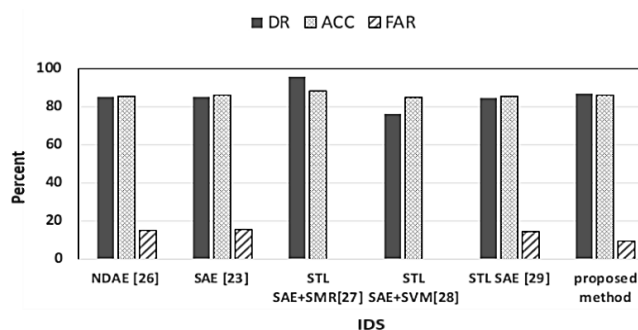


Figure 9. Performance comparison of proposed method with related works

## 5. CONCLUSION

In this work, a self-taught learning approach combining SSAE and LSTM network is proposed to recognize intrusion detection in cloud environment. The integration of SSAE within STL framework has enabled to address the requirement for large amount of labeled network traffic which is challenging in real practice. Further, the pretraining of SSAE network in an unsupervised learning manner with available normal network traffic samples has enables to boost the performance of intrusion detection. On other hand, the integration of LSTM has proved to be a candidate solution for learning the robust features representation to efficiently improve the performance of the IDS with respect to FAR, DR and ACC. The ablation and comparative analyses results have confirmed the contribution of SSAE and LSTM towards gain in intrusion detection against the existing related works.

## ACKNOWLEDGEMENTS

This study is supported via funding from Prince sattam bin Abdulaziz University project number (PSAU/2023/R/1445)

## REFERENCES




- [1] S. Wangfi, W. Wang, and Y. Tan, "Internet cross-border service model based on 5G environment and cloud computing data platform," *Microprocessors and Microsystems*, p. 103520, Nov. 2020, doi: 10.1016/j.micpro.2020.103520.
- [2] S. Shamshirband, M. Fathi, A. T. Chronopoulos, A. Montieri, F. Palumbo, and A. Pescapè, "Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues," *Journal of Information Security and Applications*, vol. 55, Dec. 2020, doi: 10.1016/j.jisa.2020.102582.
- [3] T. Vaiyapuri, "Deep Learning Enabled Autoencoder Architecture for Collaborative Filtering Recommendation in IoT Environment," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 487–503, 2021, doi: 10.32604/cmc.2021.015998.
- [4] K. Panetta, "Widespread artificial intelligence, biohacking, new platforms and immersive experiences dominate this year's Gartner Hype Cycle," *5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies*, 2018, pp. 4–7, 2018. [Online]. Available: <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>
- [5] C. A. Lee, R. B. Bohn, and M. Michel, *The NIST Cloud Federation Reference Architecture*. United States: National Institute of Standards and Technology, 2020. doi: 10.6028/nist.sp.500-332.
- [6] J. Oubaha, N. Lakkai, and A. Ouacha, "QoS routing in cluster OLSR by using the artificial intelligence model MSSP in the big data environment," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 2, pp. 458–466, Jun. 2021, doi: 10.11591/ijai.v10.i2.pp458-466.






- [7] D. Zhang, "Research on the Problems and Countermeasures of Computer Security under Cloud Computing Background," *Journal of Physics: Conference Series*, vol. 1648, no. 2, Oct. 2020, doi: 10.1088/1742-6596/1648/2/022046.
- [8] P. K. Suri, J. Sengupta, and P. Sharma, "Survey of intrusion detection techniques and architectures in cloud computing," *International Journal of High Performance Computing and Networking*, vol. 13, no. 2, pp. 184–198, 2019, doi: 10.1504/ijhpcn.2019.10018691.
- [9] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, pp. 110–120, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp110-120.
- [10] S. Sharipuddin *et al.*, "Intrusion detection with deep learning on internet of things heterogeneous network," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 3, pp. 735–742, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp735-742.
- [11] A. Binbusayyis and T. Vaiyapuri, "Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection," *Heliyon*, vol. 6, no. 7, Jul. 2020, doi: 10.1016/j.heliyon.2020.e04262.
- [12] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Jul. 2019, doi: 10.1186/s42400-019-0038-7.
- [13] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient Learning of Sparse Representations with an Energy-Based Model," in *Advances in Neural Information Processing Systems 19*, Sep. 2007, pp. 1137–1144, doi: 10.7551/mitpress/7503.003.0147.
- [14] T. Vaiyapuri and A. Binbusayyis, "Enhanced Deep Autoencoder Based Feature Representation Learning for Intelligent Intrusion Detection System," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3271–3288, 2021, doi: 10.32604/cmc.2021.017665.
- [15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
- [16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [17] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4507–4518, Jul. 2021, doi: 10.1109/tits.2020.3017882.
- [18] T. A. Rashid, P. Fattah, and D. K. Awla, "Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms," *Procedia Computer Science*, vol. 140, pp. 324–333, 2018, doi: 10.1016/j.procs.2018.10.307.
- [19] T. A. Rashid, M. K. Hassan, M. Mohammadi, and K. Fraser, "Improvement of Variant Adaptable LSTM Trained With Metaheuristic Algorithms for Healthcare Analysis," in *Advanced Classification Techniques for Healthcare Analysis*, IGI Global, 2019, pp. 111–131, doi: 10.4018/978-1-5225-7796-6.ch006.
- [20] A. Binbusayyis and T. Vaiyapuri, "Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach," *IEEE Access*, vol. 7, pp. 106495–106513, 2019, doi: 10.1109/access.2019.2929487.
- [21] A. Agarwal, P. Sharma, M. Alshehri, A. A. Mohamed, and O. Alfarrag, "Classification model for accuracy and intrusion detection using machine learning approach," *PeerJ Computer Science*, vol. 7, p. e437, Apr. 2021, doi: 10.7717/peerj-cs.437.
- [22] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/tetci.2017.2772792.
- [23] A. Javaid, Q. Niyaz, W. Sun, and M. 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)*, 2016, pp. 1–6, doi: 10.4108/eai.3-12-2015.2262516.
- [24] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/access.2018.2869577.
- [25] A. S. Qureshi, A. Khan, N. Shamim, and M. H. Durad, "Intrusion detection using deep sparse auto-encoder and self-taught learning," *Neural Computing and Applications*, vol. 32, no. 8, pp. 3135–3147, Mar. 2019, doi: 10.1007/s00521-019-04152-6.

## BIOGRAPHIES OF AUTHORS



**Thavavel Vaiyapuri**    is currently an Assistant Professor with the College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University. Her research interests include the fields of data science, security, computer vision, and high-performance computing. With nearly 20 years of research and teaching experience, she has published more than 50 research publications in impacted journals and international conferences. She is also a member of the IEEE Computer Society, and also a Fellow of HEA, U.K. She can be contacted at email: t.thangam@psau.edu.sa.



**Adel Binbusayyis**    is currently an Assistant Professor at the College of Engineering and Computer Science, Prince Sattam Bin Abdulaziz University, where he is a specialist in cybersecurity and technology transfer. He is also the Vice-Dean of e-learning with the Deanship of Information Technology and Distance Learning, Prince Sattam Bin Abdulaziz University. He is also an Advisor of the Vice-Rector with Prince Sattam Bin Abdulaziz University, where he is responsible for monitoring the performance executions of the university strategic goals. He can be contacted at email: a.binbusayyis@psau.edu.sa.