Naif Arab University for Security Sciences
Journal of Information Security and Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR

JISCR

Analyzing Autoencoder-Based Intrusion Detection System Performance: Impact of Hidden Layers

Seiba Alhassan[1,2,*], Gaddafi Abdul-Salaam[1], Michael Asante[1], Yaw Missah[1], Ernest Ganaa[2]

[1]Department of Computer Science, Kwame Nkrumah University of Science and Technology, Ghana.

[2]Department of ICT, Dr Hilla Limann Technical University, Ghana.

Received 31 Aug. 2023; Accepted 10 Dec. 2023; Available Online 17 Dec. 2023.

Abstract

The rise in cyberattacks targeting critical network infrastructure has spurred an increased emphasis on the development of robust cybersecurity measures. In this context, there is a growing exploration of effective Intrusion Detection Systems (IDS) that leverage Machine Learning (ML) and Deep Learning (DL), with a particular emphasis on autoencoders. Recognizing the pressing need to mitigate cyber threats, our study underscores the crucial importance of advancing these methodologies. Our study aims to identify the optimal architecture for an Intrusion Detection System (IDS) based on autoencoders, with a specific focus on configuring the number of hidden layers. To achieve this objective, we designed four distinct sub-models, each featuring a different number of hidden layers: Test 1 (one hidden layer), Test 2 (two hidden layers), Test 3 (three hidden layers), and Test 4 (four hidden layers).We subjected our models to rigorous training and testing, maintaining consistent neuron counts of 30 and 60. The outcomes of our experimental study reveal that the model with a single hidden layer consistently outperformed its counterparts, achieving an accuracy of 95.11% for NSL-KDD and an impressive 98.6% for CIC-IDS2017. The findings of our study indicate that our proposed system is viable for implementation on critical network infrastructure as a proactive measure against cyber-attacks.

I. Introduction

The rapid growth of computer network users has resulted in individuals, organizations, and businesses storing and transmitting sensitive information on these networks. These sensitive data have witnessed increased cyber-attacks, leading to data breaches, financial loss, intellectual property theft, reputational damage, identity theft, and other security issues. There is a need for countermeasures to minimize the spate of these attacks. Academia and industry have proposed measures such as Cryptography, Access control, firewalls, anti-virus, and Intrusion Detection Systems (IDS). IDS is the only technique that can be deployed to prevent insider and external attacks. Insider attacks emanate from people within an organization, while external attacks are those from outside the organization. According to [1], intrusion detection system is a crucial component of cybersecurity.

The intrusion detection system is hardware or software implemented to monitor a network or a

Keywords: Cybercrime, Autoencoder, Decode, Encode, Bottleneck, Neurons, IDS.

Production and hosting by NAUSS

* Corresponding Author: Seiba Alhassan
Email: alhseiba@gmail.com

doi: 10.26735/YLXB6430

host for unauthorized access and report for action to be taken. The importance of IDS has led to several researchers proposing various methods to improve the performance of existing IDS. [2] stated that Artificial Intelligence methods with data science have been proposed to solve the issue of network security.

[3], [4] Proposed Decision Trees classifier for NIDS. [5]–[8] proposed Naïve Bayes as a NIDS classifier. The aforementioned researchers recorded a significant improvement in their respective machine learning models. However, machine learning algorithms have weaknesses that make them unsuitable for NIDS research. The classical machine learning algorithm cannot handle the high dimensional data traffic witnessed by the current network[7]. This weakness has prompted researchers to focus on deep learning techniques. According to [9], deep learning algorithms can handle labeled and unlabeled data. The issues of false positive and false negatives associated with anomaly-based intrusion detection can be significantly reduced with deep learning[10]. The advantages of deep learning and the growing interest in its application in NIDS are evident from the increasing studies rate. [11]–[13] All applied Long Short-Term Memory (LSTM) to design intrusion detection systems. Deep neural networks have also been explored by [9], [14]–[16] to enhance the functionality of the intrusion detection systems in place. The autoencoder is another deep learning technique that has drawn a lot of interest. [2], [17]–[21] All mentioned the exceptional functioning of the systems they provided and offered various recommendations for improving the autoencoder algorithm's performance for NIDS. Depending on where the implementation is done, intrusion detection systems that use machine learning and deep learning techniques can be divided into two categories: host-based intrusion detection systems and network intrusion detection systems. A network intrusion detection system tracks data packets as they enter a computer network segment and uses data point analysis to report on any intrusions or attacks. While host-based intrusion detection requires installation on each host, NIDS has the benefit of just requiring one workstation to monitor the whole network. Additionally, host-based intrusion detection has the benefit of detecting encrypted attacks, but NIDS cannot detect encrypted attacks.

Depending on how they are implemented, intrusion detection systems (IDS) can also be divided into anomaly-based and signature-based categories. Any incoming traffic with a signature that differs from those in the database is regarded as an attack by signature-based intrusion detection systems, which keep a database of all known attack signatures.

There are some disadvantages associated with this type of IDS. Signature-based IDS is not capable of detecting unknown types of attacks. Besides, maintaining a database of all known attacks increases the computational cost. One major advantage of this type of IDS is the minimal number of false alarms rate. Anomaly-based intrusion detection, on the other hand, maintains a normal profile, and any incoming data packets that deviate from this normal profile are considered an attack. This type of IDS has the advantage of detecting new attacks, but its weakness is the high level of false alarms associated with it [22].

The industrial application of IDS is mostly achieved through signature-based IDS because of the false alarm associated with anomaly-based IDS. However, the inability of signature-based IDS to detect novel or new types of attacks makes it unsuitable for the current dynamic nature of network attacks. Anomaly-based intrusion detection system has become a hot research area[23]. The advantages of anomaly-based intrusion detection system call for improving the detection accuracy of existing anomaly-based IDS to reduce false alarm rates. Previous studies on IDS using autoencoders, according to [24], always report on manual and random turning of parameters, making them inconvenient for practical application.

In [24], a study was conducted to determine the possibility of the number of layers and bottlenecks affecting the performance of the autoencoder algorithm for IDS. They concluded that these parameters indeed affect the autoencoder performance. Their study, therefore, creates the need for effective ways of optimizing these parameters to improve the autoencoder algorithm to increase detection accuracy and reduce the false alarm rate.

A. Purposes of the study

This study aims to address the increasing frequency of cyberattacks targeting critical network infrastructure. We delved into the exploration of effective and accurate Intrusion Detection Systems (IDS) utilizing the capabilities of machine learning and deep learning, specifically emphasizing the use of autoencoders. The pressing need to mitigate cyber threats highlights the importance of advancing methodologies in the field of IDS.

The contributions of this paper include:

1) *The experimental design of the autoencoder networking, consisting of the number of neurons and the number of hidden layers, specifically starting from layer one.*
2) *Results from this experimental study will provide an optimized number of autoencoder layers for improved NIDS based on a constant number of neurons.*

II. Literature Review

A. The autoencoder algorithm

Autoencoders represent a category of artificial neural networks employed in unsupervised learning—a type of learning algorithm focused on analyzing and clustering unlabeled datasets. This class of algorithms is commonly known as unsupervised learning algorithms.

There are several advantages associated with using autoencoder as an IDS system learning algorithm. The most important one is its ability to learn useful representations of data in an unsupervised manner, making it very important when labelled training data is expensive. Its major components consist of an encoder and a decoder.

The autoencoder's dimensionality reduction feature enables it to learn a concise representation of input data, effectively reducing computational complexity. By focusing solely on relevant features, the use of an autoencoder diminishes the risk of overfitting. Additionally, autoencoders offer benefits such as facilitating transfer learning and adapting to various data types.

The specifics of the autoencoder's architecture are detailed in Fig. 1.

1) *Encoder:*
   a) *Input layer: This layer holds the input data.*
   b) *Hidden layers: The layers between the bottleneck and the input layer.*
   c) *Activation functions: ReLu (Rectified Linear Unit) and Sigmoid are commonly applied in an autoencoder.*

2) *Latent Space:*
   a) *It is the layer where the compressed version of the input data is stored.*
   b) *The size of the latent space is determined by the type of problem that has been solved.*

3) *Decoder:*
   a) *This layer progressively reconstructs the output layer, which is similar to the size of the input data.*
   b) *The nature of the input data determines the activation function used for the decoding phase. Binary data requires a sigmoid activation function, while continuous data is better represented using a linear activation function.*

Fig. 1 below shows the architecture of an autoencoder consisting of one hidden layer.

B. Related works

Numerous studies have applied the autoencoder algorithm to enhance network intrusion detection systems. This section provides an overview of these studies.

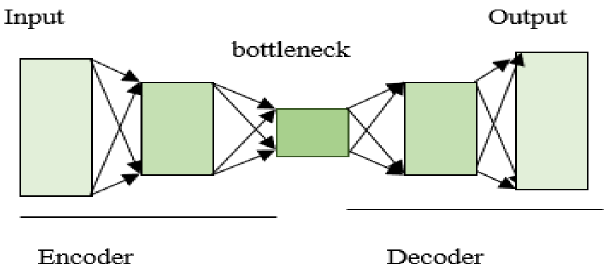

Fig. 1. Architecture of an Autoencoder.

[25] Illustrates the development of an efficient intrusion detection system for 5G and IoT networks based on deep learning. The authors introduce a hybrid system combining deep learning and data mining techniques. The system employs a deep autoencoder to perform unsupervised pre-training on the data, generating a concise and less noisy representation of the input space. The final component is a dense neural network acting as a supervised classifier for intrusion detection. The study provides insights into configuring and optimizing the hybrid intrusion detection system, including specific details on the parameter values of the applied model.

In [17], an effective deep learning method based on autoencoders is employed. The study underscores the challenges posed by the vast volume of data generated across networks and emphasizes the critical need for swift intrusion detection to prevent cyberattacks. It compares the superior intrusion detection capabilities of deep learning algorithms to traditional machine learning methods. To tackle class imbalance in intrusion detection datasets, the research utilizes the SMOTE approach. The deep autoencoder model is developed using the latest benchmark dataset, "CSE-CIC-IDS 2018", representing contemporary assaults. The study showcases promising results, considering all records and assault types in the dataset, claiming an average accuracy of 97.79%. Overall, it highlights the escalating demand for effective intrusion detection systems in response to the extensive network-generated data and emphasizes the potential of deep learning methods, particularly autoencoders, in enhancing intrusion detection precision. The application of SMOTE addresses class imbalance issues, preventing overfitting and ensuring reliable model performance.

While the above study provides valuable insights, it lacks specific technical details, including the architecture of the deep autoencoder model, evaluation measures, and a more in-depth analysis of experimental findings. The absence of references supporting the assertions made in the aforementioned paper makes it challenging to verify the accuracy of the research and the details of the dataset.

In [24], extracting features using a non-symmetric deep autoencoder is recommended. The essay discusses the challenges created by the exponential growth of network size and data, which has led to an increase in novel network attacks and called for the creation of accurate intrusion detection systems (IDS). The study emphasizes the importance of monitoring network traffic to thwart potential intrusions, highlight the value of IDS, and ensure network confidentiality, integrity, and availability. Despite extensive research efforts, IDS still needs to work on detecting new intrusions, reducing the number of false alarms, and increasing detection accuracy. To overcome these difficulties, machine learning (ML) and deep learning (DL) are emerging as viable methods for efficient intrusion detection across the network.

In [25], a novel IDS, termed AutoIDS, is introduced, utilizing an autoencoder-based approach within a semi-supervised machine learning framework. This IDS effectively discerns abnormal packet flows from normal ones by employing two efficient detectors, both constructed as encoder-decoder neural networks. These networks are trained to generate compressed and sparse representations of normal flows. During the testing phase, an intrusion is identified if the networks fail to produce the anticipated compressed or sparse representation from an incoming packet flow. To optimize computational costs while preserving accuracy, the first detector processes numerous flows, and the second detector is exclusively employed for challenging samples where the first detector exhibits uncertainty. The proposed AutoIDS undergoes evaluation on the widely-used NSL-KDD benchmark dataset, achieving an impressive accuracy of 90.17%, thereby demonstrating its superiority over alternative approaches.

III. Methodology

A. Overview

This section describes the use of autoencoders in the context of intrusion detection systems and presents a proposed system architecture. The key components of this section include the autoencoder model, datasets used, and metrics for evaluating the performance of the intrusion detection system.

B. Autoencoder

A deep autoencoder is an autoencoder with more than one hidden layer.

Autoencoder is a deep learning algorithm used to take input data (X) and compress it to a lower dimensional space known as a bottleneck (Z), and this whole process is known as encoding. Fig. 2 shows an autoencoder architecture showing the encoding and decoding phases. Equation (1) below shows the mathematical representation of the encoder.

$$Z = encoder(X) \qquad (1)$$

The bottleneck is then converted back to approximately the original data size, known as decoding. Equation (2) below shows the mathematical representation of the decoder.

$$(\hat{Y})\hat{} = decoder\ (Z) \qquad (2)$$

An error exists between the data fed as input(Y) and the output ((Y)) ˆ in training the autoencoder. The error is known as the reconstruction error. This error is an objective that needs to be optimized during training using loss functions. Common loss functions include Binary Cross Entropy (BCE) and the Mean Square Error (MSE). Equation (3) shows the mathematical representation of the Mean Square Error used in this study.

$$MSE = 1/n\ {*}\sum(Y-\hat{Y}\hat{}) \qquad (3)$$

This work's main objective is to minimize this reconstruction error and increase the detection accuracy for IDS.

C. Proposed system

The proposed system consists of a constant input neuron of size 30, and several varied hidden layers starting from layers one, two, three, and four, as shown in Fig. 3. The model is trained with two main datasets: the NSL-KDD dataset and the CIC-IDS2017 dataset. The performance of each layer in terms of detection accuracy is recorded. The process continues until the best possible accuracy is obtained.

D. Datasets

The datasets used to train, test, and validate our proposed system are the CIC-IDS2017 and NSL-KDD datasets. These datasets, in their raw form, cannot be run on deep learning algorithms such as Autoencoders. Therefore, both datasets were first prepared using the following steps:

1) Data cleaning, which involves removing duplicate data and handling missing values.

2) Normalization. The datasets are first normalized to enhance the performance and reliability of our model by converting all numeric columns to a common scale in the form of 1s or 0s. Equation (4) shows how the min-max technique performs the normalization task.

$$y = x\text{-}min/max\text{-}min \qquad (4)$$

Where y = new value of each entry

min = minimum value for each data point

max = maximum value for each data point

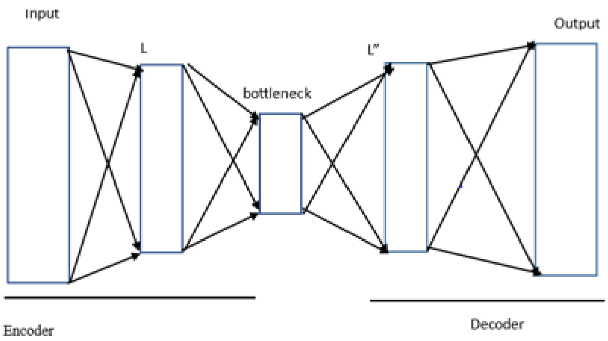

Fig. 2. Deep Autoencoder.

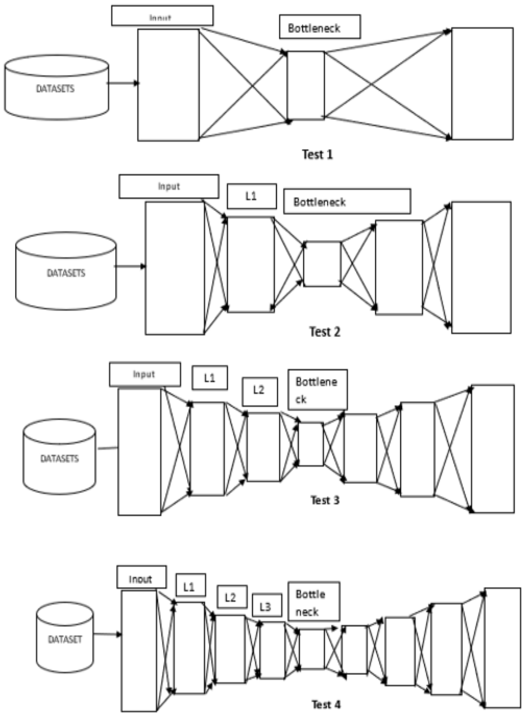

Fig. 3. Proposed system.

3) *Data Splitting. The data that has been transformed is then split into the ratio 75:25 for training and testing, respectively.*

E. Rationale for the use of NSL-KDD and CIC-IDS2017

The study used the NSL-KDD dataset because it is the most used dataset, providing an opportunity to compare our study with similar studies. This assertion is backed by [26], where a study was conducted on IDS and concluded that the most used datasets are KDDCup99 and NSL-KDD. However, according to [27], KDDCup99 contains redundant and duplicate records, which tends to make the result biased. Given the limitations of KDDCup99, NSL-KDD was selected for this study.

The NSL-KDD stated above is good for comparison, but it is an old dataset that does contain new attack types. An effective intrusion detection system requires that researchers in IDS use updated datasets such as CIC-IDS2017 [26]. The updated nature of CIC-IDS2017 is the main reason for this study.

F. Metrics of evaluation

The performance of intrusion detection systems is assessed using a variety of metrics, including accuracy, precision, F1-score, and recall. The others include:

True positive: Accurately categorized in a sample of data.

True negative: Normal traffic in an appropriately classified normal data sample.

False positive: A data sample's normal traffic was incorrectly categorized as an anomaly.

False negative: Malicious traffic mistakenly categorized as normal in a sample of data.

Mathematical representation of metrics of evaluation

Accuracy is the total number of data samples that were correctly identified. Equation (5) shows how the accuracy is calculated.

$$Accuracy\ (ACC) = \frac{TP+TN}{TP+TN+FP+FN} \qquad (5)$$

Recall, also called true positive rate, is the proportion of correctly predicted positive instances of a class to the overall instance of the same class. A higher recall rate that ranges from 0 to 1 indicates a better model performance. Equation (6) below shows how the Recall is calculated.

$$Accuracy\ (ACC) = \frac{TP+TN}{TP+TN+FP+FN} \qquad (6)$$

Precision is the ratio of positive instances correctly predicted to the ratio of all predicted samples for a class. Recall and Precision are always paired when evaluating model performance. Equation (7) shows how the Precision is calculated.

$$Precision = \frac{TP}{TP+FN} \qquad (7)$$

F1-score is computed by taking the harmonic mean of precision and recall. F1-score normally calculates the tradeoff between precision and recall. F1-score is calculated as shown in equation (8).

$$F1\text{-}score = 2^{*} \frac{Precision^{*}Recall}{Precision^{*}Recall} \qquad (8)$$

IV. Results and Discussion

A. Analyzing the performance of our proposed system

Four independent sub-experiments-Test 1, Test 2, Test 3, and Test 4-were set up as separate models to examine the performance of the autoencoder based on the number of hidden layers. NSL-KDD and CICIDS2017 preprocessed datasets were used to train the models. Our investigation maintained a consistent bottleneck size of X, X+2, X+4, and X+6, where X = 3 for the 30 and 60 neurons. This is because [1] conducted a similar study and found that the bottleneck impacts the model performance. In addition, our study kept the total number of neurons for each of the four sub-models at 30. In the next subsection we present the results depicting the loss versus the training and testing data for each deployed model. Fig. 4, 5, and 6 collectively illustrate a consistent reduction in the training and test data error rate across all models. This decline persists until the models have completed the learning process.

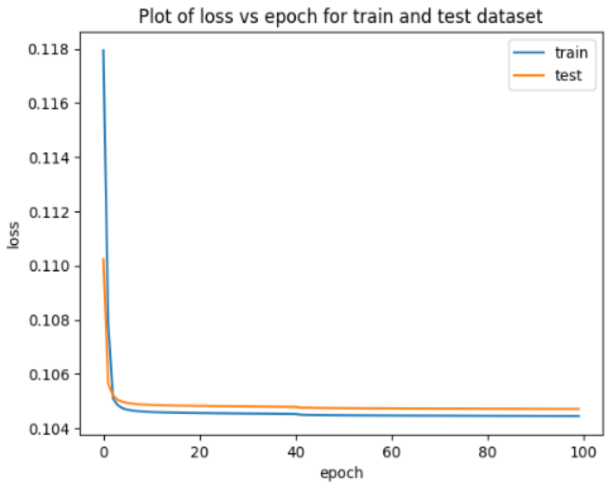

Fig. 4. Loss vs epoch for train and test dataset for two hidden layers autoencoder.

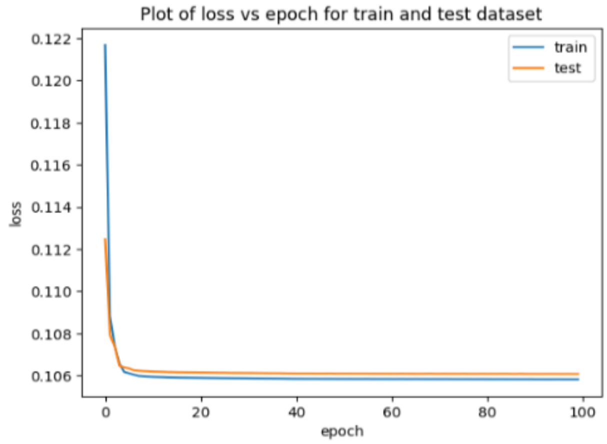

Fig. 5. Loss vs epoch for train and test dataset for three hidden layers autoencoder.

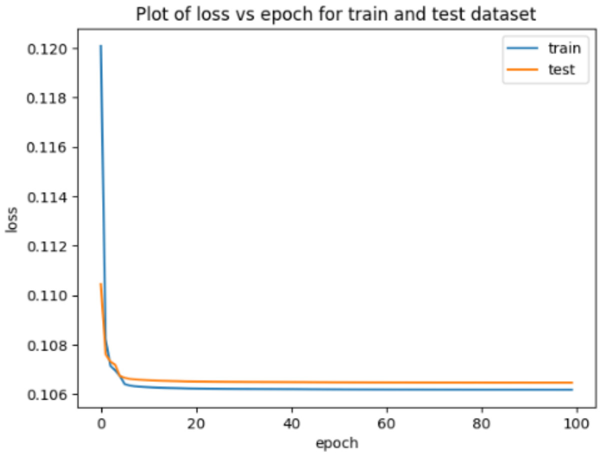

Fig. 6. Loss vs epoch for train and test dataset for four hidden layers autoencoder.

B. Results from NSL-KDD for 30 and 60 neurons

The output of the suggested system using the NSL-KDD dataset is shown in this section. The configuration was tested once, and the outcomes were recorded because this study aims to determine the model's performance based on various autoencoder hidden layers. The accuracy, precision, recall, and F1-Score of each model were noted throughout training and testing. The effectiveness of the suggested system based on the 30 neurons is displayed in Table I. According to Table I, Test 4, the model's performance was as follows: detection accuracy = 70.00%, precision = 65.55%, and recall = 75.00%. Furthermore, F-Score = 89.11% indicates the four models in the experimental configuration with the poorest performance. Accuracy, precision, recall, and F1-Score for Test 3 are 80.00%, 75.80%, 91.12%, and 93.33%, respectively. Test 2 also recorded an accuracy of 90.36, a precision of 88.49, a recall of 95.45, and an F1-Score of 96.82. Test 1, the model with only bottleneck or one-layer autoencoder, recorded the best performance with an accuracy of 92.45%, precision of 91%, recall of 96.02%, and F1-Score of 97.45%.

In addition to these metrics, the mean reconstruction error measured throughout training and testing was 0.104567 for Test 1, 0.104854 for Test 2, 0.104854 for Test 3, and 0.106460 for Test 4. The mean reconstruction error for each model is displayed in Fig. 7. The biggest reconstruction error was reported by Test 4, followed by Test 3, Test 2, and Test 1, which had the lowest reconstruction errors.

Similar results are shown in Table II for the proposed system utilizing the NSL-KDD dataset and 60 neurons. The suggested system obtained the following results from Table II for Test 4: accuracy = 80.91%, precision = 78.22, recall = 85.00%, and F1-Score of 92.50. Accuracy = 85.54%, precision = 80.77%, recall = 91.12%, and F1-Score = 97.00% were also recorded for Test 3. The best sub-model, Test 1, outperformed all other sub-models with an accuracy of 96.45%, precision of 93.00%, recall of 98.01%, and F1-Score of 98.63%. Test 2 maintained the second highest performing sub-model with an accuracy of 94.92%, precision of 91.00%, recall of 97.45, and F1-Score of 98.01.

TABLE I
Results of Ae Using 30 Neurons With Nsl-Kdd Dataset

Model	Accuracy	Precision	Recall	F1-Score
Test 4	70.00%	65.55	75.00%	89.11
Test 3	80.00%	75.80	91.12%	93.33
Test 2	90.36%	88.49	95.45%	96.82
Test 1	92.45%	91.00%	96.02%	97.45%

TABLE II
Results of Ae Using 60 Neurons With Nsl-Kdd Dataset

Model	Accuracy	Precision	Recall	F1-Score
Test 4	%.80.91	78.22%	85.00%	92.50%
Test 3	85.54%	80.77%	91.12%	97.00%
Test 2	94.92.36%	90.00%	97.45%	98.01
Test 1	96.54%	.93.00%	98.97%	98.63%

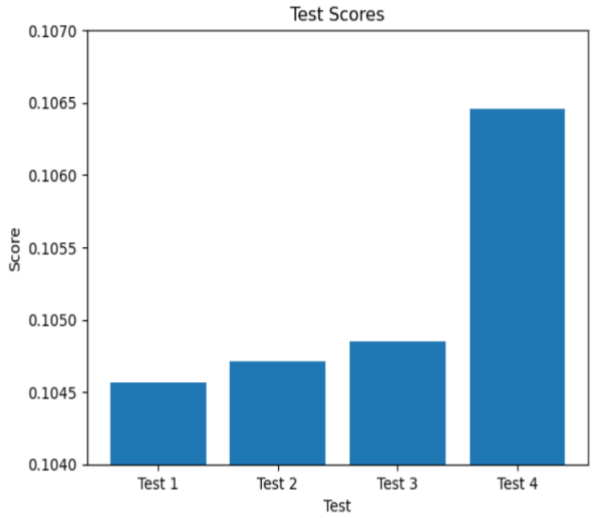

Fig. 7. Test vs mean reconstruction error.

TABLE III
The Performance of Ae Based on The Cic-Ids2017 Dataset For 30 Neurons

Model	Accuracy	Precision	Recall	F1-Score
Test 4	90.70%	84.55	86.00%	92.11
Test 3	92.60%	89.80	82.12%	90.33
Test 2	92.95%	82.49	93.45%	94.82
Test 1	95.11%	94.00%	98.88%	98.85%

C. Results from CIC-IDS2017

This section also shows the outcomes of the proposed system's training and testing using the CIC-IDS2017 dataset. The various models, including Tests 1, 2, 3, and 4, were each run once, with the results being recorded. Again, the study kept the number of neurons at 30 as a constant. The results achieved with the NSL-KDD dataset were comparable to those observed for this dataset however, the performance of the measured metrics improved in the CIC-IDS2017. For instance, the Test 1 score increased slightly from 96.45 for the NSL-KDD dataset to 97.11 for the CIC-IDS2017 dataset. The accuracy of Test 2 increased from 92.15 to 94.65, Test 3 from 92.15% to 92.45%, and Test 4 improved, going from 70.0% to 90.70%, as shown in Table III.

The autoencoder's performance using the CIC-IDS2017 dataset with 60 neurons is shown in Table IV. Based on the CIC-IDS2017 datasets, we evaluate the effect of the various layers on the functionality of our suggested system. We tested the performance of our suggested system using the sub-models, and we noted a similar pattern to that seen in phases 1 and 2 of the NSL-KDD and the CIC-IDS2017. Test 1 was the sub-model that performed the best, and Test 4 was the one that performed the worst. From Test 1 to Test 4, the reconstruction error increases by the prior pattern.

TABLE IV
The Performance Of Ae Based On Cic-Ids2017 Dataset For 60 Neurons

Model	Accuracy	Precision	Recall	F1-Score
Test 4	95.70%	90.00%	91.00%	92.11
Test 3	97.30%	94.40%	92.12%	90.33
Test 2	97.95%	95.11%	93.45%	94.82
Test 1	98.61%	97.00%	98.88%	98.15%

D. Discussion

According to the study's findings, given the same number of neurons, a single hidden-layer autoencoder performs better than a multi-layer autoencoder. In this work, 30 neurons, for example, were fed into the model for a single hidden layer autoencoder and the same number for the remaining multi-layer autoencoder layers. The single-layer autoencoder demonstrated excellent performance for every assessment parameter mentioned in this study, including recording the lowest reconstruction error. Given that single-layer autoencoders and multi-layer autoencoders use the same number of neurons, the maximum detection accuracy of 98.61% of Test 1 suggests that a single-layer autoencoder can detect intrusion more effectively than a multi-layer autoencoder. The results also demonstrate that, with the same number of neurons, the model's performance decreases as the number of layers increases.

For instance, the autoencoder with two hidden layers is recording the second-best performance, followed by the autoencoder with three hidden layers, and the autoencoder with four hidden layers is the least performing. The obtained reconstruction error verifies the proposed system's effectiveness.

The result from our study suggests that lesser reconstruction error will lead to improved model performance, while higher reconstruction error will lead to decreased model performance. The worst reconstruction error was achieved by Test 4, with Test 1 maintaining the lowest reconstruction error. We can, therefore, conclude with the following key findings:

1. The reconstruction error increases as the number of layers increases, given the same number of input neurons.
2. As reconstruction error increases, model performance will decline.

E. Methods comparison

A very good performance was registered when comparing the suggested system to another state-of-the-art research applying autoencoder with NSL-KDD and CIC-IDS2017 dataset. The performance of our suggested solution utilizing the NSL-KDD dataset and the CIC-IDS2017 dataset is shown in Table V.

TABLE V
Comparing Our Proposed System With Similar Studies

Author	Dataset	Accuracy
[28]	CIC-IDS2017	92.90%
[24]	NSL-KDD	84%
[29]	NSL-KDD	90.70%
[2]	NSL-KDD	90.61%
[30]	NSL-KDD	84.21%
Proposed system	NSL-KDD AND CIC-IDS2017	95.11%and 98.61%

V. Conclusion

In summary, the outcomes of this study strongly indicate that a single hidden-layer autoencoder surpasses multi-layer counterparts when utilizing the same number of neurons. The single-layer autoencoder consistently demonstrated remarkable performance across all evaluation metrics, particularly by attaining the lowest reconstruction error. Notably, the high detection accuracy of 98.6% in Test 1 further underscores the superiority of the single-layer autoencoder in intrusion detection compared to its multi-layer counterparts. Furthermore, the results suggest that as the number of layers in the autoencoder increases, the model's performance declines, even when maintaining an identical number of neurons. The configuration with a single hidden layer proved the most effective, achieving a reconstruction error of 0.106460 in Test 1, thereby confirming that a lower reconstruction error correlates with improved model performance.

Funding

This article did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

Authors declare that they have no conflict of interest.

References

[1] N. Alam and M. Ahmed, “Zero-day Network Intrusion Detection using Machine Learning Approach,” no. April, pp. 194–201, 2023.

[2] W. E. N. Xu, J. Jang-jaccard, A. Singh, and F. Sabrina, “Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset,” *IEEE Access*, vol. 9, pp. 140136–140146, 2021, doi: 10.1109/ACCESS.2021.3116612.

[3] R. Panigrahi *et al.*, “A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets,” *Mathematics*, vol. 9, no. 7, 2021, doi: 10.3390/math9070751.

[4] A. Ahmim, M. Derdour, and M. A. Ferrag, “An intrusion detection system based on combining probability predictions of a tree of classifiers,” *Int. J. Commun. Syst.*, vol. 31, no. 9, pp. 1–17, 2018, doi: 10.1002/dac.3547.

[5] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, and Y. L. He, “Fuzziness based semi-supervised learning approach for intrusion detection system,” *Inf. Sci. (Ny).*, vol. 378, pp. 484–497, 2017, doi: 10.1016/j.ins.2016.04.019.

[6] M. A. Rezvi, S. Moontaha, K. A. Trisha, S. T. Cynthia, and S. Ripon, “Data mining approach to analyzing intrusion detection of wireless sensor network,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 516–523, 2021, doi: 10.11591/ijeecs.v21.i1.pp516-523.

[7] S. N. Mighan and M. Kahani, “A novel scalable intrusion detection system based on deep learning,” *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, 2021, doi: 10.1007/s10207-020-00508-5.

[8] D. Karthikeyan, V. Mohanraj, Y. Suresh, and J. Senthilkumar, “An efficient stacking model with SRPF classifier technique for intrusion detection system,” *Int. J. Commun. Syst.*, vol. 34, no. 10, pp. 1–15, 2021, doi: 10.1002/dac.4737.

[9] M. R. Ayyagari, N. Kesswani, M. Kumar, and K. Kumar, “Intrusion detection techniques in network environment: a systematic review,” *Wirel. Networks*, vol. 27, no. 2, pp. 1269–1285, 2021, doi: 10.1007/s11276-020-02529-3.

[10] Z. M. Khan and H. Singh, “Deep Neural Network Solution for Detecting Intrusion in Network,” no. August, pp. 160–171, 2023.

[11] A. Issa and Z. Albayrak, “DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM,” no. January, 2023, doi: 10.12700/APH.20.3.2023.3.6.

[12] J. Han and W. Pak, “applied sciences Hierarchical LSTM-Based Network Intrusion Detection System Using Hybrid Classification,” 2023.

[13] J. Han and W. Pak, “High Performance Network Intrusion Detection System Using Two-Stage LSTM and Incremental Created Hybrid Features,” 2023.

[14] A. A. Megantara and T. Ahmad, “A hybrid machine learning method for increasing the performance of network intrusion detection systems,” *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00531-w.

[15] P. Pitre, “An Intrusion Detection System for Zero-Day Attacks to Reduce False Positive Rates,” pp. 1–6, 2022.

[16] S. Dwivedi, M. Vardhan, and S. Tripathi, “Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection,” *Cluster Comput.*, vol. 24, no. 3, pp. 1881–1900, 2021, doi: 10.1007/s10586-020-03229-5.

[17] C. Haripriya and M. P. P. Jagadeesh, “An Efficient Autoencoder-based Deep Learning Technique to Detect Network Intrusions,” vol. 13, no. 7, pp. 1–10, 2022, doi: 10.14456/ITJEMAST.2022.142.

[18] M. Sabir, J. Ahmad, and D. Alghazzawi, “A Lightweight Deep Autoencoder Scheme for Cyberattack Detection in the Internet of Things,” 2023, doi: 10.32604/csse.2023.034277.

[19] O. Tabanlı and D. Öğrenme, “Detection of Attacks in Network Traffic with the Autoencoder-Based Unsupervised Learning Method,” vol. 6, no. 2, pp. 199–207, 2022, doi: 10.26650/acin.1142806.

[20] Z. Gu, L. Wang, C. Liu, and Z. Wang, “Network Intrusion Detection with Nonsymmetric Deep Autoencoding Feature Extraction,” vol. 2021, 2021.

[21] M. Schmidt, “theRepository at St . Cloud State Autoencoder-Based Representation Learning to Predict Anomalies in Computer Networks,” 2020.

[22] H. Hosseinvand, “Intrusion Detection System Using SVM as Classifier and GA for Optimizing Feature Vectors,” vol. 10, no. 1, pp. 26–35, 2018.

[23] M. Awais, “Deep Learning Based Anomaly Detection for Fog-Assisted IoVs Network,” *IEEE Access*, vol. 11,

no. January, pp. 19024–19038, 2023, doi: 10.1109/ACCESS.2023.3246660.

[24] Y. Song, S. Hyun, and Y. G. Cheong, “Analysis of autoencoders for network intrusion detection†,” *Sensors*, vol. 21, no. 13, pp. 1–23, 2021, doi: 10.3390/s21134294.

[25] S. Rezvy, Y. Luo, M. Petridis, and A. Lasebae, “An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks”.

[26] E. M. Maseno, Z. Wang, and H. Xing, “A Systematic Review on Hybrid Intrusion Detection System,” vol. 2022, 2022.

[27] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set in Computational Intelligence for Security and Defense Applications,” *Comput. Intell. Secur. Def. Appl.*, no. Cisda, pp. 1–6, 2009.

[28] S. Sindian and S. Sindian, “An Enhanced Deep Autoencoder-based Approach for DDoS Attack Detection 3 Deep Neural Network 2 Related Work,” vol. 15, pp. 716–724, 2020, doi: 10.37394/23203.2020.15.72.

[29] M. Gharib and B. Mohammadi, “AutoIDS : Auto-encoder Based Method for Intrusion Detection System,” pp. 1–9.

[30] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, “A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach,” *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2019.11.016.