Soft Computing (2024) 28:4503–4517
https://doi.org/10.1007/s00500-023-09408-x

APPLICATION OF SOFT COMPUTING

A network intrusion detection framework on sparse deep denoising auto-encoder for dimensionality reduction

B. A. Manjunatha[1] · K. Aditya Shastry[1] · E. Naresh[2] · Piyush Kumar Pareek[1] · Kadiri Thirupal Reddy[3]

Accepted: 27 October 2023 / Published online: 30 November 2023

Abstract
In today's internet-driven world, a multitude of attacks occurs daily, propelled by a vast user base. The effective detection of these numerous attacks is a growing area of research, primarily accomplished through intrusion detection systems (IDS). IDS are vital for monitoring network traffic to identify malicious activities, such as Denial of Service, Probe, Remote-to-Local, and User-to-Root attacks. Our research focused on evaluating different auto-encoders for enhancing network intrusion detection. The proposed method sparse deep denoising auto-encoder approach produces the dimensionality reduction used to predict and classify attacks in datasets. With the most records among the datasets by training the auto-encoder on normal network data, this utilized reconstruction error as an indicator of anomalies. We tested our approach using standard datasets like KDDCup99, NSL-KDD, UNSW-NB15, and NMITIDS. Remarkably, our sparse deep denoising auto-encoder achieved an accuracy of over 96% based solely on reconstruction error. The primary aim of this work is to improve intrusion detection by achieving higher detection accuracy compared to existing methods.

Keywords Intrusion detection · Auto-encoder · Dimensionality reduction · Network security · NMITIDS · Support vector machine

1 Introduction

An intrusion detection system (IDS) is a specialized tool for analyzing and interpreting network and/or host behavior. These data can come from a variety of places, including network packet analysis, router, firewall, and server log files, local system logs and access calls, network traffic statistics, and other sources. An IDS may also compare patterns of activity, traffic, or behavior found in the data it monitors to those signatures to detect when a signature and current or recent behavior are virtually same.

Intrusion detection is a safety component that scans and analyzes web traffic for threats and alerts the system/set of connections administrator to take appropriate action. It is considered as second security gate between the firewall is given in Fig. 1. In the figure, IDS represents a critical component in ensuring the safety and security of networked systems. They play a pivotal role in scanning and analyzing web traffic, continuously monitoring for potential threats, and promptly alerting system administrators when suspicious activity is detected. In the hierarchy of network security, IDS stands as the second line of defense, following the firewall. The significance of IDS has grown exponentially in the realm of network security. As cybercriminals and hackers develop increasingly sophisticated techniques for infiltrating systems, the capabilities of IDS must evolve in tandem. It is imperative for an IDS to strike

✉ E. Naresh
naresh.e@manipal.edu

B. A. Manjunatha
manjunatha.ba@nmit.ac.in

K. Aditya Shastry
adityashastry.k@nmit.ac.in

Piyush Kumar Pareek
piyush.kumar@nmit.ac.in

Kadiri Thirupal Reddy
k.thirupalreddy2009@gmail.com

[1] Nitte Meenakshi Institute of Technology, Bengaluru, India

[2] Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India

[3] Bharat Institute of Engineering and Technology, Hyderabad, India

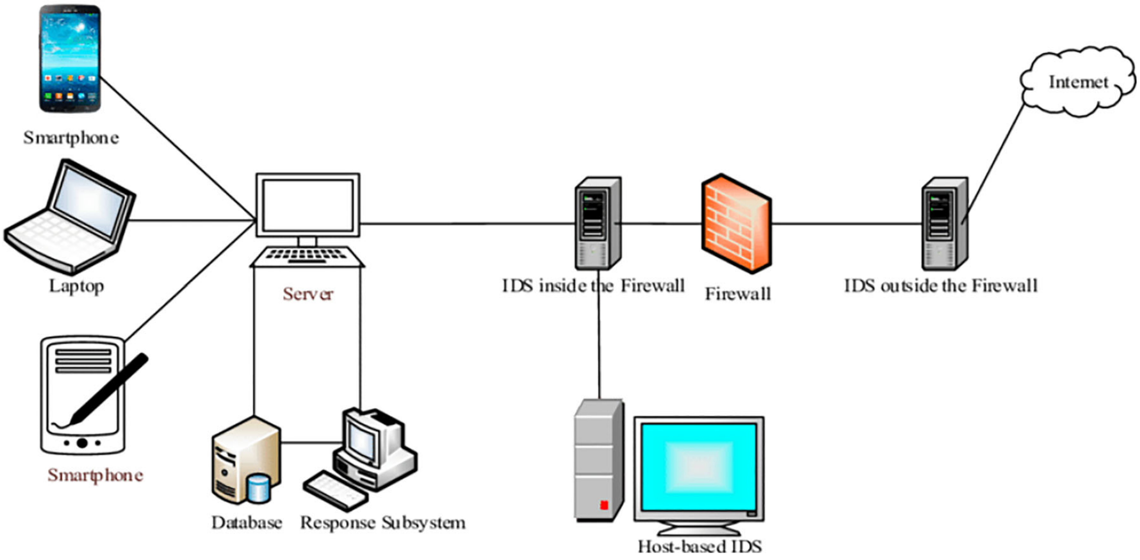


Fig. 1 Intrusion detection system architecture

a delicate balance between precision and speed. In terms of precision, an IDS must possess the capability to accurately identify and classify various types of intruders and malicious activities. This involves the continuous refinement of its detection algorithms and the ability to recognize both known and novel threats. It should be able to distinguish between normal network behavior and anomalies that may signify an attack or intrusion.

Moreover, the speed at which an IDS operates is of paramount importance. In today's digital landscape, threats can materialize swiftly, and an IDS must be agile enough to make rapid assessments and judgments. Delayed responses can result in severe security breaches and data loss.

Types of intrusion detection systems

1. Network intrusion detection system (NIDS) is deployed within a system's architecture to analyze incoming and outgoing traffic. Any deviations from the expected traffic patterns are reported to the network security team for action.
2. Host intrusion detection systems (HIDS) monitor all computers and devices within a network, capable of detecting threats originating from within the local system.
3. Signature-based IDS employs predefined rules to detect traffic deviations. It is a common method used in both NIDS and HIDS, known for its simplicity but reliant on rule quality.
4. Anomaly-based IDS passively analyze network traffic, typically using hardware or software tools for monitoring. These systems often have two network links for listening and control purposes.

Data intake, data pre-processing, feature reduction, and classification are all part of the IDS process. The KDDCup99 (Pervez and Farid 2014), NSL-KDD (Kddcup99 Public Dataset 2023), UNSW-NB15 (A. C. for Cyber Security 2023), and NMITIDS (Manjunatha and Gogoi 2020a) are publicly privately available standard datasets that will be utilized to develop and assess the system. The system's data pre-processing unit will execute data encoding by transforming symbolic data into numeric values, followed by data normalization for quick and accurate results.

The paper is structured as follows: the initial section covers the fundamentals of intrusion detection and its requirements, while the subsequent segment delves into the existing literature on intrusion detection. Section 3 elaborates on the methodologies employed in our research, and the outcomes are analyzed in Sect. 4. The paper concludes with Sect 5, which presents our conclusions and outlines potential avenues for further research.

2 Literature review

In this section, we have discussed most current research works which belongs to dimensionality reduction, and classification of intrusion detection.

Zhang et al. (2017) develop a system that uses LapSVM to integrate labeled and unlabeled data to improve classification results. The research gap is that the approach requires more memory and processing speed. For categorization, this strategy does not take into account all of the attack labels. For network intrusion detection, Al-Qatf et al. (2021) utilized a Deep Learning Approach combining sparse auto-encoder with SVM. The suggested method is used to learn features and reduce dimensionality. It significantly decreases training and testing time while significantly improving the prediction accuracy of support vector machines (SVM) in terms of assaults. For good representation features and dimensionality reduction processes, many stages of STL and a hybrid feature learning model were used. The accuracy for U2R and R2L attacks with their true-positive rate and false-positive rate is low. Tayel and Rizk (2021) propose the hybrid model for feature selection by combining the filter and wrapper-based approaches. The proposed model uses the best available clustering techniques and radian-based neural networks approaches in building the system. The IDS system is built using the clustering techniques, the artificial neural networks techniques and their types, such as feed forward neural network and radial basis neural network. Author proves that the proposed model improves the system performance. Complex and classification accuracy is not in line in with the selected artificial neural network models. Shakya and Makwana (2021) have used the combination of DBSCAN, K-mean++, and SMO algorithms for feature extraction. Not much work done on to find best and optimal value for classifier parameters. Needs to set and configure the appropriate value. Test needs to be done with different datasets. Comparison can be extended for other latest and generalized classifiers. It is observed that obtained accuracy is 96.922%. Nkiama et al. (2016a) address the elimination of irrelevant and redundant features, thereby producing the better classification accuracy. The selected features using this model will contribute to improve the detection rate, based on the score of each feature achieved during the selection process. Uses NSL-KDD dataset, no new dataset used for test and validation. As it performs the recursive operations, it takes long time for the process in achieving the optimized feature set. Lu et al. (2017) propose a hybrid feature selection method combining mutual information maximization and genetic algorithms, and the hybrid version is named as MIMAGA feature selection algorithm (Kumar et al. 2018). This method reduces the dimension of the original dataset features and removes the redundant records. The multiple classifiers are applied and evaluated for the performance results for the selected feature set, and the results shows the effectiveness of the selected model. Takes long time in processing the records as the gene expression data grow exponentially in size. Therefore, limited in resource application and memory space. Anbar et al. (2018) worked on analyzing the IPv6-based attacks and ICMPv6 DoS flooding, and classification was done using the decision tree, random forest, and k-nearest neighbor (k-NN) classification algorithms (Stefanova and Ramachandran 2017). The author analyzed the performance of three classification algorithms to detect the IPv6-based attacks. Not much work done on to find best and optimal value for classifier parameters. Needs to set and configure the appropriate value. Hoque et al. (2016) introduce the greedy feature selection method using mutual information in building the IDS system (Nkiama et al. 2016a). The combination of feature–feature and feature class mutual information is used to find get the suitable and optimized subset with low redundancy and maximization of relevancy across the features. This approach can be extended to other application types. Hybrid enhancements for optimized feature list, better detection rates. Kumar et al. (2018) propose a Machine Learning Classification Model in building the network-based intrusion detection system, and is mainly for the threats induced in mobile devices network. As we know, the threats in the mobile world increase rapidly, and the attackers steals the sensitive information, exploiting the users by sending unwanted SMS. The evaluation results show that the ML model can detect and classify known and unknown attacks with 99.4% accuracy. It can be combined with the other IDS feature selection and classification models in detecting and classifying the advanced and new threats, thereby reducing the false alarms. Shah et al. (2017) present network intrusion detection using sparse regression techniques and discriminative feature selection. SPLR may integrate feature extraction and categorization into a cohesive framework, unlike features extraction methods such as filter (ranking) and wrapper methods, which separate the feature selection and classification concerns. In identifying and categorizing sophisticated and novel threats, IDS uses selection and classification models, which reduces false alarms. Abualigah et al. (2021a) proposed Aquila optimizer (AO) is a novel population-based optimization method, which is inspired by the Aquila's behaviors in nature during the process of catching the prey, this algorithm simulates the behaviors of Aquila in nature. The author Abualigah et al. (2021b) proposed the Arithmetic Optimization Algorithm (AOA) excels in solving complex optimization problems, outperforming 11 other algorithms in various scenarios and applications. Abualigah (2019) presents an effective text document clustering method with broad applicability, demonstrating superior performance compared to comparable methods in various domains, including biomedical sciences. Zheng et al. (2020a) introduces a novel two-level data augmentation approach for automatic modulation classification in cognitive radios. It leverages interference-based spectrum augmentation to enhance the performance, showing superiority over existing methods on RadioML 2016.10a dataset. Qinghe Zheng's paper (Zheng et al. 2021) introduces the MR-DCAE model for identifying unauthorized radio broadcasting. It employs a specially designed auto-encoder with manifold regularization, achieving state-of-the-art performance on the AUBI2020 dataset. The paper (Zheng et al. 2022) presents the multi-scale radio transformer (Ms-RaT) for fine-grained modulation classification. It incorporates dual-channel representation and multi-scale analysis, outperforming the existing deep learning methods with comparable or lower computational complexity, as confirmed by simulation results and ablation studies. Zheng et al. (2020b) paper introduces Drop-path, a novel pruning method for 2D deep CNNs to reduce model parameters, addressing the computational cost challenge. Drop-path is evaluated on benchmark datasets, showing substantial model compression and acceleration with minimal accuracy loss. Zheng et al. (2023) introduces DL-PR, a priori regularization method for deep learning in automatic modulation classification (AMC). DL-PR enhances inter-class distance, reducing intra-class distance while maintaining signal information, improving AMC accuracy on diverse signal-to-noise ratios (SNRs). It outperforms other methods on the RadioML 2016.10a dataset with various deep learning models.

The discussed research works cover various aspects of intrusion detection, dimensionality reduction, and classification. Authors propose innovative methods, such as LapSVM, Deep Learning with Sparse Autoencoder, Hybrid Models for Feature Selection, and more, to enhance intrusion detection and classification accuracy. These approaches address issues like memory utilization, feature extraction, and model optimization. While some methods improve accuracy significantly, others focus on resource-efficient solutions. Additionally, research extends to diverse domains, including network-based intrusion detection, IPv6-based attack analysis, and mobile device network threats. These contributions aim to bolster network security and optimize performance in detecting both known and emerging threats. Many papers are provided that use a variety of data mining and deep learning strategies. Even though the correctness of the detection limit of abnormalities is good, there is always room for development in terms of intrusion detection accuracy and other metrics.

3 Proposed framework

For the IDS system, the suggested solution in this study effort employs a deep learning strategy. Artificial neural networks can filter through massive volumes of data to identify and categorize a variety of abnormal behaviors. An auto-encoder is a form of artificial neural network that can learn both linear and non-linear input representations and then use those representations to recreate the original data. When the auto-encoder is trained on the conventional network data, the reconstruction error (the difference between the original input and the reconstructed output) is often utilized to identify aberrant behavior. Using a suggested sparse deep denoising auto-encoder, a high degree of accuracy was achieved with low reconstruction error.

Auto-encoder An auto-encoder is a specific type of artificial neural network employed in machine learning and deep learning for the purpose of creating efficient representations or codings of unlabeled input data. This process is typically referred to as unsupervised learning, because the network learns to encode data without the need for labeled examples.

The primary objective of an auto-encoder is to capture essential features or patterns within the input data while filtering out noise or irrelevant information. By training the network to disregard input samples that do not contribute significantly to the representation, the auto-encoder strives to generate a compact encoding of the data. This often involves reducing the dimensionality of the data, which can be advantageous for various applications, such as feature extraction, compression, or denoising.

The architecture of an auto-encoder comprises two essential components: the encoder and the decoder. The encoder is responsible for mapping the input data into a new representation, typically of lower dimensionality than the input itself. This encoding is designed to capture the most salient features of the input. The decoder, on the other hand, aims to reconstruct the output as closely as possible to the original input data using the encoded representation.

The figure shown in Fig. 2 illustrates the architectural layout of an auto-encoder, showcasing its encoder and decoder components. The "code" mentioned refers to the middle or bottleneck layer of the artificial neural network, which represents the encoded data in a compressed form with a chosen dimensionality.

In essence, an auto-encoder is a versatile tool in machine learning that can uncover meaningful patterns in data, reduce its dimensionality, and facilitate various downstream tasks such as data compression or feature extraction. It is particularly valuable when dealing with unlabelled data and holds applications across multiple domains, including image processing, natural language processing, and more.

Denoising auto-encoder Its design resembles that of a standard auto-encoder. The major distinction is that the inputs are distorted to guarantee that the neurons/layers acquire more robust characteristics, resulting in greater generalization as shown in Eq. (1). Corruption may take in the input data, i.e., x. The corrupting procedure is not done during the testing phase

$$\tilde{x} = q_{\mathrm{D}}(\tilde{x}/x) \tag{1}$$

where q_{D} is an additive noise function, x is noise data, and x is input data. The loss is determined in this case as well, though, the issue is with the input layer rather than with the

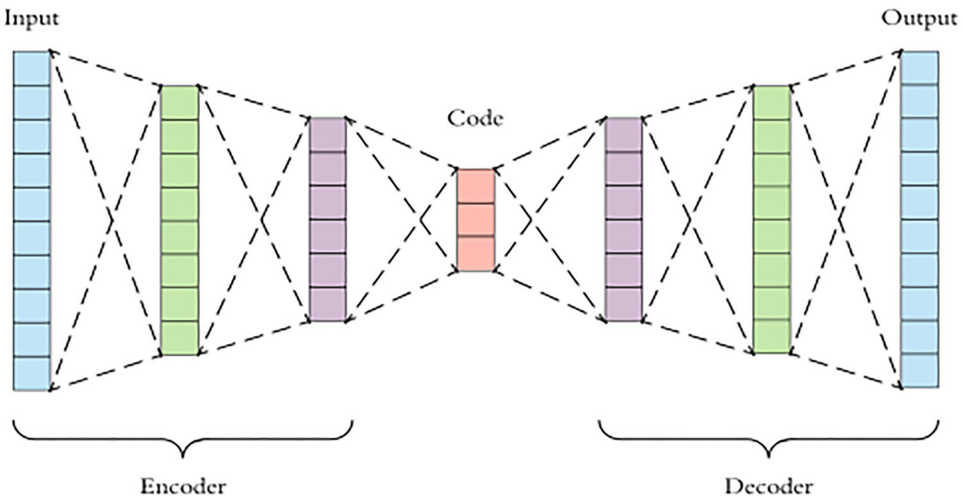


Fig. 2 Architecture of auto-encoder

defective inputs. Because it learns to rebuild the genuine inputs from corrupted inputs, the model's generalization has been proven to improve. Reconstruction error was employed to detect broken data packets once again. The prime work of DA is reconstructing from noise input data to noise-free output data.

3.1 Sparse auto-encoder for dimensionality reduction

Transferring an auto-encoder to the output layer simplifies the input data, which is an issue, because no relevant information is extracted. This encourages the auto-encoder to pick up and create succinct features with lower dimensionality. To put it another way, sparse restrictions are used to increase the precision of the input characteristics shown in Eq. (2). It took into account the average activation function, which is defined in the hidden layer

$$\widehat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right] \tag{2}$$

The auto change which might happen and which the hope is that the average activation function $\widehat{\rho}_j$ approaches ρ which is close to zero.

Kullback–Leibler (KL) divergence is added to the auto-encoder's loss function as a regularizer term this is given by Eq. (3)

$$\mathrm{KL}(\rho||\widehat{\rho}_j) = \rho\log\frac{\rho}{\widehat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\widehat{\rho}_j} \tag{3}$$

This is the result of combining entropy and cross-entropy. The KL divergence turns data point similarity into joint probability. In the next sections, we will look at how this term was added to the error function and how it helps with dimensionality reduction.

3.2 Deep sparse auto-encoder

Numerous sparse auto-encoders make up a deep sparse auto-encoder. The feed forward network operates by supplying the contribution of the next layer of the self-encoder as the yield of the previous layer. It enables the auto-encoder to recognize finer details while eliminating duplication. The weights and biases of the network are reduced, and a minimal squared error function value is obtained. This implies that altering the weights and bias will result in beneficial results. The Adam optimization method is utilized to accomplish dynamic parameter adjustment, and the greedy layer-by-layer pre-preparing technique is used to prepare each layer of the successively.

3.3 Proposed framework of sparse deep denoising auto-encoder (SDDA) dimensionality reduction method

This is a denoising auto-encoder with a sparse auto-encoder and a deep auto-encoder wrapped into one. The proposed SDDA, which may be a generative model used to solve a variety of issues. The SDDA will correctly reconstruct normal data after training, but will fail to do so with unexpected anomalous input. To discover anomalies, the reconstruction error (the difference between the actual data and its re-constructed data) is utilized to be the anomaly percent. The proposed SDDA method reconstructs from noise input data to noise-free output data. Our proposed method to take away the noise and produce the basic significant information such as accuracy in the data. In SDDA, after the input layer, there is a noise layer that corrupts the input and adds noise or masking a few of the input data. The input data x are converted into corrupted input x. The encoder uses a non-linear transformation for the input data x from High dimension P to low dimension Q in the most basic form. The name for this representation of the input is called encoder function or latent representation, as shown in Eq. (4)

$$g = \varepsilon(W\tilde{x} + b) \tag{4}$$

where W is the weight matrix, b is the bias vector, and ε is the active function, where 'g' is the encoder function and x is noisy input data as given in Eq. (1). The SDDA method include additional dropout layer, parameterized rectified linear activation function, cross-entropy, and L2 regularization term given next.

In the proposed SDDA approach, a dropout layer was added after the input layer. Dropout is a technique for preventing a model from overfitting. At each update of the training stage, Dropout indiscriminately sets the active edges of hidden units (neurons that make up hidden layers) to 0. With a likelihood (probability) of 0.5, there is a half change that the yield of a given neuron will be compelled to 0. It means for the specific probability at which layers of the output dropped out in hidden layers. The rescaling of the loads (weight) can be performed at training time considered, at the end of each concealed layer after each weight update. The output layer must be kept in the decoder the probability is close to 0.8 for retaining the output layers. From this dropout layer increases the generalization performance on each dataset. Denoising autoencoder with dropout and parameterized rectified linear activation function to achieve state-of-the-art results on an anomaly identification task.

3.4 Parameterized rectified linear activation function

An activation function is nothing but transfer function or squashing function, many active functions are non-linear. The various active functions give a huge impact on the accuracy and performance of the neural network. Generally, all hidden layers typically use the active function. Rectified linear activation (ReLU), logistic (Sigmoid), and hyperbolic tangent (Tanh) are the three most widely utilized activation functions for hidden layers (Anbar et al. 2016). ReLU activation function is simple and effective at overcoming the constraints of other existing famous activation functions, like Sigmoid and Tanh. ReLU functions at a hidden layer overcoming the limitations of vanishing gradients problem, but it can experience different issues like dying ReLU or dead units. To overcome these drawbacks in our proposed work used parameterized rectified linear activation (PReLU) function. The PReLU function is calculated as follows in Eq. (5):

$$\varepsilon = \max(0, \tilde{x}) + K\min(0, \tilde{x}) \tag{5}$$

where K is learning parameter. The SDDA decoder section that reconstruction of hidden patterns into visible representations (reconstruction of original feature set) is calculated as shown in Eq. (6)

$$x' = \varepsilon'(W'g + b') \tag{6}$$

where x' is reconstructed output, parameter W' is decoding weight, b' is decoding bias, g is encoder function, and ε' is shown in Eq. (7)

$$\varepsilon' = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{7}$$

For non-linear active functions, the reconstruction loss function is measured based on cross-entropy, as shown in Eq. (8)

$$L(x_i, x') = -\sum_{i=1}^{n}[x_i \log(x') + (1 - x_i)\log(1 - x')] + \alpha\sum_{i=1}^{n_i}\mathrm{KL}\left(\frac{\rho}{\hat{\rho}}\right) + \frac{\lambda}{2}||w||^2 \tag{8}$$

where n is number of samples, λ is coefficient of weight, ||w|| is L2-weight regularizer, α is weight coefficient of punishment, ni is number of hidden layers, and KL is divergence of Kullback–Leibler. To achieve sparsity, the Kullback–Leibler (KL) divergence is used as a regularizer term in the autoencoder's squared loss function. The sparse factor control using coefficient of weight factor error (λ). KL divergence changes close to similar data points focus on joint probabilities. The expansion of this term to loss function and it benefits dimensionality decrease as shown in Eq. (9)

$$\mathrm{KL}\left(\frac{\rho}{\hat{\rho}}\right) = \rho\log\frac{\rho}{\hat{\rho}} + (1 - \rho)\log\frac{1 - \rho}{1 - \hat{\rho}} \tag{9}$$

where $\hat{p}$ is an average activation of neurons in hidden layer, and ρ is a desired activation value of random neuron, if ρ is small value showing without a redundant features can be obtained in deep abstract set.

Next, to avoid overfitting, add the weight regularizer to the loss function. It is difficult to choose an acceptable learning rate for all network parameters when using these equations to address stochastic and mini-batch gradient descent issues. Use the adaptive moment (Adam) estimation technique proposed by Kingma and Ba to tackle this problem. Calculate the first-order moment estimate and the second-order moment estimate, such as m_{t} and v_{t}, in Algorithm 1 to update the dynamic network parameters. Following that, the formulae reveal first-order exponential damping decrements e_1 and second-order exponential damping decrements e_2 (10). In the loss function, the gradient parameter g_t is at timestamp t

$$m_t = e_1 m_{t-1} + (1 - e_1) g_t \tag{10}$$

$$v_t = e_2 v_{t-1} + (1 - e_2) g_t^2$$

where m_t and v_t is first and second moment estimate for computer bias corrected.

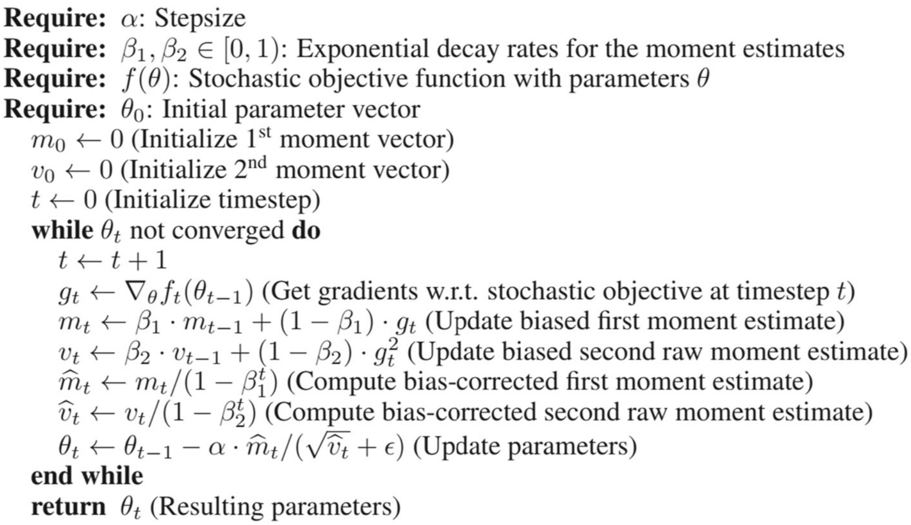
$$g_t \leftarrow \Delta_\theta J_t(\theta_{t-1})$$

Updated parameters

which focuses on joint probabilities between similar data points, aiding in dimensionality reduction.

3.4.1 Adam optimization algorithm

Algorithm 1 Adam optimization algorithm



```
Require: $\alpha$: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters $\theta$
Require: $\theta_0$: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while $\theta_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep $t$)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return $\theta_t$ (Resulting parameters)
```

$$\theta_{t-1} = \theta_t - \frac{\psi}{\sqrt{v_t} + \xi} m_t$$

where ψ is updated stepsize, ξ is small constant to avoid denominator to be 0(zero). For every iteration, the Adam optimizer optimizes the entire process reducing the weights and the bias units, taking out all the unwanted information out of the dataset. In the proposed SDDA approach, a dropout layer is introduced after the input layer to prevent overfitting during training. Dropout randomly deactivates a fraction of hidden units (neurons) during each training update, enhancing the model's generalization performance.

A key innovation in this framework is the use of the Parameterized Rectified Linear Activation Function (PReLU). While common activation functions like ReLU, Sigmoid, and Tanh are widely used, PReLU is introduced to address issues like vanishing gradients and dead units, thus enhancing the model's learning capabilities. The decoder in SDDA aims to reconstruct hidden patterns into visible representations, essentially mapping the encoded data back to its original form. The reconstruction process involves weight matrices (W'), bias vectors (b'), and the encoder function (g). To train SDDA effectively, several loss functions are introduced, including cross-entropy for non-linear activation functions, L2-weight regularization, and a Kullback–Leibler (KL) divergence term to achieve sparsity. Sparsity is controlled by a weight coefficient (λ),

After applying the feature reduction technique, the dataset's dimensionality is significantly reduced. The model is trained using fewer than 10 epochs, with the first hidden layer comprising 144 neurons and the second containing 150 neurons. Both of these layers incorporate L2 activity regularization with a coefficient of 10e−4. The model construction and training follow similar procedures to the previous networks. In our model, parameters, such as $p = 0.50$, $\lambda = 0.01$, $\beta = 3$, and the number of epochs is less than 10 for multiclass classification. This SDDA model improves upon the OLS-SVM approach, which is employed for categorizing the dataset records.

3.5 Classification of intrusion using OLS-SVM

To accelerate the training process and gain better accuracy and less false-positive rate, an instance will be considered as intrusion, since more anomalous classes within attack classes are present as compared to those that are benign or attack free. Our technique may be naturally modified to become cost-sensitive using OLS-SVM (Abualigah et al. 2021b), making it ideal for intrusion datasets. This proposed SDDA-OLS-SVM gives better accuracy and less false-positive rate with empirical results given next.

4 Experimental results

4.1 Standard measure: confusion matrix

A confusion matrix is a widely used statistic for classifier results. Actually, it is a table that describes how a classifier's test results are shown when the real values are known. There are two possible classes to expect: yes and no.

True-negative rate (TNR): The TNR counter is increased by one when the dataset record's actual class is abnormal, and the dataset record is classed as abnormal.

True-positive rate (TPR): If the dataset's real and categorized classes are the same (normal), the counter is increased.

False-positive rate (FPR): If an actual abnormal class record is classed as a normal record, the FPR counter is increased.

False-negative rate (FNR): When a normal class record is classed as an abnormal record, the FNR counter is incremented.

	Predicted no	Predicted yes
Actual no	TN	FP
Actual yes	FN	TP

Confusion matrix.

Accuracy Accuracy is the degree of information that is correct or precise is given in the equation

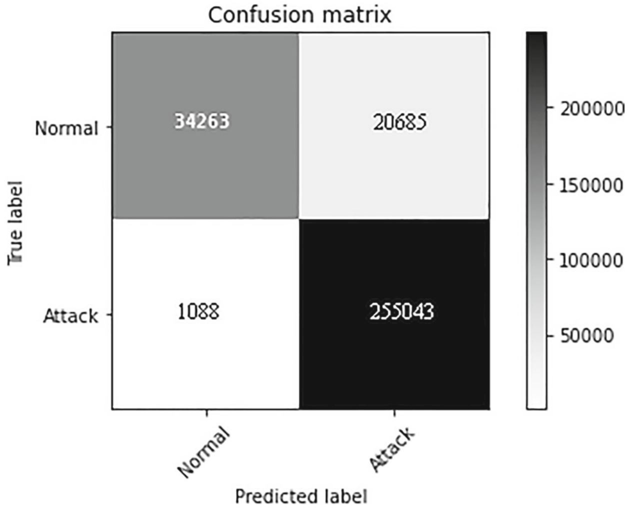


Fig. 3 Confusion matrix of deep auto-encoder

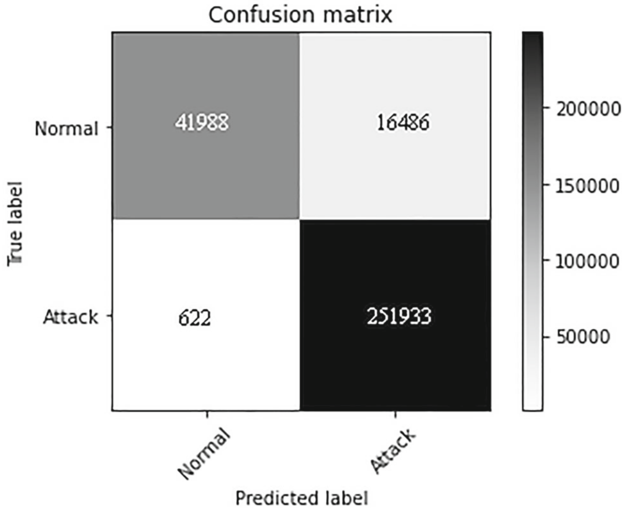


Fig. 4 Confusion matrix of denoising auto-encoder

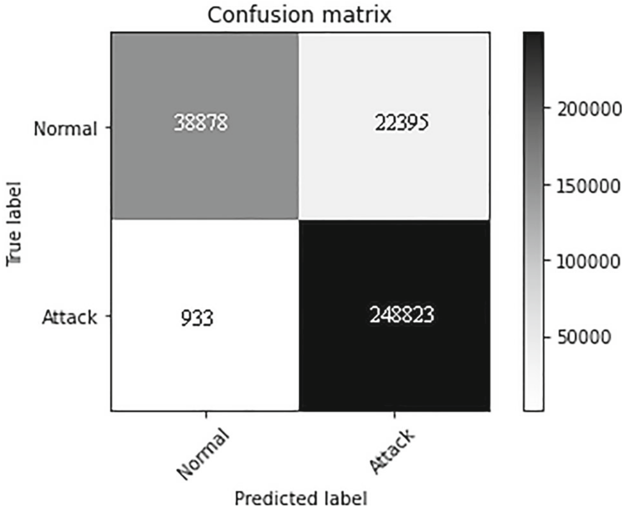


Fig. 5 Confusion matrix of sparse auto-encoder

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FN} + \text{FP})} \quad (11)$$

$$\text{False-positive rate} = \frac{\text{FP}}{\text{Actual no}}. \quad (12)$$

4.2 Results for KDDcup99

The KDDcup99 dataset (Kddcup99 Public Dataset 2023) is the most often used dataset in IDS research and is publicly available. The dataset was generated by MIT's Lincoln labs. It includes all records of both normal and attack types, and it makes up 10% of the original dataset as training data. In these datasets, each record is labeled as normal or attack, and each record provides information on 41 distinct

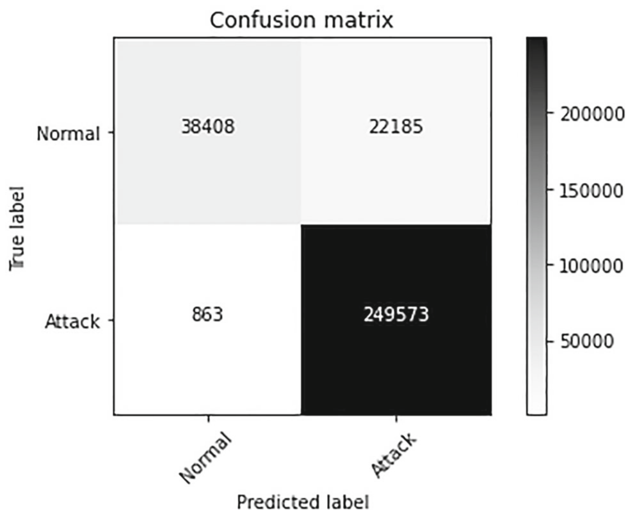


Fig. 6 Confusion matrix of sparse deep denoising auto-encoder

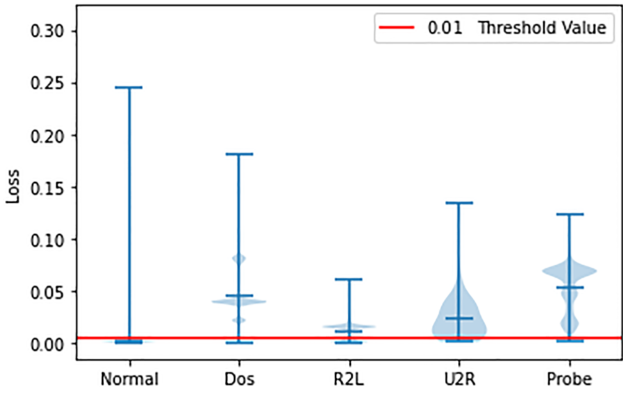


Fig. 7 Sparse deep denoising auto-encoder reconstruction error distribution on KDDcup99

attributes. The three categories of features are basic features, content-based features, and traffic-based feature groups.

The loss values of the individual models 10 epoch are used and establish the threshold value is 0.01. The cost vs epoch graph on KDDcup99 dataset is shown in Fig. 14a. A data point is considered as a regular data point if its reconstruction error is smaller than the threshold. If it is not, it is classified as an aberrant data point. A violin plot graph gives the whole distribution along with the probability density function, median, and mode information. It is a combination of box plot and probability density of the data as shown in results (Fig. 3).

The deep auto-encoder achieves an accuracy of 85.78% and false-positive rate is 07.50% on this KDDcup99 dataset. U2R packets were detected as attack packets with 99.24% accuracy, as shown in Fig. 3.

On the whole, the denoising auto-encoder is 84.91% accurate and false-positive rate is 6.14%. It is 99.55% accurate in identifying U2R packets as assault packets as shown in Fig. 4.

The sparse auto-encoder correctly classifies packets as attack packets with an accuracy of 85.27%, 8.25% is false-positive rate on the whole dataset, and 83.58% on the U2R packets, as shown in Fig. 5.

On the dataset, the hybrid auto-encoder obtains an accuracy of 94.68%, with a less false-positive rate accuracy of 5.16% on the whole dataset. The 84.28% on U2R packets is shown in Fig. 6. The violin graph depicts the distribution of reconstruction error for sparse deep denoising auto-encoder in relation to this attack shown in Fig. 7.

The sparse deep denoising auto-encoder clearly beats the other Auto-encoder kinds which is shown in Table 1. It was able to accomplish so with only a reconstruction inaccuracy. The capacity of the model to recognize the (virtual) absence of U2R packets from the training data has no effect on their attack packets. One of the key benefits of a denoising auto-encoder-based anomaly detection system is that it learns the distribution of a certain type of data and utilizes it to identify other data types from it.

4.3 Results for NSL-KDD dataset

Despite the widespread usage of KDDcup99, several limitations, such as a large amount of data, duplicate records, and so forth, will make getting high-performance results difficult. The efficiency of the IDS system will be harmed as a result of these issues. The revised version of NSL-KDD (Pervez and Farid 2014) will fix these issues. The dataset is analyzed, and duplicate and superfluous records are removed. As a consequence, in terms of operation speed and accuracy, this dataset surpasses the KDDcup99. This dataset has the same 41 normal and attack label features as KDDcup99.

Table 1 Comparison of results for all classification algorithms on KDD-cup 99 dataset

Models	Total	Normal	Dos	R2L	U2R	Probe	FPR
Deep-OLS-SVM	86.78	76.56	93.03	88.69	99.24	99.95	7.50
Denoising-OLS-SVM	84.91	70.62	93.54	98.04	99.55	99.95	6.14
Sparse-OLS-SVM	85.27	72.35	96.12	87.85	83.58	99.87	8.25
SDDA-OLS-SVM	94.68	87.41	96.33	80.89	84.28	99.87	5.16

Table 2 Comparison of results for all classification algorithms on NSL-KDD dataset

Models	Total	Normal	Dos	R2L	U2R	Probe	FPR
Deep-OLS-SVM	95.98	91.05	93.78	86.36	97.17	99.10	2.89
Denoising-OLS-SVM	96.91	90.82	94.14	89.48	96.92	99.99	2.78
Sparse-OLS-SVM	93.87	89.65	96.12	90.84	98.98	97.07	1.59
SDDA-OLS-SVM	98.21	91.24	93.54	90.64	98.89	99.99	1.01

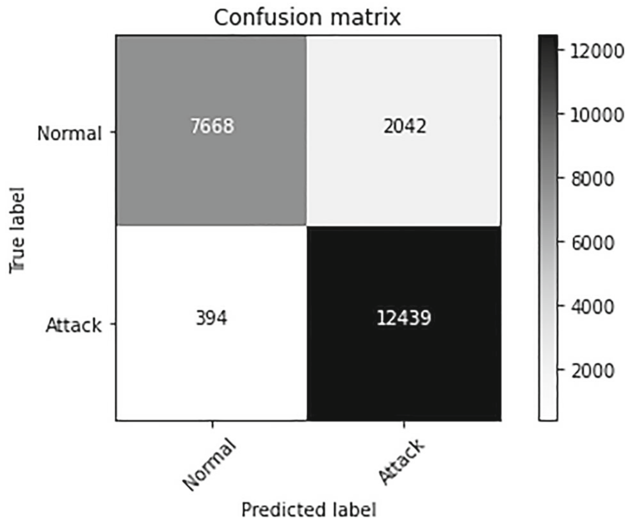


Fig. 8 Confusion matrix using SDDA on NSL-KDD

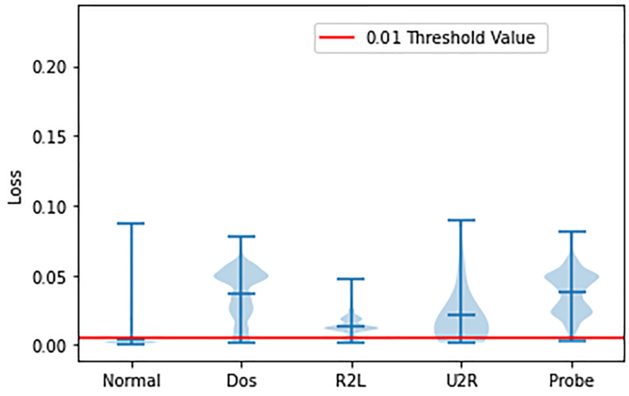


Fig. 9 Sparse deep denoising auto-encoder reconstruction error distribution on NSL-KDD

The loss values of the individual models 10 epochs are used and establish the threshold value is 0.01. The cost vs epoch graph on NSL-KDD dataset as shown in Fig. 14b. A data point is considered as a regular data point if its reconstruction error is smaller than the threshold. If it is not, it is considered an anomalous data point. Because of duplicated records, the NSL-KDD dataset has a greater accuracy rate than the KDDcup99 dataset. The accuracy percentage is greater than 98%, but not quite 99%, as shown in the Table 2.

The sparse deep denoising auto-encoder (SDDA) clearly beats the other autoencoder kinds is shown in the Table 2. With just the reconstruction error, it was able to do so. The model's ability to identify U2R and R2L packets as attack packets was not hampered by their (virtual) absence in the training data. One of the main advantages of a denoising auto-encoder-based anomaly detection system is that the model learns the distribution of a certain type of data and uses it to distinguish other data types from it.

More specifically, in deep auto-encoder, total accuracy of 95.98% and U2R and R2L attacks packets detect 86.36 and 97.17%. The denoising auto-encoder is 96.91% accurate, and it is 96.92% accurate in identifying U2R packets as assault packets. The sparse auto-encoder correctly classifies packets as attack packets with an accuracy of 93.87% on the whole dataset and 90.84% on the R2L packets, as shown in Table 2. The proposed SDDA method total accuracy of 98.21 percent and false-positive rate is 1.01%. U2R and R2L attacks detect 90.64 and 98.89% accuracy for NSL-KDD. The confusion matrix using SDDA on NSL-KDD is shown in Fig. 8. The graph depicts the distribution of reconstruction error for sparse deep denoising auto-encoder in relation to this attack shown in Fig. 9.

4.4 Results for UNSW-NB15 dataset

Moustafa and Slay created this dataset in 2015 (A. C. for Cyber Security 2023), which is a mix of real-time and simulated network traffic attack activities. In comparison to KDDcup99, this dataset has nine different attack types. There are 49 distinct features in all, vs 41 in the KDDcup99. The loss values of the individual models 10 epochs are used and establish the threshold value is 0.01. The cost vs epoch graph on UNSW-NB15 dataset, as shown in Fig. 14c. When the reconstruction error of a data point is less than the threshold, it is classified as a regular data point. If it is not, it is classified as an aberrant data point. The UNSW-NB15 accuracy rates on various auto-encoder models are more than 92% as shown in Table 3.

The proposed method sparse deep denoising auto-encoder (SDDA) clearly beats the other Auto-encoder kinds is shown in Table 3. The accuracy of SDDA model is 96.57 and 2.20% false-positive rate. More specifically, in deep auto-encoder, total accuracy of 92.87 and 3.6% false-

Table 3 Comparison of results for all classification algorithms on UNSW-NB15 dataset

Models	Total	Normal	Worms	Shell code	Reconnaissance	Analysis	Generic	Backdoor	DOS	Exploits	fuzzer	FPR
Deep-OLS-SVM	92.87	89.65	86.12	90.84	92.95	91.07	89.65	92.12	90.84	91.98	92.07	3.60
Denoising-OLS-SVM	94.89	89.82	88.14	94.48	98.92	91.99	91.82	88.14	93.48	86.92	91.99	3.01
Sparse-OLS-SVM	94.01	87.05	86.78	92.36	97.17	92.10	89.05	93.78	86.36	93.17	92.10	3.00
SDDA-OLS-SVM	96.57	95.24	93.54	94.64	97.98	94.99	96.24	93.54	94.64	97.89	97.99	2.20

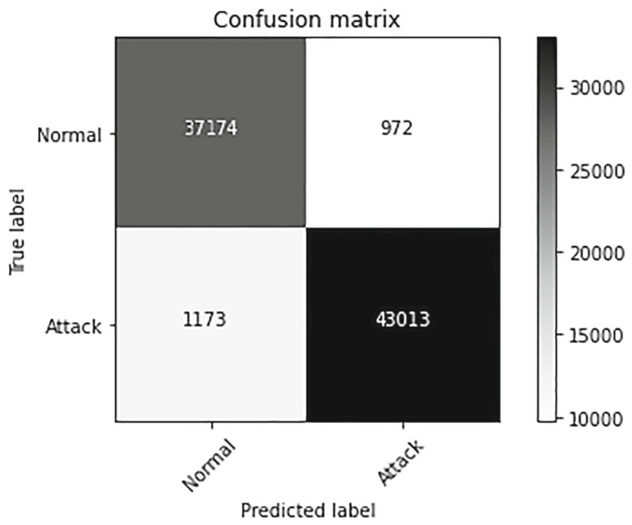


Fig. 10 Confusion matrix using SDDA on UNSW-NB15

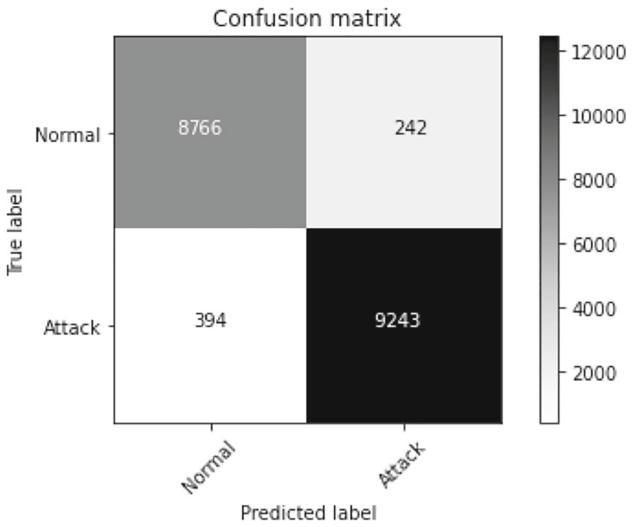


Fig. 12 Confusion matrix using SDDA on NMITIDS

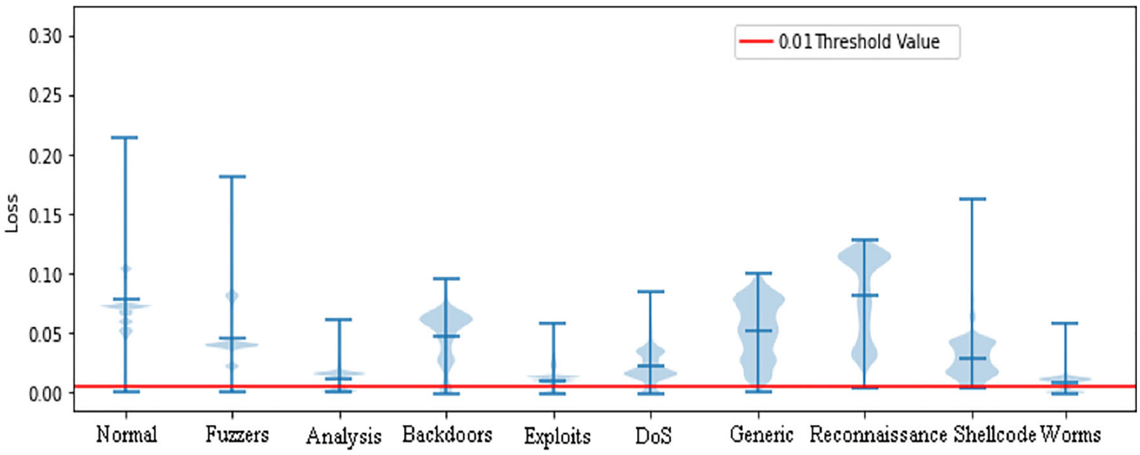


Fig. 11 Sparse deep denoising auto-encoder reconstruction error distribution on UNSW-NB15

Table 4 Comparison of results for all classification algorithms on NMITIDS dataset

Models	Total	Normal	Dbot	Mydoom	SSH	FPR
Deep-OLS-SVM	99.00	98.95	98.78	99.36	99.99	0.28
Denoising-OLS-SVM	99.30	99.12	98.14	99.48	99.99	0.23
Sparse-OLS-SVM	99.01	98.99	97.51	99.84	99.99	0.27
SDDA-OLS-SVM	99.35	99.24	98.54	99.64	99.99	0.20

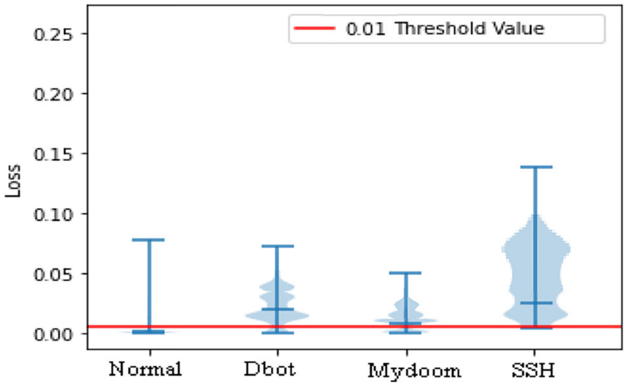


Fig. 13 Sparse deep denoising auto-encoder reconstruction error distribution on NMITIDS

positive rate. Similarly, for denoising auto-encoder is 94.89 and 3.01% false-positive rate. The sparse auto-encoder correctly classifies packets as attack packets with an accuracy of 94.01% and false-positive rate is 3% on the whole dataset as Table 3 shows the results. Figure 10 shows the confusion matrix obtained using SDDA on UNSW-NB15. Figure 11 depicts the distribution of reconstruction error for the sparse deep denoising auto-encoder in relation to this assault. One of the main advantages of a denoising auto-encoder-based anomaly detection system is that the model learns the distribution of a certain type of data and uses it to distinguish other data types from it.

4.5 Results for NMITIDS dataset

The NMITIDS dataset (Manjunatha and Gogoi 2020a) is consistent, consists of real-time network data. This dataset consists of 8,97,182 records, six type of attacks, 31 features, and several protocols, such as IP, TCP, UDP, ICMP, SSH, DNS, FTP, HTTP, ARP, etc. Finally, the NMITIDS dataset is split into train and test subsets. The loss values of the individual models 10 epochs are used and establish the threshold value is 0.01. The cost vs epoch graph on NMITIDS dataset is shown in Fig. 14d. A data point is considered as a regular data point if its reconstruction error is smaller than the threshold. If it is not, it is considered an anomalous data point.

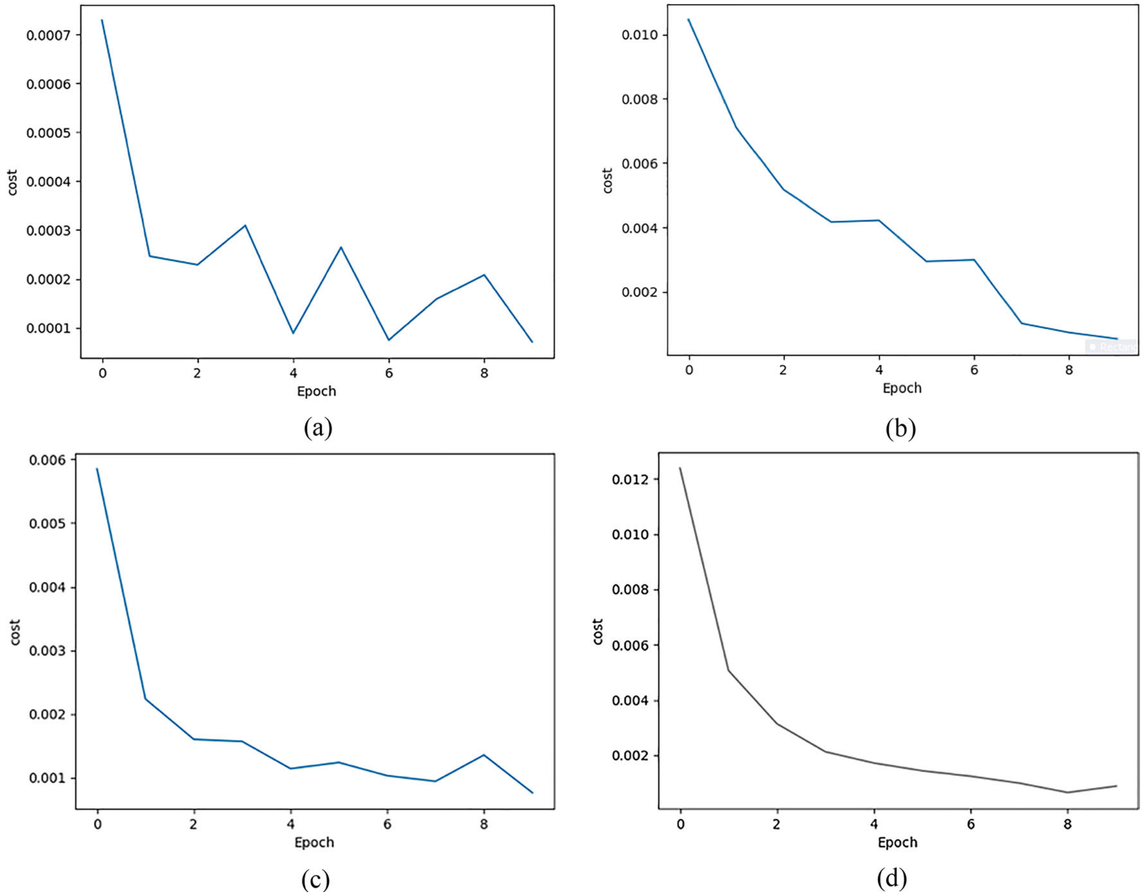


Fig. 14 **a** Cost vs epoch graph on KDD-cup 99. **b** Cost vs epoch graph on NSL-KDD. **c** Cost vs epoch graph on UNSW NB15. **d** Cost vs epoch graph on NMITIDS

Table 5 Comparison of the proposed technique with the existing methods

Classifiers	NSL-KDD		KDDcup- 99		UNSW NB15		NMITIDS	
	Accuracy	FPR	Accuracy	FPR	Accuracy	FPR	Accuracy	FPR
J48 (Qureshi et al. 2020)	81.05	NA	NA	NA	NA	NA	NA	NA
Naıve Bayes (Qureshi et al. 2020)	76.56	NA	NA	NA	NA	NA	NA	NA
NB tree (Qureshi et al. 2020)	82.02	NA	NA	NA	NA	NA	NA	NA
Random tree (Qureshi et al. 2020)	81.59	NA	NA	NA	NA	NA	NA	NA
MLP (Qureshi et al. 2020)	77.41	NA	NA	NA	NA	NA	NA	NA
NLPCA (Qureshi et al. 2020)	76.73	NA	NA	NA	NA	NA	NA	NA
DBN (Qureshi et al. 2020)	76	NA	93.49	NA	NA	NA	NA	NA
AE + DBN (Narayana Rao et al. 2021)	NA	NA	92.10	NA	NA	NA	NA	NA
CFA (Narayana Rao et al. 2021)	NA	NA	91.98	3.9	NA	NA	NA	NA
Multi-scale CNN (Narayana Rao et al. 2021)	NA	NA	94.11	2.18	NA	NA	NA	NA
STL-IDS (Al-Qatf et al. 2018)	80.48	NA	93.96	NA	NA	NA	NA	NA
SAE-SVM (Narayana Rao et al. 2021)	80.48	NA	NA	NA	NA	NA	NA	NA
SVM + ELM (Al-Qatf et al. 2018)	84.96	NA	95.75	NA	NA	NA	NA	NA
s-NDAE (Narayana Rao et al. 2021)	85.42	14.5	NA	NA	NA	NA	NA	NA
ID-CAVE (Narayana Rao et al. 2021)	80.10	8.18	NA	NA	NA	NA	NA	NA
Savaer-DNN (Narayana Rao et al. 2021)	89.36	4.70	NA	NA	93.01	5.67	NA	NA
DAE-IDS (Al-Qatf et al. 2018)	NA	NA	94.71	8.12	NA	NA	NA	NA
TSDL (Narayana Rao et al. 2021)	NA	NA	NA	NA	89.13	0.74	NA	NA
Cascade-ANN (Narayana Rao et al. 2021)	NA	NA	NA	NA	86.40	13.1	NA	NA
Random forest (Manjunatha and Gogoi 2020b)	97.53	0.9	95.49	0.3	95.11	0.1	95.67	0.8
PCA (Manjunatha and Gogoi 2020b)	97.53	0.8	93.23	0.1	92.77	0.8	91.10	0.6
SVM (Manjunatha and Gogoi 2020b)	89.63	0.1	93.60	0.7	89.99	0.27	93.25	0.9
Deep-OLS-SVM	95.98	2.89	86.78	7.50	92.87	3.60	99.00	0.28
Denoising-OLS-SVM	96.91	2.78	84.91	6.14	94.89	3.01	99.30	0.23
Sparse-OLS-SVM	93.87	1.59	85.27	8.25	94.01	3.00	99.01	0.27
SDDA-OLS-SVM	98.21	1.01	94.68	5.16	96.57	2.20	99.35	0.20

The sparse deep denoising auto-encoder (SDDA) clearly beats the other auto-encoder kinds as shown in Table 4. With just the reconstruction error, it was able to do so. The model's ability to identify Dbot, Mydoom, and SSH packets as attack packets was not hampered by their (virtual) absence in the training data. One of the main advantages of a denoising auto-encoder-based anomaly detection system is that the model learns the distribution of a certain type of data and uses it to distinguish other data types from it.

More specifically, in deep auto-encoder, total accuracy of 99.00% and false-positive rate is 0.28% packets detects. The denoising auto-encoder accuracy of 99.30% and false-positive rate is 0.23% packet detects. The sparse auto-encoder correctly classifies packets as attack packets with an accuracy of 99.01 percent on the whole dataset and 0.27% for false-positive rate as shown in Table 4. The Proposed SDDA method's total accuracy of 99.35% and false-positive rate is 0.20 percent for NMITIDS. The confusion matrix using SDDA on NMITIDS is shown in Fig. 12. The graph depicts the distribution of reconstruction error for sparse deep denoising auto-encoder in relation to this attack shown in Figs. 13 and 14.

4.6 Conducted additional performance comparisons with several related approaches

The superiority of our model by comparing its detection accuracy with that of other classification algorithms found in related studies. In Qureshi et al. (2020), Al-Qatf et al. (2018), Narayana Rao et al. (2021), the authors reported their model, constructed using various classifiers, and evaluated on KDD-cup99, NSL-KDD, UNSW NB15, and NMITIDS datasets. They compared their results with various classification algorithms discussed in Qureshi et al. (2020), Al-Qatf et al. (2018), Narayana Rao et al. (2021), as illustrated in Table 5.

We evaluated the efficiency and performance of our proposed sparse deep denoising auto-encoder (SDDA) approach using publicly accessible intrusion detection training and testing datasets. The model SDDA-OLS-SVM learned low-dimensional features to enhance classification performance of the classifiers. SDDA-OLS-SVM can retain the information in the data and achieve optimum low-dimensional features. The proposed model is well above and above the given KDDcup-99, NSL-KDD, UNSW-NB15, and NMITIDS test datasets, which is a significant indicator for efficiency, because the model has never before been seen. The proposed experiment produced optimal number of low-dimensional features 10 for KDDCup99 and NSL-KDD dataset and 11 for UNSW-NB15 and 9 for NMITIDS dataset. We built three classification models with SDDA-OLS-SVM, named sparse OLS-SVM, denoising OLS-SVM, and deep-OLS-SVM. Figures 7, 8, 9, 10, 11, 12 and 13 show the comparison results on KDDcup-99, NSL-KDD, UNSW-NB15, and NMITIDS datasets.

The model SDDA-OLS-SVM classifier obtained highest detection rate using four datasets. While we use KDDCup-99 and NSL-KDD, the model achieved significant detection rate especially in U2R and R2L attack categories. The proposed SDDA-OLS-SVM model overall detection performance for KDDCup99, NSL-KDD, UNSW-NB15, and NMITIDS datasets is illustrated in Table 5 as regards accuracy, and FPR. Table 5 shows that, in all publicly available datasets, implemented SDDA-OLS-SVM has done a good performance compared to existing methods.

5 Conclusion and future scope

The most current research works which belongs to dimensionality reduction, classification of intrusion detection. Our goal in this study is to identify intrusions with a high degree of accuracy and a low percentage of false positives. The datasets KDD-cup 99, NSL-KDD, UNSW-15nb, and NMITIDS were utilized in the analysis. These databases are highly regarded by research groups all throughout the world. The extracted features are lowered the dimensionality of the feature due to elimination of unusual features. Due to low dimensionality of the dataset train and test time of the classification reduced and improved the attack classification accuracy. Result of this research showed that sparse deep denoising auto-encoder-OLS-SVM can not only detect known and unknown attacks but can also produce good detection rate on lowered number records, such as R2L and U2R in KDDCup99 and NSL-KDD dataset. Besides that, the model is outperformed by the comparative research results on the UNSW-NB15 and NMITIDS dataset to detect complex network attacks. As compared to other existing feature learning methods, the proposed model outperformed with overall accuracy and detection rate. It is obvious from the examination of the results that all of the algorithms identify intrusions at a rate of greater than 96%. In the future, we will use more deep learning approaches to filter data and increase the accuracy of intrusion detection.

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal.

Data availability The sample data set information is included in the article that support the findings of this research.

Declarations

Conflict of interest The authors do not have any conflicts of interest with this article.

Ethical approval The submitted work is an original research work, and it is neither published nor submitted to any journal. The Research is not involving experiments on any human participants and/or animals.

References

A. C. for Cyber Security (ACCS). https://www.unsw.adfa.edu.au/unsw-canberracyber/cybersecurity/adfa-nb15-datasets/

Abualigah L (2019) Feature selection and enhanced krill herd algorithm for text document clustering. 816 (**ISBN: 978-3-030-10673-7**)

Abualigah L, Yousri D, Elaziz MA, Ewees AA, Al-qaness MAA, Gandomi AH (2021a) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng. https://doi.org/10.1016/j.cie.2021.107250

Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021b) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609 (**ISSN 0045-7825**)

Al-Qatf M, Lasheng Y, Alhabib M, Al-Sabahi K (2018) Deep learning approach combining sparse auto-encoder with SVM for network intrusion detection. 2169–3536

Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabah K (2021) Sparse auto encoder driven support vector regression based deep learning model for predicting network intrusions. Peer-to-Peer Netw Appl 14(1):2419–2429

Anbar M, Abdullah R, Hasbullah IH, Chong Y-W, Elejla OE (2017) Comparative performance analysis of classification algorithms for intrusion detection system. In: 2016 14th annual conference on privacy, security and trust (PST), 2017

Anbar M, Abdullah R, Al-Tamimi BN, Hussain A (2018) A machine learning approach to detect router advertisement flooding attacks in next-generation IPv6 networks. Cogn Comput 10(2):201–214. https://doi.org/10.1007/s12559-017-9519-8

Hoque N, Bhattacharyya DK, Kalita JK (2016) A fuzzy mutual information-based feature selection method for classification. Fuzzy Inf Eng 8:355–384

Kddcup99 public dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Kumar S, Viinikainen A, Hamalainen T (2018) A network-based framework for mobile threat detection. In: 2018 1st international conference on data intelligence and security (ICDIS), 2018

Lu H, Chen J, Yan K (2017) A hybrid feature selection algorithm for gene expression data classification. Neurocomputing 256(7):56–62

Manjunatha BA, Gogoi P (2020) An improved stacked sparse auto-encoder method for network intrusion detection. In: 7th international conference on emerging research in computing, information, communication and applications (ERCICA), vol 1. Springer. (Scopus) (**ISBN 978-981-16-1337-1**)

Narayana Rao K, Venkata Rao K, Prasad Reddy PVGD (2021) A hybrid intrusion detection system based on sparse auto-encoder and deep neural network. Comput Commun 180:77–88. https://doi.org/10.1016/j.comcom.2021.08.026

Nkiama H, Said SZM, Saidu's M (2016) A subset feature elimination mechanism for intrusion detection system. Int J Adv Comput Sci Appl (IJACSA) 7(4)

Pervez MS, Farid DM (2014) Feature selection and intrusion classification in nslkdd cup 99 dataset employing svms. In: The 8th international conference on software knowledge information management and applications (SKIMA 2014), pp 1–6, 2014

Qureshi AS, Khan A, Shamim N, Durad MH (2020) Intrusion detection using deep sparse auto-encoder and self-taught learning. Neural Comput Appl 32:3135–3147. https://doi.org/10.1007/s00521-019-04152-6

Shah RA, Qian Y, Kumar D, Ali M, Alvi MB (2017) Network intrusion detection through discriminative feature selection by using sparse logistic regression. Future Internet 9(4):81. https://doi.org/10.3390/fi9040081

Shakya V, Makwana RRS (2021) Intrusion detection system based on k-means and RBF kernel function. 7(11)

Stefanova Z, Ramachandran K (2017) Network attribute selection, classification and accuracy (NASCA) procedure for intrusion detection systems. In: 2017 IEEE international symposium on technologies for homeland security (HST), 2017

Tayel MB, Rizk MRM (2021) A new automated CNN deep learning approach for identification of ECG congestive heart failure and arrhythmia using constant-Q non-stationary Gabor transform. Biomed Signal Process Control 65:102326

Zhang X, Tian J, Zhu P, Zhang J (2017) An effective semi-supervised model for intrusion detection using feature selection based LapSVM. In: Conference: 2017 international conference on computer, information and telecommunication systems (CITS), 2017. https://doi.org/10.1109/CITS.2017.8035323

Zheng Q, Zhao P, Li Y, Waang H, Yang Y (2020a) Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification. Neural Comput Appl 33:7723–7745

Zheng Q, Tian X, Yang M, Wu Y, Su H (2020b) PAC-Bayesian framework based drop-path method for 2D discriminative convolutional network pruning. Multidimens Syst Signal Process 31(3):793–827

Zheng Q, Zhao P, Zhang D, Waang H (2021) MR-DCAE: manifold regularization-based deep convolutional autoencoder for unauthorized broadcasting identification. Int J Intell Syst 36(12):7204–7238

Zheng Q, Zhao P, Wang H, Elhanashi A, Saponara S (2022) Fine-grained modulation classification using multi-scale radio transformer with dual-channel representation. IEEE Commun Lett 26(6):1298–1302. https://doi.org/10.1109/LCOMM.2022.3145647

Zheng Q, Tian X, Yu Z, Wang H, Elhanashi A, Saponara S (2023) DL-PR: generalized automatic modulation classification method based on deep learning with priori regularization. Eng Appl Artif Intell 122:106082

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.