Penerbit
Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Regularized Stacked Autoencoder with Dropout-Layer to Overcome Overfitting in Numerical High-Dimensional Sparse Data

Abdussamad[1,*], Hanita Daud[1], Rajalingam Sokkalingam[1], Iliyas Karim Khan[1], Abdus Samad Azad[1], Muhammad Zubair[2], Farrukh Hassan[3]

[1] Fundamental and Applied Science Department Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia
[2] Department of Computer Sciences, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia
[3] Department of Computing and Information System, School of Engineering and Technology, Sunway University, 47500 Petaling Jaya, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 2 August 2024
Received in revised form 14 August 2024
Accepted 4 April 2025
Available online 30 April 2025

Keywords:
Autoencoder; deep learning; high-dimensional data; stacked autoencoder; sparse data

ABSTRACT

High-dimensional sparse numerical data are normally encountered in machine learning, recommender systems, finance and medical imaging. The problem with this type of data is that it has high dimensions (many features) and highly sparse (most values are zero), which is prone to overfitting. The data visualization can be achieved through a neural network architecture called stacked autoencoders. These multilayer autoencoders are designed to reconstruct input data, but overfitting is a major problem. To overcome this problem novel L1 Regularization-dropout technique is introduced to reduce overfitting and boost stacked autoencoder performance. L1 regularization penalizes large weights, simplifying data representations whereas the dropout technique randomly turns off neurons during training and makes the model dependent only on the selected turn-on neurons. The model employs batch normalization to improve the performance of the autoencoder. The approach was implemented on a high-dimensional sparse numerical dataset in the field of cybersecurity to minimize the loss function, measured by Mean Square Error (MSE) and Mean Absolute Error (MAE). The findings were compared to the conventional stacked autoencoder. The study revealed that the suggested method effectively mitigated the issue of overfitting. Stacked autoencoders, when combined with L1 regularisation and the dropout approach, are very successful in handling high-dimensional sparse numerical data in a diverse range of applications.

1. Introduction

Recently, the analysis of high-dimensional data has become essential due to its application across several domains, including bioinformatics, image processing, natural language processing, and cybersecurity. A dataset is classified as high-dimensional when the number of features (p) exceeds the number of observations (N). For instance, a dataset with six features (p = 6) and only three observations (N = 3) are considered high-dimensional due to the greater number of features relative

[*] *Corresponding author*
E-mail address: abdussamad_22009779@utp.edu.my

https://doi.org/10.37934/ard.129.1.6074

to data points [1]. High-dimensional data can be categorized into two types: sparse data and dense data. A dataset is dense if most of its values are non-zero; otherwise, it is classified as sparse [2], [3]. This distinction is crucial because it fundamentally impacts how the data is processed and analyzed. Sparse data is prevalent in many large-scale internet applications, such as search engines, recommendation systems, and online advertising. However, most deep learning frameworks are designed for dense data and tend to perform poorly on sparse datasets [4]. The exponential growth in the quantity, variety, complexity, and dimensions of digital data presents unique challenges, particularly in managing high-dimensional sparse data [5]. Various fields, including biology, computer vision, and text processing, frequently utilize sparse, high-dimensional vectors for data representation [6].

Several methodologies have been proposed to address these challenges. For instance, Kuan et al. [7] introduced a new approach for learning similarity measures in high-dimensional sparse data, aiming to overcome the limitations of traditional methods. While innovative, these approaches often suffer from computational inefficiency when handling large, complex datasets and generally offer limited theoretical guarantees. Another significant development is the industrial deep learning framework (XDL), a distributed, scalable, and high-performance system designed specifically for high-dimensional data. However, the lack of open-source availability of XDL restricts its accessibility for academic research and broader contributions from the scientific community [3]. Similarly, the Fast Autoencoder (FAE) model investigates high-dimensional structural (HiDS) data and reduces computational costs. Nevertheless, empirical evidence supporting its effectiveness across diverse datasets remains limited, raising questions about its generalizability and robustness [8]. The SL-LF model, which employs a smooth L1-norm approach, is designed for predicting missing data in high-dimensional sparse matrices. Despite its strengths, it struggles with automatic hyperparameter tuning and maintaining non-negative constraints, affecting its optimal performance [9]. Meanwhile, the Multi-Metric Latent Factor (MMLF) technique enhances performance by uncovering latent structures in complex data, but introduces additional computational complexity due to its intricate design [10]. Deep learning (DL) has gained significant popularity for high-dimensional data analysis because of its ability to uncover low-dimensional subspaces. Deep feedforward networks and convolutional neural networks have achieved remarkable results in image processing, natural language interpretation, and robotic control [11], [12]. In these models, a multivariate function is modelled through a hierarchical structure of features, each representing nonlinear transformations that manage high-dimensional challenges effectively. However, training deep networks typically demands large-scale datasets, which may be prohibitively expensive for practical engineering applications [13][14]. Compared to traditional machine learning (ML) approaches, deep learning represents a distinct research paradigm that has demonstrated outstanding success in multiple fields. Feature engineering, a critical bottleneck in standard ML pipelines, often limits scalability due to the heavy reliance on human expertise [15]. In contrast, DL algorithms naturally extract hierarchical representations from raw data through multiple nonlinear transformations, minimizing the need for manual feature selection [16]. Advances in GPU technology and computational infrastructure have further facilitated the training of deep learning models. Methods such as the Stacked Autoencoder (SAE) have proven highly effective in learning critical data representations, making them valuable for classification and other applications. However, SAEs are vulnerable to overfitting, particularly when trained on limited datasets, due to their complex architectures and large numbers of trainable parameters [4]. Table 1 provides a comparative summary of representative state-of-the-art approaches proposed for high-dimensional sparse data analysis.

In this study, we propose a Regularized Stacked Autoencoder (RSAE) model specifically designed to address overfitting issues associated with high-dimensional sparse data. The proposed method integrates L1 regularization and dropout layers to promote sparsity and reduce model complexity, thereby enhancing generalization performance. The RSAE model demonstrates outstanding performance on a cybersecurity dataset, highlighting its potential applicability across other domains involving high-dimensional sparse data, such as image processing, natural language processing, and bioinformatics.

The contributions of this paper are summarized as follows:

i. The RSAE model, combining L1 regularization with dropout layers, significantly improves upon the traditional Stacked Autoencoder (SAE) for handling high-dimensional sparse data.
ii. The RSAE model outperforms conventional techniques, including the classic SAE, by effectively mitigating overfitting and achieving superior error metrics.

The rest of this paper is organized as follows: Section 2 details the workings of the Stacked Autoencoder (SAE) enhanced with L1 regularization.

Table 1
Comparative summary for state-of-the-art approaches

Literature	Method	Limitation	Conclusion
Kuan *et al.,* [7]	Frank-Wolfe	Scalability and generalization are limited because of high computing costs, reliance on labelled data and probable overfitting.	This approach increases similarity learning in sparse data, resulting in better performance but requiring additional scalability enhancements.
Jiang *et al.,* [3]	XDL Framework	Large-scale dataset optimization is complex and requires a lot of processing power.	Demonstrates good handling of high-dimensional sparse data, with potential for industrial-scale use.
Jiang *et al.,* [8]	Fast Deep AutoEncoder	Computationally intensive, reconstruction accuracy and efficiency must be carefully tuned.	Effectively handles high-dimensional sparse matrices in recommender systems, improving speed and scalability.
Wu *et al.,* [9]	Robust Latent Factor Analysis	Hyperparameter selection can be critical, and optimal performance may need significant adjustment.	Accurately and robustly represents high-dimensional sparse data, boosting data analysis and modelling precision.
Wu *et al.,* [10]	Multi-Metric Latent Factor Model	The integration of many measurements is complex, and parameter adjustment may be tough.	Improves analysis of high-dimensional sparse data by using numerous metrics to increase accuracy and understanding.
Zhang *et al.,* [17]	Stacked Sparse Autoencoder (SSAE) and Improved Gaussian Mixture Model (GMM)	The model is computationally demanding and necessitates significant parameter adjustment, which might affect scalability and performance in big or noisy datasets.	The model successfully enhances intrusion detection accuracy in high-dimensional data by utilizing the Stacked Sparse Autoencoder and Improved Gaussian Mixture Model, however, it may be restricted by computational complexity and tuning issues.

2. Methodology

2.1 Data Preprocessing

Raw datasets often present several challenges, including the presence of outliers, missing values, varying feature dimensions, and lack of comparability [18]. Data must undergo thorough cleaning and preprocessing before it can be effectively utilized as input for model training [19]. Furthermore, because the input to the Stacked Sparse Autoencoder (SSAE) network must be in the form of a numerical matrix, symbolic attributes must be converted into corresponding numerical features. To ensure feature values are comparable and standardized, a min-max normalization technique is applied to rescale the original feature values into a common range, facilitating consistent interpretation across different features [20].

2.2 Dataset

The UNSW-NB15 dataset was selected for the evaluation of the RSAE model. It was generated in 2015 by the Australian Centre for Cyber Security (ACCS) laboratory using the IXIA Perfect Storm too [21]. Table 2 provides a detailed breakdown of the dataset's 49 features. In total, the dataset comprises 2,540,044 traffic samples distributed across four CSV files. For experimental purposes, a training set and a testing set were created from the original samples. The dataset was uploaded to Google Drive to ensure efficient data management and ease of access. Experimental procedures were conducted using Google Colab, leveraging the free GPU environment provided by Google Cloud. This setup enhanced computational efficiency by enabling seamless dataset access and significantly accelerating processing tasks.

Table 2
The UNSW-NB15 dataset features

Feature category	Feature name
Low features	scrip,sport,dstip,dsport,proto
Base features	state,dur,sbytes,dbytes,sttl,dttl,sloss,dloss,service,sload,dload,spkts,dpkts
Content features	swin, dwin,stcpb,dtcpb,smeansz, dmeansz, trans_depth,res_bdy_len
Time features	sjit,djit,stime,ltime,sintpkt,dintpkt,tcprtt,synack,ackdat
Additional generated features (general purpose features)	is_sm_ips_ports,ct_state_ttl,ct_flw_http_mthd,is_ftp_login,ct_ftp_cmd
Additional generated features (connection features)	ct_srv_src,ct_srv_dst,ct_dst_ltm,ct_src_ltm,ct_src_dport_ltm,ct_dst_sport_ltm,ct_dst_src_ltm
Labelled features	attack_cat,Labe

2.3 Numeralization

One-hot encoding is employed to convert categorical attributes into a numerical format. The symbolic features present in the high-dimensional dataset include “proto”, “service”, “state”, and “attack_cat”. As a result of this numeralization process, the dimensionality of the dataset increases, since each categorical value is transformed into a binary vector representation. This step is essential for enabling compatibility with deep learning models that require numerical input [22].

2.4 Normalization

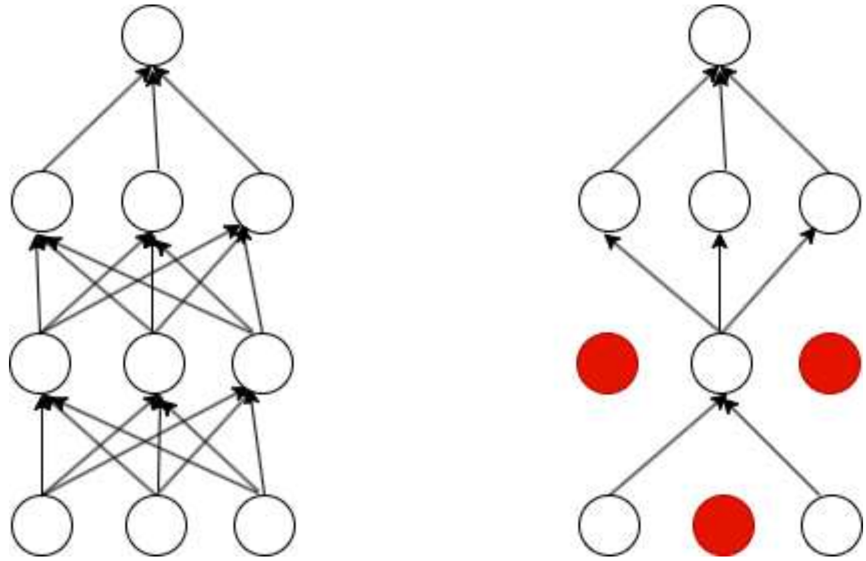
The maximal-minimum normalization approach provided in Eq. (1) is used to normalize the feature values in the dataset to make it easier to compare the findings [17]. The value of x is scaled into the numeric range [0,1] using the min-max normalization method,

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (1)$$

Where X' = normalized value, $X =$ Original value, min(X) = minimum value of X, max(X) = maximum value of X

2.5 Dropout Layer

A neural network model can employ a dropout strategy to learn more robust features and reduce interdependent learning among neurons. In this context, dropout units refer to nodes that are temporarily removed from the network along with all their incoming and outgoing connections [23]. During training, random units are dropped from the network, helping to break up complex co-adaptations among neurons. In this work, dropout is incorporated during the unsupervised learning phase to mitigate overfitting and prevent redundant feature extraction. When dropout is applied, specific nodes are assigned zero values during a training iteration and are effectively removed from the network, meaning they do not contribute to the prediction or backpropagation processes. Consequently, each training run results in a slightly altered network architecture, encouraging the model to develop redundant-free and generalized feature representations. When configuring the dropout layer, a drop probability must be defined, specifying the proportion of nodes to be set to zero in each layer. It is important to note that dropout is only active during the training phase and is disabled during testing to ensure full network capacity is utilized for inference [24]. Figure 1 illustrates the structural difference between a standard neural network and one modified with dropout.



(a) Standard neural network (b) After applying dropout

Fig. 1. Dropout applied to a standard neural network

2.6 Autoencoder Model

The structure of an unsupervised three-layer network, known as an Autoencoder, is illustrated in Figure 2 and Figure 3, representing the input layer, hidden layer, and output (reconstruction layer) [25]. An autoencoder accomplishes a nonlinear transformation from a high-dimensional space to a low-dimensional one by sequentially mapping synthetic feature vectors to abstract feature representations [26]. The autoencoder architecture can be conceptually divided into two main stages: encoding and decoding, which are formally defined as follows:

Penerbit Akademia Baru

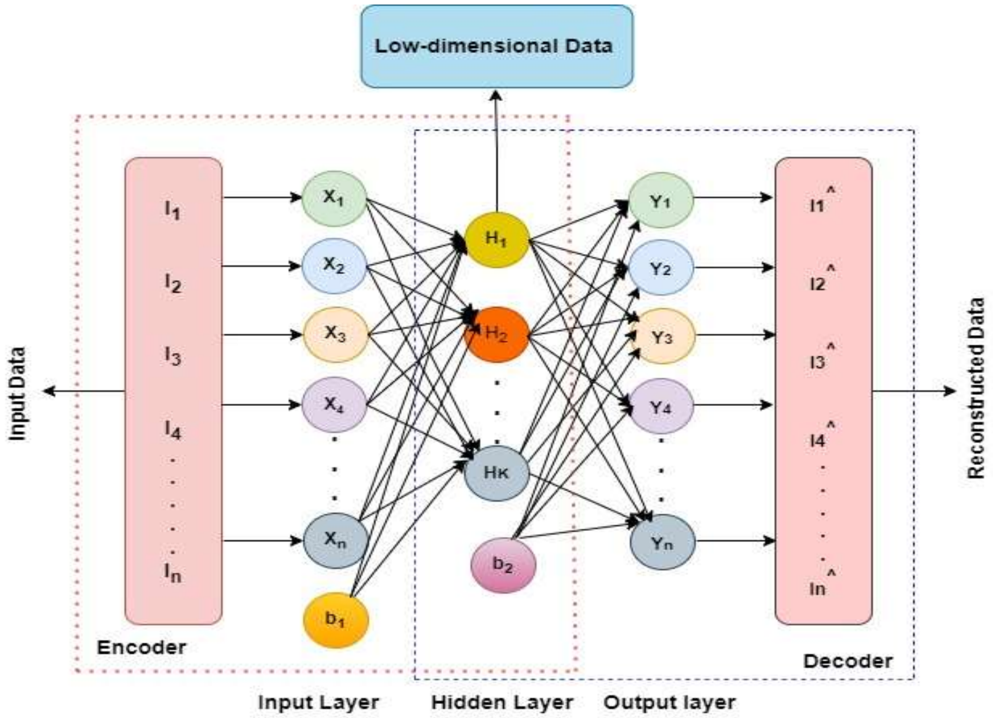


Fig. 2. Basic autoencoder model

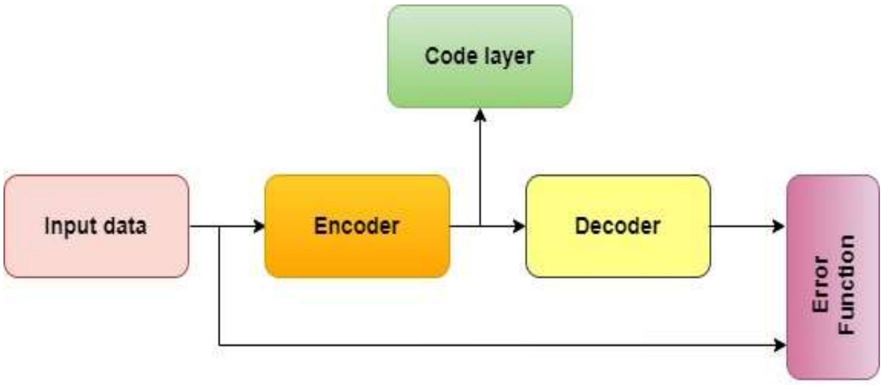


Fig. 3. Autoencoder model representation

The encoding process from the input layer to the hidden layer is as in Eq. (2),

$$H = f\theta_1(X) = \sigma(W_{ij}X + b_1) \tag{2}$$

The procedure for the decoding from the reconstruction layer to concealed layer is as in Eq. (3),

$$Y = f\theta_2(H) = \sigma(W_{jk}X + b_2) \tag{3}$$

The input data vector in this formula denoted by $X = (x_1, x_2, x_3, \dots \dots x_n)$, the reconstruction vector of the input data is represented by $Y = (y_1, y_2, y_3, \dots \dots y_n)$ and the low dimensional output from the hidden layer is denote by $H = (h_1, h_2, h_3, \dots \dots h_m)$. Thus, $X \in R^n, Y \in R^n$, $H \in R^m$ (where n is the input vector's dimension and m are the number of hidden units). The weight connection matrix between the input layer and hidden layer is denoted by $W_{ij} \in R^{m\times n}$. The weight connection matrix between the output layer and hidden layer is denoted by $W_{jk} \in R^{n\times m}$. $W_{ij} = {W_{jk}}^T$ often occurs in the experiment to reconstruct the input data as precisely as feasible while minimizing the resource consumption during model training. $b_1 \in R^{n\times 1}$ and $b_1 \in R^{m\times 1}$ are the bias vectors of input layer and hidden layer respectively. $f\theta_1(\cdot)$ and $f\theta_2(\cdot)$ are the activation functions of hidden layer neuron and output layer neurons respectively. We use Relu activation function and sigmoid activation function in this paper as in Eq. (4) and (5) respectively,

$$f\theta_1(\cdot) = max(0, x) \tag{4}$$

$$f\theta_2(\cdot) = \frac{1}{1+e^{-x}} \tag{5}$$

The Autoencoder makes the reconstruction of original data through training by minimizing the resulting error between reconstructed output and actual values. At this stage we assume that the data provided by hidden layer units aggregates all information which was present in initial dataset and is optimal low-dimensional representation of it. Eq. (6) illustrates the application of the mean squared-error function in the reconstruction error function $J_E(W, b)$ between H and Y, where N is the number of input samples.

$$J_E(W, b) = \frac{1}{2N}\sum_{r=1}^{N}\|Y^r - X^r\|^2 \tag{6}$$

2.7 Stacked Autoencoder (SAE)

The concept of sparse coding to model the computational learning of basic cell receptive fields in the primary visual cortex of mammals was first introduced by Olshausen *et al.,*[27]. For instance, the input data is transferred to the output layer by straightforward copying because of the autoencoder's inevitable issue. In this instance, the autoencoder does not extract any useful features, even though the original input data can be reconstructed properly. To make the autoencoder generate more concise and efficient low-dimensional data features under sparse constraints to better depict the input data, the author used a method of adding L1 penalty terms on hidden layers in an effort. The term "L1-norm," also known as "Lasso regression," refers to the weight vector W′s sum of the absolute values of each of its elements. It is defined as follows: $\mathrm{L1(W)} = \|\mathrm{W}\| = \sum_i \| \mathrm{W} \|_i$,. It can therefore be applied to select more significant representations. Choosing features that provide greater value to the model during training is hampered by an abundance of characteristics in the sample. As a result, we eliminate the connections that add very little to the model and do not affect the classification performance at all. With high dimensional data, it can extract more valuable features in less time.

The mean square error term and the regularization term make up the first and second terms of the error function at this point. As may be seen in Eq. (7):

$$J_E(W, b) = \frac{1}{2N}\sum_{r=1}^{N}\|Y^r - X^r\|^2 + \alpha \sum \| \mathrm{W}^r{}_{ij} \| \tag{7}$$

Here, α represents a user-adjustable hyperparameter that controls the strength of L1 regularization, allowing precise regulation of sparsity within the model. This regularization mechanism is integrated into the autoencoder architecture to enhance feature learning and reduce overfitting. The encoding and decoding layers of the architecture work together to build hierarchical feature representations from the input data. Dropout layers are incorporated after each encoding layer, where neurons are randomly deactivated during training to further prevent overfitting and encourage robustness. The structure of the Regularized Stacked Autoencoder (RSAE) network comprising multiple regularized autoencoders connected sequentially is illustrated in Figure 4.

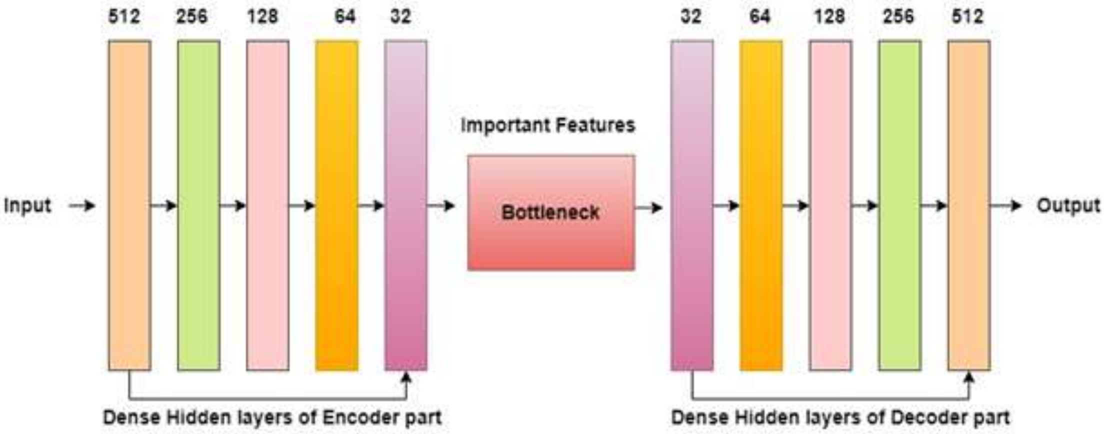

512
256
128
64
32
32
64
128
256
512
Input
Important Features
Bottleneck
Output
Dense Hidden layers of Encoder part
Dense Hidden layers of Decoder part

Fig. 4. Regularized Stacked Autoencoder model

Higher-level feature representations of the input data are generated by each successive layer of the sparse autoencoder, utilizing the output from the previous layer as input. The optimal connection weights and bias values of the stacked sparse autoencoder network are obtained through the sequential training of each layer, employing a greedy layer-wise pretraining strategy. Subsequently, the RSAE model undergoes fine-tuning using error backpropagation, optimizing the parameters until the reconstructed output closely approximates the original input data. The error function used for this fine-tuning process is defined in Eq. (6) becomes:

$$\frac{\partial}{\partial \mathrm{W}^{r}{}_{ij}} J_E(W,b) = \frac{1}{2N}\frac{\partial}{\partial \mathrm{W}^{r}{}_{ij}} \sum_{r=1}^{N} \|Y^r - X^r\|^2 + \alpha \cdot \mathrm{sign}(\mathrm{W}^{r}{}_{ij}) \tag{8}$$

$$\frac{\partial}{\partial \mathrm{b}^{r}} J_E(W,b) = \frac{1}{2N}\frac{\partial}{\partial \mathrm{b}^{r}} \sum_{r=1}^{N} \|Y^r - X^r\|^2 \tag{9}$$

Consequently, the following Eq. (10) and (11) is the weight and bias update processes,

$$\mathrm{W}^{k}{}_{ij} = \mathrm{W}^{k}{}_{ij} - \mu \frac{\partial}{\partial \mathrm{W}^{k}{}_{ij}} J_E(W,b) \tag{10}$$

$$\mathrm{b}^{r} = \mathrm{b}^{r} - \mu \frac{\partial}{\partial \mathrm{b}^{r}} J_E(W,b) \tag{11}$$

Where, Y^r and X^r are respectively the original vector and its corresponding reconstruction vectors. μ represents the learning rate.

Due to the sparse structure of the RSAE network, it is beneficial to assign distinct learning rates to individual parameters. Features that are infrequently activated require fewer updates, aligning to minimize unnecessary parameter adjustments. However, most conventional gradient descent algorithms, including mini-batch and stochastic gradient descent, apply a uniform learning rate across all parameters, which complicates the process of selecting an appropriate rate and efficiently reaching a local minimum [28]. To address this challenge, the adaptive moment estimation (Adam) optimization algorithm, as proposed by Zhang [29], is employed in this work. Adam dynamically adjusts learning rates for each parameter based on the first and second moments of the gradients, thereby facilitating faster convergence and improving the training efficiency of the RSAE network model. By calculating the gradient first-order moment estimate m_t and second-order moment estimate v_t as shown in Eq. (12) to (14), the Adam algorithm allows for the dynamic adjustment of various parameters. $\beta_1$and β_2 stand for the first order and second-order exponential damping decrements, respectively. The gradient of the parameters at the time step t in the loss function $J_E(W, b)$ is denoted by g_t.

$$m_t = \beta_1 \mathrm{m}_{t-1} + (1 - \beta_1) \,.\, g_t \tag{12}$$

$$v_t = \beta_2 \mathrm{v}_{t-1} + (1 - \beta_2) \,.\, {g_t}^2 \tag{13}$$

$$g_t \leftarrow \nabla_\vartheta J_t(\vartheta_j - 1) \tag{14}$$

Computer bias-corrected for m_t andv_t as in Eqs. (15) and (16) respectively,

$$m_t{}' = \frac{m_t}{1 - {\beta_1}^t} \tag{15}$$

$$v_t{}' = \frac{v_t}{1 - {\beta_2}^t} \tag{16}$$

The update step size is denoted by τ and ϵ is constant to prevent the denominator from zero as in Eq. (17),

$$\vartheta_{t-1} = \vartheta_t - \frac{\tau}{\sqrt{v_t{}' + \epsilon}} \,.\, m_t{}' \tag{17}$$

3. Results

3.1 Model Parameters and Sensitivity Analysis

In this work, a Regularized Stacked Autoencoder (RSAE) architecture is employed to extract significant features and reconstruct input data. The RSAE model consists of an encoder and a decoder, each comprising five interconnected layers. The encoding layers progressively reduce the dimensionality of the input data, with dense units utilizing rectified linear unit (ReLU) activation functions, batch normalization, and dropout mechanisms to mitigate overfitting. As described in Section 2.1, after preprocessing, the features in the UNSW-NB15 dataset expand from 49 to 202 dimensions. Consequently, the input layer of the RSAE model is configured with 202 neurons. Extensive experimentation and a comprehensive literature review guided the selection of critical hyperparameters, including the learning rate, number of neurons in hidden layers, batch size, and L1 regularization strength (α). A grid search technique was employed to optimize these hyperparameters, ensuring a balanced trade-off between model complexity and performance.

Further investigation confirmed that a five-layer RSAE network structure yielded the best experimental results, as detailed in Table 3. Within this architecture, the dense layer with 32 units and ReLU activation plays a pivotal role in capturing the most salient features. This critical layer is additionally regularized using L1 regularization with varying α values, illustrating the effect of sparsity enforcement on feature extraction. The model is trained using Mean Squared Error (MSE) and Mean Absolute Error (MAE) as reconstruction loss functions, with optimization performed by the Adam optimizer using a learning rate of 0.0001 over 100 epochs. Sigmoid activation is applied to the final output layer, constraining reconstructed values between 0 and 1. Mini batches of 128 samples are used to improve generalization and reduce overfitting, while the regularization term further enhances model robustness. Performance evaluation is conducted by monitoring MSE and MAE across both training and validation phases. The final experimental configuration and parameters of the RSAE model are summarized in Table 3.

Table 3
Hyperparameter summary of RSAE

Algorithms	Parameter	Value
RSAE	The number of nodes in the input layer	202
	Number of neurons in the initial hidden layer	512
	Number of neurons in the second hidden layer	256
	Number of neurons in the third hidden layer	128
	Number of neurons in the fourth hidden layer	64
	Number of neurons in the fifth hidden layer	32
	Learning rate	0.0001
	Alpha (α)	0.0001,0.001,0.01,0.1,1
	Batch size	128
	Epochs	100
	Activation functions	ReLU, Sigmoid
Adam	First-order exponential damping decrement	0.9
	Second-order exponential damping decrement	0.999
	Non-zero constant	10^{-8}

An extensive sensitivity analysis was conducted to evaluate the impact of varying the L1 regularization intensity α on the RSAE model's performance. This study systematically examined the effects of different α values on the MSE and MAE for both training and validation datasets. The findings reveal that the best performance, reflected by the lowest error metrics, is achieved with α values of 0.0001 and 0.001.

Conversely, higher α values led to a significant increase in both MSE and MAE, indicating that excessive regularization degrades model performance. Over-regularization restricts the model’s flexibility excessively, resulting in reduced accuracy, diminished stability, and impaired generalization capabilities. In contrast, lower α values contribute to better error minimization and enhance the model's ability to generalize across unseen data. These results emphasize the critical importance of carefully tuning the L1 regularization parameter to achieve an optimal balance between sparsity and predictive accuracy.

3.2 Quantitative Results

The RSAE model's ability to accurately identify structural similarities is demonstrated in Figures 5 and 6, which present the training and validation loss curves. These figures illustrate the RSAE’s strong generalization capabilities and its effectiveness in mitigating overfitting.

The MSE and MAE metrics are employed to evaluate the reconstruction quality of the autoencoder under various α configurations. These metrics provide a comprehensive assessment of the model's training and validation performance as learning progresses. As the number of epochs increases, both the MSE and MAE for the training dataset steadily decrease, indicating that the model is successfully learning meaningful representations from the input data. Similarly, a consistent decline in the validation MSE and MAE suggests that the model effectively generalizes the acquired knowledge to unseen data. In general, lower values of MSE and MAE correspond to a more optimal fit and improved model performance.

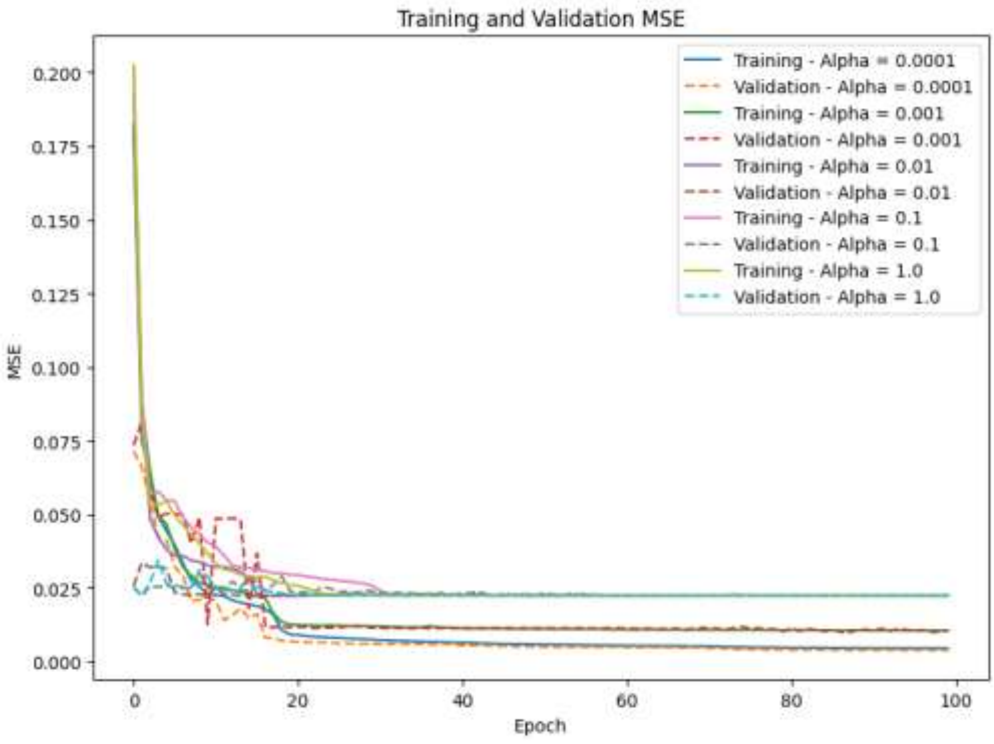


Fig. 5. Training and validation MSE loss with L1 regularization

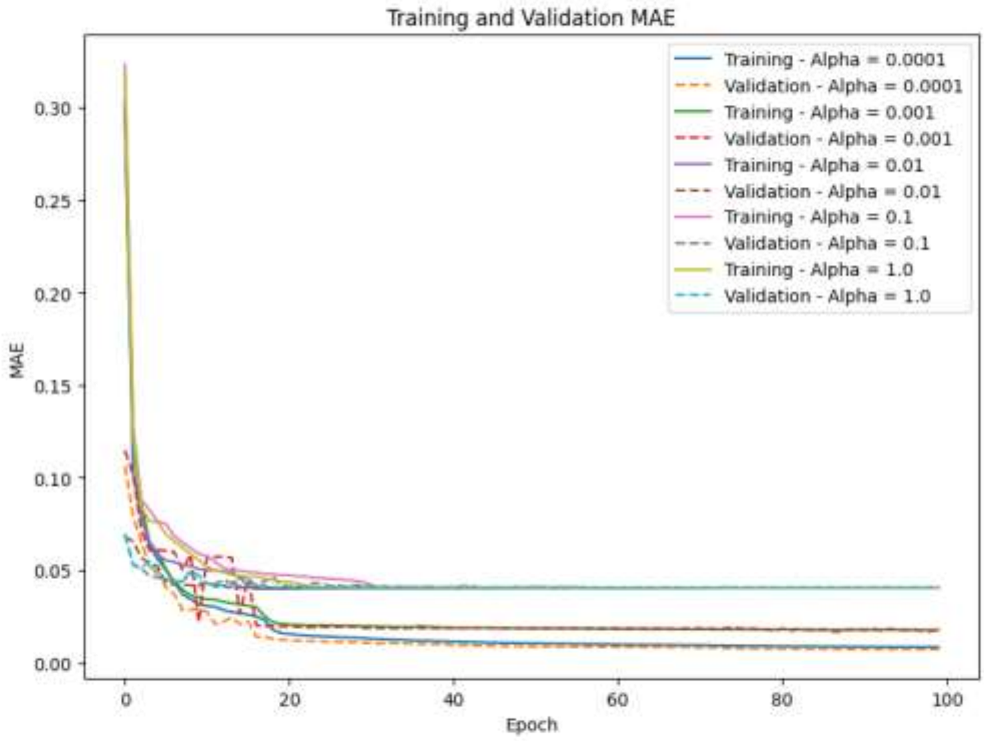


Fig. 6. Training and validation MAE loss with L1 regularization

Table 4 presents a concise summary of how varying α values impact the performance of the RSAE model. Specifically, it reports the MSE and MAE for both training and validation datasets under different levels of L1 regularization. The hyperparameter α controls the strength of L1 regularization applied to the model, where increasing α intensifies regularization, potentially mitigating overfitting but also restricting the model's learning capacity. The sensitivity analysis highlights that an α value of 0.0001 yields the best performance, achieving the lowest training and validation errors. In contrast, higher α values (e.g., 0.01, 0.1, and 1.0) result in significantly elevated MSE and MAE values, indicating that over-regularization adversely affects the model's accuracy and stability. These findings confirm that smaller α values are more effective in minimizing reconstruction errors and enhancing the model's ability to generalize to unseen data.

Table 4
Sensitivity analysis of RSAE performance metrics with Varying α

Alpha (L1 strength)	Training MSE	Validation MSE	Training MAE	Validation MAE
0.0001	0.0043	0.0038	0.0083	0.0072
0.001	0.0108	0.0099	0.0181	0.0167
0.01	0.0225	0.0225	0.0405	0.0405
0.1	0.0225	0.0225	0.0405	0.0405
1.0	0.0225	0.0225	0.0405	0.0405

To validate the effectiveness of the RSAE model against the classical Stacked Autoencoder (SAE), a comparative analysis is presented in Figure 7. The figure illustrates that the classical SAE, which lacks regularization, exhibits significant overfitting. This is evident when the training MSE and MAE are substantially lower than the corresponding validation metrics. Such a discrepancy indicates that the model has memorized specific features of the training data too closely, thereby compromising its ability to generalize to unseen data. In contrast, the RSAE model demonstrates improved generalization by mitigating overfitting through the incorporation of L1 regularization and dropout mechanisms.

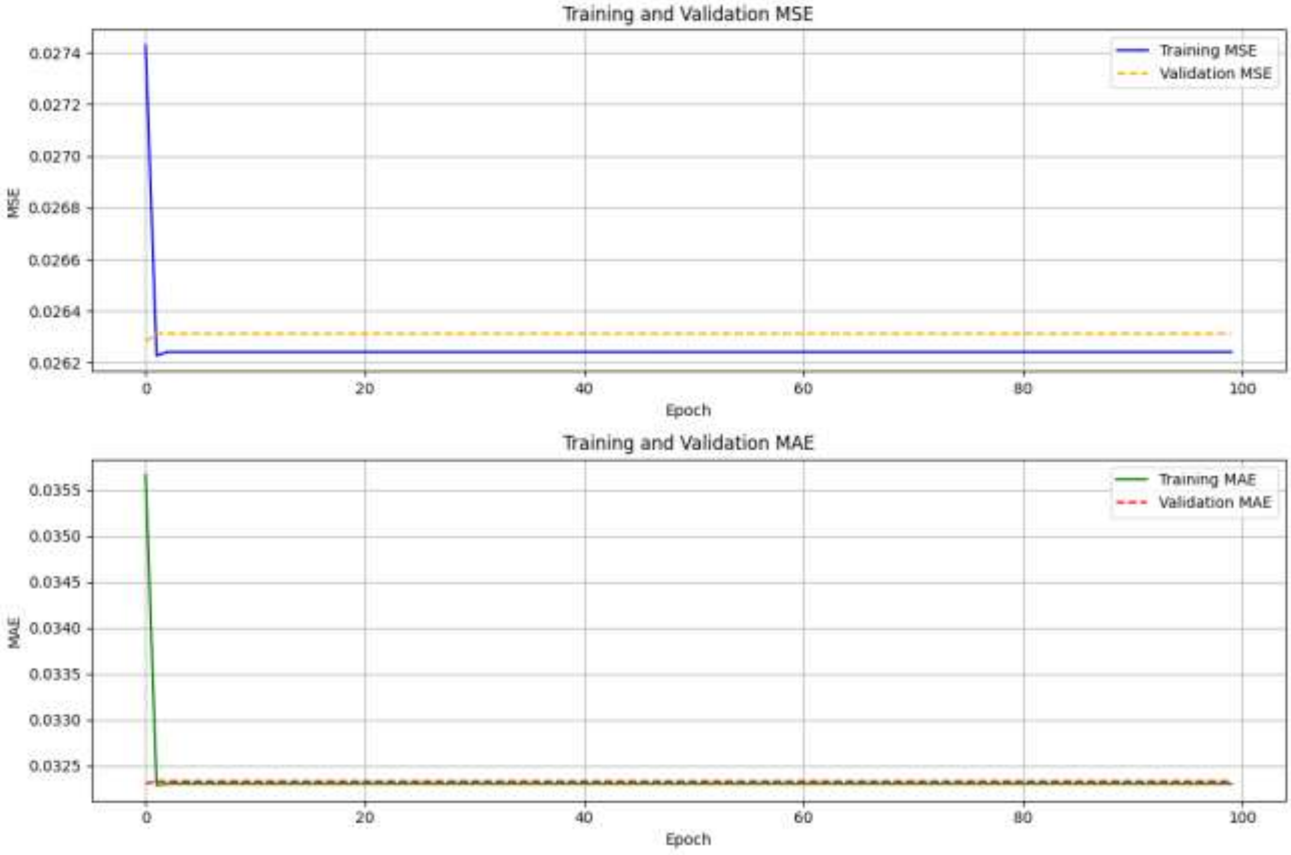


Fig. 7. Training and validation loss without L1 regularization

Table 5 summarizes the outcomes for MSE and MAE of the classic stacked autoencoder, respectively.

Table 5
SAE without regularization

S/No	MSE	MAE
1	0.0262	0.0329

A comparison of the training and validation loss curves demonstrates that the application of L1 regularization effectively prevents overfitting and enables stacked autoencoders to achieve significantly greater generalizability when applied to high-dimensional sparse data. L1 regularization thus establishes a paradigm for developing example-oriented models that not only fit the training data but also successfully capture underlying patterns, making them extendable to real-world applications. By promoting sparsity, facilitating feature selection, and enhancing generalization, L1 regularization proves to be a critical component in improving model robustness and performance.

4. Discussion

The outcomes of the proposed RSAE model demonstrate its effectiveness in mitigating overfitting and enhancing performance on the cybersecurity dataset. Beyond representing a technological advancement, these developments carry significant practical implications across multiple domains, including cybersecurity, bioinformatics, image processing, and natural language processing.

4.1 Cybersecurity Context

In the context of cybersecurity, the improved performance of the RSAE model leads to more consistent and reliable threat detection. By reducing overfitting, the model becomes better at distinguishing between legitimate activities and potential threats, thereby lowering both false positive and false negative rates. This enhanced accuracy in anomaly and threat detection is critical for enabling faster response times and preventing security breaches. Furthermore, the increased efficiency of the RSAE model allows for more effective utilization of computational resources, potentially reducing operational costs and minimizing the time required for threat detection and incident response.

4.2 Financial Sector

The RSAE model also holds significant potential for enhancing fraud detection in financial institutions, where security and accuracy are paramount. By improving the model's ability to recognize anomalous patterns within transactional data, banks and financial organizations can strengthen their defences against fraudulent activities and insider threats. This advancement contributes directly to enhancing the security of financial transactions and safeguarding sensitive customer information.

4.3 Health Sector

The advancements offered by the RSAE model also contribute to strengthening patient data privacy within the healthcare sector. Enhanced anomaly detection capabilities enable the secure storage of sensitive health information and support compliance with regulations such as the Health Insurance Portability and Accountability Act (HIPAA). These improvements not only protect patient confidentiality but also foster greater trust in digital healthcare solutions, promoting broader adoption of secure, data-driven healthcare technologies.

4.4 Manufacturing

In industries where operational technology and critical infrastructure are increasingly targeted by cyberattacks, the enhanced performance of the RSAE model can help prevent costly disruptions. By shielding industrial control systems from potential cyber threats, the model promotes operational continuity and safety while minimizing the significant financial and safety risks associated with cyberattacks.

5. Conclusions

This study demonstrates that the Regularized Stacked Autoencoder (RSAE) model effectively addresses the challenges associated with high-dimensional sparse data. By incorporating L1 regularization, controlled through the hyperparameter α, the RSAE model promotes sparsity in the learned representations, reducing the risk of overfitting and enhancing model interpretability. Careful tuning of α proved critical to optimizing performance, enabling the model to balance feature selection and generalization. Experimental results confirm that the RSAE model can efficiently learn from training data while maintaining strong generalization capabilities on unseen samples. The model achieved notable improvements in reconstruction accuracy, as evidenced by reductions in Mean Squared Error (MSE) and Mean Absolute Error (MAE), across various α configurations. Furthermore, the RSAE architecture demonstrated robustness in cybersecurity applications, with potential applicability across other domains involving high-dimensional sparse datasets, such as finance, healthcare, and industrial control systems.

Future research should focus on enhancing the RSAE model's adaptability to evolving cyber threats, extending its application to prediction tasks such as binary classification, and exploring its integration into real-time anomaly detection systems. Investigating hybrid regularization strategies and optimizing the network structure further could also contribute to performance gains in broader operational environments. Ultimately, the RSAE model, guided by the strategic tuning of L1 regularization α, emerges as a powerful and scalable framework for extracting features from complex, high-dimensional, sparse data, enabling more intelligent and secure solutions across critical industries.

Acknowledgement

This project was funded by a YUTP-FRG Grant under the cost centre: 015LC0-442, Universiti Teknologi PETRONAS, Malaysia.

References

[1] Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." *arXiv preprint arXiv:1603.04467* (2016).

[2] Abhaya, Abhaya and Bidyut Kr Patra. "An efficient method for autoencoder based outlier detection." *Expert Systems with Applications* 213 (2023): 118904. https://doi.org/10.1016/j.eswa.2022.118904

[3] Aouedi, Ons, Kandaraj Piamrat and Dhruvjyoti Bagadthey. "A semi-supervised stacked autoencoder approach for network traffic classification." In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/ICNP49622.2020.9259390

[4] Ayesha, Shaeela, Muhammad Kashif Hanif and Ramzan Talib. "Overview and comparative study of dimensionality reduction techniques for high dimensional data." *Information Fusion* 59 (2020): 44-58. https://doi.org/10.1016/j.inffus.2020.01.005

[5] Baldi, Pierre, Peter Sadowski and Daniel Whiteson. "Searching for exotic particles in high-energy physics with deep learning." *Nature communications* 5, no. 1 (2014): 4308. https://doi.org/10.1038/ncomms5308

[6] Daneshfar, Fatemeh, Sayvan Soleymanbaigi, Ali Nafisi and Pedram Yamini. "Elastic deep autoencoder for text embedding clustering by an improved graph regularization." *Expert Systems with Applications* 238 (2024): 121780. https://doi.org/10.1016/j.eswa.2023.121780

[7] Han, Jiequn and Arnulf Jentzen. "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations." *Communications in mathematics and statistics* 5, no. 4 (2017): 349-380. https://doi.org/10.1007/s40304-017-0117-6

[8] Erfani, Sarah M., Sutharshan Rajasegarar, Shanika Karunasekera and Christopher Leckie. "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning." *Pattern Recognition* 58 (2016): 121-134. https://doi.org/10.1016/j.patcog.2016.03.028

[9] Ghaddar, Bissan and Joe Naoum-Sawaya. "High dimensional data classification and feature selection using support vector machines." *European Journal of Operational Research* 265, no. 3 (2018): 993-1004. https://doi.org/10.1016/j.ejor.2017.08.040

[10] Jiang, Biye, Chao Deng, Huimin Yi, Zelin Hu, Guorui Zhou, Yang Zheng, Sui Huang *et al.,* "Xdl: an industrial deep learning framework for high-dimensional sparse data." In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pp. 1-9. 2019. https://doi.org/10.1145/3326937.3341255

[11] Jiang, Jiajia, Weiling Li, Ani Dong, Quanhui Gou and Xin Luo. "A fast deep autoencoder for high-dimensional and sparse matrices in recommender systems." *Neurocomputing* 412 (2020): 381-391. https://doi.org/10.1016/j.neucom.2020.06.109

[12] Jin, Lina, Jiong Yu, Xiaoqian Yuan and Xusheng Du. "Fish classification using DNA barcode sequences through deep learning method." *Symmetry* 13, no. 9 (2021): 1599. https://doi.org/10.3390/sym13091599

[13] Ketkar, Nikhil. "Introduction to tensorflow." In *Deep Learning with Python: A Hands-on Introduction*, pp. 159-194. Berkeley, CA: Apress, 2017. https://doi.org/10.1007/978-1-4842-2766-4_11

[14] Liu, Kuan, Aurélien Bellet and Fei Sha. "Similarity learning for high-dimensional sparse data." In *Artificial Intelligence and Statistics*, pp. 653-662. PMLR, 2015.

[15] Lai, Xiaochen, Xia Wu, Liyong Zhang, Wei Lu and Chongquan Zhong. "Imputations of missing values using a tracking-removed autoencoder trained with incomplete data." *Neurocomputing* 366 (2019): 54-65. https://doi.org/10.1016/j.neucom.2019.07.066

[16] Kim, Jihyun and Howon Kim. "An effective intrusion detection classifier using long short-term memory with gradient descent optimization." In *2017 International Conference on Platform Technology and Service (PlatCon)*, pp. 1-6. IEEE, 2017. https://doi.org/10.1109/PlatCon.2017.7883684

[17] Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver and Daan Wierstra. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971* (2015).

[18] Pinaya, Walter Hugo Lopez, Sandra Vieira, Rafael Garcia-Dias and Andrea Mechelli. "Autoencoders." In *Machine learning*, pp. 193-208. Academic Press, 2020. https://doi.org/10.1016/B978-0-12-815739-8.00011-0

[19] Moustafa, Nour and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." In *2015 military communications and information systems conference (MilCIS)*, pp. 1-6. IEEE, 2015. https://doi.org/10.1109/MilCIS.2015.7348942

[20] Olshausen, Bruno A. and David J. Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." *Nature* 381, no. 6583 (1996): 607-609. https://doi.org/10.1038/381607a0

[21] Spoorthy, G. and S. G. Sanjeevi. "Multi-criteria–recommendations using autoencoder and deep neural networks with weight optimization using firefly algorithm." *International Journal of Engineering* 36, no. 1 (2023): 130-138. https://doi.org/10.5829/IJE.2023.36.01A.15

[22] Vaziri, Pouya, Sanyar Ahmadi, Fatemeh Daneshfar, Behnam Sedaee, Hamzeh Alimohammadi and Mohammad Reza Rasaei. "Machine learning techniques in enhanced oil recovery screening using semisupervised label propagation." *SPE Journal* 29, no. 09 (2024): 4557-4578. https://doi.org/10.2118/221475-PA

[23] Wu, Di and Xin Luo. "Robust latent factor analysis for precise representation of high-dimensional and sparse data." *IEEE/CAA Journal of Automatica Sinica* 8, no. 4 (2020): 796-805. https://doi.org/10.1109/JAS.2020.1003533

[24] Wu, Di, Peng Zhang, Yi He and Xin Luo. "A Multi-Metric Latent Factor Model for Analyzing High-Dimensional and Sparse data." *arXiv preprint arXiv:2204.07819* (2022).

[25] Yan, Binghao and Guodong Han. "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system." *IEEE Access* 6 (2018): 41238-41248. https://doi.org/10.1109/ACCESS.2018.2858277

[26] Zhang, Guoqiang Peter. "Neural networks for classification: a survey." *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 30, no. 4 (2000): 451-462. https://doi.org/10.1109/5326.897072

[27] Zhang, Tianyue, Wei Chen, Yuxiao Liu and Lifa Wu. "An intrusion detection method based on stacked sparse autoencoder and improved gaussian mixture model." *Computers & Security* 128 (2023): 103144. https://doi.org/10.1016/j.cose.2023.103144

[28] Zhang, Zijun. "Improved adam optimizer for deep neural networks." In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pp. 1-2. Ieee, 2018. https://doi.org/10.1109/IWQoS.2018.8624183