Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-03077-0

ORIGINAL RESEARCH

Network intrusion detection using sparse autoencoder with swish-PReLU activation Model

Phanindra Reddy Kannari[1] · **Noorullah C. Shariff**[2] · **Rajkumar L. Biradar**[3]

Received: 5 November 2020 / Accepted: 2 March 2021

Abstract
In computer networks, the massive amount of data increases the challenges for intrusion detection systems, because of its high dimensionality. To overcome this problem, a four-phase system is developed for intrusion detection based on encoding techniques and deep learning. In the first phase, input data are collected from NSL-KDD, Canadian Institute for Cybersecurity-Intrusion Detection System 2017 (CIC-IDS2017), and Aegean Wi-Fi Intrusion Dataset (AWID). The collected data are converted into the machine-readable form by using label and one hot encoding technique that reduces the human intervention process and increases the accuracy of data classification. Next, the top percentile and recursive features are selected utilizing second percentile methodology and recursive feature elimination. The undertaken feature selection techniques; second percentile method and recursive feature elimination selects the relevant or active features from the pre-processed data that effectively diminishes the computational time and complexity of the proposed model. In the final phase, sparse autoencoder with swish-PReLU activation model is proposed to classify the normal and traffic types in the NSL-KDD, CIC-IDS2017, and AWID datasets. In the experimental phase, the proposed sparse autoencoder with swish-PReLU activation model achieved effective performance in intrusion detection in light of false alarm rate, detection rate and classification accuracy. From the experimental result, the proposed model showed the maximum of 4.77% improvement in classification accuracy compared to the existing models; correlation-based feature selection with bat algorithm, artificial neural network, chi-square and information gain-random tree, and sequential Search-Bayesian network.

Keywords Intrusion detection · Label encoder · One hot encoder · Recursive feature elimination · Second percentile method · Sparse autoencoder

1 Introduction

The computer network has pervaded almost all aspects of human life, due to the rapid internet growth and its worldwide spread (Manimurugan et al. 2020). In recent decades, network systems are used for several online services like online social interactions, online shopping, online education, internet banking, business transactions, etc. (Liu et al. 2020). The exponential growth of network size increases the intrusions and attacks, so the detection of attacks from the networks has turned out to be an emerging task. However, a firewall is a first-line defense process that is utilized for intrusion detection, which is not robust in detecting and preventing the intrusions (Prasad et al. 2020a, b; Almasoudy et al. 2020). Intrusion Detection System (IDS) is a new security technology that protects the network systems from illegal external attacks like Distributed Denial-of-Service (DDoS) (Chen et al. 2020). By detecting malicious

✉ Phanindra Reddy Kannari
phanindrareddyk@gmail.com

Noorullah C. Shariff
cnshariff@gmail.com

Rajkumar L. Biradar
rajkumar_lb@yahoo.com

[1] Department of Electronics and Communication Engineering, Rao Bahadur Y Mahabaleswarappa Engineering College, Ballari, India

[2] Department of Electronics and Communication Engineering, SECAB Institute of Engineering and Technology, Vijayapur, India

[3] Department of Electronics and Telematics Engineering, G. Narayanamma Institute of Technology and Science, Hyderabad, India

Published online: 13 March 2021

behaviour, IDS effectively improves the security and reliability of the systems (Wu et al. 2020; Li et al. 2020). Usually, IDS is divided into two types, such as a signature based detection system (misuse detection system) and profile based detection system (anomaly detection system) (Lv et al. 2020; Gupta and Agrawal 2020). By monitoring the system activities, the anomaly detection system detects computer and network intrusions, and then classify it as anomalous or normal based on rules or heuristics. The misuse detection system is used to detect the computer attack, where abnormal system behaviour is initially defined, and then the remaining behaviors are defined as normal (Rose et al. 2020; Kumar et al. 2020). Currently, the use of machine learning and deep learning techniques in IDS is a growing research area that analyses and extracts useful information from the data (Zong et al. 2020; D'hooge et al. 2020). Some of the existing machine learning and deep learning IDS techniques are based on the genetic convolutional neural network (Nguyen and Kim 2020), deep neural networks (RM et al. 2020), ensemble of discriminant classifiers (Bhati et al. 2020), rough set theory (Prasad et al. 2020a, b), etc.

In the prior research works, non-linear characteristics and higher dimensionality of data make machine and deep learning techniques unfit to solve multiple classification tasks. Hence, feature selection is considered as an essential requirement for the learning techniques that eliminate the redundant or irrelevant features in the original data to enhance the learning procedure. In this study, an effective feature selection and classification technique are proposed to improve intrusion detection performance. At first, the original data are collected from NSL-KDD (Tavallaee et al. 2009), CIC-IDS2017 (Sharafaldin et al. 2018a, b), and AWID (Aminanto et al. 2017) databases to validate the performance of the proposed model. Next, data pre-processing is accomplished using label encoder, and the use of one hot encoder improves the quality of collected data. The label encoder converts the collected labels into numeric form (machine-readable form), and one hot encoder splits the columns based on the number of categories for better representation. The data pre-processing techniques provide details about how the data labels are operating. Then, feature selection is accomplished using the second percentile method and recursive feature elimination to select the most relevant features from the pre-processed data that significantly reduce the computational time of attack detection, and computational complexity of the proposed model. Finally, the sparse autoencoder with swish-PReLU activation model is proposed to classify the attack types in NSL-KDD, CIC-IDS2017, and AWID databases. If the data is structured or linear, sparse autoencoder with swish-PReLU activation model is an effective choice for classification. In Autoencoder, swish-PReLU activation function is used as an activation function that improves both classification performance and learning speed related to other activation functions. In the experimental phase, the proposed model performance is evaluated in light of detection rate, accuracy and false alarm rate. Compared to the existing models like correlation-based feature selection with bat algorithm, artificial neural network, chi-square and information gain-random tree, and sequential Search-Bayesian network, the proposed model showed a maximum of 4.77 % enhancement in intrusion classification.

This research paper is prepared as follows; a few recent research papers on "intrusion detection" is surveyed in Sect. 2. The proposed sparse autoencoder with swish-PReLU activation model is briefly explained in Sect. 3. Experimental investigation of the proposed sparse autoencoder with swish-PReLU activation model is denoted in Sect. 4. The conclusion about the present research study is given in Sect. 5.

2 Literature review

Zhou et al. (2020) presented a new intrusion detection system based on ensemble learning and feature selection techniques. In this study, Correlation-based Feature Selection (CFS) with Bat Algorithm (BA) was introduced to select the optimal subset of features to deal with higher dimensional and unbalanced network traffics. Next, an ensemble technique was developed based on random forest, C4.5 and forest by penalizing attributes along with an average of probability combination rule to construct the classification model. In the experimental section, the developed intrusion detection system was investigated using 10-fold cross validation on three online databases such as NSL-KDD, CIC-IDS2017, and AWID. The simulation result reveals that the developed system exhibits effective performance related to the existing systems under several performance metrics like false alarm rate, precision, f-measure, accuracy and attack detection rate. However, the developed system is ineffective in dealing with the rare attacks, which are derived from the massive network traffics. Additionally, Ahmad et al. (2018) used Extreme Learning Machine (ELM), random forest, Support Vector Machine (SVM) to improve the intrusion detection rate and to minimize the false alarm rate. Related to random forest and SVM techniques, the ELM performs well on large databases that effectively handles an enormous amount of traffic data. In this study, NSL-knowledge discovery and data mining databases were used to evaluate the intrusion detection mechanism. The Extensive experiment showed that the ELM achieved better performance in intrusion detection related to the existing techniques in terms of accuracy, recall, and precision. The experimental results showed that the ELM technique was highly suitable to analyse the vast amount of data. In this literature, different traffic types were

required to model several network attacks that were expensive and complex.

Shone et al. (2018) introduced an unsupervised feature learning technique; Non-Symmetric Deep Autoencoder (NDAE) for intrusion detection. The novel classification model was built by utilizing random forest and stacked NDAEs. The Two benchmark databases; NSL-KDD and KDD Cup 99 were used for experimental evaluation, where the developed technique achieved promising results in intrusion detection. The performance metrics like precision, accuracy, recall, false alarm and f-score were utilized to evaluate the effectiveness of NDAE technique over other deep learning techniques. However, the developed technique is ineffective in handling the zero-day attacks that degrade the performance of intrusion detection. Further, Ghasemi et al. (2020) developed a Genetic Algorithm (GA) (Maulik and Bandyopadhyay 2000) and Kernel ELM (KELM) for optimal feature selection and classification. In this literature study, NSL-KDD and KDD Cup 99 databases were used for experimental analysis. The simulation outcome showed that the developed model achieved better performance in intrusion detection, but the normal records were not detected appropriately by the GA-KELM model, which was a significant drawback in this study. Cavusoglu (2019) developed a hybrid layered intrusion detection system using different machine learning techniques. The main objective of this literature study was to establish a system that performs attack detection with lower error rates related to the existing studies. Initially, the normalization technique was used for data pre-processing, and then CfsSubsetEval (Chou et al. 2008) and WrapperSubsetEval feature selection algorithms were used for optimal feature selection. The machine learning techniques such as naive Bayes, random forest, J-48, and random tree were used for traffic classification. The developed system performance was evaluated on NSL-KDD database in light of detection rate, f-measure, and overall accuracy. Based on the attack types, run-time and processing time of the system were evaluated, where processing time was very high in user to root attack type, because of its stacking structure.

Gu et al. (2019) developed an intrusion detection system based on feature augmentation with SVM classifier. In this study, a better quality transformed training data were achieved by applying logarithm marginal density ratio transformation in the original features. Compared to the existing systems, the developed system achieved effective performance on NSL-KDD, KDD'99 and Kyoto 2006 + databases in light of detection rate, false alarm rate, accuracy, and training speed. The developed system is ineffective in solving the problems like handling of the huge volume of data, and high dimensionality features. Hajimirzaei, and Navimipour (2019) presented a new intrusion detection system based on the Artificial Bee Colony (ABC) and Multilayer Perceptron (MLP). In this literature, the MLP technique was utilized to identify normal and abnormal network traffic packets. Further, the ABC algorithm was used to optimize the value of biases and linkage weights in MLP that improves the testing and training performance. The developed system performance was evaluated on NSL-KDD database in light of root mean square error and mean absolute error, which was better related to the existing systems. However, the computational overhead of MLP is higher that was considered as a major concern in this study.

Asad et al. (2020) used feed forward backpropagation algorithm to classify the network flows as normal flows or attack. The developed deep learning algorithm protects the services from application-layer DDoS attacks by determining malicious behaviour. In this literature study, a new malicious pattern was used to detect the malicious behaviour from packets, which serve as a secondary database to train the deep learning algorithm. In the experimental phase, the developed algorithm performance was evaluated on CIC-IDS 2017 database for DDoS detection through recall, f-score, and precision. The developed deep learning algorithm is ineffective in imbalanced data problem, and also it is not feasible in the DDoS scenario. Thanthrige et al. (2016) developed a two-phase system for intrusion detection. In the first phase, chi-square and information gain were used to determine the relevant features, which has an effect on the accuracy of intrusion detection and also improves the speed of classification. In the second phase, the random tree classifier was applied to classify the finer-grained and higher level class distributions. The extensive experiment showed that the developed system achieved effective performance in intrusion detection on AWID using accuracy, detection rate and false alarm rate. During classification, the random tree is sensitive to the outliers and the parameters and also has a higher computational burden. Zhang and Wang (2013) developed an effective wrapper feature selection algorithm on the basis of the Bayesian network for intrusion detection. In the experimental section, the developed algorithm performance was tested on NSL-KDD database to validate its effectiveness. The Empirical results showed that the developed algorithm significantly increased the classification accuracy and decreased the time to detect the attacks, where network security was a widely concerned issue in this literature study. To address the issues mentioned above the sparse autoencoder with swish- PReLU activation model is proposed to improve the performance of intrusion detection.

3 Methodology

In computer networks, it is necessary to develop a robust IDS to monitor the activities of network traffics. The IDS is a security management system that collects and analyses

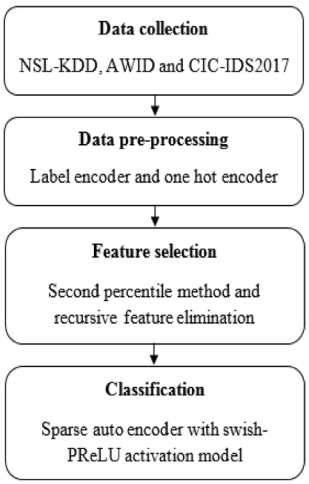

Fig. 1 The workflow of the proposed system

information from computer and networks to check the abnormal behaviors of the system (Almiani et al. 2020; Alazzam et al. 2020; Davahli et al. 2020). In this article, the sparse autoencoder with swish- PReLU activation model is proposed to improve the classification performance of intrusion detection. The proposed intrusion detection system includes four phases; data collection: NSL-KDD, AWID and CIC-IDS2017, data pre-processing: label encoder, and one hot encoder, feature selection: second percentile method, and recursive feature elimination and classification: sparse autoencoder with swish- PReLU activation model. The workflow of the proposed system is presented in Fig. 1.

3.1 Data collection and pre-processing

In this research study, the proposed sparse autoencoder with swish-PReLU activation model performance is validated on NSL-KDD, AWID, and CIC-IDS2017 databases. The NSL-KDD database is an improved version of the KDD cup 1999 database that is extensively utilized in intrusion detection experiments. NSL-KDD database reasonably alters the number of records for testing and training and also resolves the inherent redundant records problems in KDD cup 1999 database. NSL-KDD database consists of KDD-Test+, KDD-Train+, and KDD-Test-21 sets to make a better comparison with different models. As shown in Table 1, NSL-KDD database includes four types of abnormal records (DoS, U2R, R2L and probe attack) and a normal record (Gurung et al. 2019; Su et al. 2020). In this study, KDD-Test + set comprises of 22,544 instances that include 9,711 normal instances and 12,833 traffic instances. Also, KDD-Train + set contains 125,973 instances in that 67,343 instances are normal traffic, and 58,630 instances are attack traffic. As a subset of KDD-Test + set and KDD-Train + set, KDD-Test-21 set includes 11,850 instances. The list of attacks presented in NSL-KDD database is given in Table 2.

AWID database comprises of both intrusive and normal data, which are collected from a real network environment. In the AWID database, each record includes a vector of 155 attributes, and every attribute has nominal and numerical values (Lopez-Martin et al. 2020; Thang and Pashchenko 2019). AWID database is sub-divided into two sets; AWID-ATK and AWID-CLS based on the number of target classes. AWID-ATK has seventeen target classes and AWID-CLS groups the instances into four major classes like injection, flooding, normal, and impersonation. The data statistics of AWID-CLS database is denoted in Table 3.

The CIC-IDS2017 database contains up-to-date attacks and the results of network traffic analysis with labelled flows based on the source and destination internet protocols and time stamp. The CIC-IDS2017 database is the advanced intrusion detection database that covers important criteria with updated attacks like Botnet, SQL injection, port scan, DDoS, brute force, infiltration and XSS.

Table 1 Data statistics of NSL-KDD database

Sets	Total	Normal	DoS	Probe	R2L	U2L
KDD-Test+	22,544	9711	7458	2421	2754	200
KDD-Train+	125,973	67,343	45,927	11,656	995	52
KDD-Test-21	11,850	2152	4342	2402	2754	200

Table 2 List of attacks presented in NSL-KDD database

Attack category	Attack name
U2R	Httptuneel, Perl, buffer-overflow, load module, Ps, rootkit, SQL attack, and Xterm
DoS	Back, UDP-storm, Teardrop, Pod, Apache-2, Process-table, Smurf, land, Neptune, and mail-bomb
Probe	Ip-sweep, M-scan, Satan, N-map, Saint, and Port-sweep
R2L	Snmp-guess, X-lock, X-snoop, worm, Snmp-get-attack, send-mail, spy, phf, multi-hop, named, Ftp-write, Imap, warez-master, guess-password, and warez-client

Table 3 Data statistics of AWID-CLS database

Classes	Number of attributes
Normal	530,785
Injection	16,682
Flooding	8097
Impersonation	20,079
Total	575,643

Table 4 Data statistics of CIC-IDS2017

Classes	Number of attributes
Normal	439,683
DoS slowloris	5796
DoS slowhttptest	5499
DoS hulk	230,124
DoS goldeneye	10,293
Heartbleed	11
Total	691,406

The CIC-IDS2017 database consists of 2,830,743 records, where each record includes 78 features with its label. In this research, Wednesday working hour set is considered for experimental analysis to maintain a similar order of magnitude for every database (Khammassi and Krichen 2020; Sharafaldin et al. 2018a, b). The Wednesday working hours set comprises of 691,406 instances that belong to six categories, as detailed in Table 4.

After data collection, pre-processing is accomplished by utilizing label encoding and one hot encoding techniques to reduce the system complexity. Label encoding is a simple technique which converts every value in a categorical column (label) into a nominal or numerical value (machine-readable form) (Mottini and Acuna-Agost 2016). The Label encoding technique converts every category of a specific feature with a value between 0 and $n-1$, where n is denoted as the number of distinct categories of the feature (Shahriar et al. 2020). Label encoding is an important pre-processing technique in the structured database in supervised learning. Besides, one hot encoding technique is utilized to convert categorical features into numerical values (Cerda et al. 2018; Tang et al. 2020). In one hot encoding technique, a new variable is created for each level of categorical features, where each category is mapped with a binary variable containing either 0 or 1. Particularly, each categorical attribute is stated as a binary value; for instance; the protocol type feature has three attributes in NSL-KDD database like ICMP, TCP, and UDP. One hot encoding technique converts the features into binary values like [0,0,1], [1,0,0] and [0,1,0].

3.2 Feature selection

After pre-processing the data, second percentile method and recursive feature elimination technique are used to select the relevant features for network intrusion detection. By selecting the optimal features, computational complexity of the system is extremely reduced. The second percentile methodology is a modified version of the K-best feature selection technique, where top f ($f = 70$) percentiles of the best scoring features are selected from the total features F (attributes). The selected top percentiles features f are given as the input to recursive feature elimination technique to select the recursive feature vectors x by considering smaller sets of features. Initially, an estimator is trained on the selected top percentiles features f, and the importance of every feature is obtained using feature importance or coefficient attributes. Next, the least important features are eliminated from the selected top percentiles features f, where 10% of the important features are selected. This mechanism is repeated recursively on the eliminated feature sets until the desired number of features x are selected eventually. After obtaining the recursive features x, selected features are fed to sparse autoencoder with swish- PReLU activation model for intrusion attack classification, where the selected features are determined below;

Features selected for DoS ['logged_in', 'count', 'serror_rate', 'srv_serror_rate', 'same_srv_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'service_http', 'flag_S0', 'flag_SF']

Features selected for Probe ['logged_in', 'rerror_rate', 'srv_rerror_rate', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_rerror_rate', 'dst_host_srv_rerror_rate', 'Protocol_type_icmp', 'service_eco_i', 'service_private', 'flag_SF']

Features selected for R2L ['src_bytes', 'dst_bytes', 'hot', 'num_failed_logins', 'is_guest_login', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'service_ftp', 'service_ftp_data', 'service_http', 'service_imap4', 'flag_RSTO']

Features selected for U2R ['urgent', 'hot', 'root_shell', 'num_file_creations', 'num_shells', 'srv_diff_host_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'service_ftp_data', 'service_http', 'service_telnet'].

3.3 Classification

After selecting the optimal features, intrusion attack classification is carried out utilizing sparse autoencoder with swish-PReLU activation model. The Autoencoder is a type of artificial neural network that reflects its input at the output, and it primarily comprises of a decoder and an encoder. The selected features are fed into a deep network for training by using the input layer of the encoder. Autoencoder learns from the low-level representation of the selected features (input data), and then it deformed back for projecting the original data. In this scenario, encoder maps the selected features into a new data representation, and then the data representation is decoded at the output end to reconstruct the input data x' based on the Eqs. (1) and (2).

$$Z = h(Wx + b) \tag{1}$$

$$X' = g(W'z + b') \tag{2}$$

where z is denoted as new data representation, x is indicated as input data, W' and $W$are denoted as weight matrices, b' and b are represented as decoder and encoder bias vector, g is indicated as output layer neurons, and h is denoted as activation function of hidden layer neurons. In this article, the swish-PReLU activation function is used to select the optimal number of hidden neurons, because several hidden neurons result in over fitting. The hybrid swish-PReLU activation function slightly showed better performance related to ReLU, and it is graphically depicted in Fig. 2. The proposed swish-PReLU activation function includes a few advantages like easy to compute, unsaturated and does not cause a vanishing gradient problem. General equation of swish and PReLU activation function are $swish = f(x) = max(\beta \times x, x) \times x$ and $PReLU = Max(0, x) + \alpha \times Min(0, x)$.

The reconstruction error function E between the reconstructed data x' and the original input data x is calculated using the Eq. (3). In this study, sparse autoencoder is used to obtain the lower-level representation of the selected features (input data) under sparse constraint. Though, sparsity is introduced by regularizing the cost function, where $\widehat{p}_i$ is denoted as average activation of neurons in the hidden layer, and it is mathematically defined in Eq. (4).

$$E = \frac{1}{N}\sum_{i=1}^{N} x_i + x'^2_i \tag{3}$$

$$\widehat{p}_i = \frac{1}{n}\sum_{j=1}^{n} z_i(x_j) \tag{4}$$

where N is represented as the number of input samples, I is indicated as *ith* neuron, n is represented as number of training samples, j is represented as *jth* training samples and the average activation function $\widehat{p}_i$ value is constant, which is closer to zero. Here, Kullback-Leibler (KL) divergence is utilized for regularizing the cost function to achieve sparsity that is mathematically defined in Eq. (5). The Kullback-Leibler (KL) divergence architecture is graphically stated in Fig. 3 (Qi et al. 2017).

$$\Omega_{sparsity} = \sum_{i=1}^{d} p\log\left(\frac{p}{\widehat{p}_i}\right) + (1-p)\log\left(\frac{1-p}{1-\widehat{p}_i}\right) \tag{5}$$

where d is represented as the number of neurons in the layer and p is indicated as sparsity proportion. Then, L2 regularization is included in the cost function to control the weights and to prevent the overfitting problem, and it is mathematically defined in Eq. (6). L2 regularization forces the weight towards zero, but it does make the weights exactly zero. So, L2 regularization acts like a force, which eliminates small percentage of weights at each iteration. In addition, choosing an optimal value for λ is important. If λ is set to zero,

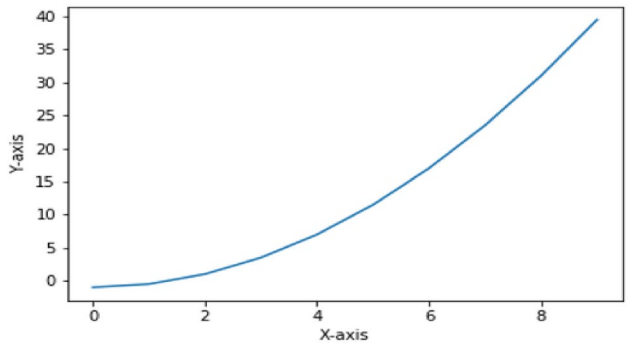

Fig. 2 Graphical depiction of the swish-PReLU activation function

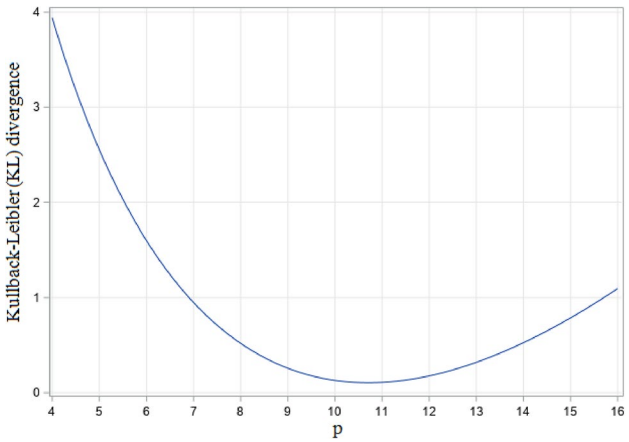

Fig. 3 Architecture of kullback-leibler (KL) divergence

then regularization is completely removed (higher risk of overfitting).

$$\Omega_{weights} = \frac{1}{2}\sum_{I}^{L}\sum_{j}^{N}\sum_{i}^{K}\left(w_{ji}^{(I)}\right) \tag{6}$$

where K is represented as the number of features in the sample and L is denoted as the number of hidden layers. Further, include the weight attenuation units in the cost function, where an Eq. (3) is updated, as shown in Eq. (7).

$$E = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(x_{kn} + \hat{x}_{kn})^2 + \lambda \times \Omega_{weights} + \beta \times \Omega_{sparsity} \tag{7}$$

Three optimization parameters are used in this study such as λ, β, and p, λ is a coefficient of L2 regularization that prevents the overfitting problem, β is a sparsity regularization parameter, and p is a sparsity proportion that controls the sparsity level. Hence, the optimization parameter values are fixed as $\lambda = 0.0001$, $\beta = 0.01$ and $p = 0.5$. Furthermore, the adam optimization algorithm is used to realize the dynamic adjustments of parameters with the help of gradient 1st and 2nd order moment estimates m_t and v_t, which is defined in the Eqs. (8–10).

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) \times g_t \tag{8}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) \times g_t^2 \tag{9}$$

$$g_t \leftarrow \nabla_\theta J_t\left(\theta_{t-1}\right) \tag{10}$$

where β_1 and β_2 are represented as 1st and 2nd order exponential damping decrement, g_t is stated as gradient parameters at time step t in the cost function E. By using the Eqs. (11), (12), and (13), calculate bias corrected for m_t and v_t.

$$m_t' = \frac{m_t}{1-\beta_1^t} \tag{11}$$

$$v_t' = \frac{v_t}{1-\beta_2^t} \tag{12}$$

Updated parameters;

$$\theta_{t+1} = \theta_t - \frac{\gamma}{\sqrt{v_t'} + \zeta} m_t' \tag{13}$$

where γ is denoted as updated step size and ζ is stated as a constant value that stops the denominator value from becoming zero. The parameter setting of a sparse autoencoder with swish- PReLU activation is given as follows; the number of hidden layers is 32, the learning rate is 0.0025, the batch size is 64, epochs is 10, $\lambda = 0.0001$, $\beta = 0.01$ and $p = 0.5$.

The Pseudocode of sparse autoencoder with swish-PReLU activation is given as follows;

- **Input**: Training matrix.
- **Output**: Autoencoder parameters.
- **Initialize the parameters**: Swish PReLU activation function for the hidden layer h, p is a sparsity parameter,W and W' are weight matrices, and b and $b'$are encoder and decoder bias vector.
- Obtain the error function using Eq. (3).
- By adding the sparse regularity parameter, the cost function is updated using Eq. (7).
- Further, Adam optimizer is applied in Autoencoder to realize the dynamic adjustments of parameters using the Eqs. (8) and (9).
- Train the sparse encoder model.
- Predict the sparse encoder model.
- **Def** swish-PReLU ($x, \beta, p,$ and α):

$$swish = f(x) = \max(\beta \times x, x) \times x$$
$$\text{and} PReLU = Max(0, x) + \alpha \times Min(0, x)$$

4 Experimental analysis

In this research, the proposed model is simulated using anaconda navigator 3.5.2.0 (64-bit), python 3.7 environment on the Windows 10 (64 bit) OS, with a RAM of 16GB, and Intel Core i7 processor as the system specifications. The proposed model performance is related with a few benchmark models like CFS-BA (Zhou et al. 2020), Artificial Neural Network (ANN) (Asad et al. 2020), chi-square and information gain-random tree (Thanthrige et al. 2016) and sequential Search-Bayesian network (Zhang and Wang 2013) to validate its efficiency. In this research work, the proposed sparse autoencoder with swish- PReLU activation model performance is investigated through detection rate, accuracy and False Alarm Rate (FAR). In detection rate, accuracy and FAR are mathematically stated in the following Eqs. (14–16),

$$Detectionrate = \frac{TP}{TP + FP} \times 100 \tag{14}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{15}$$

$$FAR = \frac{FP}{FP + TN} \times 100 \tag{16}$$

Where True Positive is indicated as TP, True Negative is represented as TN, False Positive is stated as FP, and False Negative is denoted as FN.

Table 5 Performance evaluation of sparse autoencoder with swish-PReLU activation model on NSL-KDD database

Database	Classifiers	Accuracy (%)	Detection rate (%)	FAR (%)
NSL-KDD	ANN	90.66	98.58	9.34
	Simple autoencoder	90.86	98.80	9.14
	Sparse autoencoder	98.29	98.55	1.71
	Sparse autoencoder with swish-PReLU activation	99.95	99.82	0.05

4.1 Quantitative investigation

In this section, the performance of a sparse autoencoder with swish-PReLU activation model is analysed in light of detection rate, accuracy and FAR on NSL-KDD database. Here, 22,544 data attributes are used for testing, and 125,973 data attributes are used for training. By inspecting Table 5, the proposed sparse autoencoder with swish-PReLU activation model performance is compared with a few benchmark models such as ANN, simple Autoencoder, and the sparse Autoencoder. From the experimental investigation, sparse autoencoder with swish-PReLU activation model achieved a maximum accuracy of 99.95%, the detection rate of 99.82 %, and minimum FAR value of 0.05 %. The proposed sparse autoencoder with swish-PReLU activation model showed a maximum of 9.29% and a minimum of 1.66% improvement in intrusion traffic classification. The graphical comparison of a sparse autoencoder with swish-PReLU activation model on NSL-KDD dataset in light of FAR, detection rate, and accuracy is represented in Figs. 4 and 5.

In Table 6, the performance of a sparse autoencoder with swish-PReLU activation model is analysed on the AWID database using detection rate, accuracy, and FAR. By analysing Table 6, the proposed sparse autoencoder with swish-PReLU activation model showed better intrusion traffic classification performance related to the comparative models like ANN, simple Autoencoder, and the sparse Autoencoder. In this scenario, 575,643 attributes are used for experimental investigation with 80% training and 20% testing of data. In the AWID database, the proposed sparse autoencoder with swish-PReLU activation model achieved a maximum accuracy of 99.89%, a detection rate of 99.54% and minimum FAR value of 0.11 %. The graphical comparison of a sparse

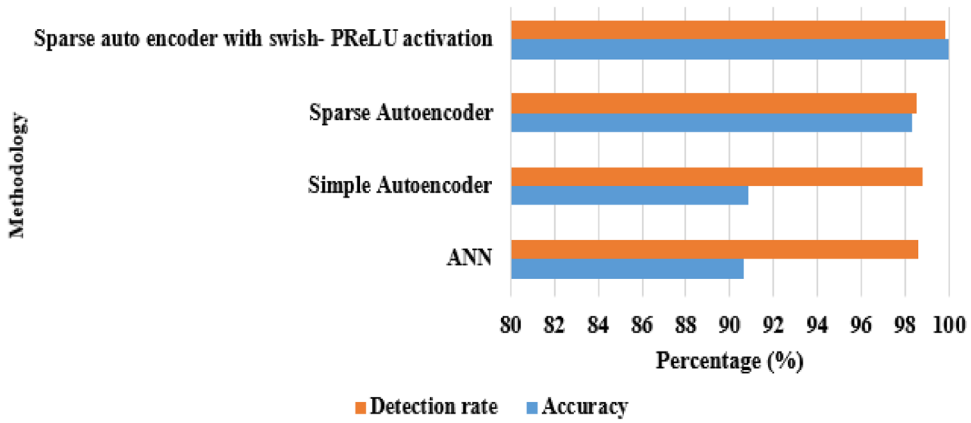

Fig. 4 Graphicalcomparison of sparse autoencoder with swish-PReLU activation model on NSL-KDD databasein light of detection rate and accuracy

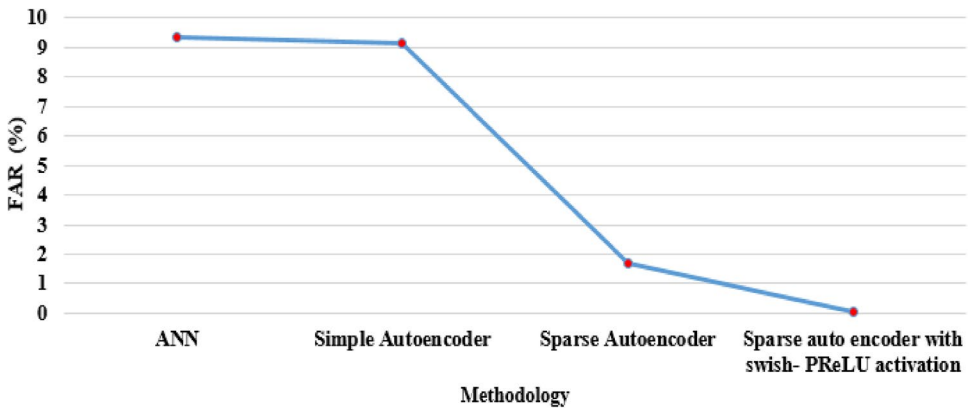

Fig. 5 Graphicalcomparison of sparse autoencoder with swish-PReLU activation model on NSL-KDD databasein terms of FAR

Table 6 Performance evaluation of sparse autoencoder with swish-PReLU activation model on AWID database

Database	Classifiers	Accuracy (%)	Detection rate (%)	FAR (%)
AWID	ANN	92.56	92.48	7.44
	Simple autoencoder	91.78	92.85	8.22
	Sparse autoencoder	99.43	98.99	0.57
	Sparse autoencoder with swish-PReLU activation	99.89	99.54	0.11

autoencoder with swish-PReLU activation model on AWID database through FAR, detection rate and accuracy are indicated in Figs. 6 and 7.

Correspondingly in Table 7, the performance of a sparse autoencoder with swish-PReLU activation model is analysed on CIC-IDS2017 database using detection rate, accuracy, and FAR value. In this scenario, 691,406 attributes are used for experimental investigation with 80% training and 20% testing of data. Similar to the other two databases, the proposed sparse autoencoder with swish-PReLU activation model achieved effective performance in intrusion detection related to the comparative models like ANN, simple Autoencoder, and sparse Autoencoder. By inspecting Table 7, the proposed sparse autoencoder with swish-PReLU activation model showed a maximum of 9.05% and a minimum of 0.47% improvement in intrusion traffic classification. The proposed sparse autoencoder with swish-PReLU activation model has decreased overfitting problem, which results in higher classification accuracy related to the comparative models. The graphical comparison of a sparse autoencoder with swish-PReLU activation model on CIC-IDS2017 database in light of FAR, detection rate and accuracy are denoted in the Figs. 8 and 9.

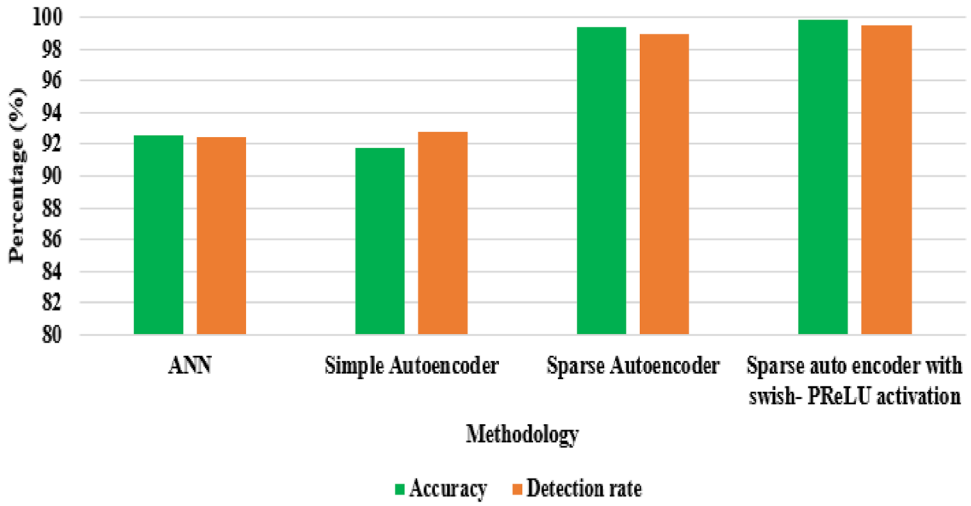

Fig. 6 Graphicalcomparison of sparse autoencoder with swish-PReLU activation model on AWID databasein terms of detection rate and accuracy

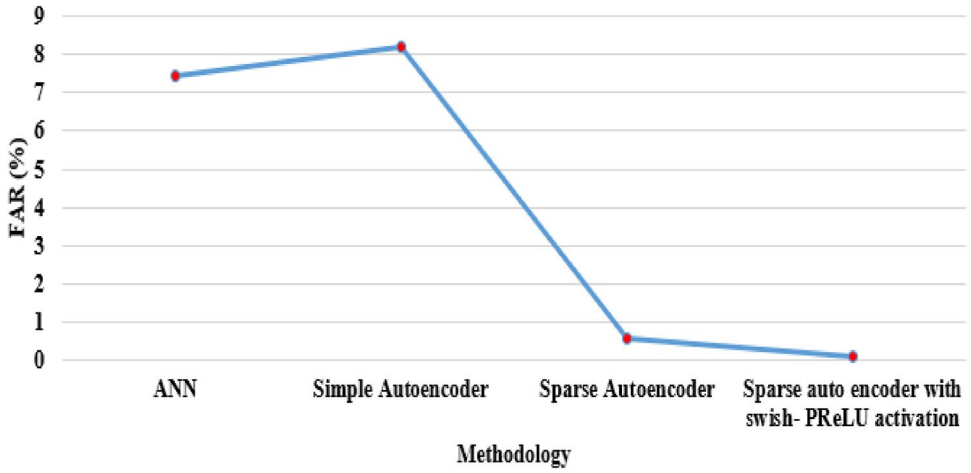

Fig. 7 Graphicalcomparison of sparse autoencoder with swish-PReLU activation model on AWID databasein terms of FAR

Table 7 Performance evaluation of sparse autoencoder with swish-PReLU activation model on CIC-IDS2017 database

Database	Classifiers	Accuracy (%)	Detection rate (%)	FAR (%)
CIC-IDS2017	ANN	91.17	91.75	8.83
	Simple autoencoder	90.88	91.43	9.12
	Sparse autoencoder	99.46	99.86	0.08
	Sparse autoencoder with swish-PReLU activation	99.93	99.96	0.07

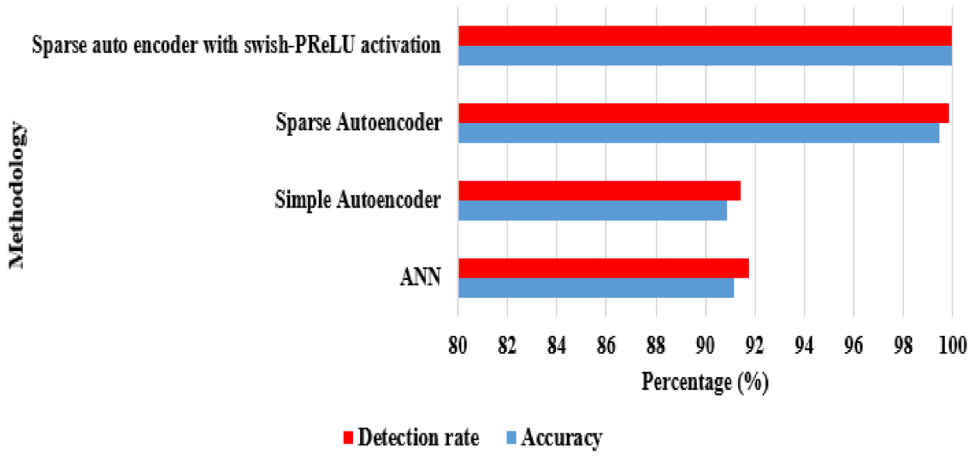

Fig. 8 Graphical comparison of sparse autoencoder with swish-PReLU activation model on CIC-IDS2017 database in light of detection rate and FAR

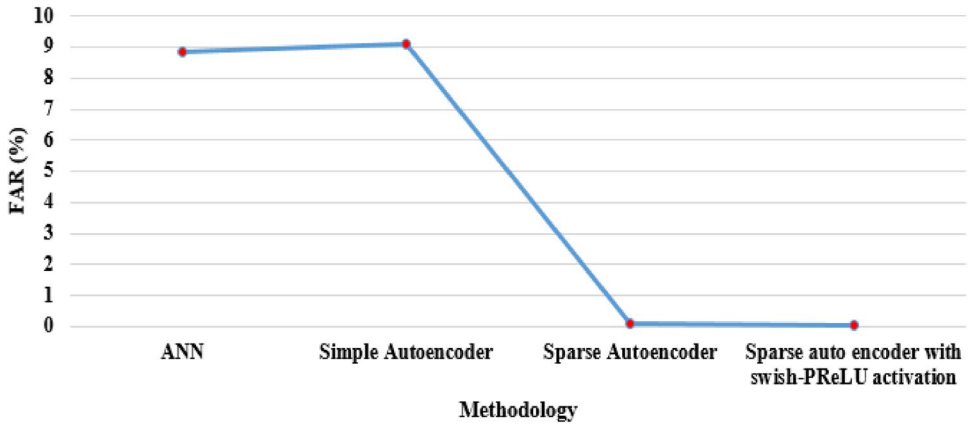

Fig. 9 Graphical comparison of sparse autoencoder with swish-PReLU activation model on CIC-IDS2017 database in terms of FAR

4.2 Comparative investigation

In this section, Table 8 represents the comparative investigation of the proposed and the existing models. Zhou et al. (2020) developed a new intrusion detection system based on ensemble learning and feature selection techniques. In this literature, CFS-BA was introduced to select the optimal feature sub-sets to deal with unbalanced network traffics and higher dimensional. The ensemble techniques like C4.5, random forest, and forest by penalizing attributes were combined with the average of probability combination rule to construct the classification model. In this literature, the developed intrusion detection system performance was validated on NSL-KDD, CIC-IDS2017 and AWID databases. The extensive experiment showed that the developed CFS-BA system achieved 99.81% of accuracy, 0.08% of the FAR value, and 99.80 % of detection rate on NSL-KDD database. In addition, the developed CFS-BA system obtained 99.52% and 99.89% of accuracy, 0.15% and 0.12% of FAR, and 99.5% and 99.9% of detection rate on AWID and CIC-IDS2017 databases respectively.

Additionally, Asad et al. (2020) presented a feed forward backpropagation algorithm; ANN to classify the network flows. In this study, a new malicious pattern was introduced to detect malicious behaviour from the packets. The extensive experiment showed that the developed method achieved

Table 8 Comparative investigation of proposed and existing models

Methods	Database	Accuracy (%)	False alarm rate (%)	Detection rate (%)
CFS-BA (Zhou et al. 2020)	NSL-KDD	99.81	0.08	99.8
Sequential search- Bayesian network (Zhang and Wang 2013)		98.98	0.60	–
Sparse auto encoder with swish-PReLU activation		99.95	0.05	99.82
CFS-BA (Zhou et al. 2020)	AWID	99.52	0.15	99.5
Chi-square and information gain-random tree (Thanthrige et al. 2016)		95.12	0.538	92
Sparse auto encoder with swish-PReLU activation		99.89	0.11	99.54
CFS-BA (Zhou et al. 2020)	CIC-IDS2017	99.89	0.12	99.9
ANN (Asad et al. 2020)		98.694	1.882	98.694
Sparse auto encoder with swish-PReLU activation		99.93	0.07	99.96

98.694% of classification accuracy, 1.882% of the FAR value and 98.694% of detection rate on CIC-IDS 2017 database. Thanthrige et al. (2016) utilized chi-square and information gain to determine the relevant feature subsets. Then, the obtained feature sub-sets were fed to a random tree classifier to classify the finer-grained and higher level class distributions. The simulation outcome showed that the developed system achieved 95.12% of classification accuracy, 0.538% of the FAR value, and 92% of detection rate on AWID database. Zhang and Wang (2013) introduced a wrapper feature selection technique based on Bayesian network for intrusion detection. From the experimental analysis, the developed technique achieved 98.98% of classification accuracy and 0.6% of the FAR value on NSL-KDD database. Compared to these existing works, the proposed sparse autoencoder with swish-PReLU activation model achieved significant performance in intrusion detection employing FAR, detection rate and classification accuracy.

5 Conclusions

In this research paper, a new intrusion detection model; sparse autoencoder with swish-PReLU activation model is proposed to deal with high dimensional data and unbalanced network traffic. Initially, the second percentile method and recursive feature elimination technique are developed to select the optimal feature subsets from the pre-processed data to reduce the computational complexity of the model. In this research study, data pre-processing is accomplished by using label and one hot encoding technique. Finally, a new deep learning model named sparse autoencoder with swish-PReLU activation model is proposed to classify the normal and attack traffic in the NSL-KDD, CIC-IDS2017 and AWID databases. In the experimental section, the proposed sparse autoencoder with swish-PReLU activation model attained better intrusion detection performance in terms of FAR, detection rate and classification accuracy compared to ANN, simple Autoencoder, and sparse Autoencoder. From the simulation result, the proposed sparse autoencoder with swish-PReLU activation model showed a maximum of 4.77% and minimum of 0.04% improvement in classification accuracy related to the existing models like correlation-based feature selection with bat algorithm, chi-square and information gain-random tree, and sequential Search-Bayesian network. In future work, a new clustering algorithm can be included in the proposed model to further enhance the performance of intrusion detection and still need to concentrated on overfitting and data sparsity problems.

Dataset links:

NSL-KDD: https://www.kaggle.com/hassan06/nslkdd

AWID: http://icsdweb.aegean.gr/awid/

CIC-IDS2017: https://www.unb.ca/cic/datasets/ids-2017.html

Funding We haven't received any funding from any sources.

Declaration

Conflict of interest On the behalf of all the authors corresponding author declares that there is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Ahmad I, Basheri M, Iqbal MJ, Rahim A (2018) Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 6:33789–33795

Alazzam H, Sharieh A, Sabri KE (2020) A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst Appl 148:113249

Almasoudy FH, Al-Yaseen WL, Idrees AK (2020) Differentialevolution wrapper feature selection for intrusion detection system. Proc Comput Sci 167:1230–1239

Almiani M, AbuGhazleh A, Al-Rahayfeh A, Atiewi S, Razaque A (2020) Deep recurrent neural network for IoT intrusion detection system. Simul Model Pract Theory 101:102031

Aminanto ME, Choi R, Tanuwidjaja HC, Yoo PD, Kim K (2017) Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. IEEE Trans Inf Forensics Secur 13:621–636

Asad M, Asim M, Javed T, Beg MO, Mujtaba H, Abbas S (2020) DeepDetect: detection of distributed denial of service attacks using deep learning. Comput J 63:983–994

Bhati BS, Rai CS, Balamurugan B, Al-Turjman F (2020) An intrusion detection scheme based on the ensemble of discriminant classifiers. Comput Electr Eng 86:106742

Cavusoglu U (2019) A new hybrid approach for intrusion detection using machine learning methods. Appl Intell 49:2735–2761

Cerda P, Varoquaux G, Kégl B (2018) Similarity encoding for learning with dirty categorical variables. Mach Learn 107:1477–1494

Chen J, Qi X, Chen L, Chen F, Cheng G (2020) Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. Knowl Based Syst 203:106167

Chou TS, Yen KK, Luo J (2008) Network intrusion detection design using feature selection of soft computing paradigms. Int J Comput Intell 4:196–208

Davahli A, Shamsi M, Abaei G (2020) Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks. J Amb Intell Humaniz Comput 11:5581–5609

D'hooge L, Wauters T, Volckaert B, De Turck F (2020) Inter-dataset generalization strength of supervised machine learning methods for intrusion detection. J Inf Secur Appl 54:102564

Ghasemi J, Esmaily J, Moradinezhad R (2020) Intrusion detection system using an optimized kernel extreme learning machine and efficient features. Sādhanā 45:1–9

Gu J, Wang L, Wang H, Wang S (2019) A novel approach to intrusion detection using SVM ensemble with feature augmentation. Comput Secur 86:53–62

Gupta AR, Agrawal J (2020) The multi-demeanor fusion based robust intrusion detection system for anomaly and misuse detection in computer networks. J Amb Intell Humaniz Comput, pp 1–17

Gurung S, Ghose MK, Subedi A (2019) Deep learning approach on network intrusion detection system using NSL-KDD dataset. Int J Comput Netw Inf Secur 11:8–14

Hajimirzaei B, Navimipour NJ (2019) Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Expr 5:56–59

Khammassi C, Krichen S (2020) A NSGA2-LR wrapper approach for feature selection in network intrusion detection. Comput Netw 172:107183

Kumar P, Gupta GP, Tripathi R (2020) A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. J Ambient Intell Humaniz Comput. pp 1–18

Li X, Chen W, Zhang Q, Wu L (2020) Buildingauto-encoder intrusion detection system based on random forest featureselection. Comput Secur 95:101851

Liu J, Zhang W, Tang Z, Xie Y, Ma T, Zhang J, Zhang G, Niyoyita JP (2020) Adaptive intrusion detection via GA-GOGMM-based pattern learning with fuzzy rough set-based attribute selection. Expert Syst Appl 139:112845

Lopez-Martin M, Carro B, Sanchez-Esguevillas A (2020) Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst Appl 141:112963

Lv L, Wang W, Zhang Z, Liu X (2020) A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine. Knowl Based Syst 195:105648

Manimurugan S, Majdi AQ, Mohmmed M, Narmatha C, Varatharajan R (2020) Intrusiondetection in networks using crow search optimization algorithm with adaptiveneuro-fuzzy inference system. Microprocess Microsyst 79:103261

Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. Pattern Recogn 33:1455–1465

Mottini A, Acuna-Agost R (2016) Relative label encoding for the prediction of airline passenger nationality. In: 2016 IEEE 16th international conference on data mining workshops, pp 671–676

Nguyen MT, Kim K (2020) Genetic convolutional neural network for intrusion detection systems. Future Gener Comput Syst 113:418–427

Prasad M, Tripathi S, Dahal K (2020a) An efficient feature selection based Bayesian and Rough set approach for intrusion detection. Appl Soft Comput 87:105980

Prasad M, Tripathi S, Dahal K (2020b) Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection. Comput Secur 99:102062

Qi ZF, Liu QQ, Wang J, Li JX (2017) Battle damage assessment based on an improved Kullback-Leibler divergence sparse autoencoder. Front Inf Tech Electron Eng 18:1991–2000

Rose T, Kifayat K, Abbas S, Asim M (2020) A hybrid anomaly-based intrusion detection system to improve time complexity in the Internet of Energy environment. J Parallel Distrib Comput 145:124–139

Shahriar MH, Haque NI, Rahman MA, Alonso M Jr (2020) G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System. arXiv preprint arXiv:2006.00676

Sharafaldin I, Lashkari AH, Ghorbani AA (2018a) A detailed analysis of the cicids2017 data set. In: International conference on information systems security and privacy, pp 172–188

Sharafaldin I, Lashkari AH, Ghorbani AA (2018b) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In ICISSP, pp 108–116

Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence 2:41–50

Sp RM, Maddikunta PKR, Parimala M, Koppu S, Reddy T, Chowdhary CL, Alazab M (2020) An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. Comput Commun 160:139–149

Su T, Sun H, Zhu J, Wang S, Li Y (2020) BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset. IEEE Access 8:29575–29585

Tang C, Luktarhan N, Zhao Y (2020) An Efficient Intrusion Detection Method Based on LightGBM and Autoencoder. Symmetry 12:1458

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications, pp 1–6

Thang VV, Pashchenko FF (2019) Multistage System-Based Machine Learning Techniques for Intrusion Detection in WiFi Network. J Comput Netw Commun. https://doi.org/10.1155/2019/4708201

Thanthrige USKPM, Samarabandu J, Wang X (2016) Machine learning techniques for intrusion detection on public dataset. In: 2016 IEEE Canadian conference on electrical and computer engineering, pp 1–4

Wu Z, Wang J, Hu L, Zhang Z, Wu H (2020) A network intrusion detection method based on semantic Re-encoding and deep learning. J Netw Comput Appl 164:102688

Zhang F, Wang D (2013) An effective feature selection approach for network intrusion detection. In: 2013 IEEE eighth international conference on networking, architecture and storage, pp 307–311

Zhou Y, Cheng G, Jiang S, Dai M (2020) Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput Netw 174:107247

Zong W, Chow YW, Susilo W (2020) Interactive three-dimensional visualization of network intrusion detection data for machine learning. Future Gener Comput Syst 102:292–306