



# Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications

Manisha Rani<sup>1</sup> · Gagandeep<sup>1</sup>

Received: 29 March 2021 / Revised: 24 August 2021 / Accepted: 22 November 2021 /  
Published online: 2 February 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

The Intrusion Detection System plays a significant role in discovering malicious activities and provides better network security solutions than other conventional defense techniques such as firewalls. With the aid of machine learning-based techniques, such systems can detect attacks more accurately by identifying the relevant data patterns. However, the nature of network data, time-varying environment, and unknown occurrence of attacks made the learning task very complex. We propose a deep neural network that utilizes the classifier-level class imbalance solution to solve this problem effectively. Initially, the network data is preprocessed through data conversion followed by the min-max normalization method. Then, normalized data is fed to neural network where the cross-entropy function is modified to address the class imbalance problem. It is achieved by weighting the classes while training the classifier. The extensive experiments are performed on two challenging datasets, namely NSL-KDD and UNSW-NB15, to establish the superiority of the proposed approach. It includes comparisons with commonly employed imbalance approaches such as under-sampling, over-sampling, and bagging as well as existing works. The proposed approach attains 85.56% and 90.76% classification accuracy on NSL-KDD and UNSW-NB15 datasets, respectively. These outcomes outperformed data-level imbalance methods and existing works that validate the need to incorporate class imbalance for network traffic categorization.

**Keywords** Data Normalization · Neural Networks · Class Imbalance · Intrusion Detection System

---

✉ Manisha Rani  
manishabhandari1993@gmail.com

Gagandeep  
gdeep.pbi@gmail.com

<sup>1</sup> Department of Computer Science, Punjabi University, Patiala 147002, India

## 1 Introduction

With the rapid developments in internet and communication technologies, the volume of stored data increases significantly, which augmented the volumes of traffic flowing through networks from all over the world [1]. With the surge in traffic over networks, cyber-attacks are also rapidly increasing due to novel attacks and mutations of older ones. The frequency of attacks exploiting systems' flaws is expected to escalate as more and more devices connect with the internet [43]. Thus, network security has become a requisite research domain to protect data and networks from malicious users and attackers. Several security solutions have been available to secure networks from external and internal attacks; however, preventing them is still challenging due to the intrinsic limitations of security policies, firewall, access control scheme, and antivirus software [31]. Intrusion Detection Systems (IDS) are deployed to monitor or analyze the ongoing events for detecting potential attacks to deal with these large-scale network threats. It provides real-time protection against internal and external attacks by blocking them upon detection [56].

Intrusion Detection Systems are classified into host-based IDS (HIDS) and network-based IDS (NIDS) primarily [82]. The HIDS relied on the signature of the known attacks to determine the vulnerability of the system. This approach is commonly restricted to a host system running a specific operating system and requires an updated database of attacks, limiting its ability to detect novel attacks [61]. On the other hand, NIDS targets network behavior by analyzing the format and content of network data packets which makes it more suitable to detect unknown attacks as well. However, this approach has its own challenges, including a high false-positive rate due to non-linear nature of the problem, data imbalance originating from attacks, lack of numerical representation of some features, and high dimensionality [27]. To overcome these challenges, machine and deep learning techniques have emerged as a promising tool in contrast to contemporary statistical methods and knowledge-based expert systems [9, 15].

Deep learning has recently emerged as a disruptive technology in this domain also and became hugely popular due to higher accuracy and flexibility with little domain knowledge [32, 63]. The neural networks can discover the relevant patterns to discriminate the attack from the regular traffic through series of non-linear transformations [40]. Also, a little domain knowledge is required to construct models since categorical and strings attributes of the network can be converted into numerical form through integer or one-hot encoding schemes. Many forms of deep learning such as deep neural network (DNN) [35], autoencoders (AE) [4], convolutional neural networks (CNN) [42, 74], and recurrent neural networks (RNN) [70, 83] have been utilized so far to provide the solutions to this challenging problem. Nevertheless, the performance bottlenecks in recognizing attacks from regular traffic still exist due to overfitting and class imbalance problems [26, 27, 74]. Overfitting is common in this domain due to the lack of highly representative training data points from real-world applications corresponding to the true data distribution. For instance, there is a huge disparity in performance on the popular but outdated dataset KDDCup'99 for training (>95%) and test sets (<90%) [69]. Consequently, a more challenging dataset NSL-KDD was developed from the KDDCup'99 dataset but suffered from the same problem. Therefore, multiple datasets are required to validate the IDS model. When insufficient training data is supplied to neural networks, the model captures noise or superficial information. Another problem associated with the IDS is the class imbalance which refers to the uneven distribution of class samples in the dataset where majority class outnumbers the minority class significantly. When such data is provided to neural networks, they tend to be biased

towards the majority class which causes poor performance on the unseen data. Therefore, further improvements can be made to obtain better-generalized models by considering the above issues while developing IDS.

This paper presents a neural network based approach to categorize the network traffic into normal and attack classes. The key contributions of this work are two-fold. Firstly, we showed data normalization and selection of neural network components such as initialization and activation functions play an essential role in identifying the anomalies from the traffic data through empirical analysis. Secondly, the crucial issue of class imbalance has been tackled at the classifier level. A cost-sensitive loss function is designed for imbalanced training and demonstrated to be more valuable than the data level approaches for obtaining generalized classification models. The experiments are performed on challenging NSL-KDD and UNSW-NB15 datasets to assess the performance. The results are also compared with the previous works to measure the improvement in classification performance. The obtained outcomes show that the balanced NN outperformed not only machine learning approaches but also advance deep learning approaches, including CNN and RNN. These outcomes also validate the data preprocessing and class imbalance as a crucial barrier impeding the development of better IDS models.

The rest of the paper is organized as follows: Section 2 discusses the previous work on IDS using NSL-KDD and UNSW-NB15 datasets. The proposed IDS methodology using NN has been provided in Section 3, and the experimental outcomes are provided in Section 4. Section 5 concludes the work.

## 2 Literature survey

Numerous approaches have been proposed to build effective IDS from network traffic data by employing data preprocessing, feature reduction with feature subset selection, PCA and AE, machine learning and deep learning techniques. Initially, we briefly review the existing machine learning based approaches to identify their shortcomings and then present the neural network based efforts to solve this problem.

The machine learning based approaches had dominated the research in NIDS for the last two decades. These works mainly employed data normalization [44, 57], class imbalance [68]–[60], feature selection [59, 72], ensemble learning [16, 76], fuzzy rules [39] and comparison of various classifiers [34, 86] over several datasets. But, these approaches fail to address the inherent complex characteristics of the data distribution and usually lack in performance. The feature selection methods were unable to improve the classification accuracy and suffer from overfitting problem as observed from many studies [59, 72], and our experiments [62]. The feature extraction also suffers from similar issues [55], and the comparisons of different learning algorithms or ensemble learning [16, 76] were also not able to yield satisfactory performance. The class imbalance problem which is inherently manifested in IDS has been addressed by a few researchers at the data level. For instance, Wu et al. [68] combined  $k$ -means clustering and SMOTE and Tan et al. [67], and Priyadarsini [60] used SMOTE on full set of features with random forest classifier. The latter authors also performed feature selection using artificial bee colony algorithm.

On the other hand, deep learning showed significant improvements comparatively. Several forms of the artificial neural network (ANN) have been used to solve this problem, including self organization map (SOM) by Ibrahim et al. [23] for classifying attacks on

KDDCup'99 and NSL-KDD datasets. Almi'ani et al. [3] clustered the outputs of SOM with  $k$ -means where largest cluster represents normal traffic while smaller one indicates the attack using NSL-KDD problem. Moukhafi et al. [51] used it with genetic algorithm over KDDCup'99 and UNSW-NB15 datasets. The works based on feedforward neural networks include comparing several backpropagation training algorithms on NSL-KDD [25] and selecting features on UNSW-NB15 dataset [48] using MLP classifier. Ding and Wang [12] and Kim et al. [35] used slightly different DNN models to detect anomalies in KDDCup'99 dataset. The convolutional-based approaches have been utilized in [42] where ResNet 50 and GoogLeNet models were used to identify attacks through conversion of NSL-KDD data into images. Using CNN, Wu et al. [77] proposed a cost-sensitive imbalance approach, Nguyen et al. [54] detected DoS attacks and Verma et al. [74] applied Adaptive Synthetic (ADASYN) sampling on NSL-KDD dataset to overcome the class imbalance. Some works focused on the feature transformation through AE, where Azar et al. [84] transformed the NSL KDD dataset and compared four classifiers. Zhang et al. [85] added a softmax layer to AE for the NSL-KDD problem whereas Khan et al. [78] proposed a two-stage semi-supervised approach using AE for KDDCup'99 and UNSW-NB15 problems. Al-qatf et al. [4] used the combination of AE and SVM for solving binary and multiclass problems. Dong et al. [13] selected features with the multivariate correlation analysis and fed them to LSTM to solve NSL-KDD and UNSW-NB15 classification problems. Yin et al. [83] used RNN to solve binary and multiclass classification problems whereas Tchakoucht and Ezziyyani [70] used multi-layered echo-state machine on NSL-KDD and UNSW-NB15 datasets.

Recently, Sethi et al. [64] used Deep-Q-Network to discriminate normal and attack samples on UNSW-NB15 dataset. Wu et al. [78] combined Deep Belief Network (DBN) and feature weighing to develop the IDS using NSL-KDD dataset. The deep neural networks have been used with PCA reduced data dimensionality in [63] for NSL-KDD dataset and extra trees ranked features of UNSW-NB15 and AWID datasets in [29]. Ashiku and Dagli [7] used CNN to classify attacks present in UNSW-NB15 dataset whereas Su et al. [66] proposed multi-layered BAT-MC model comprising of multiple CNN layers, a bi-directional LSTM layer, and an attention layer to detect anomalies in network traffic. Wu et al. [79] combined AE and kernel machine learning to solve the NSL-KDD problem. The stacked AE was used for feature transformation, kernel approximation was performed with random fourier feature selection approach, and linear SVM was used for learning. Ieracitano et al. [24] proposed combining AEs and LSTM for the same problem. Mighan and Kahani [49] extracted latent features using stacked AE and used SVM for classification. Jaing et al. [27] used the combination of under and oversampling to deal with the data imbalance problem where One-Side Selection (OSS) was used to reduce the noise samples in majority category, and SMOTE was used to increase the minority samples. A deep hierarchical model combining CNN and Bi-directional LSTM was employed for classification. Using CNN, the cost-sensitive approaches to handle class imbalance were proposed in [21, 38] with focal loss functions for NSL KDD, UNSW-NB15, and Bot-IoT datasets.

The analysis of existing works reveals that the neural network based approaches are becoming more prevalent with the recent advances in deep learning. The majority of the researchers have developed end-to-end models using deep neural networks, CNN, and LSTM approaches for classification of traffic and employing AE primarily to reduce data dimensionality. These models yield some fruitful outcomes in terms of data dimensionality as well as better classification. However, the current works focus more on advanced deep learning approaches to alleviate the performance bottlenecks in this domain instead of resolving the class-imbalance issue. As a result, this relevant issue has been tackled in few works only where data and classifier level methods have been suggested. Additionally,

recent efforts are directed towards cost-sensitive approaches. So, there is scope to develop the neural network based generalized IDS models through classifier-level balanced learning.

### 3 Materials and methods

This section provides the details of the datasets considered in the study and the proposed approach for the classification of normal and attacks network traffic.

#### 3.1 Datasets

In this paper, two challenging datasets, NSL KDD and UNSW-NB15, are collected from public sources to develop effective IDS. A brief description of both datasets is given as follows:

##### 3.1.1 NSL KDD dataset

The KDDCup'99 dataset has been commonly used to solve network intrusion with ML techniques. The classifiers tend to overfit it due to numerous redundant records. The number of repeated samples belonging to attacks far exceed the normal class sample in both training (more than 90%) and testing (more than 80%) sets of this dataset, which introduced significant bias and resulted in high accuracy. To resolve this issue, a more challenging NSL KDD set was proposed that consists of KDDTrain+ and KDDTest+ having 125,973 and 22,544 samples respectively. The details of the dataset are provided in Table 1 along with the class balance ratio.

##### 3.1.2 UNSW-NB15 dataset

Another widely used dataset is the UNSW-NB15 which is developed with IXIA Perfect Storm tool at Australian Centre for Cyber Security. Unlike conventional datasets, it contains several modern synthesized attacks, including worms, fuzzer, generic, and reconnaissance. The attributes are categorized into five groups: flow, basic, content, time, and additional generated features. Table 2 provides the details about this dataset along with the class balance ratio.

### 3.2 Proposed work

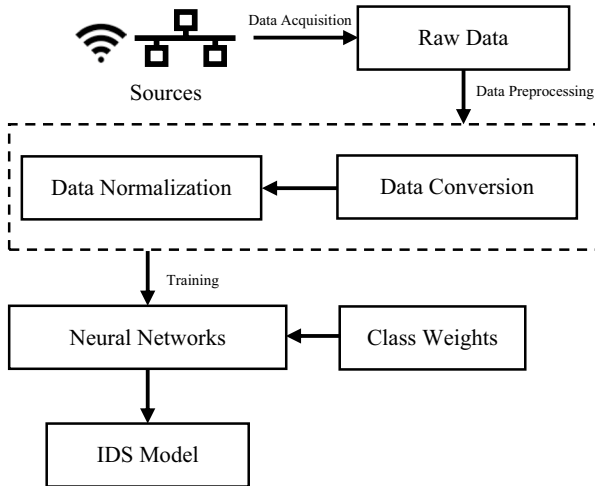
Figure 1 shows the flowchart of the proposed IDS approach with neural networks. The raw traffic data from networks contain heterogeneous feature types such as binary, numerical

**Table 1.** Description of NSL KDD dataset

Set	Instances	Classes		Balance Ratio (Attack/Normal)
		Normal	Attack	
KDDTrain+	125,973	67,343	58,630	0.87
KDDTest+	22,544	9,711	12,833	1.32

**Table 2.** Description of UNSW-NB15 dataset

Set	Instances	Classes		Balance Ratio (Attack/Normal)
		Normal	Attack	
Train	175,341	56,000	119,341	2.13
Test	82,332	37,000	45,332	1.22

**Fig. 1.** Flowchart of the proposed IDS approach

and categorical. Initially, categorical is converted to numerical form using integer encoding to attain homogeneity. Afterward, data is normalized to have a uniform data distribution for all features. Then, normalized data is fed to the neural networks model to differentiate the traffic into normal and attack. The model is trained with the cost-sensitive loss function through class weights to deal with the imbalance issue. The weight for each class is computed based on a heuristic measure which is proven to be useful for better performance. The detailed methodology is provided as follows:

### 3.2.1 Data preprocessing

Data preprocessing is a prerequisite step in modeling neural networks for analyzing complex features. It includes the transformation of categorical data and rescaling of data through normalization. Both NSL-KDD and UNSW-NB15 datasets consist of numeric and categorical attributes. In these datasets, three features are categorical while 38 features of NSL-KDD and 39 features of UNSW-NB15 are numeric. The categorical features are converted into numerals using integer encoding scheme. For instance, ‘protocol’ attribute in NSL-KDD defines TCP, UDP, and ICMP protocols to make the connection. To convert this attribute into the numeric form, each protocol is assigned with an integer. In this case, TCP, UDP, and ICMP values are converted into 1, 2, and 3 respectively.

After converting all features, it is essential to rescale them in a uniform range for better performance. The unnormalized features with different ranges introduce bias in learning as

the greater numeric feature values dominate the smaller ones [65]. The training algorithms of neural networks also fail to converge due to an uneven range of features [19]. Thus, data is normalized with the min-max method on the basis of the empirical analysis involving several normalization methods. This normalization has been widely used in existing IDS approaches also [4, 23]. In this method, the features are rescaled to the interval [0, 1], and it is given as follows:

$$\hat{x}_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (1)$$

where  $x_i$  represents the  $i^{\text{th}}$  feature of data (denoted as  $x$ ) to be learned by the classifier, min and max represents the minimum and maximum value the  $i^{\text{th}}$  feature respectively.

### 3.2.2 Neural networks

The recent breakthroughs in artificial neural networks have improved the learning capabilities of machines by manifolds. This powerful learning technique known as deep neural networks has achieved state-of-the-art performance on numerous classification problems in the field of image recognition [21, 38], speech [18, 22], and natural language processing [11, 28]. Deep neural networks consist of multiple layers of non-linear functions which are composed in series. These networks are different from the conventional shallow networks which allow better function approximation. The shallow networks correspond to the model with a single hidden layer, whereas several layers correspond to deeper networks. DNNs are preferred over shallow networks as more compact representation of the same functions can be achieved. Other critical advances in the neural network are better activation functions, parameter initialization, and backpropagation training algorithms. We discuss these components in detail as follows:

Firstly, the activation functions play a vital role in the performance of neural networks. The rectified linear unit (ReLU) is the most popular non-linear activation function that converges very quickly compared to smoother functions such as sigmoidal and hyperbolic tangent [40, 53]. It is given as  $f(z) = \max(z, 0)$  where  $z$  represents the input units. It preserves the linear nature for positive values while pruning the negative ones. Thus, the sparse activations helps to obtain superior performance as well as to avoid the vanishing gradient problem. However, dropping negative values is not always helpful and causes the dead neuron problem [81]. This problem deactivates a large portion of the network and thereby limiting the contribution of the model. To resolve this issue, better non-linear activation functions belong to the rectified unit family have been proposed. The Leaky ReLU [45] is one of the popular choices among them that allows the nonzero gradient for the negative values. It is given as follows:

$$y = \begin{cases} z & \text{if } (z \geq 0) \\ az & \text{otherwise} \end{cases} \quad (2)$$

where  $z$  indicates the input units,  $y$  indicates the output units, and  $a$  is a fixed parameter in intervals 0 and 1 that controls the negative units. This function allows all positive values and some parts of the negative units. As the derivative is not always zero, the probability of the silent neurons reduces significantly. Leaky ReLU has outperformed ReLU in numerous studies [10] and was also found to be best suited to small datasets as compared to ReLU during the empirical analysis of four ReLU based functions [80].

Secondly, the neural networks are susceptible to random initialization of networks which can cause vanishing gradient and exploding gradient problems. As a result, the optimizers such as standard gradient descent usually perform poorly with deeper networks than the statistics-based weight initializers. Thus, parameter initialization also plays an essential role in the performance of different architectures. Glorot initialization [17] is the most common approach in which the network is initialized by selecting weights from the normal distribution between  $(-s, s)$  where

$$s = \sqrt{\frac{6}{fan - in + fan - out}} \tag{6}$$

where fan-in and fan-out indicate the numbers of inputs and outputs to a neuron.

Lastly, the neural networks have been primarily trained with gradient descent (GD) optimization techniques. The gradient descent minimizes the objective function, typically cross-entropy, by updating the parameters of neural networks in the direction opposite to gradient. However, these methods suffer from convergence issues, especially when the loss functions are non-convex. They converge very slowly and require an adequate learning rate depending upon the problem which makes them highly susceptible to trap in a local minimum. Therefore, stochastic gradient descent (SGD) algorithms are employed to find solutions with low training error and provide good generalization [20]. Adam [37] is one of the most successful first order SGD optimization techniques that accelerates the training with adaptive step size and momentum. This method maintains the learning rate for every parameter by averaging the first and second moments of the gradients and also introduces the bias correction. This improves the convergence rate and makes Adam an excellent training algorithm that can realize the advantages of AdaGrad (adapts learning rate for optimization) and RMSProp (bias-correction allows sparse gradients) methods. It is given as follows:

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

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3}$$

where  $t$  indicates the time step,  $m_t$  indicate the first moment,  $v_t$  indicate the second moment, and  $\beta_1$  and  $\beta_2$  are the decay factor. The bias correction is given as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \text{ and } \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{4}$$

where  $\beta_1^t$  and  $\beta_2^t$  are the  $t^{th}$  power of  $\beta_1$  and  $\beta_2$  respectively. Each parameter is then updated as follows:

$$\theta_t = \theta_{t-1} - \frac{\alpha_t}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{5}$$

where  $\alpha_t$  is the learning rate, and  $\epsilon$  is a small constant to avoid division by zero.

The proposed multilayer neural networks consist of an input layer, two hidden layers, and one output layer. The activation functions at the hidden layers are Leaky ReLU with the parametric setting of  $a = 0.3$ , whereas *softmax* is used at the output layer. The number of neurons is set to 100 and 200 at the hidden layers. The network is trained using mini-batch of 128 training samples and by minimizing the cross-entropy between the



binary class label vector  $y = [y_A, y_N]$  and the output probability vector  $\hat{y} = [\hat{y}_A, \hat{y}_N]$ . The binary cross-entropy is given as follows:

$$E(y, \hat{y}) = -(y_A \log(\hat{y}_A) + y_N \log(\hat{y}_N)) \quad (7)$$

Adam is used to training the network with the following parameters: learning rate ( $\alpha_t$ ) is set to 0.001, and  $\beta_1$  and  $\beta_2$  are set to 0.9 and 0.999 respectively.

### 3.2.3 Class imbalance

The issue of class imbalance is prevalent in intrusion detection problems due to the lack of sufficient data pertaining to attacks. It refers to improper distribution of data in which one class contains a significantly large number of instances compared to others. While training the model with such data, the classifiers tend to be more skewed towards majority class instances than minority class ones. Although the performance of the model becomes high due to the biased classification, minority class instances suffer greatly. The results usually contain a high false alarm rate. Therefore, this problem needs to be addressed before training the model so that the underrepresented class could have the same importance in learning as does the majority class. Class imbalance can be tackled using several ways at the data and classifier level. At the data level, sampling the data is the popular technique that aims to balance the class distribution of the training data [71]. The balancing can be achieved by either selecting fewer instances of the majority class to equalize the minority class instances which is known as under sampling or adding new instances to the minority class to balance the majority class instances which is known as oversampling.

At the classifier level, this issue is handled with the cost-sensitive approach. In this approach, the penalty for the misclassification of minority classes is applied to force the learning algorithms to focus more on these classes [8]. It is attained by either assigning weights to the classes depending upon their number of instances or explicitly adjusting the prior probabilities of the classes. The first approach is more commonly used to balance different classes at the training time. It is implemented in the form of a loss function by using different error penalties for classes. On the other hand, testing time cost assignment is done either by assigning threshold values to the prior probabilities or adjusting F-score [8].

In this work, weights are assigned to the classes using heuristic measures [36], which is proven to be helpful in solving many classification problems effectively in recent times [41], [58], [47]. The weights are determined for each class as follows:

$$w_A = \frac{|A| + |N|}{2 |A|}$$

$$w_N = \frac{|A| + |N|}{2 |N|} \quad (8)$$

where  $|A|$  and  $|N|$  denote the total samples belonging to attacks and regular traffic respectively, and  $w_A$  and  $w_N$  are the corresponding loss weights. The weight for each class modifies the binary cross-entropy function for training as follows:

$$E(y, \hat{y}) = -(w_A y_A \log(\hat{y}_A) + w_N y_N \log(\hat{y}_N)) \quad (9)$$

This class-weighted function is used to train neural networks for identifying attacks from normal traffic.

## 4 Experimental results

The proposed methodology is implemented in python using tensorflow and sklearn using Intel® Core™ i5-Processor and 8GB RAM. Two widely used datasets, namely NSL-KDD and UNSW-NB15, have been considered to validate the proposed approach. The performance of the model is evaluated based on four metrics, including classification accuracy, precision, recall, F-score, area under the curve. These metrics are described as follows:

**Classification accuracy** It is defined as the percentage of correctly classified instances to the total instances. It is given as follows:

$$ACC = \frac{TP + FN}{TP + TN + FP + FN}$$

where  $TP$  indicates the True Positive, the number of attacks that are correctly identified as an attack,  $TN$  indicates the True Negative, the number of normal packets that are correctly identified as normal,  $FP$  indicates the False Positive, normal packets that are incorrectly identified as attacks and  $FN$  indicates the False Negative, attacks that are incorrectly identified as normal traffic.

**Precision** It is defined as the percentage of correctly identified attacks to the total number of records classified as attacks. It is given as follows:

$$Precision = \frac{TP}{TP + FP}$$

**Recall** It measures the percentage of correctly identified attacks versus the total number of attacks and is given as follows:

$$Recall = \frac{TP}{TP + FN}$$

**F1-score** It is the harmonic mean of precision and recall metrics and is computed as follows:

$$FS = \frac{2 * (Precision * recall)}{Precision + recall}$$

**Area under the ROC curve (AUC)** This measure is defined in terms of the receiver operating characteristic (ROC) curve to measure classifier performance. The ROC curve is obtained by plotting precision and recall at different decision threshold values. The AUC measure is estimated by computing the area under this curve.

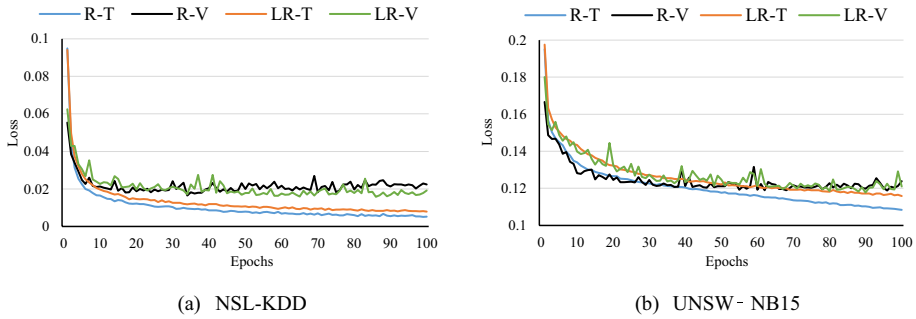


Fig. 2. Comparison of activation function and parameter initialization with two hidden layered NN

### 4.1 Fine-tuning NN architecture

To validate the proposed model for identifying attacks, we performed several experiments using the training sets of NSL-KDD and UNSW-NB15 datasets. The holdout approach is used for this purpose where train sets of both datasets are split into 70% for training and 30% for validation. Figure 2 compares the impact of activation and parameter initialization using the proposed two-layered NN. Specifically, the performance of activation function Leaky ReLU with glorot initialization has been compared with the combination of ReLU activation function and uniform initialization. From the outcomes, the overfitting with ReLU and uniform initialization can be observed on both datasets in contrast to the proposed approach. In the NSL-KDD dataset, even though ReLU attains better convergence than Leaky ReLU, the validation error exceeds the Leaky ReLU after 40 epochs. Similarly, the ReLU obtains better convergence, but validation error does not improve after 50 epochs and is comparable to Leaky ReLU in the UNSW-NB15 dataset. With Leaky ReLU, the training and validation errors are almost similar, thereby showing minor overfitting only. These outcomes showed the negative values have a crucial role in solving both problems. Leaky ReLU reduces the frequency of silent neurons by introducing leak correction in negative units, thereby allowing more neurons to have values during training. Additionally, parameter initialization with glorot scheme also complements the performance of

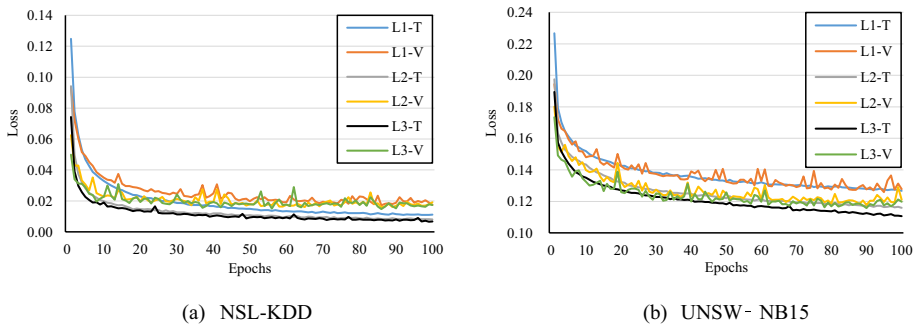


Fig. 3. Effect of network depth on NN performance for IDS

the network as compared to the uniform one. Thus, the proposed activation function and initialization is the better choice for IDS classification.

Further, we analysed the effect of network depths on obtaining generalized models for both datasets. The outcomes are shown in Fig. 3 where the comparison has been made with one, two, and three hidden layers with the Leaky ReLU activation and glorot initialization. From the outcomes, it is evident that the shallow network does not attain good performance. Further, when comparing two and three-layered architectures, the former one emerges as the better choice. In the NSL-KDD dataset, the performance is almost similar to both architectures. On the other hand, the three-layered architecture overfits in contrast to the two-layered one. Thus, two-layered architecture is the best choice to develop the IDS models.

## 4.2 Impact of data normalization

The impact of data normalization on both datasets has been measured to obtain better quality of data. However, several normalization methods are available for this purpose; the best one depends upon the data itself. Thus, these works consider the best normalization method based on empirical analysis. Four widely used methods, namely *z*-score, min-max, pareto scaling, and *tanh* have been considered for experiments. Table 1 provides the outcomes for each method on both datasets. From the outcomes, it is evident that the data normalization improves the classification of IDS with deep neural networks. Additionally, min-max normalization helps to attain better performance than the other three methods. The un-normalized data result in very poor performance. The best and worst accuracy difference is 22.67% and 11.61% on NSL-KDD and UNSW-NB15 datasets respectively, when comparing the un-normalized with min-max outcomes. Further, the improvement is more than 3% on NSL-KDD data while accuracy is improved by 2% on UNSW-NB15 data with the min-max method. Thus, it can be concluded that data preprocessing with normalization plays a crucial role in building better prediction models for detecting attacks. Subsequently, we considered the min-max normalized data for further experiments based on the empirical evidence (Table 3).

## 4.3 Impact of class imbalance

The class imbalance problem is the critical aspect of the IDS model demanding a relevant approach. For this purpose, we have measured the effects of class imbalance on classification performance with several techniques. Table 4 provides the outcomes obtained with the popular methods to deal with class imbalance. It includes imbalanced data as well as data and classifier level approaches. At the data level, three widely used approaches, namely,

**Table 3.** Effect of data normalization on the classification performance of IDS systems

Normalization Method	NSL-KDD	UNSW-NB15
Unnormalized	61.41%	75.02%
<i>z</i> -score	80.03%	84.54%
Min-Max	84.08%	86.63%
Pareto	80.91%	82.98%
<i>tanh</i>	78.48%	83.32%

**Table 4.** Comparison of imbalance data with class-balanced approaches on IDS datasets

	Imbalance Method	NSL-KDD	UNSW-NB15
Imbalanced data		84.08%	86.63%
Data Level	Under-sampling with Cluster Centroids	81.80%	90.35%
	Over-sampling with SMOTE	84.42%	90.82%
	Bagging with Majority Voting	73.09%	75.88%
	Bagging with Soft Decision	77.79%	76.73%
Classifier Level	Class Weights	85.56%	90.76%

over-sampling, under-sampling, and bagging, have been considered. Specifically, cluster centroid has been chosen as an under-sampling method in which centers of the clusters are determined from the majority class samples using  $k$ -means algorithm to reduce the instances. SMOTE has been selected as the over-sampling method where samples from the minority class are generated using  $k$  nearest neighbors randomly. In the bagging approach, multiple training sets are used where each set has the same number of instances from both classes. In this study, the training sets have been made by partitioning the majority class into five non-overlapping subsets. The equal number of minority class instances have been selected randomly to make balanced sets. The classification decision has been determined with soft and hard decision strategies. In the soft decision, the average of prediction probabilities from 5 sets has been used to determine the final class. On the other hand, the majority voting scheme has been employed in the hard decision strategy.

The obtained outcomes show the class imbalance cause lower performance on both datasets. In terms of the best class imbalance approach, the classifier level method performs better than the data level approaches. The lower performance is attained with both bagging strategies (less than 80% accuracy on both datasets). Further, oversampling with SMOTE achieves better performance than the cluster centroids method. The proposed classifier level scheme outperforms SMOTE on the NSL-KDD dataset by more than 1.12%. The higher accuracy is also observed on the UNSW-NB15 dataset, but accuracy is 0.06% less which is a minor difference. Thus, empirical evidence indicates the superiority of classifier-level class imbalance in dealing with the IDS problem.

**Table 5.** Performance of proposed approach for the prediction of attacks on different metrics

Metric	Datasets	
	NSL-KDD	UNSW-NB15
ACCURACY	85.56%	90.76%
PRECISION	97.09%	89.32%
RECALL	76.94%	94.52%
F1SCORE	85.85%	91.85%
AUC	94.29%	97.46%

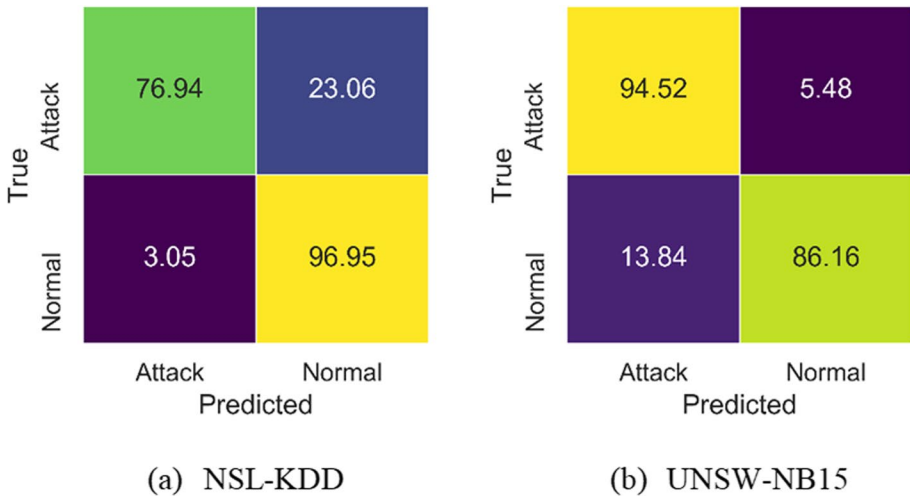


Fig. 4. Confusion matrices for the NSL-KDD and UNSW-NB15 datasets

#### 4.4 Performance with proposed approach

The performance of the proposed approach in terms of accuracy, precision, recall, F-score, and AUC for detecting attacks has been provided in Table 5. Figure 4 shows the confusion matrices for the NSL-KDD and UNSW-NB15 datasets. From the outcomes, it can be seen that the proposed approach alleviates the class imbalance problem effectively. The minority attack samples on NSL-KDD have been classified more precisely, and a similar trend has been observed on UNSW-NB15, where normal samples belong to the minority class. The proposed approach detects most of the attacks (>94%) on UNSW-NB15 dataset. In contrast, performance on NSL-KDD dataset is lower, a primary challenge that this dataset poses for machine and deep learning approaches. The F-score is above 85% and 91% for both datasets. Nonetheless, the AUC measure above 94% and 97% for NSL-KDD and UNSW-NB15 respectively show the superiority of the proposed approach in characterizing

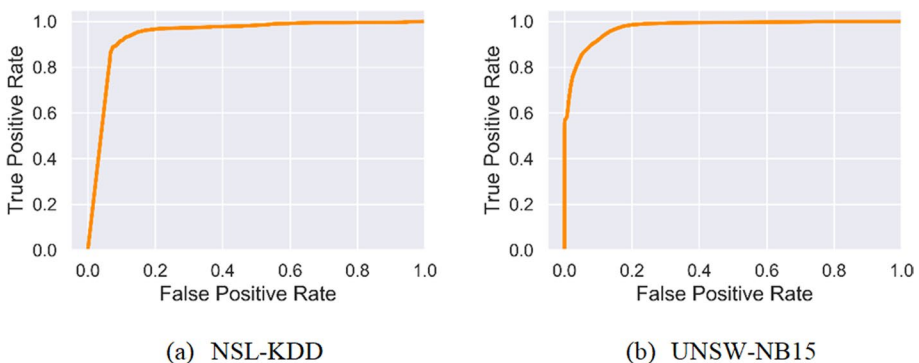


Fig. 5. ROC with the proposed approach for the NSL-KDD and UNSW-NB15 datasets

**Table 6.** Comparison of the proposed approach with the existing works on NSL-KDD dataset

Author	Year	Methodology	Data Normalization	Class Imbalance	Accuracy (%)
Tavallae [69]	2009	NBTree	No	No	82.02
Kromer [39]	2011	Fuzzy classifier + GP	No	No	82.74
Mohammadi [50]	2012	RCDFT	No	No`	80.14
Ibrahim [23]	2013	SOM ANN	Yes	No	75.49
Ingre [25]	2015	ANN	Yes	No	81.20
Yin [83]	2017	RNN-IDS	Yes	No	83.28
Yousefi-Azar [84]	2017	Auto Encoders + GNB	No	No	83.34
Ashfaq [6]	2017	Neural Network	Yes	No	84.12
Paulauskas [57]	2017	Ensemble of J48 + PART	Yes	No	84.84
Li [42]	2017	GoogLe Net	Yes	No	77.04
Li [42]	2017	ResNet50	Yes	No	79.14
Tchakoucht [70]	2018	RNN + ML-ESM	Yes	No	83.00
Almi'ani [3]	2018	Clustered SOM	Yes	No	83.46
Al-qatf [4]	2018	STL-IDS	Yes	No	84.96
Vinayakumar [75]	2019	DNN	Yes	No	80.10
Zhang [85]	2019	Deep Auto Encoder	Yes	No	79.74
Dong [13]	2019	MCA-LSTM	Yes	No	80.52
Verma [74]	2019	CNN-1D	Yes	Yes	84.29
Rawat [63]	2020	PCA+DNN	No	No	79.30
Khraisat [34]	2020	Hybrid of C5 DT + OC-SVM	No	No	83.24
Jaing [27]	2020	Deep Hierarchical Network	Yes	Yes	83.58
Alsaadi [5]	2020	NIDeMFO	No	No	84.14
Ieracitano [24]	2020	AE	Yes	No	84.21
Su [66]	2020	Bi-directional LSTM	Yes	No	84.25
Wu [79]	2020	SSAE-FSVM	Yes	No	84.60
Wu [79]	2020	JSAE-FSVM	Yes	No	85.80
Wu [78]	2020	DBN-WSVM	Yes	No	85.73
Mulyanto [52]	2021	FL-NIDS + DNN	No	Yes	83.95
Mulyanto [52]	2021	FL-NIDS + CNN	No	Yes	84.89
Proposed approach		CS-NN	Yes	Yes	85.56

GP = genetic programming, GNB = Gaussian Naive Bayes, RCDFT = Reduced Class-dependent Feature Transformation, MCA-LSTM = Multivariate Correlation Analysis - Long Short Term Memory, ML-ESM = Multilayered-Echo State Machine, OC-SVM = One Class Support Vector Machine, FL-NIDS = Focal Loss Network Intrusion Detection System, NIDeMFO = Network Intrusion Detection based on Matched Filter Optimization, SSAE-FSVM = Separate training with Stacked AutoEncoder and kernel approximation with random Fourier based SVM, STL-IDS = Self taught learning IDS, DBN-WSVM = Deep Belief Network with feature weighted SVM, JSAE-FSVM = Joint training with Stacked AutoEncoder and kernel approximation with random Fourier based SVM, CS-NN = Cost-Sensitive-Neural Networks

normal and attacks samples. Figure 5 shows the corresponding ROC curves for both datasets.

**Table 7.** Comparison of the proposed approach with the existing works on UNSW-NB15dataset

Author	Year	Methodology	Data Normalization	Class Imbalance	Accuracy (%)
Moukhafi [51]	2018	SOM + GA	No	No	82.91
Dong [13]	2019	MCA-LSTM	Yes	No	88.11
Khan [33]	2019	Two stage deep learning + AE	Yes	Yes	89.71
Gao [14]	2019	ELM and MVT	Yes	No	89.71
Mebawondu [48]	2020	IG and MLP	No	No	76.96
Jaing [27]	2020	Hybrid sampling + Deep Hierarchical Network	Yes	Yes	77.16
Sethi [64]	2020	Deep reinforcement learning	Yes	No	83.80
Kasongo [29]	2020	FFDNN	Yes	No	85.48
Kasongo [30]	2020	XGBoost algorithm + DT	Yes	No	90.85
Umar [73]	2020	Relief + DT	Yes	No	86.41
Aleesa [2]	2021	RNN-LSTM	Yes	No	85.42
Mulyanto [52]	2021	FL-NIDS + DNN	No	Yes	86.04
Mulyanto [52]	2021	FL-NIDS + CNN	No	Yes	86.73
Man [46]	2021	RLC-CNN	Yes	Yes	87.90
Man [46]	2021	RLF-CNN	Yes	Yes	88.70
Proposed approach		CS-NN	Yes	Yes	90.76

IG = Information Gain, FFDNN = Feed Forward Deep Neural Network, DT = Decision Tree, RLC-CNN = CNN with simple cross-entropy loss function, RLF-CNN = CNN with focal loss function, ELM = Extreme Learning Machine, MVT = Multi-Voting Technology, CS-NN = Cost-Sensitive-Neural Networks

#### 4.5 Comparison with existing works

The performance of the proposed approach has been compared with the existing works to validate its superiority in developing a good IDS system. The comparisons have been made on the test sets of NSL-KDD and UNSW-NB15 datasets using accuracy, and the outcomes are provided in Tables 6 and 7 respectively. The proposed approach has obtained satisfactory results on both datasets, and evidently, class imbalance plays a crucial role in better performance. Most previous works have not considered this issue, thereby unable to achieve better performance on both datasets. In NSL-KDD dataset, most of the earlier results (22 out of 29) reported accuracy between 80% and 85% as shown in Table 6. One of the data normalization methods has been used in many reported works (20 out of 29). But, only four methods deal with the class imbalance problem, and these methods reported accuracy in the range of 84% and 85% which is higher than other approaches. Two works have achieved 85.73% and 85.80% accuracy, which is 0.17% and 0.24% higher than the proposed work. However, both approaches are much complex than the simple and straightforward proposed approach. In [78], the combination of deep belief network, feature weighting, particle swarm optimization, and SVM was proposed. In [79], autoencoders, kernel approximation, and linear SVM were combined for better performance. Additionally, the class imbalance was not considered in both works

In the UNSW-NB15 dataset, most of the works reported accuracies below 90%. Also, most works used data normalization (11 out of 15) for performance gains, which is similar to



NSK-KDD dataset. On the other hand, several works incorporated the data imbalance issue in their method (6 out of 15) while working on this dataset, unlike NSL-KDD as shown in Table 7. In terms of performance, the proposed approach outperforms all approaches except one. Higher accuracy of 90.85% is reported by Kasongo [30] using Decision Tree classifier. However, the reported F1-score of 88.45% is lower than the proposed approach (91.85%). These outcomes validate the proposed approach for solving the IDS classification problem more effectively in contrast to previous works.

## 5 Conclusion

This paper presents a simple and straightforward neural-based approach for differentiating the regular and attacks traffic. To establish the superiority of the proposed approach, rigorous experiments are performed using two challenging datasets NSL-KDD and UNSW-NB15. Initially, the empirical analysis of normalization validates the higher performance with min-max normalization. It is in contrast to the  $z$ -score method which has been widely used with deep learning-based approaches but does not achieve reasonable accuracy in this domain. Secondly, the impact of class imbalance has been analysed and compared with other contemporary approaches such as oversampling, undersampling, and ensemble learning. The cost-sensitive function emerged as the better option for modeling the IDS problems as compared to data level and the combination of multiple balance models. Lastly, the outcomes are compared against the current works to validate the competitive performance on both datasets. In conclusion, the proposed method with class-weighted neural networks is helpful for effectively classifying traffic. In the future, this work can be extended by introducing the dimensionality reduction with the feature subset approach or through new feature representation with autoencoders. The recognition of attacks is another viable option to explore the impact of the proposed work.

## Declarations

**Conflict of interests** The authors declare no conflict of interest.

## References

1. Aldwairi T, Perera D, Novotny MA (2018) An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection. *Comput Networks* 144:111–119
2. Aleesa AM, Younis M, Mohammed AA, Sahar N (2021) Deep-Intrusion Detection System with Enhanced Unsw-Nb15 Dataset Based on Deep Learning Techniques. *J Eng Sci Technol* 16(1):711–727
3. Almi'ani M, Ghazleh AA, Al-Rahayfeh A, Razaque A (2018) Intelligent intrusion detection system using clustered self organized map.” in 2018 Fifth international conference on software defined systems (SDS), pp. 138–144.
4. Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K (2018) Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access* 6:52843–52856
5. Alsaadi HS, Hedjam R, Touzene A, Abdessalem A (2020) Fast Binary Network Intrusion Detection based on Matched Filter Optimization, in 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIOT), pp. 195–199
6. Ashfaq RAR, Wang X-Z, Huang JZ, Abbas H, He Y-L (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. *Inf Sci (Ny)* 378:484–497

7. Ashiku L, Dagli C (2021) Network Intrusion Detection System using Deep Learning. *Procedia Comput Sci* 185:239–247
8. Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106:249–259
9. Çavusoglu Ü (2019) A new hybrid approach for intrusion detection using machine learning methods. *Appl Intell* 49(7):2735–2761
10. Clevert D-A, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus).” *arXiv Prepr. arXiv1511.07289*
11. Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning.” in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167
12. Ding S, Wang G (2017) Research on intrusion detection technology based on deep learning.” in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1474–1478
13. Dong R-H, Li X-Y, Zhang Q-Y, Yuan H (2019) Network intrusion detection model based on multivariate correlation analysis--long short-time memory network. *IET Inf Secur* 14(2):166–174
14. Gao J, Chai S, Zhang C, Zhang B, Cui L (2019) A Novel Intrusion Detection System based on Extreme Machine Learning and Multi-Voting Technology. *Chinese Control Conference (CCC) 2019:8909–8914*
15. García-Teodoro P, Díaz-Verdejo J, Maciá-Fernández G, Vázquez E (2009) Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput Secur* 28(1–2):18–28
16. Garg S et al (2020) En-ABC: An ensemble artificial bee colony based anomaly detection scheme for cloud environment. *J Parallel Distrib Comput* 135:219–233
17. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256
18. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks.” in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649
19. Hagan MT, Demuth HB, Beale M (1997) *Neural network design*. PWS Publishing Co
20. Hardt M, Recht B, Singer Y (2016) “Train faster, generalize better: Stability of stochastic gradient descent.” in *International Conference on Machine Learning*, pp. 1225–1234
21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778
22. Hinton G et al (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process Mag* 29(6):82–97
23. Ibrahim LM, Basheer DT, Mahmood MS (2013) A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network. *J Eng Sci Technol* 8(1):107–119
24. Ieracitano C, Adeel A, Morabito FC, Hussain A (2020) A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing* 387:51–62
25. Ingre B, Yadav A (2015) Performance analysis of NSL-KDD dataset using ANN, in *2015 international conference on signal processing and communication engineering systems*, pp. 92–96
26. Injadat M, Moubayed A, Nassif AB, Shami A (2020) Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Trans Netw Serv Manag*
27. Jiang K, Wang W, Wang A, Wu H (2020) Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access* 8:32464–32476
28. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences.” *arXiv Prepr. arXiv1404.2188*
29. Kasongo SM, Sun Y (2020) A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Comput Secur* 92:101752
30. Kasongo SM, Sun Y (2020) Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *J Big Data* 7(1):1–20
31. Kaur T, Malhotra V, Singh D (2014) Comparison of network security tools--firewall, intrusion detection system and HoneyPot. *Int J Enhanc Res Sci Technol Eng* 200204
32. Khan MA (2021) HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. *Processes* 9(5):834
33. Khan FA, Gumaei A, Derhab A, Hussain A (2019) A novel two-stage deep learning model for efficient network intrusion detection. *IEEE Access* 7:30373–30385
34. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A (2020) Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. *Electronics* 9(1):173

35. Kim J, Shin N, Jo SY, Kim SH (2017) Method of intrusion detection using deep neural network. *IEEE Int Conference Big Data Smart Comput (BigComp)* 2017:313–316
36. King G, Zeng L (2001) Logistic regression in rare events data. *Polit Anal* 9(2):137–163
37. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization, arXiv Prepr. arXiv1412.6980
38. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
39. Krömer P, Platoš J, Snášel V, Abraham A (2011) Fuzzy classification by evolutionary algorithms,” in 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 313–318
40. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
41. Lee H et al (2019) An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets. *Nat Biomed Eng* 3(3):173–182
42. Li Z, Qin Z, Huang K, Yang X, Ye S (2017) Intrusion detection using convolutional neural networks for representation learning, in International conference on neural information processing, pp. 858–866.
43. Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y (2013) Intrusion detection system: A comprehensive review. *J Netw Comput Appl* 36(1):16–24
44. Liu Z et al (2011) A method of SVM with normalization in intrusion detection. *Procedia Environ Sci* 11:256–262
45. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. *Proc icml* 30(1):3
46. Man J, Sun G (2021) A Residual Learning-Based Network Intrusion Detection System,” *Secur. Commun. Networks*, vol. 2021
47. Manda B, Bhaskare P, Muthuganapathy R (2021) A Convolutional Neural Network Approach to the Classification of Engineering Models. *IEEE Access* 9:22711–22723
48. Mebawondu JO, Alowolodu OD, Mebawondu JO, Adetunmbi AO (2020) Network intrusion detection system using supervised learning paradigm. *Sci. African* 9:e00497
49. Mighan SN, Kahani M (2021) A novel scalable intrusion detection system based on deep learning. *Int J Inf Secur* 20(3):387–403
50. Mohammadi M, Raahemi B, Akbari A, Nassersharif B (2012) New class-dependent feature transformation for intrusion detection systems. *Secur Commun networks* 5(12):1296–1311
51. Moukhafi M, El Yassini K, Bri S, Oufaska K (2018) Artificial neural network optimized by genetic algorithm for intrusion detection system,” in International Conference on Advanced Intelligent Systems for Sustainable Development, pp 393–404
52. Mulyanto M, Faisal M, Prakosa SW, Leu J-S (2021) Effectiveness of Focal Loss for Minority Classification in Network Intrusion Detection Systems. *Symmetry (Basel)* 13(1):4
53. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines
54. Nguyen S-N, Nguyen V-Q, Choi J, Kim K (2018) Design and implementation of intrusion detection system using convolutional neural network for DoS detection,” in Proceedings of the 2nd international conference on machine learning and soft computing, pp. 34–38
55. Nskh P, Varma MN, Naik RR (2016) Principle component analysis based intrusion detection system using support vector machine, in 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1344–1350
56. Patel A, Taghavi M, Bakhtiyari K, Júnior JC (2013) An intrusion detection and prevention system in cloud computing: A systematic review. *J Netw Comput Appl* 36(1):25–41
57. Paulauskas N, Auskalnis J (2017) Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset, in 2017 open conference of electrical, electronic and information sciences (eStream), pp. 1–5
58. Peng P, Zhang W, Zhang Y, Xu Y, Wang H, Zhang H (2020) Cost sensitive active learning using bidirectional gated recurrent neural networks for imbalanced fault diagnosis. *Neurocomputing* 407:232–245
59. Pervez MS, Farid DM (2014) Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs,” in The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), pp. 1–6.
60. Priyadarsini PI (2021) ABC-BSRF: Artificial Bee Colony and Borderline-SMOTE RF Algorithm for Intrusion Detection System on Data Imbalanced Problem,” in Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2020, pp. 15–29
61. Rani M, Gagandeep (2019) A Review of Intrusion Detection System in Cloud Computing
62. Rani M, et al. (2021) Employing Artificial Bee Colony Algorithm for Feature Selection in Intrusion Detection System,” in 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), pp 496–500

63. Rawat S, Srinivasan A, Ravi V, Ghosh U (2020) Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network. *Internet Technol. Lett.*, p. e232
64. Sethi K, Kumar R, Prajapati N, Bera P (2020) Deep Reinforcement Learning based Intrusion Detection System for Cloud Infrastructure,” in 2020 International Conference on COMMunication Systems & NETWORKS (COMSNETS), pp. 1–6
65. Singh D, Singh B (2019) Investigating the impact of data normalization on classification performance. *Appl Soft Comput.*, p. 105524
66. 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
67. Tan X et al (2019) Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm. *Sensors* 19(1):203
68. Tao W, Honghui F, HongJin Z, CongZhe Y, HongYan Z, XianZhen H (2021) Intrusion Detection System Combined Enhanced Random Forest With Smote Algorithm
69. Tavallaei 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
70. Tchakoucht TA, Ezziyyani M (2018) Multilayered Echo-State Machine: A Novel architecture for efficient intrusion detection. *IEEE Access* 6:72458–72468
71. Thabtah F, Hammoud S, Kamalov F, Gonsalves A (2020) Data imbalance in classification: Experimental evaluation. *Inf Sci (Ny)* 513:429–441
72. Umar MA, Zhanfang C (2020) Effects of Feature Selection and Normalization on Network Intrusion Detection
73. Umar MA, Zhanfang C, Liu Y (2020) Network Intrusion Detection Using Wrapper-based Decision Tree for Feature Selection, in Proceedings of the 2020 International Conference on Internet Computing for Science and Engineering, pp. 5–13
74. Verma AK, Kaushik P, Shrivastava G (2019) A Network Intrusion Detection Approach Using Variant of Convolution Neural Network, in 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 409–416
75. Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S (2019) Deep learning approach for intelligent intrusion detection system. *IEEE Access* 7:41525–41550
76. Wang Y, Shen Y, Zhang G (2016) Research on intrusion detection model using ensemble learning methods,” in 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), , pp. 422–425
77. Wu K, Chen Z, Li W (2018) A novel intrusion detection model for a massive network using convolutional neural networks. *Ieee Access* 6:50850–50859
78. Wu Y, Lee WW, Xu Z, Ni M (2020) Large-scale and robust intrusion detection model combining improved deep belief network with feature-weighted SVM. *IEEE Access* 8:98600–98611
79. Wu Y, Lee WW, Gong X, Wang H (2020) A Hybrid Intrusion Detection Model Combining SAE with Kernel Approximation in Internet of Things. *Sensors* 20(19):5710
80. Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional network,” *arXiv Prepr. arXiv1505.00853*
81. Xu J, Li Z, Du B, Zhang M, Liu J (2020) Reluplex made more practical: Leaky ReLU,” in 2020 IEEE Symposium on Computers and Communications (ISCC), pp. 1–7
82. Yeung D-Y, Ding Y (2003) Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognit* 36(1):229–243
83. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access* 5:21954–21961
84. Yousefi-Azar M, Varadharajan V, Hamey L, Tupakula U (2017) Autoencoder-based feature learning for cyber security applications. *Int Joint Conference Neural Networks (IJCNN)* 2017:3854–3861
85. Zhang C, Ruan F, Yin L, Chen X, Zhai L, Liu F (2019) A deep learning approach for network intrusion detection based on NSL-KDD dataset,” in 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 41–45
86. Zwane S, Tarwireyi P, Adigun M (2018) Performance Analysis of Machine Learning Classifiers for Intrusion Detection. *Int Conference Intell Innov Comput Applic (ICONIC)* 2018:1–5