INTRUSION DETECTION USING DEEP (CNN) CONVOLUTIONAL NEURAL NETWORK FEATURE EXTRACTION WITH (EPCA) ENHANCED PRINCIPAL COMPONENT ANALYSIS FOR DIMENSIONALITY REDUCTION

Research Paper

Abhilash Kayyidavazhiyil, SSBM, Geneva, Switzerland, abhilash@ssbm.ch

Dr. Mario Silic, SSBM, Geneva, Switzerland, mario@ssbm.ch

Abstract

(IDS)Intrusion detection system, is an extremely significant to prevent network attacks, and in classification of network traffic to determine anomalies inside network. However, no any existing studies, explored an efficient IDS, to address the problem of low accuracy engendered by redundant, irrelevant, non-linear features and dealing with large dataset. Hence to overcome this issue, different level of features traversed in several hidden layers undergoes deep learning and extracted by two 1-dimensional models of Deep CNN. Then with the integration of (Fourier)F-transform and (PCA)Principal component-analysis transforms the non-linear features to linear feature sets, and reduces high dimensionality to low-dimensional features. Hence it aids in minimising the PCA computation time and increases model robustness. This sort of enhanced PCA with F-transform facilitates to increase accuracy of classification. The transformed features are classified efficiently through algorithms (RF)Random Forest, (GNB)Gaussian Naive Bayes, XGBoost and KNN classifier. The comparative assessment of proposed IDS model, outperformed in classifying normal and abnormal data with higher accuracy.

Keywords: *CNN-Convolutional neural network, NB-Naïve Bayes, RF-Random forest, KNN-K-Nearest Neighbour, XgBoost-Extreme Gradient boosting, PCA-Principal Component analysis, IDS-Intrusion detection system.*

I. Introduction

(IDS)Intrusion detection system arise in glory in recent decade due to its sturdiness. This IDS were designed to determine intruders in specific area [1]. This IDS consists of three major components, first component wherein the agent oversees collecting data of monitoring-event's data flow. Then second, the analysis engine determined the intrusion evidences and propagates the alarms. Then the third component, comprises of return module, performing with results of analysis engine. Different IDS had enhanced efficiency and reliability over time, however more diversified tactics of attacks were evolved in circumventing detection system. Further to this various typical IDS were not capable to deal with numerous network layers in network like IoT[2].There are different researchers urged to utilised distribute IDS in working with (ML)Machine learning processes like ANN(Artificial neural-network), (DL)Deep learning and (RL)reinforcement learning. Due to latest advancement in intelligent-system, ordinary ANN, constrained with its capability to handle with IDS intricacy(Aleesa, Younis, Mohammed, & Sahar, 2021). Hence the attempt to develop a effective neural network based IDS is necessary to address the flaws [3]. The dimension of network information had dramatically grown since from dawn of Big Data era. The high dimensional data volume will be the challenging task to tackle in different domains, like in ML, data analysis and text mining. The redundant and irrelevant characteristics adds up to the complexity of the high dimensions and these may obstruct the outcomes of proper feature classification(Saputra, Widiyaningtyas, & Wibawa, 2018). These may yield out poor results of algorithm. In this similar context, IDS were based upon significant data and it performs network-transmission controls to process and determine illegal usage resources [5]. In this scenario to

enhance the IDS qualifications, the detection accuracy in classifying and removing the intrusions had become the pressing issue which ought to get addressed.

(FS)Feature Selection stands as popular method to minimize the dimensions of data. The feature selection and various algorithm contribute to decrease the complexity of data through eliminating the extraneous elements, significant to IDS(Hussain, Neggaz, Zhu, & Houssein, 2021). FS methods reduces the network data dimensions through filtering out redundant and irrelevant features. Additionally the payload of IDS computing were also decreased and speed of detection will get increases. Such enhanced FS method in effective IDS increases the rate of detection accuracy and IDS performance results with various classifiers(Sindhu, Ngadiran, Yacob, Zahri, & Hariharan, 2017). Recently different FS techniques depending on (MH)metaheuristics approaches were adopted to address IDS model. The algorithms of MH employed to IDS are such as chaotic teaching-learning algorithm, improved Cat-swarm optimisation algorithm, (CSA)crow search optimisation algorithm(Ouadfel & Abd Elaziz, 2020), genetic algorithm(Maleki, Zeinali, & Niaki, 2021),(PSO)particle swarm optimisation algorithm etc., Those algorithms maximizes IDS performance in determination, however the algorithm possess drawbacks such that they could not get rid of local non-linear features solutions and resulting to higher (FPR)False positive-rate in IDS. Hence to addresses all the above issues like non-linearization, low performance in classification, high variance in the feature subset similarity distance, to handle the high dimensional data features, and to improve the efficiency of IDS performance in classifying the attacks, the study is proposed utilising enhanced PCA with Fourier transform method with two deep CNN 1d feature extraction models employed using two datasets NSL-KDD and UNSW-NB-15.

Based on these statements, the major contributions of the study were delineated as

- To propound an efficient intrusion detection model using enhanced (PCA) Principal Component analysis for feature fusion of the extracted features aided through Deep CNN 1D models.
- To select out the different level of optimal features using Deep CNN one dimensional models, that deep trains the different range of parameters in hidden layers, and maxpooling layers., to maximize the accuracy in classifying the intrusions.
- To merge the features of using Enhanced PCA algorithm, that transforms the F-Fourier transform features to kernel low dimensional features and yields out the linear set of features (avoiding linearization).
- To engage RF, Gaussian NB, KNN and XGB Classifiers, to bring out the best feature solutions based on voting of decision tree, Probability condition, Euclidean distance of K-count of neighbours and high feature scalability.

1.1. Paper organisation

Section I elucidates about the introductory concepts of the research and the purpose of the research. Section II propounds the review of literature related to research. Section III, deals with the research methodology of the study and the dataset description. Section IV enumerated the results gained from research implementation with discussion of the findings. Section V, explicates the conclusive statements of the study and future works were also summarized in the section.

1.2. Problem Identification

- Even though IDS provides many advantages, certain limitations are observed in IDS organizations. Since it is very expensive, it needs a lot of work and time requirement, and the unreliability of the IDS leads to poor utility (Werlinger, Hawkey, Muldner, Jaferian, & Beznosov, 2008).
- Also, face challenges like false alarm rate, unstable datasets, response duration, and decreased detection rate. The major challenge of this approach has an up-to-date view as new protocols evolve (Al-Janabi, Ismail, & Ali, 2021).
- Machine learning also has disadvantages in algorithm selection, data acquisition, time and space, shows high error-prone and one of the major problems machine learning professionals face is the lack of good quality data (DATAFLAIR, 2019).

- Some IDS based on deep learning techniques face limitations like low transparency and interpretability, huge data requirement, and artificial intelligence while transferring the data (Camilleri & Prescott, 2017).
- It observes the drawbacks in feature selection at graph-based, semi-supervised feature selection. Such method is not applicable for large-scale networks due to the presence of a hung number of training networks and also needs high time for a built graph-like matrix (Venkatesh & Anuradha, 2019).
- Even though IDS is a powerful system, it only produces the outcomes of abnormal behaviours attacks with false negative and false positive values that implies inaccurate detection range. The other limitation is that different attacks arise consecutively with various behaviours detected by IDS through high positive false rates that spoil the efficiency and lifetime of the system.

II. Review of literature

The below section enumerated the review analysis of different researchers that deals with intrusion detection methods using various approaches and their implications.

Cloud-based IDS models aids in detection and prevention of attack from unknown and sophisticated attacks related with complex cloud architecture having low error-rates(Farhat, Abdelkader, Meddeb-Makhlouf, & Zarai, 2020). This is due to the fact wherein the single IDS becomes tedious to identify all the prevailing attacks in network and to performing blocking of attacks, because of the IDS model limited attack patterns and their implication. The co-operation between IDS which belong to various cloud providers were attained through permitting the model in exchanging the analysis feedback of intrusion and in exploitation of every other expertise in covering the unknown attack or threat patterns, hence to achieve mutual advantages [5].To be reasonable supplement of firewall, the technology of IDS could support system in dealing with offensive IDS suffering from higher FPR resulting to bad rate of accuracy. Hence this work recommended to employ IDS through Recursive Feature-Elimination for feature selection and utilised DNN and (RNN)Recurrent neural-network for feature classification. The model suggested in the research provides better results to yield higher accuracy rate of 94%. The DNN is utilised in binary-classification for categorizing normal features and attack features. RNN were utilised to categorize five different classes such as DoS, U2L, Prove and Normal class. The system were implemented through NSL-KDD dataset, that seems effective for offline IDS analyses-system(Mohammed & Gbashi, 2021).Intrusion detection in IoT networks also become a major issue in today's internet technology. This work proposes a suppressed fuzzy clustering (SFC) algorithm along with principal component analysis (PCA) algorithm. This algorithm classifies data into high and low risks respectively. The principal factor is analysed by simulation experiment. The work outcome elucidates that, this algorithm was more easy to adapt comparing to traditional methods available(L. Liu, Xu, Zhang, & Wu, 2018).Similarly another research introduces a hypervisor-based cloud IDS using online multivariate statistical change analysis to find the invisible networks behaviours. This method was analysed by datasets collecting and employing a new data set with wide range of attack vectors accordingly(Aldribi, Traore, Moa, & Nwamuo, 2020). The primary technique to protect the databases obtained from internal-attacks is in limiting the database access depending on user-role. The (RBAC) role-based access-control offers the valuable abstraction level in promoting security administration, for business[12]. The hybrid system, referred as (CN-LS) Convolutional neural-based learning-classifier system, stands as hybrid system is proposed to be database IDS. This model is an integration of (GA) Genetic algorithm and CNN(Bu & Cho, 2020). This CN-LCS categorises the queries through depending on role through CNN and select more effective features automatically across time through GA model[4]. Through modelling the access patterns of normal data depending on large data volume, different robust statistical-model which were not sensitive to changes of user could be generated. The core-concept to detect internal attacks were in the classification of queries which does not match out respective user-role through statistical technique.

DL seems to be high efficient to discriminate DDoS traffic, from benign network-traffic, starting from granular, low-level packet features. Such DL-based DDoS detection design architecture , eligible for online resource-constrained network environment(Doriguzzi-Corin, Millar, Scott-Hayward, Martinez-del-Rincon, & Siracusa, 2020). The architecture, leverages CNN, to study the DDoS behaviour and to

benign the traffic-flows, with lower attack-detection computational time and less processing power. The architecture model was referred as (LUCID)Lightweight Usable CNN in DDoS-Detection. The Dataset present in the proposed design, is agnostic pre-processing mechanism, that generates traffic-observation . The detection algorithm ought to cope up with traffic flow segments, gathered over predefined time-windows. The consistency of detection outcomes over various dataset ranges , demonstrated the results stability. With the progressing growth of computing power in big-data, deep-learning techniques, getting blossomed quickly(Kasongo & Sun, 2019). The techniques were broadly used in different fields. In proceeding this approach, one of deep-learning approach is developed utilising (RDS)Recurrent Neural-networks IDS is developed, to train the system for intrusion detection, consisting of NSL-KDD dataset. The experimental outcomes reveals that proposed design RNN-IDS evolved as superior to traditional classification-techniques, in multi-class classification and binary methods(Yin, Zhu, Fei, & He, 2017). Since Deep-learning does had capability for extracting data representations (features), it creates better intrusion detection-models through inspiration of RNN. The performance of Random-forest, SVM approaches, Naïve Bayesian algorithm in multi-classification were studied. The dataset considered in the research are NSL-KDD dataset. However, the research ought to pay high attention to minimise the training time taken through GPU acceleration, learn the LSTM classification performance, performance analysis of Bi-Directional RNN-algorithm and prevent the vanishing and exploding gradients in intrusion detection mechanisms(Doriguzzi-Corin et al., 2020).

Stacked Auto-encoder utilised for latent feature-extraction, proceeded by various classification based IDS including (RF)Random forest, Naïve Bayes, (SVM)Support vector-machine and decision-tree utilised in efficient and rapid intrusion detection in massive network-traffic data(Mighan & Kahani, 2021). The real-time dataset like UNB ISCX-2012 applied in proposed method validation and performance assessment is performed in accordance to precision, f-measure, time, sensitivity and accuracy metrics(Hijazi, El Safadi, & Flaus, 2018). The IDS model utilising DL secures out ICS network. The technique in the study utilised (MLP)Multi-layer perceptron with binary-classification and does training of higher dimensional Modbus data packets after simulation of network. Then the data will be labelled as malicious and normal data to neural-network, for understanding the underlining anomalous and normal network behaviour. The performance could be improvised through adding potential to determine (DoS)Denial of service attacks and to add time-stamps to fields, to learn time interval of data packets that generally arrives in. one such study, inhibits new hybrid-approach that integrates the (EFS)Ensemble of feature-selection algorithm and (TLBO)Teaching learning-based optimisation(Singh & Shrivastava, 2021) . This model EFS-TLBO method utilises (ELM)Extreme learning-machine to select out the most efficient features and enhances the accuracy of classification. The performance evaluation of suggested approach is assessed in benchmark-dataset. The outcomes of experiment revealed that proposed method explicated high accuracy in prediction, FPR values and necessitates low relevant features(Wani & Khaliq, 2021). There is no any definite protocols or definite standards for towards IoT communication and those IoT devices possess limited resources. To enable the entire security measure for those IoT devices, seems as challenging activity and it is required. Many categories of lightweight security-protocols were not capable to yield out optimum protection mechanism, against those existing effective threats in the cyber-world. Software-defined network proposes the centralised computer network control. This SDN consists of programmable method to networking which decouples data planes and control. This SDN-based IDS were applied that utilised DL classifier to detect anomalies within IoT(C. Liu, Gu, & Wang, 2021).

As a category of security equipment, in protecting digital-assets, the IDS is quite low efficient if alert is not propagated timely and IDS would be not beneficial if accuracy rate could meet the demands. Hence a IDS that integrates ML with DL is implemented in a research. The model utilises RF algorithm and K-means algorithm for binary-classification. The computing distribution of those algorithms were employed on Spark-platform to classify the attack and normal events rapidly(Kunang, Nurmaini, Stiawan, & Suprapto, 2021). Then through CNN, DL algorithms and (LSTM)long short-term memory algorithms the judged events were classified further as abnormal and different attack-types. In this stage, (ADASYN)Adaptive synthetic-sampling method is been adopted to rectify the issue of unbalanced dataset. The CISIDS2017 and NSL-KDD-dataset utilised to assess the designed model performance. The Deep IDS through (PTDAE)pre-training method with deep auto-encoder integrated with DNN is

put forward. The IDS model is designed with optimisation procedures of hyperparameter, to enhance the performance outcomes. The research offers alternative solution to DL models by automatic hyper-parameter optimisation process, combining grid-search and random-search methods. The optimisation of hyperparameters aids to detect hyperparameter values and best categorical configuration of hyperparameters, to enhance performance detection. The model tested upon CSE-CIC-IDS2018 and NSL-KDD datasets. In model specifically in pre-training phase, the results are presented through applying three feature-extraction technique such (SAE)-Stacked auto-encoder, (AE)-auto-encoder and (DAE)_Deep auto-encoder. The best outputs yields for DAE approach. The results of performance outperformed other conventional IDS methods with respect to metrics in classifying multi-classes.

2.1. Research Gaps

Shallow learning-methods necessitates large training data quantity for operation, that turned out to be challenging for heterogeneous environment. Beside this, shallow learning stands as labour intensive and expensive and not suited to forecast the higher data dimensional learning requirements and the non-linear features, having massive data. When to deal with large count of multi-type data variables Decision tree model and (LR) Logistic regression were prone to face overfitting issues and it ignores the drawback caused by correlation of inter-data features. Similarly if SVM is applied in IDS for classifying the data features, if handling out large data samples, the SVM model becomes inefficient and it induces the more computational time in training phase. The model becomes challenging to determine kernel function, to tackle and manage missing data feature(Kocher & Kumar, 2021).

III. Research method

The dataset set UNSW-NB15 and NSL-KDD are loaded into pre-processing step. Pre-processing step, rescales the varying feature range and checks out the missing values in the data. Reducing features is meant to minimize the redundant data and make the decisions easier for extraction.

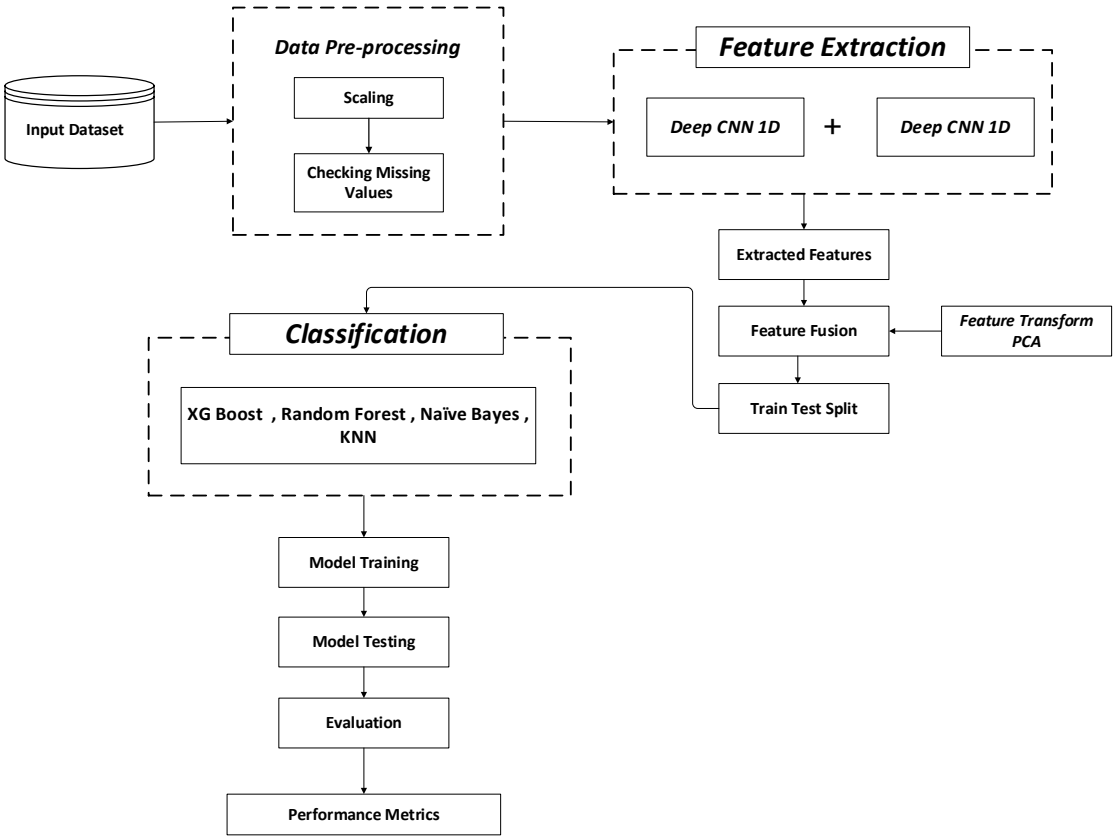


Figure 1. Flow of Proposed Model

Furthermore, the quick training time attains by reducing the enigmatic and irrelevant data. The optimal set of features is selected to develop a predictive model and it is extracted through two models of Deep

CNN one dimensional architecture. Since the pre-processed features comprises of large dimensions, those are reduced by maxpooling layer learning in Deep CNN model. As a result different level of dataset features will be extracted from each Deep CNN model. . The features extracted were learned in many hidden layers units in CNN architecture. The data precision of extracted results are enhanced. The features retrieved from two Deep CNN model are merged together as a single linear set of features by using enhanced (PCA) Principal component analysis. The linear set of features, were again reduced in low dimensional feature space, through Fourier transform function. These Fourier function with PCA fuses the features from binary files and those features were minimized through random projection algorithm, in creating low dimensional features set. This process utilised in classify the presence of intrusion or not.

Followed by that, the process of feeding the transformed features into the train test split classifier for training and testing were performed. Once linear features are chosen, classification occurs. The majority of class from each decision tree in random forest classifier are classified. The high probability of the independent feature are computed using Naïve bayes algorithm and the features classified based on this. The weights of variables incorrectly predicted by any decision tree increases and were fed to next decision tree. The XGBoost classifier ensemble to provide more precise and stronger model, turns out weak classifier and boost to become strong classified outcomes. KNN classifier retrieves the KNN neighbours that had the shorter Euclidean distance towards high probability features. The classification performed through the classifiers XGBoost, NB, RF and KNN are utilised to assess the capability of model with new set of classified linear features and it improves prediction accuracy. The performance of model is analysed through comparative analysis and metrics.

3.1. Dataset Description

UNSW-NB15-Data-set

This dataset UNSW-NB15 belongs to network traffic-basis data-set type, and it generated by 4 tools such as Tcp-dump tool, Bro-IDS-tool, Argus-tool, and IXIA-perfect-storm tool. The Raw-network-packets of this UNSW-NB 15 dataset created by IXIA Perfect Storm tool in Cyber range Lab to generate the hybrid of synthetic contemporary attack comportments and real modern normal activities. The tcpdump tool used in capturing raw traffic. The dataset consists of nine categories of attacks including Analysis DoS-attacks, Fuzzers, worms, Generic, Shellcode, and Exploits, Reconnaissance and Fuzzers. The total records count of this dataset are two million and 540044, stored in four CSV-files. The total training set records are 175341 data records and total testing set records are 82332 records obtained from different types normal and attack type.

Link: https://research.unsw.edu.au/projects/unsw-nb15-dataset

NSL-KDD

NSL-KDD is a new version dataset, of the KDD'99 data set. This is an effective benchmark data set to help researchers compare different intrusion detection methods. This dataset consists of nearly forty three features in each record, that refers to traffic input and then last two represents the labels (if it can be normal feature or the attack) and then score(traffic input severity). In this dataset there consists of four various attack classes such as Probe. (R2L) Remote to-local, (U2R) User to-Root, (DoS)Denial of-service. The total records count are 125972 entries, with consisting of training set 100777 records and test set comprising of 25194 data records.

Link: https://www.unb.ca/cic/datasets/nsl.html

3.2. Data Pre-processing

In the proposed method, the dataset loads impure data, which has to be refined during the stage of data pre-processing. At that stage, two functions are performed for purifying data, scaling and checking for missing values. During scaling, data are arranged in a fixed specific range. The original data in the data set contains data in all ranges. Scaling transforms the original data into the fitted data on a specific scale.

The next process of checking the missing values is performed. If the values are missed, they must be attributed to the most frequent value.

3.3.Deep (CNN)Convolutional Neural network for Deep Learning of different level of features of dataset in low dimensional feature space.

CNN model stands as network model, that is proposed by Lecun in year 1998. This model is a category of feed-forward neural-network, that yields out better performance in (NLP)Natural language processing and image processing. The local perception and CNN weight sharing , could greatly minimized the parameters count, to project different range of features with deep learning of features layer by layer thereby enhancing the learning model efficiency. This CNN model majorly comprises of three parts, such as convolution-layer, pooling-layer and then fully-connection layer. Every convolutional-layer consist of different convolutional kernel, and their computation. After each convolutional layer undergoes convolutional operation, the data features were extracted. However the dimensions of the extracted featured seems to be so high, hence to address this complexity, and to minimise the training cost of network, the maxpooling layer affixed after this convolutional layer. This layer hence limits out the features dimensions.

Algorithm-1

Deep Convolutional neural network 1
$\mathrm{Conv}_t = \ \mathrm{relu}\ (a_t * k_t\ + b_t\ )$ $\mathrm{Conv}_t \longrightarrow$ output of convolution $\mathrm{relu} - \rightarrow$ activation function $a_t \longrightarrow$ input vector $w_t - - \longrightarrow$ weight of the convolution kernel $b_t - - \longrightarrow$ bias of the convolution kernel

The features from pre-processing phase, enters convolutional layer. The layer extracts the parameters, filters or kernels that need to get learned from this convolutional-layer. In each layer (dense layer) the input values towards processing element, a_n were multiplied by connection-kernel weight w_t. This weight connection will simulates the neural pathways strength and it is summed up with convolutional kernel bias. Hence in feature training process, the features, trained by considering the relevant weight values and features. Each filter of convolutional-layer generates the activation map. Then the learned features traverse on to pooling layer, summarizes all the features determined feature regions generated by convolutional-layer. It may also reduces feature representation dimensions as well. The dimensions of the parameters from convolutional layer are reduced and extracted in this pooling-layer. The signal features as array representation points passed on to several pooling layers, in Deep CNN, to obtain deeper extraction of features. ReLU layer employs the function f(x) to every input data x. The input value along with computed activation-function and passes the output value to next layer's input. This function is applied to all feature input values. This activation function gets computed with feature input values, without impacting receptive convolutional layer fields and non-linear model properties are increased. Then extracted learned features are fed to dense layer, which retrieved input from entire previous layer's output. The output of all the previous layers are fed as the input to fully-connected layer. Each layer is interconnected with another layer neuron. Hence, the fully extracted features from all the layers are summarized as updated features in fully-connected layer. Simultaneously, the process is carried out in another Deep CNN architecture, and the features from two Deep CNN feature extraction model were fused in feature-fusion phase.

$$\mathrm{Conv}_t = \tanh(\ a_t * w_t + \ b_t) \qquad \text{-- (1)}$$

Wherein this Conv_t represents the output-value after this convolution, the activation function is defined by tanh. The variable a_t denotes the input-vector, and the weight of the convolutional-kernel is represented by w_t. In the equation the bias of this convolutional-kernel represented by b_t.

The feature extraction different level of features of both datasets are performed by using the Deep CNN (Deep Convolutional Neural Networks) model which are inspired by visual system structure, in specific

tops at classification and object recognition. Moreover, CNN can reserve the spatial locality and neighbourhood relation input. Deep neural network architecture shows greater non-linear nature. Subsequently, deep CNN is highly suited to manage the non-linear spectral spatial evaluation and high dimensional hyperspectral image difficulties

Layer (type)	**Output Shape**	**Param #**
conv1d (Conv1D)	(None, 85, 32)	128
conv1d_1 (Conv1D)	(None, 85, 16)	528
flatten (Flatten)	(None, 1360)	0
dropout (Dropout)	(None, 1360)	0
dense (Dense)	(None, 128)	174208
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 100)	6500

Table 1. Model: "sequential_1"

Layer (type)	**Output Shape**	**Param #**
conv1d_2 (Conv1D)	(None, 85, 32)	128
conv1d_3 (Conv1D)	(None, 85, 32)	1056
flatten_1 (Flatten)	(None, 2720)	0
dropout_2 (Dropout)	(None, 2720)	0
dense_3 (Dense)	(None, 256)	696576
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 64)	16448
dense_5 (Dense)	(None, 250)	16250

Table 2. Model: "sequential_2"

The two tables 1 and 2 above describes the learning process in the extraction of features through Deep CNN one-dimensional models. Each Deep CNN model traverses to the different layers of CNN such as conv-1d, conv-1d-1, flatten layer, drop out layer, denselayer-1 and denselayer-2 in model-1. The pre-processed features are learned through these layers, deeply in all the hidden layers and brings out the total of 100 extracted features in Deep CNN-1d model-1. Similarly, the features are simultaneously trained through conv-1d_2, conv-1d_3, flatten_1 layer, drop out_2 layer, drop out_3 layer, denselayer-3, denselayer-4 and denselayer-5 in model-2. Hence fine level of extracted features from two deep learned deep CNN-1d models are gained in this phase, with low dimensions as well.

3.4.Architecture Design of Deep CNN

Deep CNN pertains to the group of Feed Forward ANN. Generally, CNN encompasses of two main layers namely pooling and convolution which provides feature maps by computing the dot product that relates to input filter and local-area. This is followed by non-linear function which determines the complexity for squashing the outcomes of NN. Further, pooling layer processes the signals to the feature maps by computing average or maximum value. Whereas, the Fully-Connected layers will follow stacked-convolutional and pooling-layers. At this fully connected layer is Softmax-layer which performs score computation for individual classes of features. Though Deep CNN is similar to CNN, addition of dense and hidden layers increase the performance of CNN and also assist in reducing the training time due to its innate capability to accomplish in-depth learning. It is also capable of grouping the unlabeled data in accordance with the resemblances amongst the inputs and extracting data while having a labelled dataset for training. Due to these valuable merits, the present study proposes the two models of Deep CNN algorithms for feature extraction in IDS

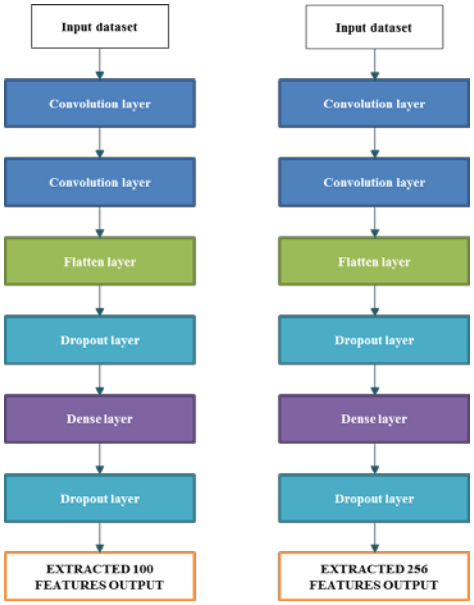


Figure 2. Deep CNN Architecture Design

The deep CNN architecture is established shown in figure 2 above to retrieve the hyper spectral-spatial features because of the imbalance among the massive parameters and limited labelled samples.

This Deep CNN architecture implementation is performed for feature extraction of pre-processed features of NSL KDD and UNSW-NB15 datasets. The architecture model of Deep CNN is represented in the below figure .The data features are represented and extracted in the form of array representation. The input X is given to the model, consisting of input features. Each feature point is extracted as pixel data-points (matrix-representations). Each three-dimensional matrix feature representations were chosen specifically through subsequent neural-network layers in Deep CNN. The data points, significant features are extracted from entire attributes in each layer. In regular CNN architecture, there may be 5 to 10 feature learning-layers, wherein in Deep CNN architecture consists of more than 50 layers to 100 layers deep, utilised for category of attack classification model. In this proposed system, Deep CNN-1d first model extracts 100 features and second model retrieves 256 features and drains out total of 356 features were fused together as single set to feature transformation phase.

3.5. Enhanced (PCA)Principal Component Analysis for Dimensionality reduction

Even though the different level of features are learned and extracted by Deep CNN-1D models, the dimensionality of some features are high, hence to overcome this issues, the PCA with Fourier transform enhances the ability of PCA, to bring out the linear set of features with low dimensions. This model is also referred as (MPCA)Multi-linear Principal component-analysis. The input feature information to MPCA were concentrated as like in PCA, it is orthonormal projection and this anticipated-feature is tensor of same-order as those feature samples with abridged-dimension. The fused features groups were signified as below.

$x = \{x_1, x_2, \ldots., x_L\}$ -- (2)

Wherein this $x_l \in R^{s_1 * \ldots * s_N}$ represents the L^{th} N-modes input-feature key-points having size $s_1 * \ldots * s_N$. The objective of MPCA is in describing the multi-linear transformation of the features, which maps out original feature scale space $R^{s_1 * \ldots * s_N}$ to scale space $R^{ev_1 * ev_2 * \ldots ev_n}$ ($ev_n <= s_N$). This sort of multi-linear features transformation projects out the non-linear high dimensional features to optimal low dimension linear features set. This Multi-linear feature transformation were demarcated to be

$U^n \in R^{s_n * ev_n}, n = 1, \ldots, N$ -- (3)

Then this variable $y_l \in R^{ev_1 * \ldots * ev_N}$ could be defined to be as below

$y_l = x_l * U^{(1)^T} * U^{(2)^T} * \ldots * NU^{(N)T} \in R^{ev_1 * ev_2 * \ldots ev_n, l=1,..,L}$ -- (4)

The objective of MPCA is that it regulates the N-projection matrices which exploits total tensor-scatter denoted by $\varphi(y)$.

Algorithm-2

Enhanced PCA algorithm
$x = \{x_1, x_2, \dots., x_L\}$ $x_l \in R^{s_1 * \dots * s_N}$ $s_1 * \dots s_N$ $R^{s_1 * \dots * s_N}$ $U^n \in R^{s_n * ev_n}, n = 1, \dots, N$ $y_l \in R^{ev_1 * \dots * ev_N}$ $y_l = x_l * U^{(1)^T} * U^{(2)^T} * \dots * NU^{(N)T} \in R^{ev_1 * ev_2 * \dots ev_n, l=1,..,L}$ $U^n, n = 1, \dots, N = \text{argmax}[\varphi(y)]U^{(1)} \dots U^{(N)}$ $\varphi(y) = \sum_{l=1}^{L} \lvert y_1 \rvert^2$ $\varphi^{(n)} = \sum_{l=1}^{L} x_l(n)..x_{l(n)}^T \quad n = 1, \dots N$ $\varphi^{(n)} = ev[:,0: x_{l(n)}^T]$ $Q^{(n)} = \frac{\sum_{i_n=1}^{ev_n} \gamma_{i_n}{}^{(n)}}{\sum_{i_n=1}^{s_n} \gamma_{i_n}{}^{(n)}} \geq 0.97$

$U^n \in R^{s_n * ev_n}, n = 1, \dots, N$ -- (5)

Wherein this variable $\varphi(y)$ denotes

$\varphi(y) = \sum_{l=1}^{L} \lvert y_1 \rvert^2$ -- (6)

In this time, initial-projection matrices were the eigenvectors (ev_n) that conforms to largest ev_n matrix eigenvalues.

$\varphi^{(n)} = \sum_{l=1}^{L} x_l(n)..x_{l(n)}^T \quad n = 1, \dots N$ -- (7)

For every n, dimensionality (ev_n) could be unwavering in accordance to ratio, illustrated below.

$Q^{(n)} = \frac{\sum_{i_n=1}^{ev_n} \gamma_{i_n}{}^{(n)}}{\sum_{i_n=1}^{s_n} \gamma_{i_n}{}^{(n)}} \geq 0.97$ -- (8)

In this equation, the $i_n{}^{th}$ Eigen values of this n-model total-scatter matrix represented by variable $\gamma_{i_n}{}^{(n)}$.The classification accuracy is improved with the optimal linear set of features transformed, and optimum accurate results are attained using this enhanced algorithm by feature fusion of Deep CNN 1-D models.

3.6. Random Forest Classifier

Random forest algorithm, classifies the results based on major voting approach, that clear the over fitting problem. It produces a stable result since the responses are taken from various trees on high dimensional data. The most relevant data in the algorithm are i (number of trees) and random vector (Si). Training data set is used for growing the tree and Si. The sample taken for the training dataset is denoted as TS. The binary tree is generated through a partitioning process (recursive) that divides into two answers, either yes or no. The training data is taken for the generation of decision trees that operates iteratively and reaches the voting process of selecting subsets and validated.

Algorithm-3

Random forest classifier
Input: TS: training_sample ni: number of input instance to be employed at tree each I： number of generated trees in random_forest 1) EM is empty

2) for i = 1 to I
3) TSi = bootstrapSample(TS)
4) Ci = BuildRandomTreeClassifiers(TSi, ni)
5) EM = EM ∪ {Ci}
6) next i
7) return EM

From the two resulting subsets, a pure class is generated from two subsets resulting after each iteration. Then, based on the rules, every subset gets separated. Thus the algorithm functions in such a way that prediction operates effectively. The bagging mechanism within RF enables algorithm to classify high dimensional data rapidly. The classification decision regarding accuracy obtained through voting from each and every classifier in ensemble.

3.7. Gaussian Naive-Bayes classification

Another algorithm that classifies data is primarily based on the Bayes theorem. The algorithm differs from Naive Bayes since it uses Gaussian normal distribution to find the independent quality among features. In this theorem, the training dataset is taken as input, and the test dataset is received as output. Gauss density function is used for classification. Each feature should be independent of the other feature. The classifiers need training data for the estimation of parameters required for classification.

Algorithm-4

Gaussian Naive Bayes

Input: Training dataset TD
$PV = (pv_1, pv_2, pv_3, \dots pv_n)$ / predictor variablevalue in testing_dataset.
Output： A class of testing_dataset.
Step：

1) Considered the dataset used for training PV;
2) Evaluate the standard deviationand mean of the predictor variables in each class;
3) Repeat
    4) Measure the probability of pi using the gauss density equation in each class;
5) Until the probability of all predictor variables $(pv_1, pv_2, pv_3, \dots pv_n)$ has been measure
6) Calculate the likelihood for each class;
7) Get the greatest likelihood;

The system's execution time is computed faster since the classifier performs based on Gaussian distribution. The continuous values correlated with every class are distributed when computing with a continuous data group. The estimation of continuous data is performed based on the mean and standard deviation calculated. The data for training will be segregated, and the equation is computed with the variable as X and class is represented as C

$$PV(X = x \mid C = c) = \frac{1}{\sqrt{2\pi\theta}} e^{\frac{-(x-\varepsilon)^2}{2\theta^2}} \quad \text{-- (9)}$$

Wherein this Pv defines the predictor variable value.

3.8. XgBoost Classification

This algorithm can be executed in different languages and differs from other algorithms since it uses a multithreaded technique in which CPU utilization is efficient. Thus it increases the speed and performance. In addition to the advantage of this boosting technique is the automatic handling of missing values. In parallel, it maintains a structure for constructing a tree, and the continuous training process can boost the fitted model into the newly comprised data.

Algorithm-5

XGBoost

Input: training set $\{(a_i, b_i)\}_{i=1}^{N}$, a differentiable loss function L(b, F(a)), a number of weak M lea

Algorithm:
Initialize model with a constant value:

$$f_{(0)}(a) = \arg\min \sum_{i=1}^{N} L\,(b_i, \emptyset)$$

For $m = 1$ to M
Compute the 'gradients' and 'hessians'

$$g_m(a_i) = \left[\frac{\partial L(b_i f(a))}{\partial f(a_i)}\right]_{f(a)=\hat{f}_{(m-1)}(a)}$$

$$h_m(a_i) = \left[\frac{\partial^2 L(b_i f(a_i))}{\partial f(a_i)^2}\right]_{f(a)=\hat{f}_{(m-1)}(a)}$$

Fit a base learner (or weak learner, e. g. tree)using the training set $\left\{a_i, -\frac{g_m(a_i)}{h_m(a_i)}\right\}_{i=1}^{N}$

$$\partial_m = \arg\min \sum_{i=1}^{N} \frac{1}{2} h_m(a_i)\left[-\frac{g_m(a_i)}{h_m(a_i)} - \emptyset(a_i)\right]^2$$

$f_m(x) = \alpha\,\partial_m\,(m)$
Update the model:
$f_m(a) = f_{(m-1)}(a) + f_{(m)}(a)$

Output $f_m(a) = f_{(M)}(a) = \sum_{m=0}^{M} f_m(a)$

The model is initialized with a value, and computation is performed using the formula for gradients and Hessians. Then, the training dataset for a base learner is calculated, and the model is updated using the function $f_m(a)$ this algorithm can be experimented with for predicting and classifying datasets. This study claimed that this algorithm obtains the best result in execution speed.

3.9. KNN Classifier – Low Variance in feature set

(KNN)K-Nearest Neighbour algorithm, is a prominent Non-parametric algorithm, on the basis of supervised learning-technique. The KNN algorithm considers the similarity between new data/case (features subsets) and the available features sets. This then put out the new feature case to a category, which will be not similar to available category of features. The similarity is determined based on the Euclidean distance of the feature. This actually means when new-feature appears in the dataset, it could be classified easily to well-suited category through KNN algorithm.

Algorithm-6

K − NN classifier algorithm

input : Dataframe DF
extracted features from DF were it contains set of features(attributes)and target
output: Class name(classification)
Begin
data spitting = 80: 20
training set = 80; testing set = 20
for each process X in the testing set do
if X
if $X \longrightarrow$ non − attack
$X \rightarrow$ attack
else then
for each process DF_j in the training set do
cal $\text{intrusion}_{\text{features}}(X, DF_j)$;
if $\text{intru}(X, DF_j)$equal 1.0 then
X is normal ; exist;

find k classification scores of intru (X, DF);
Calculate $\text{intru}_{\text{avg}}$fo the $\text{k} - \text{nn}$;
if $\text{intru}_{\text{avg}} > (0)$ then
$\text{X} - \text{non} - \text{attack}$
else then
$\text{X} - \text{attack}$

The dataframe is obtained as input. The transformed features from enhanced PCA, consists of various class features and variables. The splitting of data features to train and test data occurs. For each feature process of testing data, the condition to check it the features consists of any attack feature or non-attack feature were ensured. Hence training for classifying the features for every process of dataframe (DF_j) ,is performed. The binary values are verified for each feature variable$\text{intru}(\text{X}, \text{DF}_\text{j})$. If the value is 1, the variable returns to normal features. For every iteration, the k number of classification scores are determined of intru(X, DF). the average score of the intrusion variable is computed for KNN clusters. Then with if condition, the $\text{intru}_{\text{avg}}$ greater than, then the feature is classified as non attack, other side if the value is 0, then the feature declared to be the attack feature.

IV. Results and Discussion

These performance assessment are considered for measuring the system's capability towards intrusion detection in response to the enhancements taken in the implementation.

4.1 Performance Metrics

The performance of the proposed IDS model is determined with the performance metrics such as F1-score accuracy, recall and precision.

Here, the following terms are represented as;

T_P- True-Positive, F_P- False-Positive, T_N- True-Negative and F_N- False-Negative

a. Accuracy (A_{cc})

The term accuracy can be denoted as the model classification rate that is provided through the proportion of correctly classified instances ($T_P + T_N$) to the sum of instances in the dataset ($T_P + F_P + T_N + F_N$) , with the following equation (11) the accuracy range is evaluated.

$$A_{cc} = \frac{(T_N + T_P)}{(T_P + F_P + T_N + F_N)} \quad \text{-- (11)}$$

b. Precision

The term precision is defined as the degree of covariance of the system that is resulted from the correctly identified instances T_P to the total array of instances that are accurately classified($T_P + F_P$). It comprises reproducibility and repeatability of the resources. It is measured by equation (12)

$$Precision = \frac{T_P}{F_P + T_P} \quad \text{- (12)}$$

c. Recall

The term recall is the one of the performance metric, which quantifies the amount of correct positive classification made out of all the positive classification which could have been made. It is considered with the following equation (13).

$$Re_{call} = \frac{T_p}{F_N + T_P} \quad \text{-- (13)}$$

d. F1-score

F1 score is the weighted harmonic-mean value of recall and precision, it is calculated with the succeeding equation (14)

$$F1 - score = 2 \times \frac{re_{call} \times pre_{cision}}{re_{call} + pre_{cision}} \quad \text{-- (14)}$$

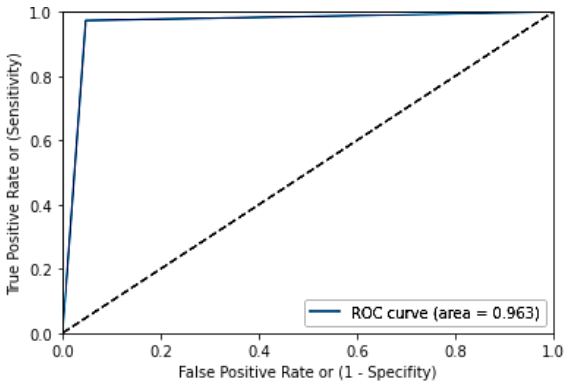
e. Receiver-Operating-Characteristic (ROC)

The ROC curve is employed for determining the appropriate threshold for the system that gives probability scores as binary classification output.

4.2 Performance Analysis

Performance analysis of the proposed system measured by few specific performance metrics, which are stated above. These calculation are considered for measuring the behavior of the system in response to the developments taken in the implementation. The analysis performed with both the dataset are highlighted. Generally, the ROC graph is generated to determine the efficiency of the classifier. The ROC is defined as the plot of sensitivity test as the y coordinate to its 1- false positive rate (or) specificity as x-coordinates. It is considered as the efficient method for determining the performance of the classifiers.

Generally, if the range of area under ROC is 0.5 suggested as no discriminations, the capability of classifying the intrusion by detecting the presence and absence of attack (or) based the applied conditions. The range between 0.7 -0.8 is denoted as acceptable, the range between 0.8 -0.9 is denoted as excellent and more than 0.9 is considered as outstanding performance.



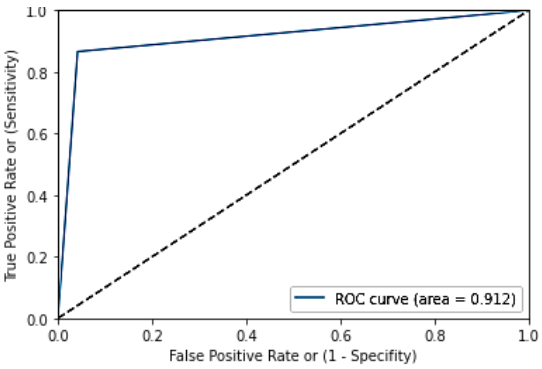
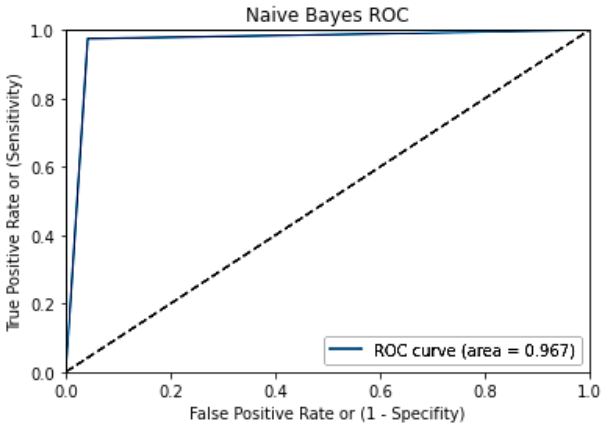


Figure-3(a) *Figure-3(b)*

Figure 3(a) and 3(b) ROC range of XG Boost and Random Forest Using UNSW-NB-15.

From figure 3(a) and 3(b), which represents result of area under ROC for XGBoost and Random forest using UNSW-NB-15 dataset. ROC curve area of XG Boost and Random Forest were 0.963 and 0.912. This denotes that the performance of XG Boost and Random forest classifiers are more efficient and the classification was accomplished competently.



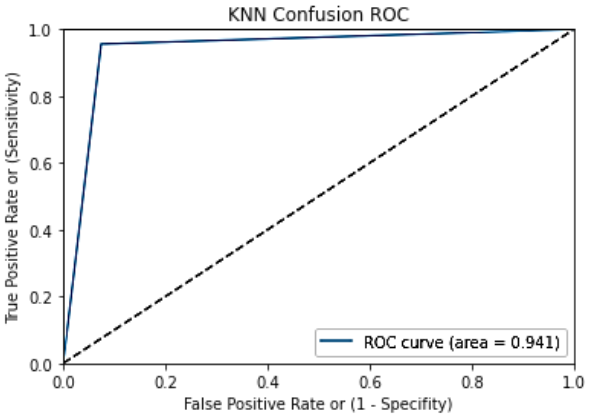
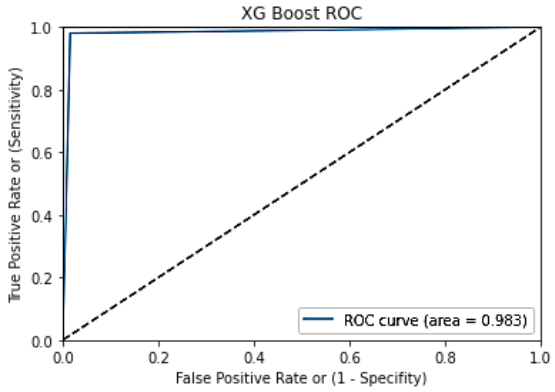


Figure-4(a) *Figure-4(b)*

Figure 4(a) and 4(b) ROC of Naïve Bayes and KNN using UNSW-NB-15.

From figure 4(a) and 4(b), which represents result of area under ROC for Naïve Bayes and KNN using UNSW-NB-15 dataset. ROC curve area of Naïve Bayes and KNN were 0.967 and 0.941. This denotes that the performance of introduced classifiers are more efficient and the classification was accomplished proficiently. Therefore, with the ROC evaluation, Naïve Bayes showed better ROC curve (area=0.967), which is higher than the other classifiers. This proves that the Naïve Bayes showed enhanced classification performance.

Similarly, the same analysis have been performed for all the classifiers using NSL-KDD dataset. The performance of classifiers using NSL-KDD dataset are showed in the following,



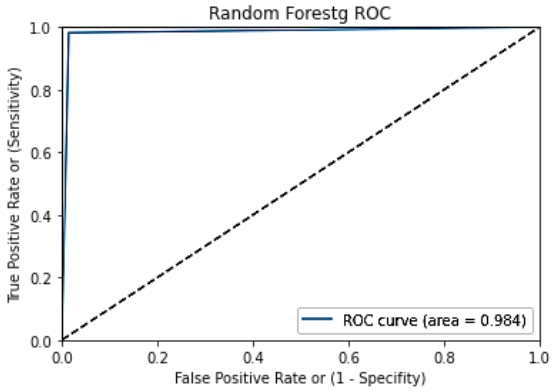
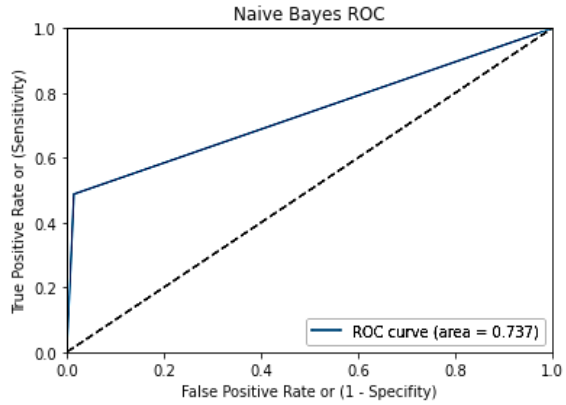


Figure-5(a) *Figure-5(b)*

Figure-5(a) and 5(b) ROC range of XG Boost and Random Forest Using NSL-KDD

From figure 5(a) and 5(b), which represents result of area under ROC for XGBoost and Random forest using NSL-KDD dataset. ROC curve area of XG Boost and Random Forest were 0.983 and 0.984. This denotes that the performance of XG Boost and Random forest classifiers are more efficient and the classification was accomplished competently.



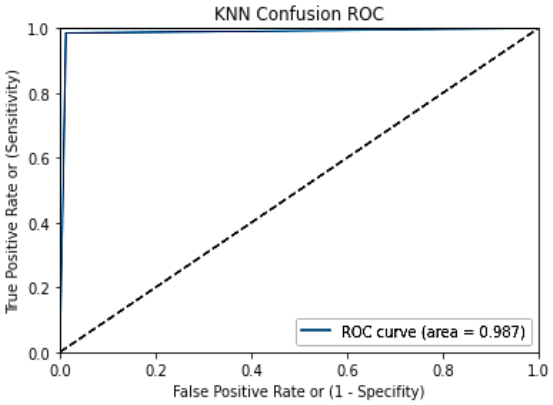


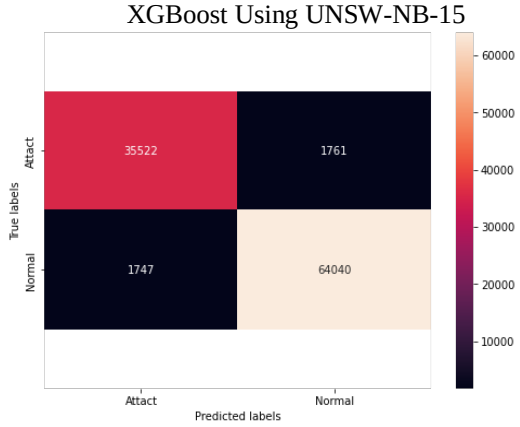
Figure-6(a) *Figure-6(b)*

Figure-6(a) and 6(b) ROC of Naïve Bayes and KNN using NSL-KDD.

From figure 6(a) and 6(b), which represents result of area under ROC for Naïve Bayes and KNN using NSL-KDD dataset. ROC curve area of Naïve Bayes and KNN were 0.737 and 0.987. This denotes that the performance of introduced classifiers are more efficient and the classification was accomplished effectively. By considering the analysis performed using NLS-KDD dataset, KNN showed excellent classification performance with 0.987 ROC curve area and the Naïve Bayes showed acceptable range of classification performance with ROC curve (area=0.737).

Followed by this, the confusion matrix of the classifiers such as XGBoost, Random forest, Naïve Bayes and KNN are stated in the following for both the dataset. The confusion matrix was employed to estimate the performance of methods involved in classification. For binary classification, the scheme of the confusion matrix is seen in the following figures 7(a) to 10(b).

From figure 7(a) and 7(b), the confusion matrix of XGBoost using UNSW-NB-15 and NSL-KDD dataset are shown. The XG Boost classifier using UNSW-NB-15 dataset 35522 attacks were classified as attacks and 1761 attacks were misclassified normal. Similarly, 6404 normal labels were classified as normal and 1747 attacks were misclassified as normal. Likewise, the XGBoost classifier using NSLK dataset, 15079 attacks were classified as attacks and 235 attacks were misclassified as normal. Similarly, 14111 normal labels were classified as normal and 278 attacks were misclassified as normal. Where, the correctly classified rates were higher than the misinterpreted classification. This shows that the XG-Boost showed efficient classification performance in both the dataset.



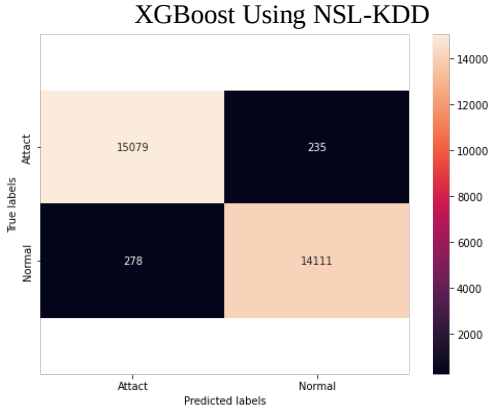
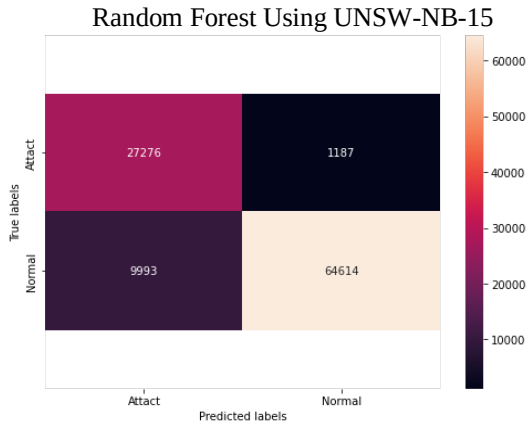


Figure 7 (a) *Figure 7 (b)*

Figure 7(a) and 7(b) Confusion-Matrix of XG Boost using UNSW NB 15 and NSL-KDD.

From figure 8(a) and 8(b), the confusion-matrix of Random forest using both the dataset are shown. The Random forest classifier using UNSW-NB-15 dataset, 27276 attacks were classified as attacks and 1187 attacks were misclassified normal. Similarly, 64614 normal labels were classified as normal and 9993 attacks were misclassified as normal. Likewise, the Random Forest classifier using NSLK dataset, 15203 attacks were classified as attacks and 221 attacks were misclassified as normal. Similarly, 14125 normal labels were classified as normal and 254 attacks were misclassified as normal. Where, the correctly classified rates were higher than the misinterpreted classification. This shows that the Random Forest showed efficient classification performance in both the dataset.



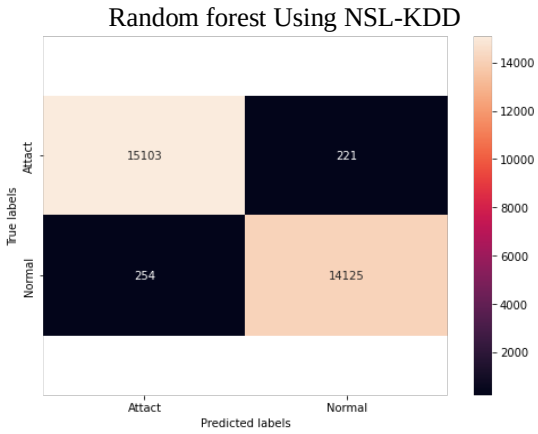
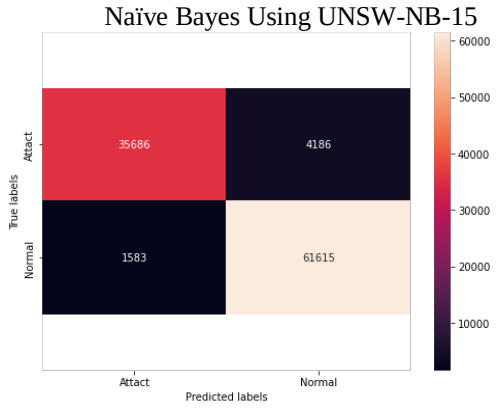


Figure 8(a) *Figure 8(b)*

Figure 8(a) and 8(b) Confusion-Matrix of Random Forest using UNSW-NB-15 and NSL-KDD.

From figure 9(a) and 9(b), the confusion-matrix of Naïve Bayes using UNSW-NB-15 and NSL-KDD dataset are shown. The Naïve Bayes classifier using UNSW-NB-15 dataset, 35686 attacks were classified as attacks and 4186 attacks were misclassified normal. Similarly, 61615 normal labels were classified as normal and 1583 attacks were misclassified as normal. Likewise, the Naïve Bayes classifier using NSLK dataset, 315 attacks were classified as attacks and 12 attacks were misclassified as normal. Similarly, 14334 normal labels were classified as normal and 15044 attacks were misclassified as normal. Where, the correctly classified rates were higher than the misinterpreted classification. This shows that the Naïve Bayes showed efficient classification performance in both the dataset.



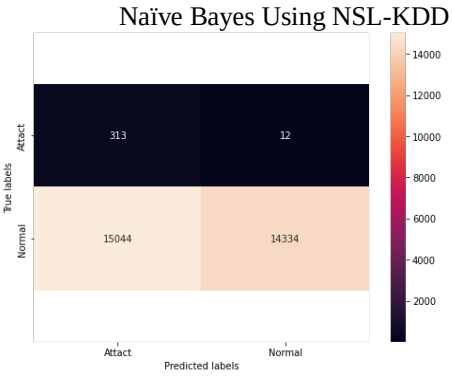


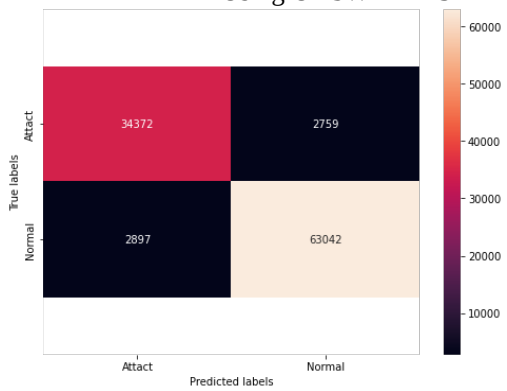
Figure. 9(i) *Figure 9(ii)*

Figure 9(i) and 9(ii) Confusion-Matrix of Naïve Bayes using UNSW-NB-15 and NSL-KDD

From figure 10(i) and 10(ii), the confusion-matrix of KNN using UNSW-NB-15 and NSL-KDD dataset are shown. The KNN classifier using UNSW-NB-15 dataset, 34372 attacks were classified as attacks and 2759 attacks were misclassified normal. Similarly, 63042 normal labels were classified as normal and 2897 attacks were misclassified as normal. Likewise, the KNN classifier using NSLK dataset, 15149 attacks were classified as attacks and 189 attacks were misclassified as normal. Similarly, 14157

normal labels were classified as normal and 208 attacks were misclassified as normal. Where, the correctly classified rates were higher than the misinterpreted classification. This shows that the KNN showed efficient classification performance in both the dataset.

KNN Using UNSW-NB-15



KNN Using NSL-KDD

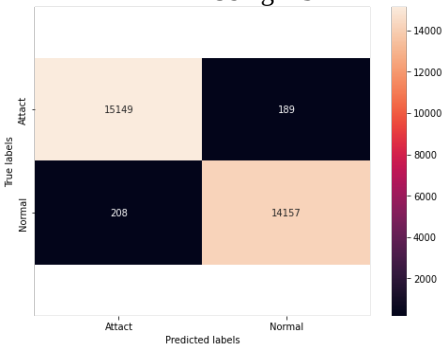


Figure 10(a) *Figure 10(b)*

Figure 10(a) and 10(b) Confusion Matrix of KNN using UNSW-NB-15 and NSL-KDD

From figure 11, the performance evaluation of XG Boost, Random Forest, Naïve Bayes and KNN are done in terms of considered performance metrics, F1-score, precision, recall, and A_{cc} using UNSW-NB-15 dataset. The XG Boost showed 0.965 A_{cc}, 0.97 recall, 0.97 precision and 0.97 F1-score. Where, the random forest showed 0.89 A_{cc}, 0.89 recall, 0.9 F1-score and 0.91 precision. In a similar context, Naïve Bayes showed, Acc (0.94), F1-score (0.94), recall (0.94), and precision (0.94). Then, KNN classifier resulted, A_{cc} (0.94), F1 score (0.95), recall (0.95) and precision (0.95). From this evaluation, XG-Boost showed better classification rate in terms of F1-score, precision, recall, and A_{cc} than the other classifiers.

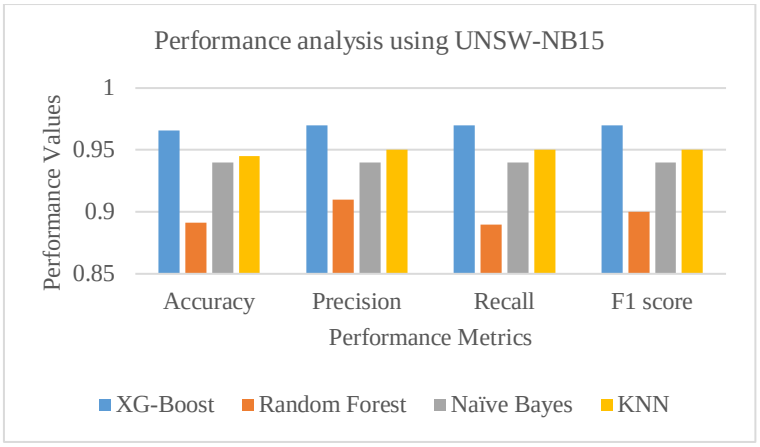


Figure 11 Performance analysis of the designed model using UNSW-NB-15.

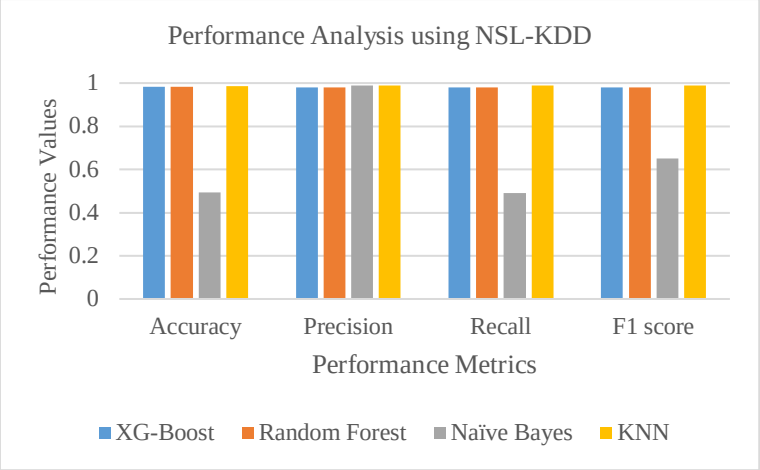


Figure 12. Performance analysis of the designed model using NSL-KDD

The performance analysis of the designed model using NSL-KDD is shown in figure 12. The experimental evaluation was done by using NSL-KDD dataset. The XG Boost showed 0.98 A_{cc}, 0.98 recall, 0.98 precision and 0.98 F1-score. Where, the random forest showed 0.984 A_{cc}, 0.98 recall, 0.98 F1-score and 0.98 precision. In a similar context, Naïve Bayes showed, 0.4931 $A_{cc,}$ 0.65 F1-score, 0.49 recall, and 0.99 precision. Then, KNN classifier resulted, A_{cc}(0.986), F1-score(0.99), recall(0.99) and precision(0.99). From this evaluation, KNN showed better classification rate in terms of F1-score, precision, recall, and A_{cc} than the other classifiers. The Naïve Bayes showed low $A_{cc,}$ recall and F1-score.

4.3 Comparative Analysis

The performance evaluation of proposed IDS system is performed by comparing various machine learning methods listed below with the implemented system, and validation accuracies are obtained. Figure 13 and 14 represents the comparative analysis of the proposed work with the existing system by using both UNSW-NB-15 and NSL KDD datasets.

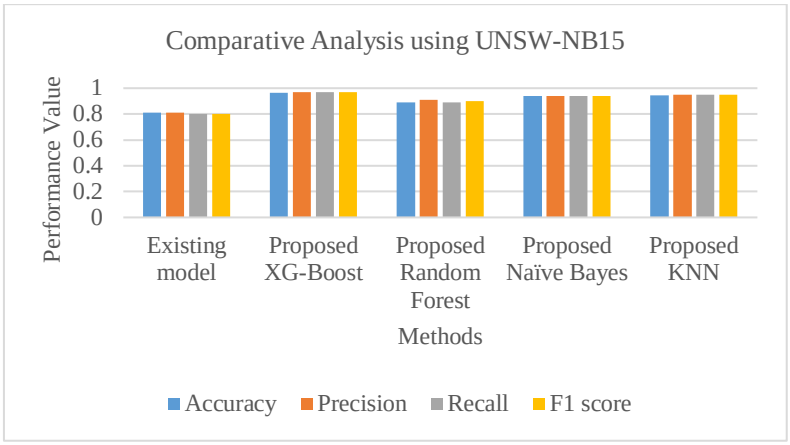


Figure.13. Comparative Analysis of proposed ML algorithms with existing ML algorithms(Du, Cheng, Wang, & Han, 2022)

Figure 13 represents the comparative analysis of the proposed model with the existing system (Du et al., 2022) using UNSW-NB-15 and the comparison is performed in terms of F1-score, precision, recall, and $A_{cc.}$ Existing work showed A_{cc} (0.81), F1-score (0.81), recall (0.81) and precision (0.81). Then, the proposed XG Boost showed A_{cc} (0.965), F1-score (0.97), recall (0.87) and precision (0.97). The Random Forest showed A_{cc} (0.89), F1-score (0.90), recall (0.89) and precision (0.91), and Naïve Bayes showed F1-score (0.94), recall (0.94), A_{cc}(0.94)and precision(0.94),. Similarly, KNN showed 0.95 recall, 0.95 precision, 0.94 A_{cc} and 0.95 F1-score. From this evaluation, it has been determined that the

proposed work showed better result than the existing work in terms of all the considered performance metrics.

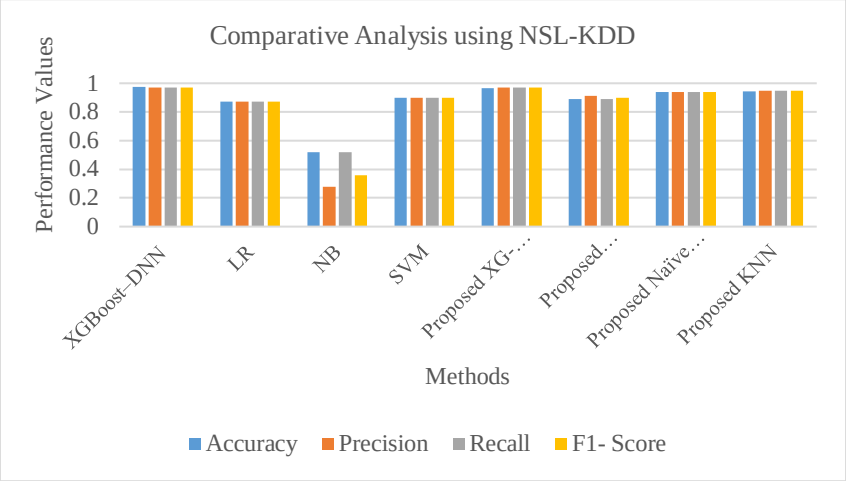


Figure 14. Comparative Assessment of proposed framework with existing work (Devan & Khare, 2020) using NSL-KDD

The comparative analysis of the proposed work with various ML algorithm (Devan & Khare, 2020) implemented in development of IDS by using NSL-KDD is shown in figure 14. The existing XG Boost-DNN, LR, NB and SVM showed 0.976 A_{cc}, 0.87 A_{cc}, 0.52 A_{cc} and 0.9 A_{cc}. The proposed XG Boost, Random forest, Naïve Bayes and KNN showed 0.96 $A_{cc,}$ 0.89 $A_{cc,}$ 0.94 $A_{cc,}$ and 0.94 A_{cc}. The proposed model showed better results than the existing model in all the considered performance metrics, which is clearly represented in above shown figure 14.

V. Conclusion

The study focussed to implement a prominent Intrusion detection model performing extraction of features through deep layer learning of feature subsets in more hidden layers by two models of Deep CNN 1-dimensional models. Through this Deep CNN-1D model different level of features are obtained. Those optimal features were merged and fed to feature transformation phase, applied with enhanced PCA approach, to maintain linearization of the extracted features. The single set of linear features yielded and were reduced to low-dimensional vectors, by PCA with F-transform algorithm. Further to this, F-Transform combined with PCA method, significantly minimises the computational time of PCA and increases the accuracy of classifying the attacks or non-attack features. The linear features sets, classifying the normal and abnormal data in IDS with four different classifiers RF algorithm, XgB, NB and KNN classifier. The effectiveness of the classifiers deliberated through ROC measure, showing outstanding performance of NB classifier (ROC = 0.967) in UNSW-NB-15 dataset. Similarly, using NSL-KDD dataset, KNN classifier explicated high classification performance with high value of ROC (0.987). The performance further examined by confusion matrix, and all four classifiers, exhibited to yield out better classifications, relying in both the datasets. The performance of classifiers are assessed using two datasets dataset in terms of accuracy, recall, F1-score and precision. KNN classifier, stands out best to classify the features with 0.986 A_{cc}, 0.99 F1-score, 0.99 recall and 0.99 precision score higher than other classifier. Hence comparative analysis of proposed Deep CNN-1d feature selection with EPCA feature transformation model exhibited the outstanding outcomes in intrusion detection in terms of accuracy (96.66%), precision (97%), F1-Score (97%) and Recall (97%) through XGBoost classifier. However future direction to design a effective IDS model with less time consumption and resources can be built.

References

Al-Janabi, M., Ismail, M. A., & Ali, A. H. (2021). Intrusion Detection Systems, Issues, Challenges, and Needs. *Int. J. Comput. Intell. Syst., 14*(1), 560-571.

Aldribi, A., Traore, I., Moa, B., & Nwamuo, O. (2020). Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking. *Computers & Security, 88*, 101646.

Aleesa, A., Younis, M., Mohammed, A. A., & Sahar, N. (2021). Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques. *Journal of Engineering Science and Technology, 16*(1), 711-727.

Bu, S.-J., & Cho, S.-B. (2020). A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. *Information Sciences, 512*, 123-136.

Camilleri, D., & Prescott, T. (2017). *Analysing the limitations of deep learning for developmental robotics.* Paper presented at the conference on Biomimetic and Biohybrid Systems.

DATAFLAIR, T. (2019). Advantages and disadvantages of machine learning language: DataFlair© Available at. https://data-flair. training/blogs/advantages-and ….

Devan, P., & Khare, N. (2020). An efficient XGBoost–DNN-based classification model for network intrusion detection system. *Neural Computing and Applications, 32*(16), 12499-12514.

Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del-Rincon, J., & Siracusa, D. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Transactions on Network and Service Management, 17*(2), 876-889.

Du, X., Cheng, C., Wang, Y., & Han, Z. (2022). Research on Network Attack Traffic Detection HybridAlgorithm Based on UMAP-RF. *Algorithms, 15*(7), 238.

Farhat, S., Abdelkader, M., Meddeb-Makhlouf, A., & Zarai, F. (2020). *Comparative study of classification algorithms for cloud IDS using NSL-KDD dataset in WEKA.* Paper presented at the 2020 International Wireless Communications and Mobile Computing (IWCMC).

Hijazi, A., El Safadi, A., & Flaus, J.-M. (2018). *A Deep Learning Approach for Intrusion Detection System in Industry Network.* Paper presented at the BDCSIntell.

Hussain, K., Neggaz, N., Zhu, W., & Houssein, E. H. (2021). An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. *Expert Systems with Applications, 176*, 114778.

Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE Access, 7*, 38597-38607.

Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing, 25*(15), 9731-9763.

Kunang, Y. N., Nurmaini, S., Stiawan, D., & Suprapto, B. Y. (2021). Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications, 58*, 102804.

Liu, C., Gu, Z., & Wang, J. (2021). A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning. *IEEE Access, 9*, 75729-75740.

Liu, L., Xu, B., Zhang, X., & Wu, X. (2018). An intrusion detection method for internet of things based on suppressed fuzzy clustering. *EURASIP Journal on Wireless Communications and Networking, 2018*(1), 1-7.

Maleki, N., Zeinali, Y., & Niaki, S. T. A. (2021). A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Systems with Applications, 164*, 113981.

Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. *International Journal of Information Security, 20*(3), 387-403.

Mohammed, B., & Gbashi, E. K. (2021). Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination. *Engineering and Technology Journal, 39*(7), 1069-1079.

Ouadfel, S., & Abd Elaziz, M. (2020). Enhanced crow search algorithm for feature selection. *Expert Systems with Applications, 159*, 113572.

Saputra, M. F. A., Widiyaningtyas, T., & Wibawa, A. P. (2018). Illiteracy classification using K means-Naïve Bayes algorithm. *JOIV: International Journal on Informatics Visualization, 2*(3), 153-158.

Sindhu, R., Ngadiran, R., Yacob, Y. M., Zahri, N. A. H., & Hariharan, M. (2017). Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. *Neural Computing and Applications, 28*(10), 2947-2958.

Singh, D. K., & Shrivastava, M. (2021). Evolutionary Algorithm-based Feature Selection for an Intrusion Detection System. *Engineering, Technology & Applied Science Research, 11*(3), 7130-7134.

Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and information technologies, 19*(1), 3-26.

Wani, A., & Khaliq, R. (2021). SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). *CAAI Transactions on Intelligence Technology, 6*(3), 281-290.

Werlinger, R., Hawkey, K., Muldner, K., Jaferian, P., & Beznosov, K. (2008). *The challenges of using an intrusion detection system: is it worth the effort?* Paper presented at the Proceedings of the 4th symposium on Usable privacy and security.

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.