ELSEVIER

Lightweight Intrusion Detection System using QRNetV2

Ilfa Shaheed Valiyavallappil

Keywords:
Cyber security
Intrusion detection system
Harris Hawk Optimization
CSE-CIC-IDS2018
MobileNetV2
ShuffleNet

a b s t r a c t

The evolution of technologies has led to an increasing concern in the cyber security of personal data stored in various devices such as desktop, cloud, and mobile applications. Such intrusions has led to huge financial losses for the commercial industry and personal data loss for individuals. Due to the rising concerns, an intrusion detection system (IDS) was introduced to predict any upcoming anomalies in the servers. The traditional intrusion detection systems used to store the previous cyber attacks in a particular database and compare them with the new recent attacks. However, the technique seemed futile due to the increasing skill base of hackers, and the increasing diversity of the attack types. In this study, a multi-class attack classification model is proposed utilizing lightweight intrusion models such as MobileNetV2 and ShuffleNet. The dataset utilized in this study is CSE-CIC-IDS2018 which contains a diverse amount of attack types with their metadata. The first stage involves evaluating the pre-processed dataset on MobileNetV2 and ShuffleNet. The second stage involves conducting feature selection using the SoftMax layer on both the models. The selected features are then evaluated on SVM and KNN classifiers. The third stage involves adding an extra step i.e. Harris Hawks Optimization Algorithm (HHO) to get the best feature set and again evaluate on SVM and KNN classifiers. The best performing model is from the first stage namely QRNetV2 (MobileNetV2) with an accuracy of 96.6% with a training time of 42 minutes.

1. Introduction

With emerging technologies, there has been an unprecedented affinity for data in all established and emerging fields. Data stored in devices that are connected to the internet can be labelled as 'Internet of Things' (IoT). A forecast predicts the number of IoT devices to massively inflate to 64 billion devices by 2025 [1]. Since the quantity of IoT devices are escalating, the complexity of networks as well as the vulnerability of data stored within them increases. To ensure protection against cyber security attacks that may result in infringement of classified data a joint system known as 'Network Intrusion Detection System' (NIDS) was introduced. The main objective of such a system is to monitor the network activities and traffic in the internet space to avoid and detect any upcoming cyber attacks to IoT devices. NIDS are well known to keep cyber criminals away from unauthorized access, data breaches, data corruption and other vindictive activities.

A NIDS is classified into two major divisions: Anomaly-based (ANIDS) and Signature-based (SNIDS) systems. A SNIDS system relies on a database that stores previous information of any kind of cyber attacks to identify a future unseen test case of a network flow attack. Whereas ANIDS detects an attack using methods to identify any deviation from the normal flow of data in IoT devices. SNIDS may be able to identify any known attack quickly such as SNORT and SURICATA [2] with very few false positives. However it fails to discover any new unseen attacks. This paper follows an anomaly-based intrusion detection system using a deep learning model. ANIDS performs better in a real-time network flow environment due to its potential to identify and detect any concealed attacks defined unique from the predefined behavioural activity. Since it can do such a unique task, naturally there will be an existence of false positives as well [3].

NIDS is advised to be developed using machine learning (ML) and deep learning (DL) which are subsets of artificial intelligence (AI) [4]. A deep learning model may be preferred compared to a machine learning model due to its capability to capture the complex network activities and identifying an unforeseen attack efficiently and accurate. However, deep learning models are computationally expensive due to which this paper introduces a novel lightweight intrusion detection system. This will enable the lower utilization of computational resources and reduces the time effectively to detect any anomaly in the network of an IoT device.

The work done in this paper utilizes dataset in the form of QR code images to enhance visual representation that can work well with deep learning models such as CNN's since they capture complex patterns the best in case of images. The metadata captured in the QR code images will further aid in classification of the attacks in an effective manner [5]. The standardized format of QR code images restrains any variability in the images giving the models a standardized

format to learn from and accurately detect any unseen attack in the future. Such a technique can also encrypt the data protecting any sensitive information it might carry. Due to its non-analysable nature, it proves to be difficult for third parties to capture information on a glimpse of it.

2. Related works

Under this section, a precise and concise review is provided on the framework of various learning techniques used in building an intrusion detection system for a multi-class attack classification.

2.1. Literature review

Until the present time, multiple IoT attacks have been identified using the Intrusion Detection System (IDS) through different methodologies. The summary of selected works is as follows:

In this study [6], a network intrusion detection system is developed to identify and classify any type of IoT network attacks. This paper follows a method of feature clustering using classes, Message Queuing Telemetry Transport (MQTT), Flow and Transmission Control Protocol (TCP) with the help of the features in the UNSW-NB15 dataset. Supervised machine learning algorithms such as artificial neural networks, random forests and support vector machine are employed. It helps in overcoming dimensionality curse, overfitting, and class imbalance in dataset. The feature clusters help in acquiring minimum training time and maximum accuracy compared to the other simple machine learning algorithms. Nevertheless, feature clusters might overlook the minority classes creating a slight imbalance and it is unable to identify unknown IoT network attacks.

Another work [7] discusses the development of a new model using the random forests algorithm with synthetic minority over-sampling technique (RF-SMOTE) to detect the IoT malicious attacks in a network. The experimentation is done on NSL-KDD dataset and N-BaIoT dataset having different types of IoT attacks. The accuracy showed improvement on four classes in NSL-KDD dataset. The model showed additional improved accuracy on N-BaIoT dataset. The proposed model also helped in the creation of a non-biased model taking care of data imbalance. However, it can create complexity in the model due to increase in the size of the dataset. Overfitting is also a possibility in the SMOTE technique.

Through this study [8], it suggests the use of a machine learning based intrusion detection system (ML-IDS) for detecting IoT network attacks. Various supervised machine learning algorithms are applied to the UNSW-NB15 dataset such as XGBoost, CatBoost, KNN, SVM, QDA and NB classifiers. Data preprocessing was applied including feature scaling using the minimum-maximum method of normalization to minimize the amount of data being leaked. Dimensionality reduction is done using the Principal Component Analysis (PCA) method. The paper succeeded in reducing the communication overhead and creating an intelligent IDS system. Despite the proposed system, there exist certain disadvantages such as the absence of a dynamic nature in the IoT IDS system and it is also unable to exploit large quantities of data.

From the study [9], it can be inferred that botnets have the capability of disrupting multiple systems and releasing malicious attacks on IoT networks. To tackle this intriguing problem, a deep learning model is proposed to identify the botnet attacks in real-time. The model is evaluated on a CTU-13 dataset having multiple hidden layers. The model shows a decrease in the mean square error during performance evaluation as well as improved accuracy. In cases where the validation dataset overfits the training dataset, there can be an increase in the mean square error. Random noise or outliers might be present during a case of overfitting in the proposed model.

Another paper [10] describes the various machine learning algorithms like decision tree, gradient boosting machine (GBM), and random forests to interpret and classify various categories of IoT network attacks. Performance indicators are applied to each of the machine learning model and a comparative study is conducted. From the analysis it can be inferred that high accuracy is shown by decision tree algorithm but random forest have satisfactory area under curve (AUC) scores due to the combination of many weak decision trees. Although gradient boost might perform well, it may not be the optimal option in terms of accuracy and timing aspects. The gradient boosting machine takes longer training time which is not efficient. Feature selection process is not simplified for the classification models.

The journal article [11] aims to create a supervised machine learning model to predict any IoT attacks in the network based on past data that can be integrated into the real world to keep all data secure and private in the future. The dataset from Kaggle public repository undergoes preprocessing through various encoding techniques. Two different approaches are applied to the dataset. The initial approach involves applying the classification algorithms such as decision tree, naïve bayes, logistic regression random forest and artificial neural network (ANN) to whole dataset. The second approach involves applying the same classification algorithms to the dataset after excluding the binary values 0 and 1 in the value feature. The models shows high accuracy and efficiency in the second approach. However, it does not consider the micro-services causing variations in the anomalies in the IoT network.

Through the work [12], it suggests the utilization of ensemble learning in creating an efficient intrusion detection system (IDS). It presents a novel IDS system by classifying IoT attacks using ensemble learning with a model called IDS-SIoEL. The model incorporates AdaBoost and integrates various feature selection techniques which include Pearson correlation, Boruta, and mutual information. The model is experimented on Edge-IIoT, BoT-Iot and IoT-23 datasets with GPU. Compared to existing state-of-the-art ML models, the ensemble model shows higher accuracy, better F1-score, recall and precision. While the quantity of trees increases, the accuracy also increases. The performance time achieved is a very good value as the model is deployed on GPU. However, the article fails to conduct multi-classification of IoT attacks and a more enhanced model such as a deep learning model is absent.

Another study [13], proposes an enhanced anomaly-based Intrusion Detection Deep learning Multiclass classification model (EIDM) which is capable of classifying 15 traffic behaviours including 14 attack types. The models are built using, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), multilayer perceptron (MLP) and a combination of CNN and LSTM. The dataset preprocessing was done using the SMOTE technique. A large-scale comparison study is conducted between EIDM and other existing ML algorithms. The study shows EIDM has achieved higher accuracy and precision. It contains advanced layers that differentiate between targeted classes accurately. The model is very complex hence take more computational time and is not an efficient factor.

This paper [14] develops a filter-based feature selection deep neural network (DNN) where the highly correlated features are displayed. Parameter tuning and tuning of various hyperparameters also take place. Generative Adversarial Networks (GANs) were utilized in generation of synthetic data of minority attacks in order to solve the issue of class imbalance. As the number of packets of minority attacks increased using GAN-DNN, the accuracy increased. It has highest accuracy when compared to other simple models. Only five types of traffic has been considered in this model, the accuracy is not technically the highest when compared to other models.

In this paper [15], the objective is to develop a novel deep learning model for different types of IoT attack datasets using an Intrusion Detection System (IDS). The need for IDS comes from the demand for security of data in IoT networks as the number of devices connected to the internet increases. The datasets selected are KDD'99, NSL-KDD and UNSW-NB15 which undergoes preprocessing by using methods such as numericalization and normalization. Numericalization converts textual data to numerical features. Models such as RNN, LSTM, and DNN are used to train on the selected datasets. On evaluation, RNN and DNN have good accuracy and detection rates on KDD'99 and NSL-KDD datasets. However, the proposed models are unable to provide good results on UNSW-NB15 dataset.

In the work [16], it employs a deep neural network (DNN) and federated learning (FL) for the classification of IoT attacks. In addition, it also suggests mutual information (MI) for the accurate detection of anomalies in the IoT network. The technique makes use of decentralized on-device information to identify any IoT network threats. The data that is placed on localized IoT gadgets is used for the model training phase and only the weights are shared in the FL server which is centralized. The dataset IoT-Botnet goes through the

training phase of the model. The model shows promise and efficiency due to higher accuracy and a decrease in the false alarm rate. The model shows more satisfactory results than basic deep learning models. It shows a high F1 score and true positive and negative rates. However, privacy concerns and communication overhead due to the sharing of global models are still challenges to be solved.

The study [17] discusses the use of federated learning which is novel and unique from any other simple models as it encrypts the edge devices to provide complete encryption of data. The study adapts the Pelican Optimization Algorithm with Federated Learning Driven Attack Detection and Classification (POAFL-DDC) method. The technique is performed on decentralized on-device data for classifying the IoT network threats. The data is kept on the local devices by substituting updated weight in the central FL server and then the federated training phase is employed using a Deep learning model. The deep learning model utilized in this study is the Deep Belief Network (DBN) and POA is used for optimizing the hyperparameters. The evaluation of the model is evaluated on the TON_IOT dataset. The performance metrics indicate good accuracy over other models. However, the POA algorithm might not capture the complexities in the IoT attacks.

This work [18], indicates the privacy and security concerns as well as regulatory restrictions as challenges that call for attention using federated learning. This technique allows distributed clients to come together and build a shared model in collaboration with each other while keeping privacy intact. The paper employs a Fed-ANIDS that integrates anomaly detection and federated learning to solve the problem of privacy in centralized models. The intrusion core can be calculated using various models such as simple autoencoders, adversarial autoencoders, and variational autoencoders. The model outperforms other models and shows high accuracy and efficiency while preserving the privacy of users. It has fewer false alarms. The FedProx algorithm that generalizes and re-parameters the standard FL algorithm, FedAvg shows better results. However, it lacks domain generalization to identify a wide range of IoT attacks which is a drawback.

The paper [19], states the use of black-box deep learning is inefficient due to its lack of interpretability, hence introducing integration of an Extreme Gradient Boosting classifier (XG-Boost) with explainable artificial intelligence to interpret the detections. The model not only improves the accuracy and efficiency but also the trust in the model. The model is evaluated on the IOTD20 dataset and evaluation of each feature using XAI indicates the accurate detection of susceptible IoT network attacks. The model shows high accuracy and overcomes the issues of overfitting. Although there are concerns of user data leakage and oversimplification of interpretations by XAI.

Another study [20] discusses the use of deep learning with explainable artificial intelligence to create a model and interpret its prediction using XAI. A filter-based approach is followed to decrease the range of features and build a CNN and DNN deep learning models. Two datasets, NSL-KDD and UNSW-NB15 are the datasets used for the evaluation of the model. For both datasets a satisfactory accuracy is attained with less computational cost. A satisfactory model explanation is also shown using XAI. In order to increase confidence in the model, Shapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) and methods are utilized. However, the issue of class imbalance persists which needs to be taken care of.

3. Research Gap

The various techniques and methods used by authors in creating an efficient intrusion detection system are to be discussed. Maximum classifiers have been implemented as stated in the literature review above. Strategies such as machine learning, ensemble learning, deep learning, federated learning, and explainable AI were utilized in the novel development of an intrusion detection system. Ensemble learning and Explainable AI were implemented by very few research papers. Explainable AI is an emerging domain hence not many research papers have covered the entirety of it. The datasets used majorly were NSL-KDD and UNSW-NB15.

In recent years there has been more interest in hybrid deep learning models and federated learning models than single classifiers. They have resulted in more accurate classifications. Federated learning is emerging due to the ability to decentralize the training process in local devices preserving privacy and security. Feature selection techniques include correlation selection, PCA, mutual information, and several other methods were used to reduce the dimensionality. Autoencoders showed high precision and accuracy.

Most of the datasets were CSV files whereas certain federated learning papers had files in PCAP (Packet Capture) format that were data files created using a program. Many models have tried to do real-time classification however only a certain few were accurately efficient. Many shallow machine-learning techniques were employed. No dataset was properly labelled due to unpredictive nature of network traffic and needed data preprocessing using mostly the SMOTE technique. Traditional machine learning techniques fall short on large datasets.

In conclusion data preprocessing is a major segment before creating a model due to redundancy, corrupted and unlabelled data. The SMOTE technique might be efficient to enhance the minority classes. Feature reduction also can result in higher performance and less computational overhead. It is advised to employ a hybrid learning methods for more benefits in efficiency and accuracy. The GPU can help in the advancement of the model prediction and detection. Explainable AI are also an efficient method in interpretation of the model prediction.

4. Dataset

The dataset for the intrusion detection system was acquired from Kaggle [6] open-source website. The CSE-CIC-IDS2018 dataset was created by the Communications Security Establishment (CSE) and the Canadian Institute of Cybersecurity (CIC). The dataset consists of a wide range of attacks and contains metadata about real-time network traffic. It contains both Benign and malicious attack types such as Brute Force, Distributed Denial of Service (DDoS), Botnets, etc. The dataset encapsulates the network traffic at both flow and packet level. The dataset is divided into numerous .csv files each specifying a particular attack.

4.1. Data Cleaning and Preprocessing

Six Different attack classes namely Benign, DDOS attack-HOIC, FTP-BruteForce, DoS attacks-GoldenEye, DoS attacks-Slowloris and DDOS attack-LOIC-UDP, each having a particular .csv file were combined to form a single data frame. The dataset was cleaned by removing the null and infinity values embedded within the .csv files. Random subsampling was conducted to sample 1000 records from each class with a total of 6000 records in the dataframe. The dataset was further saved as a new .csv file named 'cleaned_ids2018_sampled.csv'.

4.2. QR Code Image Transformation

The dataset was converted to QR code images using the 'qrcode' and 'os' libraries within Python. Each record of the dataset was converted into QR code images under a common directory with class labels saved as images in a new common directory. The images were resized in the form 1410 x 1410 (width X height). The images were also converted to a one-bit depth form in the standard grayscale form.

5. Proposed Methodology

The Figure below illustrates the path to find the proposed lightweight intrusion detection system QRNetV2.

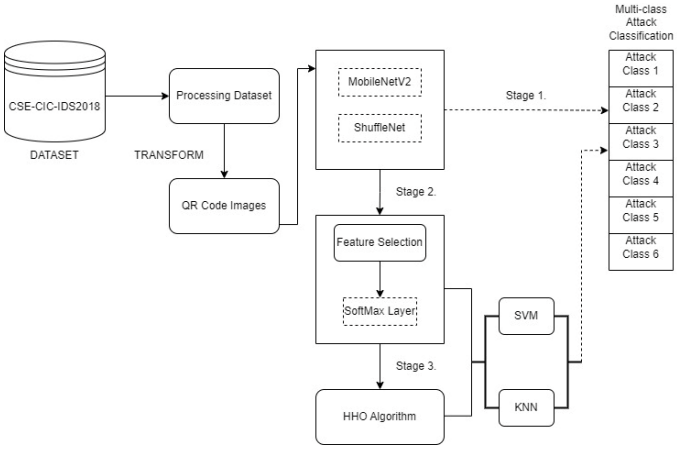

Fig. 1. Proposed Framework

The proposed methodology follows three primary stages with a view to find the best performing lightweight model. The first stage involves passing the QR code images into the MobileNetV2 and ShuffleNet individually. The second stage involves introducing feature selection using the SoftMax layer and conduct classification using ML models such as SVM and KNN. The final stage acts as a continuation to the previous stage i.e. apply the Harris Hawkins Optimization Algorithm (HHO) and again conduct multi-classification using SVM and KNN. A train test split of 70% training and 30% testing is utilised in this proposed framework with an input shape of 224 x 224.

5.1. Hardware requirements

The proposed model was run on a 12th Gen Intel(R) Core(TM) i7-12700H @2.3GB with 16GB RAM. The experiment is run on Python 3.11 with the model implementation done on Jupyter Notebook.

5.2. MobileNetV2

MobileNetV2 is a lightweight and fast model utilized for computer vision tasks such as object detection and classifying images. The model utilizes few computational resources such as memory and processing capacity. A technique known as inverted residuals is adopted by this framework where the number of features or channels are reduced in the beginning followed by their expansion in the conclusion. Such a technique pays heed to utilization of minimal computational resources.

During dimensionality reduction, a challenge lies in retaining the features to reduce the fall in accuracies. Such a challenge is overcome by introducing linear bottlenecks. It utilizes another peculiar technique known as depth wise separable convolution which divides the process of convolution into simpler steps that will speed up the process of classification The model has 53 layers in its standard format.

The model architecture consists of 17 bottleneck layers in each sub-layer containing multiple depth wise separable layers, linear bottleneck, and pointwise convolutions. The architecture diagram of MobileNetV2 is given in the figure below.

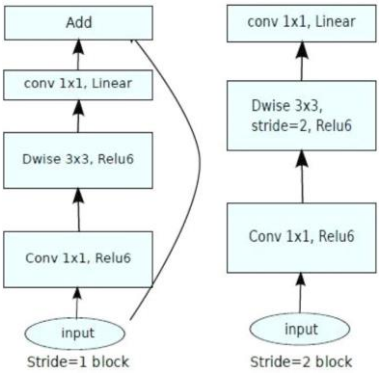

Fig. 2. MobilNetV2 architecture (Sandler et al., 2018).

5.3. ShuffleNet

ShuffleNet model is similar to a MobileNetV2 model in terms of computational resource utilization. It is fast and achieves high accuracy with minimization of resources. Shufflenet uses group convolutions by dividing the input features or channels into groups. The convolution process is applied to each group individually. Such a modification reduces the computational calculation by a major margin.

However, the use of group convolution can limit the information flow among the groups hence the channels are mixed in between the groups after the convolution process collectively. This can ensure the information about the features is distributed evenly across all groups minimizing any loss of information.

A ShuffleNet model has approximately 53 to 101 layers in its architecture. It contains maximum eight residual blocks with three sub layers each accommodating group convolution, residual connection and channel shuffle.

The figure (a) depicts a normal Bottleneck which is used as a depth convolution. (b) indicates the ShuffleNet segment with a 1 x 1 group convolutional layer. (c) depicts the ShuffleNet model with an extra 3 x 3 average pooling layer with stride = 2. Concatenation operation takes place in this part where numerous multi-dimensional arrays are combined in a specific dimension (channel dimension).

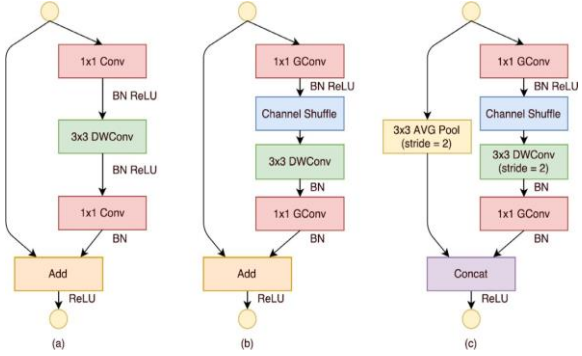

Fig. ShuffleNet CNN architecture (Zhang et al., 2018).

5.4. Harris Hawk Optimization

Harris Hawk Optimization algorithm is a nature-based optimization algorithm that mimics the behaviour of Harris's hawks. The Harris hawk with a scientific name Parubuteo unicinctus is a prey hailing from the southern USA. One unique feature of the hawks is that they hunt for a prey with a strategy known as 'seven kills' strategy. Using the strategy, these hawks hunt in groups using a surprise pounce technique and simultaneously hunt from various locations. The two stragies followed are shown below.

$$X(t+1)=\begin{cases} X_{rand}(t)-r_1\left|X_{rand}(t)-2r_2X(t)\right|, q\geq 0.5 \\ \left(X_{rabbit}(t)-X_m(t)\right)-r_3\left(LB+r_4\left(UB-LB\right)\right), q\leq 0.5 \end{cases} \quad (1)$$

The prey in the above equation is known as the 'rabbit' which is hunt upon at random locations initially. q is the random variable with a value between 0 and 1. Suppose the value of q is less than 0.5 then first hunting strategy is applied otherwise the second strategy is implemented. The above equation represents the first phase i.e., exploration phase. The second phase i.e., exploitation phase is switched with the first phase simultaneously depending in the escape energy of the prey. The energy is given in the form of an equation as shown below.

$$E = 2E_0\left(1-\frac{t}{T}\right) \quad (2)$$

In the equation above E depicts the escaping energy, E_0 is the initial energy state, and T is the maximum number of iterations. If the above resultant is greater than one the algorithm is said to exist in exploration phase. Otherwise, it exists in the exploitation phase.

In this proposed methodology, the number of features is reduced to 500 using PCA to give a best fitness value of -0.615.

5.5. Support Vector Machines

Support Vector Machines is a supervised learning algorithm are used for both classification and regression machine learning tasks. The objective of Support Vector Machines is to find the optimal hyperplane that separates the data points of each class. The support vectors can be defined as points of data that are in close proximity to the hyperplane. The margin is calculated on the basis of support vectors which is the distance between the most adjacent data points and hyperplane.

Since the dataset contains non-linearity in them hence kernel functions are implemented to classify the images. The two kernel function used in this project are Linear and Polynomial Kernel. The linear kernel is known as the base kernel function used for linearly separable data. The equation for a linear kernel is given below.

$$K(x_i, x_j) = x_i \cdot x_j \quad (3)$$

The Polynomial kernel helps in the calculation of similarity in support vectors in the feature space. The equation of the polynomial kernel is as shown below.

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d \quad (4)$$

In the above equation d is the degree of polynomial and c is a constant.

5.6. K Nearest Neighbour

K Nearest Neighbour is a supervised learning algorithms which makes prediction based on the instances and is used for both regression and classification problems. The cluster centres are calculated using a distance measure then then the test sample is predicted based on the neighbouring labels.

The distance measures used in this are Euclidean, Manhattan and Minkowski distances.

Euclidean distance is the distance calculated between two endpoints on a straight line. The formula for Euclidean is given below.

For $p = (x_1, x_2, \ldots, x_n)$ and $q = (y_1, y_2, \ldots, y_n)$.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (5)$$

Minkowski distance measures include both the Euclidean and Manhattan distance based on the p value. If p = 1, then the distance measure is Manhattan in nature. If p =2, then the distance is Euclidean in nature. The formula for the distance measure is given below.

$$D(X,Y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}. \quad (6)$$

Manhattan distance is also known as taxicab or L1 distance that is sum of the difference between the two endpoints. i.e. absolute distance raised to the reciprocal of p value. If p = 2 then the square root of the absolute differences gives the Manhattan distance. The equation for Manhattan distance is given below.

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \quad (7)$$

5.7. Feature Selection

The most important or relevant features are selected using feature selection using the SoftMax layer of MobileNetV2 and ShuffleNet architectures. The SoftMax layer was chosen due to its proximity to the output layer hence giving a more effective and accurate classification. The main aim of feature selection is to identify the most suitable features from trained feature maps to enhance the ability of the model, also aid in reduction of model architecture complexity.

5.8. Hyperparameter Description

The number of epochs can be specified as 16 and number of batches as 32. The optimizer implemented in this proposed framework is Adam (Adaptive Moment Estimation) which combines the benefits of two optimizers, RMSProp and AdaGrad. It provides the feature of adaptive learning rate for various parameters. Each parameter have its own adaptive learning rate utilized during the training phase of the methodology depending on the historical gradients.

The loss function implemented is Sparse Categorical Crossentropy where the target variables are integers instead of categorical target variables. It is majorly implemented in muti-class classification contextual problems. Cross entropy score is calculated using the probability of the predicted class and the true label. The formula for calculation of cross-entropy loss is the following.

$$Loss = -\log(p_y) \quad (8)$$

In the above equation p_y is the predicted probability for the class y which is known to be true. The total average loss is defined by taking an average of the discrete losses of the individual samples. The equation for the former is as follows.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\log(p_{yi}) \quad (9)$$

In the above equation, p_{yi} is the probability predicted for the true class of the 'i' sample and N is the number of samples in the batch.

The activation function implemented in the framework is the ReLU i.e. Rectified Linear Unit which is widely used in many neural networks. It performs comparatively superior to sigmoid and hyperbolic tangent functions. The function is given as follows.

$$ReLU(x) = \max(0, x)$$

If the input is positive the output is the input itself otherwise it is zero. Non linearity is introduced used for capturing complex patterns in the network.

6. Evaluation Metrics

The model is evaluated on the test data set using a confusion matrix and using specific evaluation metrics such as accuracy, sensitivity, specificity, precision and F1 score. The equations for the evaluation metrics are given below.

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

$$Specificity = \frac{TN}{TN + FN}$$

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

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

$$F1 - score = \frac{2\,.TP}{2\,.TP + FP + FN}$$

7. Results

The first stage with direct passage of images into MobileNetV2 and ShuffleNet shows great promise in terms of both accuracy and training time. The MobileNetV2 model takes around 42 minutes of training time and gives an accuracy of 96.60%. The ShuffleNet model also shows a promising accuracy of 88% with a training time of merely 30 minutes.

The second stage involves feature extraction with the help of softmax layer and additionally conducting muti-class classification through SVM and KNN respectively. As shown in Table 2, The MobileNetV2 model performs the best with SVM embedded in a linear kernel with an accuracy of 86%. Whereas in Table 3, ShuffleNet model performs the maximum with an SVM embedded in a quadratic or polynomial kernel with an accuracy of 66%.

During the third stage, it is unfortunate to see any growth in the accuracy in spite of the inclusion of an HHO algorithm with the best performing model to have an accuracy of 64% only.

Table 1
Performance results obtained with CNN models.

CNN Model	Accuracy	Sensitivity	Specificity	Precision	F1_score	Train Time (minute)
MobileNetV2	**0.96**	**0.97**	**0.96**	**0.98**	**0.96**	**42**
ShuffleNet	0.88	0.88	0.83	0.87	0.86	30

Table 2
Classification results of feature selection of MobileNetV2 CNN model with SVM and KNN.

Algorithm	distance & kernel function	Accuracy	Sensitivity	Specificity	Precision	F1_score
SVM	Linear	0.86	0.86	0.96	0.65	0.86
	Quadratic	0.85	0.85	0.88	0.87	0.85
kNN	Euclidean	0.84	0.84	0.83	0.95	0.84
	City Blok	0.85	0.85	0.82	0.89	0.85
	Minkowski	0.84	0.84	0.96	0.8	0.84

Table 3
Classification results of feature selection of ShuffleNet CNN modelwith SVM and KNN.

Algorithm	distance & kernel function	Accuracy	Sensitivity	Specificity	Precision	F1_score
SVM	Linear	0.62	0.62	0.66	0.65	0.64
	Quadratic	0.66	0.66	0.65	0.64	0.67
KNN	Euclidean	0.61	0.61	0.64	0.62	0.63
	City Blok	0.62	0.63	0.65	0.6	0.64
	Minkowski	0.62	0.62	0.61	0.65	0.63

Table 3
Classification results of HHO algorithm with SVM and KNN

Algorithm	Kernel &distance function	Accuracy	Sensitivity	Specificity	Precision	F1_score
SVM	Linear	0.64	0.64	0.60	0.63	0.63
	Quadratic	0.64	0.62	0.64	0.59	0.65
KNN	Euclidean	0.62	0.62	0.60	0.65	0.63
	City Blok	0.62	0.64	0.66	0.65	0.64
	Minkowski	0.62	0.63	0.61	0.65	0.65

8. Discussion

Table 5
Comparison of data sets and features used in cyber security.

Authors	Methods	Datasets	No. of class	Accuracy
2020,Farhan et al.(Farhan et al., 2020a)	Deep Neural Network (DNN)	CSE-CIC-IDS2018	7	90.25
2020,Farhan et al.(Farhan et al., 2020b)	Binary Particle Swarm Optimization (BPSO)	CSE-CIC-IDS2018	7	95.00
2021,Cil et al.(Cil et al., 2021)	Deep Neural Network (DNN)	CICDDoS2019	4	94.57
2023, Noever et al.(Yusuf et al., 2023b)	MobileNetV2, ShuffleNet, HHO	CSE-CIC-IDS2018	6	95.89
2022, This paper	MobilNetV2	CSE-CIC-IDS2018	6	96.60

It can be confirmed from the above proposed methodology the outperforming model is the MobileNetV2 which is named further as QRNetV2. The training time is merely 42 minutes that is impressive for a 6 class classification with an accuracy of 96.60%. The proposed methodology utilizes the lightweight architecture of a MobileNetV2 in comparison to the other state-of-the-art models mentioned in Table 5. The computationally minimized proposed framework outweighs the traditional DNN and BPSO techniques (Farhan et al., 2020a). The dataset used by (Cil et al., 2021) in their paper with proposed model as DNN is specific to only DDoS attacks making it inferior to the proposed model utilizing six different classes.

It is notable from Table 5, that the hybrid technique implemented as the third stage in the proposed methodology (Yusuf et al., 2023b) shows an accuracy of 95.89% which is 0.71% less than the much lightweight singular model i.e. QRNetV2. The study conducted lack any structural foundation to support their evaluation findings. Hence, the proposed model QRNetV2 with all foundations applied outperforms the rest.

Overall, it is safe to say that this paper in all aspects performs to its optimal capability compared to the previous works where they either lack in terms of accuracy of structural foundations as stated in the research works published, Hence the QRNetV2 model passes the standard benchmark and shows reliability along with efficiency in most aspects.

9. Conclusion

A Network Intrusion Detection System was developed using lightweight deep learning model, QRNetV2 which is computationally inexpensive utilizing minimum resources. The proposed methodology was carried out in three stages, The first stage showed remarkable performance by the outperforming model i.e. QRNetV2 and ShuffleNet showed an accuracy of 96.6% and 88% with training time 42 and 30 minutes respectively. The second stage with an inclusion of feature selection and further classification with SVM and KNN showed an accuracy of 86% and 66% with MobileNetV2 and ShuffleNet respectively. The final stage showed no further increase in accuracy after deploying the HHO algorithm with a mere accuracy of 64%. The adaptability of QRNetV2 shows a promising approach of its deployment in real time environment with better accuracy and training time.

The future work includes, experimenting different lightweight models with various other optimization algorithms to improve the accuracy and decrease the training time achieved. New novel techniques such as ensemble and hybrid models can be experimented for future promising results. Furthermore, Explainable AI can be used to interpret the results and allowing to understand the feature importance in a particular dataset.

References

[1] K. Riad, T. Huang, L. Ke, A dynamic and hierarchical access control for iot in multi-authority cloud storage, J. Netw. Comput. Appl. 160 (2020) 102633.

[2] Performance Comparison and Detection Analysis in Snort and Suricata Environment | Wireless Personal Communications (springer.com)

[3] Sharma B, Sharma L, Lal C. Anomaly detection techniques using deep learning in IoT: A survey. In: 2019 international conference on computational intelligence and knowledge economy. 2019, p. 146–9. http://dx.doi.org/10.1109/ICCIKE47802.2019.9004362.

[4] Rishnaveni, S., Vigneshwar, P., Kishore, S., Jothi, B., Sivamohan, S.: Anomaly-based intrusion detection system using support vector machine. In: Dash, S.S., Lakshmi, C., Das, S., Panigrahi, B.K. (eds.) Artificial intelligence and evolutionary computations in engineering systems, pp. 723–731. Springer, Singapore (2020)

[5] Alaca, Y., & Çelik, Y. (2023). Cyber attack detection with QR code images using lightweight deep learning models. *Computers & Security*, *126*, 103065.

[6] Ahmad, M., Riaz, Q., Zeeshan, M. *et al.* Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *J Wireless Com Network* **2021**, 10 (2021). https://doi.org/10.1186/s13638-021-01893-8

[7] Karthik, M.G., Krishnan, M.B.M. Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks. *J Ambient Intell Human Comput* (2021). https://doi.org/10.1007/s12652-021-03082-3

[8] Kayode Saheed, Y., Idris Abiodun, A., Misra, S., Kristiansen Holone, M., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, *61*(12), 9395–9409. https://doi.org/10.1016/j.aej.2022.02.063

[9] Ahmed, A.A., Jabbar, W.A., Sadiq, A.S. et al. Deep learning-based classification model for botnet attack detection. *J Ambient Intell Human Comput* 13, 3457–3466 (2022). https://doi.org/10.1007/s12652-020-01848-9

[10] Su, J., He, S., & Wu, Y. (2022a). Features selection and prediction for IOT attacks. *High-Confidence Computing*, *2*(2), 100047.

[11] Mukherjee, I., Sahu, N.K. & Sahana, S.K. Simulation and Modeling for Anomaly Detection in IoT Network Using Machine Learning. *Int J Wireless Inf Networks* 30, 173–189 (2023). https://doi.org/10.1007/s10776-021-00542-7

[12] Hazman, C., Guezzaz, A., Benkirane, S. et al. IIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning. *Cluster Comput* **26**, 4069–4083 (2023). https://doi.org/10.1007/s10586-022-03810-0

[13] Elnakib, O., Shaaban, E., Mahmoud, M. *et al.* EIDM: deep learning model for IoT intrusion detection systems. *J Supercomput* **79**, 13241–13261 (2023). https://doi.org/10.1007/s11227-023-05197-0

[14] Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023a). Anomaly based network intrusion detection for IOT attacks using Deep Learning Technique. *Computers and Electrical Engineering*, *107*, 108626. https://doi.org/10.1016/j.compeleceng.2023.108626

[15] Silivery, A. K., Rao Kovvur, R. M., Solleti, R., Kumar, L. S., & Madhu, B. (2023, December). A model for multi-attack classification to improve intrusion detection performance using deep learning approaches. *Measurement: Sensors*, *30*, 100924. https://doi.org/10.1016/j.measen.2023.100924

[16] Wang, X., Wang, Y., Javaheri, Z., Almutairi, L., Moghadamnejad, N., & Younes, O. S. (2023b). Federated deep learning for anomaly detection in the internet of things. *Computers and Electrical Engineering*, *108*, 108651.

[17] Al-Wesabi, F. N., Mengash, H. A., Marzouk, R., Alruwais, N., Allafi, R., Alabdan, R., Alharbi, M., & Gupta, D. (2023). Pelican Optimization algorithm with Federated Learning Driven Attack Detection Model in internet of things environment. *Future Generation Computer Systems*, *148*, 118–127. https://doi.org/10.1016/j.future.2023.05.029

[18] Idrissi, M. J., Alami, H., El Mahdaouy, A., El Mekki, A., Oualil, S., Yartaoui, Z., & Berrada, I. (2023). Fed-Anids: Federated Learning for Anomaly-based network intrusion detection systems. *Expert Systems with Applications*, *234*, 121000. https://doi.org/10.1016/j.eswa.2023.121000

[19] Muna, R. K., Hossain, M. I., Alam, Md. G., Hassan, M. M., Ianni, M., & Fortino, G. (2023). Demystifying machine learning models of massive IOT attack detection with explainable AI for sustainable and Secure Future Smart Cities. Internet of Things, 24, 100919. https://doi.org/10.1016/j.iot.2023.100919

[20] Sharma, B., Sharma, L., Lal, C., & Roy, S. (2024). Explainable artificial intelligence for intrusion detection in IOT Networks: A deep learning based approach. Expert Systems with Applications, 238, 121751. https://doi.org/10.1016/j.eswa.2023.121751

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-.C., 2018. Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 .

Zhang, X., Zhou, X., Lin, M., Sun, J., 2018. Shufflenet: an extremely efficient convolu-tional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6 84 8–6 856 .

LITERATURE SURVEY. Lightweight Intrusion Detection System using QRNetV2.

Ref. No.	Objective	Problem Statement	Methodology	Advantages	Disadvantages	Software Tool used	Performance Measure
1.	1. Construct a Network Intrusion System (IDS) using feature clusters. 2. Conduct feature selection and extraction to select the feasible features required. 3.Create feature clusters in terms of flow, Transmission Control Protocol (TCP) and Message Queuing Telemetry Transport (MQTT) by using the features of the dataset.	1. Security of data in IoT threatened due to unauthorized transmission of data between connected devices. 2.Recurrent Existence of malicious packets in connected networks. 3.Absence of a monitoring system to identify and classify different types of IoT attacks.	Acquired the public dataset, 'UNSW-NB15' which is pre-processed to clean the data and resolve the problems such as data imbalance, data type mismatch and missing/null values. Six feature clusters are created according to different network layers, to which classification algorithms such as Random Forests (RF), Support Vector Machine (SVM) and Artificial Neural Networks (ANN) are applied in both binary and multi-class to classify the IoT attack.	Issues such as over-fitting, curse of dimensionality and an imbalance in the dataset are eliminated. Higher accuracy is achieved and requires less training compared to the standard supervised machine learning algorithms.	Lack of appropriate features related to other IoT protocols. Unable to identify and classify unknown IoT attacks. Feature clusters might not enhance the representation of minority instances.	1.Google Collab with pre-installed libraries and GridSearchCV for hyperparameters. 2.Intel (R) Xeon (R) CPU E3-1285 v6 @ 4.10 GHz with 8 CPUs, 4 cores per CPU and 2 threads per core. 3.The system runs Ubuntu 18.04.1 LTS and has 64 GB of RAM.	Binary Classification: 1. RF – 98.67% 2. SVM -97.69% 3. ANN -94.78% Multi-Class Classification: 1. RF – 97.37% 2. SVM -95.67% 3. ANN -91.67%
2.	1.Detect a new model using Random Forests and synthetic minority over-sampling technique to classify the IoT attacks. 2.Evaluate the NSL-KDD and N-BaIoT datasets. 3.Improve the accuracy of the machine learning model.	1. Various IoT breaches lead to security lapses such as consumption of server resources, saturating link bandwidth, etc. 2.An absence of secure monitoring systems and lack of protection. 3.Resource constrained nature of IoT devices leave them vulnerable.	The methodology consists of four phases such as data collection, data pre-processing, splitting of data and attack detection of using the RF-SMOTE model. The NSL-KDD and N-BaIoT datasets includes different attacks and protocol types. The model used is random forests along with the SMOTE technique that enhances the representation of the minority classes and make their decision region general. This technique balanced the normalized data vectors giving higher accuracy in classification.	Improvement in classification accuracy is achieved. Helped to include the minority classes and make the model a non-biased model. Provided better visibility across IoT networks.	Overfitting is possible in terms of noisy data or outliers by which the SMOTE method by create synthetic instances. Creates complexity in dataset by increasing the size of the dataset.	1.Simulated using Anaconda navigator and python 3.6 software with windows 10 operating system,123 GB RAM, 1TB memory, 22GB 2070 Ti GPU and i9 processor.	1.NSL-KDD dataset: 98.31% 2. N-BaIoT dataset: 99.98%

3.	1. Develop a ML-IDS system for detecting IoT network attacks. 2.Apply supervised machine learning algorithm to develop the IDS system. 3.Conduct feature scaling on the dataset and reduce dimensionality.	1. Privacy and security issues due to the limitation in energy and scalability of IoT devices. 2.Need for IDS due to the vigorous increase in devices connected to the internet. 3.Constraints of IoT devices in terms of memory capacity.	Conduct feature scaling using min-max concept of normalization the UNSWNB15 dataset. Output of the feature scaling is fed to the algorithm, Principal Component Analysis (PCA) is performed for dimensionality reduction. The resultant dataset is trained by XGBoost, CatBoost, KNN, SVM, QDA and NB classifiers.	Achieved the milestone of building an intelligent IDS system. Communication overhead is reduced. Foreign key is not required in encryption methods for the security of IoT networks.	Absence of a dynamic and adaptable nature in the IoT environment. Unable to exploit large quantities of data.	1.Windows 10 IoT Core Services.	1.XGBoost: 99.99% 2.CAT Boost: 99.99% 3.KNN: 99.98% 4.SVM: 99.98% 5.QDA: 99.97% 6.NB: 97.14%
4.	1.Assemble a deep learning model for IoT intrusion detection. 2.Identify zero-day botnet attacks in real time. 3.Identify botnets accurately and in an efficient manner.	1.Botnets gives an opportunity to hackers to seize control of multiple systems. 2.Botnet can turn multiple computers into zombie systems which leaves it vulnerable to attacks. 3.Compromise of personal information security through IoT.	A Deep Neural Network model techniques are using to identify the botnet attacks. The CTU-13 dataset from the Botnet Capture Facility Project undergoes feature selection and normalization. The output data is fed ads input to the neural networks and testing phase id deployed.	A decrease in mean square error is reflected during the performance measures. An improvement in accuracy is visible.	An increase in mean square error can take place when the validation dataset overfits the training data. Identification of random noise due to overfittig.	1.Matlab 2016 version 9 using 10,000 randomly selected flows for dataset training, validation and testing. 2.Model developed using Tensorflow framework.	1.DNN: 99.6%
5.	1.Analyze Different machine learning algorithms such as random forest and gradient boosting machine (GBM) to classify the IoT attacks. 2.Conduct model evaluations to find the most accurate algorithm. 3.Use performance indicators to conduct model evaluations.	1. Rapid increase in different types of IoT network attacks. 2.Property and trust of users and developers compromised due to the attack. 3.Important data lost due to IoT network attacks.	IoTID20 dataset acquired from IEEE Dataport are sent for data preprocessing. Gradient boosting machine ensemble in python is applied to select the most important features. To the resultant dataset machine learning algorithms such as Logistic regression, Linear Discriminant analysis, Quadratic Discriminant analysis, Naïve Bayes, Random Forest, Decision Tree and Gradient Boosting Machine are applied.	Best accuracy and area under curve (AUC) provided by random forest, decision tree and GBM models. Random Forest are more accurate on classifying unseen attacks.	Hyperparameters not tuned well enough to improve the model. GBM model takes relatively long training time. Feature selection in classification models not simplified.	Not specified	1.Decision Tree: 97.8% 2.Random Forest: 97.84% 3.GBM: 96.36%

6.	1.Predict anomalies from data collected on IoT network attacks to block such attacks in the future. 2.Apply Machine learning classification algorithms to the whole dataset. 3.Apply the same classification algorithms after eliminating the data points having binary values, 0 and 1.	1.Increased use of IoT infrastructure and automation in the present world. 2.Increased failure of nodes, increase in threats, spying and abnormalities. 3.Maximum probability of attacks due to the 24/7 work hours of IoT system.	The dataset is acquired from Kaggle public repository. Data cleaning conducted followed by categorical data transformation using the encoding technique such as dummy encoding and one-hot encoding. Classification algorithms such as Logistic Regression, Naïve Bayes, Decision Tree, Random Forests and Artificial Neural Network (ANN) are applied on the dataset as the first case. In the second case, all 0's and 1's are removed from the dataset and the above classification algorithms are applied again and compared.	In case 1, only DOS class was mislabelled, and other classes were correctly classified accurately for most of the algorithms. In case 2, the accuracy had a proficient increase and all the classes were classified accurately.	Does not consider the aspect of operation of micro-services in the IoT networks causing variations in the anomalies. Unable to find a correlation between the networks and attacks as well as precautions to prevent such attacks.	Not specified	Case 1: 1.LR, DT and RF: 99.4% 2.ANN: 99.37% 3.NB: 94% Case 2: 1.LR, ANN: 99.99% 2.DT, RF: 100% 3.NB: 94%
7.	1.Develop a novel intrusion detection system with ensemble learning called IDS-SIoEL. 2.Combine ensemble learning techniques and feature selection techniques to construct an efficient model. 3.Evaluate the model on various IoT attack type datasets using the GPU.	1. A smart city needs to provide improved services to customers. 2.Rapid increase in integration of information, communication technologies and devices throughout a network. 3.Real times detection of IoT threats, immense volume and time restrictions have posed a challenge.	The datasets IoT-23, BoT-IoT and Edge-IIoT undergo PCA for dimensionality reduction. The resultant categorical features are encoded using catboost Encoder. The reduced features undergo feature selection methods such as mutual information, Boruta and Pearson correlation. The best features selected is then passed through an AdaBoost model. The evaluation metrics are applied on the model.	The accuracy increases as the number of trees increases. Numerous feature selection methods used to enhance the data quality. The performance time by adapting the model on GPU and to detect the IoT attacks are a good efficient value.	Lacks multi-classification for detecting multiple threats to IoT network at a time. More enhanced model such as a deep learning model is absent. Dependency on weak learners might lead to suboptimal results.	1.Kaggle machine with 15 GB of GPU memory and 64-bit operating system. 2.Jupyter Lab and Python 3.9.7 are used in model building.	1. Edge-IIoT: 100% 2.BoT-IoT: 99.99% 3.IoT-23: 99.98%

8.	1.Propose an enhanced anomaly-based Intrusion Detection Deep learning Multi-class classification model (EIDM). 2.Customize the deep learning models to identify and classify six classes of network traffic behaviour. 3.Conduct an extensive comparative study on the basis of accuracy and efficiency.	1. Due to the omnipresent nature of IoT, it is susceptible to various cyber threats. 2.There are more than 13.8 billion devices connected to the internet worldwide and this figure is expected to increase in the future. 3.Corruption of encrypted data calls for an enhanced intrusion detection system.	The dataset was cleaned using the SMOTE technique. Six classes of network traffic such as: Normal, Distributed Denial of Service (DDOS), Slowloris, Slowhttptest, Hulk and GoldenEye from the CICIDS2017 dataset are used in this model. Initially custom simple learning deep learning models are used for classification such as Multi-layer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and a combination of CNN and LSTM. Later EIDM was used for classification by considering all the 14 attack types of the CICIDS2017 dataset. It can classify all the classes without grouping close classes of similar features.	EIDM achieves high precision in classification of all attacks. It includes advanced layers that appropriately differentiate between targeted classes. Time cost is better than most models. It is a good solution for both accuracy and complexity.	Although the CICIDS2017 dataset has a large set of data, some of them were unlabelled, corrupted and contains duplicated data. The classifiers' accuracy is constrained by the total number of classes and samples per class. Simple models find it hard to attain high accuracy. More complex the model, the running time increases since time is a crucial factor in IDS system.	1.Machine equipped with Intel Core i7-8700 LGA1151 @ 3.70 GHz, 16 GB DDR4 RAM 2400 MHz, and Nvidia RTX 2080 Ti 11 GB GDDR6 PCIe. 2.Anaconda3 and python language were used for implementation. 3.Deep learning models were developed, trained and tested using Keras package.	1.MLP: 88.7% 2.CNN: 93.7% 3.LSTM: 96.4% 4.CNN+LSTM: 97% 5.EIDM: 99.48%
9.	1.Develop a novel anomaly-based IDS system. 2.Conduct a filter-based feature selection to represent the highly correlated features. 3.Generate synthetic data of minority attacks to solve the issue of class imbalance.	1.Increase in the heterogeneous physical devices connected to the network has brought numerous security challenges. 2.An immeasurable amount of data is created due to the number of devices connect to the internet leaving it vulnerable to IoT attacks. 3.The gadgets connected to the network have limited storage capacity and computational power.					

10.	1.Propose novel deep learning models for multiple IoT attack classification. 2.Increase the accuracy, detection rate and decrease the false alarm rate. 3. Models must self-learn the feature classify the attacks as multi-attack classification.	1.With rapid growth of the use of the internet, data is exposed to various threats. 2.Any loopholes within the computer system can lead to dire consequences. 3. A high occurrence of false alerts that IDS experiences are recognized.	Recurrent neural network (RNN), Long Short-Term Memory Recurrent Neural Network (LSTMRNN), and Deep Neural Network (DNN) are the proposed models. The models go through pre-processing, feature extraction, training and testing phases on the KDD'99, NSL-KDD and UNSW-NB15 datasets. Pre-processing is done through numericalization and normalization. The evaluation is done using a comparative study.	Overall performance of the proposed models in terms of detection rate and accuracy is satisfactory. The comparison of the models with simple deep-learning models show higher accuracy. The models outperform on KDD'99 and NSL-KDD datasets.	The model is unable to provide accurate results on UNSW-NB15 dataset. The model does not accurately classify the attacks from the UNSW-NB15 dataset.	1.Required environment is jupyter notebook. 2.Packages required are anaconda, pandas, numpy, scikit-learn, and Matplotlib.	KDD'99: 1.RNN: 98.73% 2.LSTM:96.85% 3.DNN:98.20% NSL-KDD: 1.RNN: 98.68% 2.LSTM:95.69% 3.DNN:98.95% NSW-NB15: 1.RNN: 69.24% 2.LSTM:80.39% 3.DNN:80.39%
11.	1. Integrate federated learning and deep learning to develop a novel intrusion detection system. 2.Retain the information on localized IoT devices for model training. 3.Resolve the issues of privacy, latency, bandwidth and connectivity.	1.Privacy is an apprehensive concern due to an increase IoT connected devices. 2.Regular internet do not provide any protection of data being transmitted through the network from one device to another. 3.Security threats are an alarming challenge leading to an increased concern in developing and IDS system.	Using centralized learning, the IoT data is uploaded to the centralized cloud servers with ML approaches. The dataset used is IoTBoTNet2020. In Federated learning, the local data is processed using the local model mutual inclusion with deep learning neural network (MI-DNN) and this result is uploaded on the server for aggregation. The aggregated models are sent back to the devices. The average weights shared between the federated server and IoT device is used in the training phase.	The model shows good accuracy, F1-score, true positive and negative rates. The ROC curve (receiver operating characteristic curve) obtained is better than the other federated deep learning models. The model refines the security of the data by sharing of global models and centralization.	Uses a basic deep learning model in combination with federated learning. Requires continuous communication between the central server and localized devices. Overhead can be generated. During modification of model the data is exposed to threats and attacks.	Python version (3.6)	FMI-DNN: 99.4% Error rate:0.142

12.	1.Propose a federated learning model to help identify malicious attacks in IoT network. 2.Propose a pelican optimization algorithm and integrate it with federated learning to give optimized results. 3.Evaluate the model and examine the results on different measures.	1.Low-level devices in an IoT network have restricted computational power. 2.Accumulation of unprotected data in cloud leads to an increase in unauthorized access. 3.Centralized data repositories attract malicious IoT attacks.	A novel Pelican Optimization Algorithm with federated learning Driven attack detection (POAFL-DDC) using the TON_IOT dataset was proposed to identify the IoT attacks. The model is operated on decentralized on-device data. The update weight is substituted to the central FL server, the data is transferred to the local devices and use FL on DL model. The attack detection is done using POA with Deep Belief Network (DBN) model.	Protects data on end devices ensuring privacy and encryption. The accuracy of the DNN model has been boosted. The attack detection results of the POAFL-DDC technique are fairly good.	Tuning the parameters is quite challenging as the problem domain differs. The POA might lack the features needed to handle the complication in the IoT attacks.	Not specified	1.MIM: 99.72% 2.Ping DDoS: 99.41% 3.Query Flood: 99.37% 4.SYN DDoS: 99.34%
13.	1.Detect IoT intrusions using federated learning to tackle the issues associated with privacy concerns with centralized models. 2.Compute intrusion score using various anomaly detection models. 3.Reconstruct the error of normal traffic using models including autoencoders, variational autoencoders and adversarial autoencoders.	1.Providing efficient cybersecurity has become an eminent concern in the booming IoT industry. 2.State-of-the-art challenges such as privacy concerns and regulatory restrictions are appalling. 3.Increase in the number of IoT network attacks has been recorded.	The paper proposes a federated anomaly intrusion detection system (Fed-ANIDS). The learning technique consists of four main components such as Global model initialization, Local training, Model aggregation, and Model dissemination. Fed-ANIDS utilizes the FedProx algorithm to update the local models. The features are extracted from the datasets: USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018.	The model surpass other GAN based models for each dataset. Autoencoders are more effective in identify any type of IoT attacks. Model involving autoencoders are light, simple and efficient in term of computation.	Lacks domain generalization to improve the performance and efficiency of the model across various network domains. Other FL algorithms are not explored in this study.	1.Python and machine learning libraries such as PyTorch, Numpy and Pandas are used to perform all experiments. 2.The study was carried out using African SuperComputing Center HPC service.	1. USTC-TFC2016: 97.66% 2.CIC-IDS2017: 76.28% 3. CSE-CIC-IDS2018: 80.77%

14.	1.Tackle large scale threats and attacks on IoT devices with the help of explainable AI. 2.Combine the algorithm, Extreme Gradient Boosting (XG-Boost) classifier and Explainable Artificial Intelligence (XAI) to create an IDS system. 3.Construct an effective model with high accuracy.	1.Use of various IoT applications in a smart city has raised challenges of security and privacy. 2.Existing classification systems only classify limited IoT attacks. 3.Lack of accountability in the models at presents exists hence making it quite difficult to understand the decision making process.	The dataset used in the model creation is IoTID20. Data preprocessing involves label encoding, and replacement of infinite values and missing values. Feature selection is apply using the PCA algorithm followed by oversampling using the SMOTE technique. On data split the XG Boost algorithm is applied which sequentially creates decision trees. Later cross-validation is done, and then explainable AI is used on the detection model to make the results understandable. LIME, ELI5 and SHAP are applied to the models.	The extreme gradient boosting machine provides higher accuracy, surpass the overfitting issues and parallelization of the detection process. The model is built on a centralized data repository.	Explainable AI might oversimplify the interpretations of the model. It may lead to the exposure of centralized data leading security breaches.	1.Used 107.72 GB of disk space and 12 GB RAM to construct the model in Google Collab. 2.Programming was done using Python and libraries such as Numpy, Pandas, Matplotllib, sklearn and XAI packages.	XG Boost: 86%
15.	1.Build two unique deep-learning models for intrusion detection in an IoT network. 2.Apply Explainable Artificial Intelligence to explain the detection model. 3.In addition for better understanding apply sophisticated extensions of Explainable artificial Intelligence.	1.Detection of malicious attacks in IoT network due to rapid growth of big data in the IoT field. 2.Challenges such as computation of big data, storage, and cyber security in IoT networks exists. 3.Signature-based IDS are ineffective due to the requirements of known knowledge of attacks.	The datasets used are NSL-KDD and UNSW-NB15 in the evaluation of models. Data preprocessing is done using label encoding and one hot encoding. Min-Max normalization is used to normalize the data. Feature selection is done using correlation-based filter, wrapper, and embedded methods. The DNN and CNN models go through training and testing phases. The prediction is interpreted using XAI with surrogate model LIME and SHAP to understand the models better.	Achieved highest accuracy in less training time. The 2D-CNN model attained the highest accuracy and outperformed other models. Explanation of prediction is very efficient using XAI. Decrease in computational cost due to dimensionality reduction	Dataset has an imbalance in classes due to the high number in the majority classes. It does not contain GANs to resolve data imbalance.	TensorFlow library is used for developing the model after it uploading in Google Collab.	1.DNN: 80% 2.1D CNN: 80% 3.2D-CNN: 81%