Enhanced Intrusion Detection for Internet of Things Using Hybrid Algorithms

Arpitha S,
Dept of CSE, UVCE,
Bangalore, India,
arpithas70@gmail.com

R Tanuja,
Dept of CSE, UVCE,
Bangalore, India
tanujar.uvce@gmail.com

ABSTRACT

As a foundational technology in the security net environment, detection of network intrusions has a great deal of attention and use. Network Intrusion Detection (NID) still has difficulties when it comes to installing on devices with limited resources, despite the tremendous efforts of research & advanced technologies. We provide a lightweight detection of intrusion technique based on knowledge of extraction is called Lightweight Neural Network (LNet), which strikes the balance between efficiency and accuracy by concurrently lowering computing costs and model storage. To be more precise, we stack DeepMax blocks to create the LNet after carefully designing the DeepMax blocks to extract compressed representation effectively. Additionally, in order to compensate for the lightweight network's performance decrease, we apply batchwise knowledge of oneself distillation to regularise training consistency. Our suggested Lightweight Neural Network (LNN) and XGboost methodology is shown to be successful on Allflowmeter_Hikari datasets through experiments, outperforming the existing methods with less compute burdens and fewer parameters.

Keyword - *Distillation of self-knowledge, deep learning, intrusion detection, lightweight networks, Internet of Things.*

INTRODUCTION

It is challenging to assess the extent of advancements made by malicious detection addresses in the Intrusion Detection Systems (IDSs) area. Finding a reliable dataset to compare is a challenge for machine learning-driven intrusion detection systems (IDSs), which need training using existing datasets.

A few of the things that hinder the process of comparing datasets include inadequate method documentation [1], a lack of a comparison method [2], and the absence of critical aspects like ground-truth labelling, publicly accessible traffic, and real-world environment data. Furthermore, only a small number of datasets accurately reflect the fact that network traffic is primarily encrypted these days to preserve security and privacy. The dataset is a crucial component in the development of IDS models based on machine learning. The first step in the procedure is to gather internet traffic, either in the form of packets or flows. Subsequently, the recorded traffic is assembled into a particular kind of data that has attributes associated to networks, such as labelling. Fig. 1 depicts a generic machine learning-based IDS. An essential step for the dataset is labelling. Managing ground truth is extremely difficult, particularly when specialists are unable to identify when the traffic is benign or malicious. Researchers employ synthetic traffic for this reason. This suggests, nonetheless, that the created traffic is not typical of the actual world. To put it briefly, gathering traffic is the first step in creating a dataset, and the last step is preprocessing. A labelled dataset is the end product of the preprocessing stage. Every data point is categorised as benign or malevolent. The

file provides tabular data in binary form (IDX file) or human-readable format (CSV file). The dataset may be benchmarked based on the quantity of malicious activity or false alarms discovered. The current datasets are not realistic enough to serve as the foundation for developing a complete model for the identification of novel attacks, nor do they contain consistently encrypted traces. The majority of the research that has been done so far using encrypted traffic is concentrated on various domains, such traffic analysis and categorization [3]. Despite the existence of this research [4], this dataset is not accessible to the general public because of data sensitivity.

Benchmark datasets provide an essential basis for evaluating and comparing the quality of different intrusion detection systems. Based on the detection algorithms employed, there are three types for intrusion detection systems (IDS): hybrid, anomaly-based, and signature-based. Each of these IDS types evaluate their systems using the deprecated KDD99 dataset. The signature-based strategy focuses on establishing automatic signature production, whereas the anomaly-based approach focuses on spotting an aberration from the legitimate profile [5–6]. The signature-based type recognises and tries to match against the signatures database using a pattern-matching technique. An alert is triggered when the signature of an attempted attack is matched. The most accurate and least likely to cause false alarms is the signature-based kind; nevertheless, it is not able to identify unknown threats. The ratio of alarms that are false is still large, even though the anomaly-based type may be able to identify unknown assaults by comparing anomalous traffic with regular traffic. In this study, we describe a tool and specifications for creating a new dataset in a real-world setting through the generation of encrypted network traffic. We are contributing in two ways. Firstly, we provide new specifications for building new datasets. Secondly, we build a novel intrusion detection system dataset that includes encrypted traces of network activity. The dataset has attacks labelled, including probing and brute force login. The ground-truth data, background traffic, and packet traces with message are all supplied.

This paper is organised as follows: We provide relevant literature in Section II; we build the system description and outline the optimisation issue in Section III. The two-phase alternate optimisation strategy is developed in Section IV, and its efficacy is assessed in Section V. It is eventually concluded in Section VI.

LITERATURE SURVEY

IoT device security is highly vulnerable because of the rise in cyberattacks. With the combined use of machine learning algorithms for the identification and detection of various assaults, the state of the art offers several options for their prevention. Some of the work in this direction is covered in this section.
In [7], an ensemble-based system for identifying intrusions was developed utilising a variety of machine learning classification algorithms, including Decision-Tress (DT0, J48, and Support Vector Machine-(SVM). The Knowledge Discovery in Databases (KDD99) intrusion detection dataset's nine most pertinent and significant attributes were chosen via particle swarm optimisation. The results of the suggested model showed a lower of 0.9% and a greater accuracy of 90%. In [8], a different hybrid IDS model built on NB and SVM was introduced. For this investigation, the real-time historic log dataset underwent preprocessing and normalisation. The suggested model yielded 95% precision and accuracy after improvement. Studies show that incorporating session-based features improved the classifier's performance. In order to detect attack traffic, an evaluation of the performance of many traditional ML algorithms on various ID-based datasets has been carried out in [9]. Three machine learning techniques—SVM, K-Nearest Neighbours (KNN), and DT—were used after each set of data (CICIDS2018, UNSW-NB15, ISCX2012, NSLKDD, and CIDDS001) were normalised. DT produces detection accuracy rates among 99 & 100% for all datasets, outperforming other classifiers in the process. Another work that uses the NSL-KDD dataset to develop an IDS using the classification algorithm RF is given in [10]. The z-score for the tree depth was determined by taking the entropy score and the Gini-index into account. The Boruta approach was employed to pick 34 significant characteristics from the dataset.

The suggested model [11] yielded 99% detection accuracy for assaults. Using SVM, a lightweight intrusion detection system (IDS) has been created to identify unknown and attempted misuses in IoT networks. Several tests were carried out in this work to detect DDoS assaults using various functions, including linear, polynomial, and radical base. SVM processing duration and complexity were decreased as a result of the input's well-chosen characteristics. The primary flaw in this suggested algorithm was its inability to identify intrusions without affecting the rate of network flow. A sequential detection paradigm for intrusion detection system (IDS) that utilises machine learning has been unveiled.

The need for processing power was reduced by employing a suitable feature selection method. Utilising the N-BaIoT dataset, this study generated detection performance of 99% using three machine learning algorithms: neural networks, decision trees. Each sub-engine employed hybrid classification to get the best accurate results possible from a variety of classifiers. This categorization provides an extra advantage to expand the detection process by adding more sub-engines to accommodate novel types of assaults.

A stacking classifier using ensemble learning uses DT, Logistic Regression (LR), and gradient-boosting data as inputs, and an ensemble-based AIDS framework has been suggested in [12]. On the CICIDS2018 dataset, the chi-squared correlation approach was used to identify 23 significant features. The proposed model beats seven separate classifiers and obtained 98.8% detection success rate with a 97.9% F-measure score. In [13], an anomaly detection method for cloud computing was suggested.
A superior machine learning technique is SVM, which has several kernels. Key characteristics for the NSL-KDD dataset were selected using the information gain ratio. The results show that the kernel's RBF functionality has the greatest accuracy of 96.24% and the lowest number of false alarms (FAR). The ratio of testing to training was 80/20%. According to the study's conclusion, SVM offers a number of advantages for IDS evaluation in cloud computing. In [14], a new IDS on a multi-agent system with a hybrid technique has been presented.

Deep neural networks (DNNs) were used to investigate network and transport-level protocols, with a focus on transmitted control protocol (TCP). The effectiveness of DNN was examined for both detection and training agents. On the NSL-KDD dataset, the proposed model was tested with various optimizers, Init_modes, and activation functions. It achieved 98% performance for anomaly detection and 97% for classifying distinct attack types. Another work that uses the NSL-KDD dataset to develop an IDS using the classification algorithm RF is given in [15].
In this work, many machine learning (ML) algorithms were assessed for Ransomeware Attack (RA), including knearest neighbours (KNN), Naïve-Bayes (NB), gradient boost, gradient tree, Random Forest (RF), & decision tree (DT). The model's performance was predicated on machine learning techniques with effective intrusion predictivity. Using the CICIDS2017 dataset, the predictive model was an application of machine learning techniques that yielded improved outcomes. For RA, the projected outcomes of 15 models were used to analyse the risk matrix.
A work that was published in [16] suggested a paradigm for applying ML algorithms to identify Distributed Denial of Service (DDOS). Using DT & KNN classifiers, the model's performance was examined on two datasets, namely NSLKDD and KDDCup99. Eight characteristics were retrieved for this investigation using the correlation technique. With a detection accuracy & error rate that are 98.51% & 1.5%, respectively, KNN fared better in this study than DT. An investigation of the efficacy of machine learning methods for anomaly-based intrusion detection within Internet of Things field has been carried out [17]. AdaBoost (AB), random forest algorithm (RF), & multilayer perceptron (MLP) are just a few of the single and ensemble algorithms whose efficacy against DDoS (distributed denial-of-service) attacks has been studied. The study aimed to ascertain the significance of a single classifier and the possibility of significant performance from one. The results demonstrated the effectiveness of the XG-Boost (XGB) the algorithm with regard to regression tree construction and classification. Three popular data sets were utilised as benchmarks in this work: NSL-KDD, UNSW-NB15, and CIDDS-001.
[18] conducted a thorough evaluation of the effectiveness of various machine learning techniques, including logistic regression, Naïve-Bayes, K-Nearest Neighbour, Support vector machine, the Decision Tree, and Random Forest, in detecting protocol-based attacks on the Internet of Things (Message Queuing Telemetry Transport, or MQTT). A fresh dataset based on MQTT was created and then made available to the scientific community. The various requirements of MQTT-based as well as conventional attack detection are also examined in this study. True-positive (TP), true-negative (TN), and accuracy measures were used to evaluate the experiments for fivefold cross validation. For the bidirectional flow feature, the average weighted recall and accuracy increased to 98.85% & 99.04%, respectively, whereas for the unidirectional flow feature, they increased to 93.77% with 97.19%, respectively.

The investigation comes to the conclusion that flow-based features have an advantage over MQTT-based attacks in distinguishing between human entrance (Benign) and comparable characteristics. [19] Discusses the fundamental requirement for feature selection and suggests an IDS for identifying Denial of Service (DOS) assaults that makes use of Machine Learning (ML) methods including NB, KNN, RF, and SVM. Results from experiments demonstrate that when characteristics in any dataset are reduced, accuracy increases. With a 99.63% accuracy rate, RF performs better than other algorithms in this investigation.

An ensemble IDS model using KNN, extreme learning machine, and multilayer extreme learning machine approaches was also described in a different work [20]. The suggested model detected zero-day attacks with a

77.18% detection rate and an accuracy of 84.29%. The research described in [21] also assesses four machine learning methods using the Canadian Institute of Cybercrime Intrusion Detection System 2017 dataset (CICIDS2017): RF, decision tree C5.0, naïve Bayes (NB), and support vector machines (SVM).
The two main goals of this study were to detect DDoS assaults and identify higher performing machine learning algorithms. The others are surprised by the success chance of 99% with an average precision of 86.80% & 96.45% of RF & C5.0, respectively. They discover that SVM was classifying wrongly in 75% of cases, or False-Positive-Rate(FPR). Algorithmic complexity was determined by the quantity of training samples and characteristics.

SYSTEM MODEL

We go into great depth on the suggested hybrid lightweight of neural network with XGBoost intrusion detection in this section. The novel and automated Deep Learning-based attack detection system described here, shown in Fig -1, learns from data collected by the host Internet of Things (IoT) network and, once sufficiently trained, identifies network intrusions. The suggested Intrusion Detection System, or IDS, dynamic connector connects the simulated network to requests coming from the Internet of Things (IoT) network. Through an interface module, the feature extraction and network classifier are in communication with the simulated network. The network packets that make up the dataset underlying the neural network that is used in the suggested technique are extracted of their features by the feature extractor. The main goal of PCA is dimensionality reduction, which involves representing the information in a smaller-scale space while minimizing information loss. By retaining only a subset of the principal components that capture the majority of the variance in the data, PCA reduces the number of features or variables needed to represent the dataset effectively. This can lead to simpler models, faster computation, and improved generalization performance in machine learning tasks. The dataset will be split into train and test datasets and the algorithm will be trained on train dataset. Through the classifier's Updated module, the suggested IDS is constantly and continuously updated according to newly found characteristics. The network classifier forwards the intrusion to the mitigation step upon detection. The effect of the incursion is lessened at this time. The traffic in the test dataset will be classified as an Attack or No Attack.

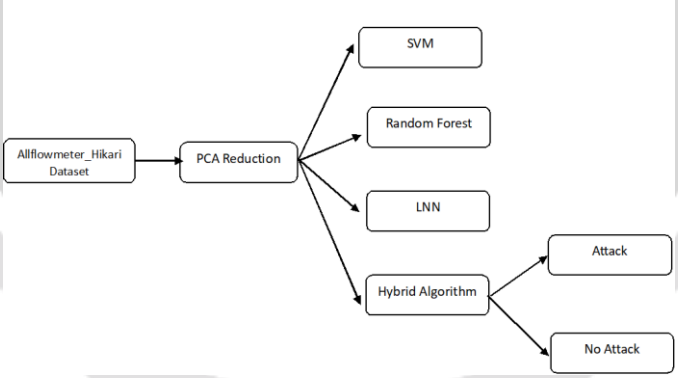

Fig -1: The Overview of the proposed methadology.

A. **Dataset**

One factor that makes evaluating malware detection systems challenging is the dearth of current datasets that are made accessible to the public. The Allflowmeter_HIKARI2021 dataset, which includes benign traffic and encrypted simulated assaults, is presented in this publication. This dataset satisfies both the content requirements, which focus on the finished dataset, and the process requirements, which focus on the dataset's development. To facilitate the generation of new datasets, we have compiled these prerequisites.

B. **Data pre-processing**

In the dataset there are 6 categories such as—flow features, time characteristics, content amenities, connection features, multipurpose features, and labelled features—are created from the total of 85 labelled features. This study takes into account seven other kinds of assaults in addition to standard data: analysis, fuzzing, code for shells, worms, denial-of-service, exploits, and backdoor assaults. In this study, 4,44,222 records are used as the training set

and 1,11,056 records are used as the testing set from the dataset. To speed up the model's training process, preprocessing the data might minimise the quantity of raw data.

The main factors that influence the quality of data are accuracy, consistency, and integrity. But inaccurate, lacking, and inconsistent data can be found in databases and data warehouses in the actual world. The suggested study pre-processes the unstructured information in order to convert it into an organised form after data collecting. Data is divided into test and training sets as part of the pre-processing stage. In the recommended research, 20% of the data is utilised for testing and 80% is used for training. At this phase, instances of overlap and duplication in the data start to appear.

A circumstance known as "data duplication" happens when an info sequence appears more than once in a collection. Conversely, data overlapping refers to the situation where a data sequence occurs in both sets. Overlapping and duplicate data might lead to an untrustworthy assessment model. The performance of the model as a whole may be jeopardised if there are sequences in the data pool that overlap. This might happen if the same sequence appears in both the sets used for training and testing. The suggested approach makes advantage of data cleaning to lessen this problem by making sure that there are no duplicate or overlaid data sequences. The original, uncleaned data is kept apart from the clean training and pristine testing data sets.

C. Feature Extraction

Feature extraction is the process of taking important features out of the dataset. As it reduces duplicate data characteristics, conserves storage, and expedites computations, this is an important step. First in the attribute selection process, pick an appropriate 0/1 string, where 1 indicates that you approve of a particular attribute and 0 indicates rejection of it. The dataset has exactly the same number of characteristics as the string's length.

PCA Dimension Reduction: After all processed records are entered, PCA will choose all pertinent features and eliminate all superfluous features in order to reduce dimensionality. The reduced parameters will be split into training and testing groups, with 20% of the dataset being used for algorithm testing within the application and 80% of the dataset being used for training.

D. Training and testing Model

In this study, a lightweight neural network termed an LNN is introduced. It uses a compressed neural network from CONV1D, which required less processing power and resources than CONV2D and 3D. We use the dimensionality reduction method of PCA (principal component analysis) to deal with large numbers of parameters by selecting only the most significant characteristics and ignoring the rest.

The proposed hybrid method has been compared with SVM, Random Forest, LNN.

ALGORITHM

In this section, we provide a two-stage alternating optimisation approach that is multitask-based and capable of effectively solving the problems in the previous works.

A. Linear SVM

In 1970, SVM was created using ideas from the theory of statistical learning [1]. In essence, it addresses regression and two-class problem with classification. A hyper-plane establishes a categorization border between two classes. The technique used to determine support vectors—which are the locations that are closest to the hyperplane—is known as the support vector machine algorithm (SVM).

Hyperplane is stated as follows in Equation 1

$$w \cdot y + b = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

where y is the value of the input vector and w and b represent the weight and bias of the input vector, respectively.

SVM is represented mathematically as Equation 2

$$If \quad w \cdot y + b \geq 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2)$$

then

The following is the final decision Equation 3

$$x = \begin{cases} +1 & if \; w.y + b \; \geq \; 0 \\ -1 & if \; w.y + b \; < \; 0 \end{cases} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

When data can be split using just one line and are preferred for a high number of features, the linear kernel form of SVM is employed.

B. Random Forest

The ensemble algorithm was first presented by Leo Breiman [13]. The author merged the technique of bagging with decision tree to construct an ecosystem for Decision Trees (DT). To construct these trees, the characteristics for separation of each node are selected at random. The DT overfitting issue is resolved by using this ensemble method. To determine sample sizes using the original data set, the bootstrap technique is applied.

C. Lightweight Neural Network

Lightweight neural networks for intrusion detection represent a specialized application of machine learning techniques to enhance cybersecurity in computer networks. In contrast to conventional intrusion detection systems, which could depend on computationally demanding models or rule-based techniques, lightweight neural networks have been developed to handle network traffic data effectively while using the least amount of memory and processing resources possible. These networks are typically characterized by simplified architectures optimized for streamlined feature extraction from network packets. The architecture often includes layers tailored to the characteristics of network traffic data. For instance, convolutional layers may be utilized to capture spatial patterns in packet headers or payload data, while recurrent layers can model temporal dependencies within traffic flows. One key aspect of lightweight neural networks for intrusion detection is their efficiency in training and inference. Training algorithms are designed to optimize the network parameters efficiently, minimizing computational overhead and training time. During inference, the network can quickly process incoming network traffic data in real-time, enabling timely detection of potential intrusions.

Moreover, lightweight neural networks exhibit adaptability to dynamic network environments and evolving threat landscapes. They can dynamically adjust their parameters based on changes in network conditions and adapt to emerging intrusion techniques. Techniques such as online learning or transfer learning may be employed to facilitate continuous model updates and knowledge transfer from related tasks or domains. Robustness to noise and adversarial attacks is another essential characteristic of lightweight neural networks for intrusion detection. These networks are designed to handle noisy data and remain resilient in the face of deliberate attempts to evade detection. Techniques such as data augmentation, regularization, and adversarial training may be employed to enhance the network's robustness and generalization performance. Furthermore, lightweight neural networks are scalable and deployable across various network environments, including edge devices, cloud servers, and network appliances. Efficient deployment mechanisms and model compression techniques enable their deployment in distributed network infrastructures without significantly impacting system performance.

D. XGBoost

Extreme Gradient Boosting, or XGBoost for short, is a well-liked and potent machine learning technique that excels in a variety of fields, including computer network intrusion detection, thanks to its scalability, efficiency, and high performance. XGBoost belongs to the ensemble learning family and is particularly well-suited for classification tasks, making it a valuable tool for detecting and classifying network intrusions.

Fundamentally, XGBoost leverages the advantages of gradient boosting methods in conjunction with an optimised and scalable implementation, making it capable of efficiently managing huge datasets with high dimensionality. The way the method operates is by creating a group of decision trees repeatedly, with each new tree being trained to fix the mistakes of the preceding ones. The end product of this iterative process is a very reliable and accurate prediction model, which is maintained until a certain number of trees (or convergence) is attained. The versatility of XGBoost in handling various data sources and feature representations frequently seen in network detection applications is one of its main features. XGBoost can efficiently learn from features and identify intricate patterns in the data, regardless of whether they are numerical, categorical, or a combination of the two. Furthermore, XGBoost manages missing values automatically, negating the requirement for intensive data preparation. Moreover, XGBoost provides a variety of hyperparameters. Among them tree depth, learning rate, regularisation parameters, including the number of trees in the ensemble, that may be adjusted to maximise model performance.

Through careful hyperparameter tuning, XGBoost can achieve exceptional performance on intrusion detection tasks, balancing between model complexity and generalization ability. Another advantage of XGBoost is its computational efficiency, which is crucial for real-time intrusion detection systems deployed in network environments. XGBoost's parallelized and optimized implementation enables fast training and inference, making it suitable for processing large volumes of network traffic data in real-time.

Moreover, XGBoost provides interpretable results, allowing analysts to understand the underlying patterns and decision-making process of the model. The most important characteristics for intrusion detection may be found using the feature significance scores produced by XGBoost, which also helps with the interpretation and justification of anomalies that are found.

EXPERIMENTAL RESULTS

We have experimented with the Allflowmeter_HIKARI2021 dataset to assess the effectiveness of the LNN and XGBoost based NID approach. The entire dataset was converted to a numeric format. The dataset has 85 characteristics or columns and 555278 records before PCA. Prepare the dataset: Both numeric plus non-numeric data are included in the dataset, but machine learning can only process numeric data. Therefore, we will use the label encoder class to turn any qualitative in nature data into data that is numerical and to replace any values that are absent with 0.

At preprocessing stage, out of 85 characteristics 83 characteristics are used. After using PCA dimension reduction, we were able to extract 20 features out of 83. Additionally, the dataset is being divided into train and test sets, with 20% of the dataset being used to evaluate the algorithm's performance and 80% of the dataset being used for training. This project has extension which proposes LLN model to Hybrid LLN model. Hybrid model will extract trained optimized features from LLN model and then retrain with XGBOOST algorithm to enhance accuracy. This is similar to training an experienced person who can employ his experience and training together to yield better result.

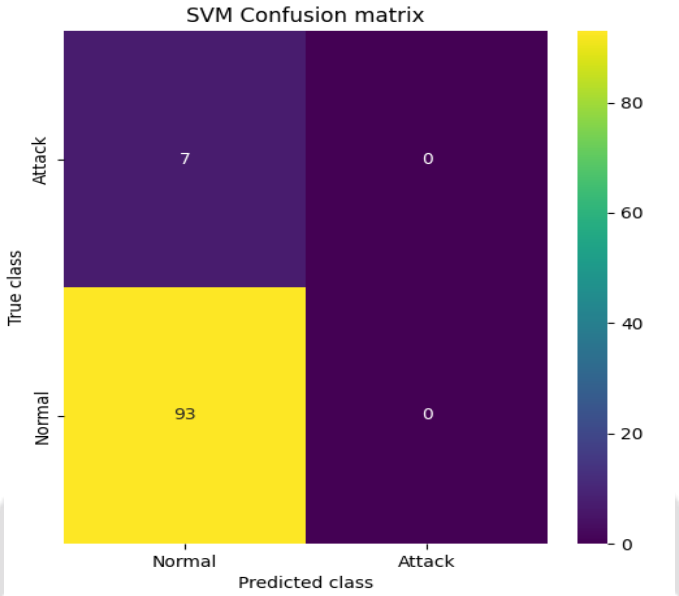

Fig -2: SVM Algorithm Confusion Matrix

The confusion matrix for the 94% accurate Support Vector Machine method is displayed in Fig -2. In the confusion matrix graph, the light green, yellow, and blue boxes contain the accurate prediction count, the blue boxes contain the wrong prediction count, and the x-axis represents the predicted labels and the y-axis represents the TRUE labels.

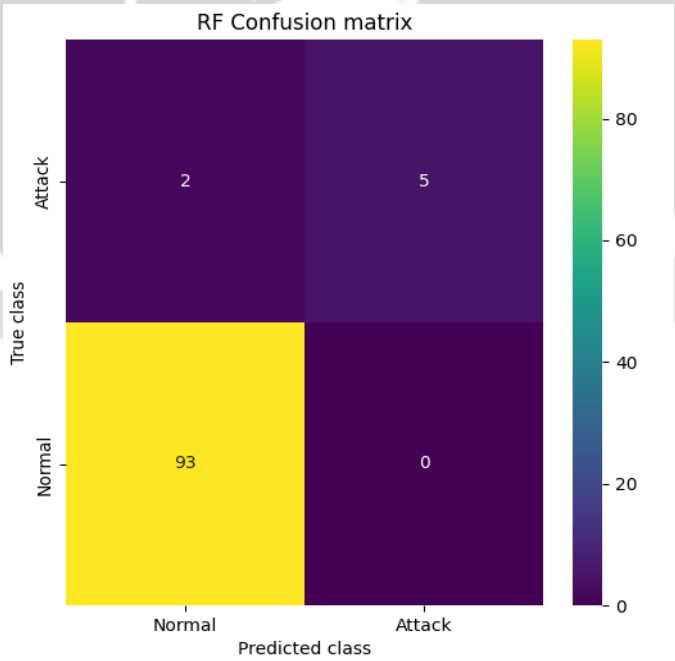

Fig -3: Random Forest Algorithm Confusion Matrix

The Fig -3, show confusion matrix for Random Forest algorithm which has the accuracy of 99 %.

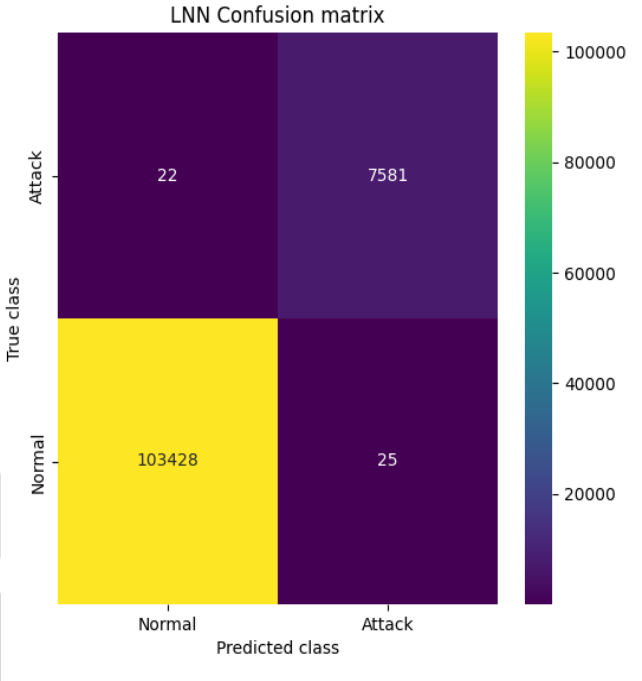

Fig -4: LNN Algorithm Confusion Matrix

The Fig -4, show confusion matrix for Lightweight Neural Network algorithm which has the accuracy of 100.95 %.

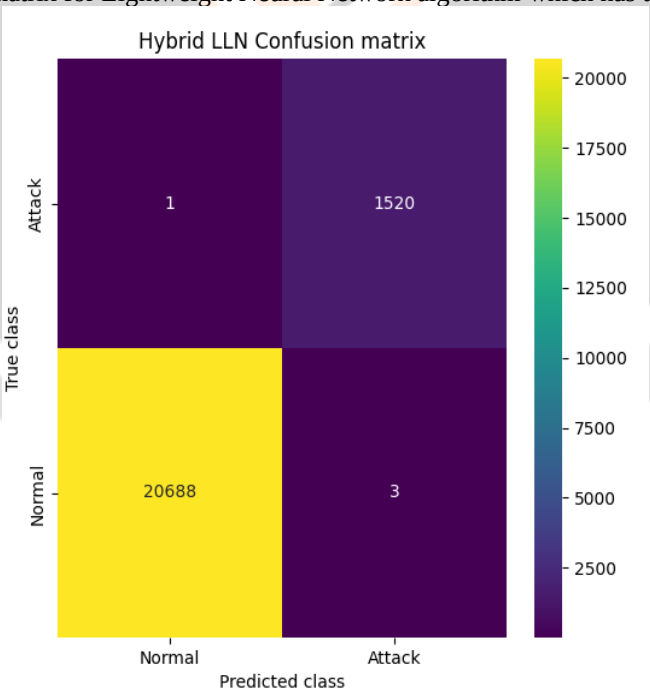

Fig -5: Hybrid Algorithm Confusion Matrix

The confusion matrix for the hybrid method, which has a 100.98% accuracy rate, as seen in Fig -5. In the confusion matrix graph, the light green, yellow, and blue boxes contain the accurate prediction count, the blue boxes contain the wrong prediction count, and the x-axis represents the predicted labels and the y-axis represents the TRUE labels.

Table -1: Comparison of Algorithms

Algorithm	Accuracy	Precision	Recall	F1-Score
SVM	94	47	51	49.1
Random Forest	99	99.94	86.71	92.13
LNN	100.95	100.82	100.84	100.83
Hybrid	100.98	100.89	100.95	100.92

In the Table -1, we can find LNN and Hybrid algorithm gives the accuracy of 100 percent. So we can consider these methods to predict Intrusion under Internet of Things IoT model.

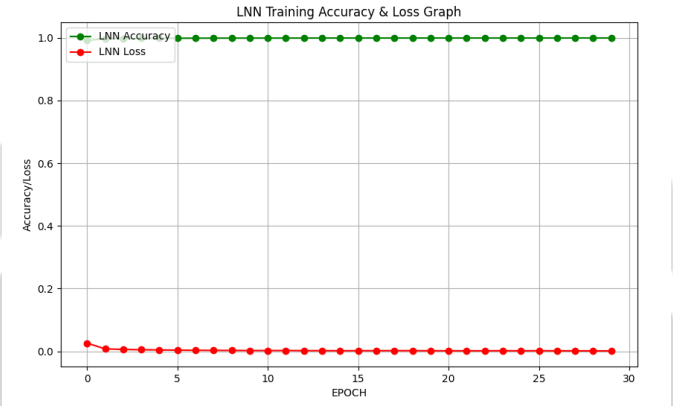

Fig -6: LNN Accuracy and Loss Graph

In the graph shown in Fig -6, the x-axis denotes training epoch, the y-axis denotes accuracy and loss. The red and green lines, respectively, show accuracy and loss. As the period progressed, accuracy increased and approached 1, while loss decreased and approached 0.

CONCLUSIONS

This work presented hybrid LNN, a lightweight detection of intrusions method, for edge devices with low resources. Our suggested technique achieves a superior trade-off between effectiveness and precision when compared to a series of circuits that employed the same topology as LNN. LNN + XGBoost specifically achieves a near 100% reduction in computational cost overall parameter size while maintaining a little higher accuracy & F1 score. Additionally, LNN performs better than the standard models and other current attack detection theories, which is quite possibly the best outcome at such a cheap resource cost. We find that our suggested method achieves a notable edge during network intrusion detection.

REFERENCES

[1] Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I.: Toward a lightweight intrusion detection system for the internet of things. IEEE Access 7, 42 (2019)
[2] Nivaashini, M.; Thangaraj, P.: A framework of novel feature set extraction based intrusion detection system for internet of things using hybrid machine learning algorithms. In: 2018 International conference on computing, power and communication technologies (GUCON). pp. 44–49 (2018)
[3] Tait, K.-A.; Khan, J. S.; Alqahtani, F.; Shah, A. A.; Khan, F. A.; Rehman, M. U.; Boulila, W.; Ahmad, J.: Intrusion detection using machine learning techniques: an experimental comparison. In: IEEE International congress of advanced technology and engineering (ICOTEN)
[4] Khan, M.A.; Khan, M.A.; Latif, S.; Shah, A.A.; Rehman, M.U.; Boulila, W.; Driss, M.; Ahmad, J.: Voting classifier-based intrusion detection for IOT networks. In: 2nd International conference of advanced computing and informatics (ICACIN) (2021)

[5] Abiodun, O.I.; Abiodun, E.O.; Alawida, M.; Alkhawaldeh, R.S.; Arshad, H.: A review on the security of the internet of things: challenges and solutions. Wireless Pers. Commun. (2021). https://doi.org/10.1007/s11277-021-08348-9.
[6] Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2, 12 (2019)
[7] Kumari, A.; Mehta, A.: A hybrid intrusion detection system based on decision tree and support vector machine. In: 2020 IEEE 5th International conference on computing communication and automation (ICCCA), pp. 396–400, (2020)
[8] Pokharel, P.; Pokhrel, R.; Sigdel, S.: Intrusion detection system based on hybrid classifier and user profile enhancement techniques. Int. Workshop Big Data Inf. Secur. 2020, 137–144 (2020)
[9] Kilincer, I.F.; Ertam, F.; Sengur, A.: Machine learning methods for cyber security intrusion detection: datasets and comparative study. Comput. Netw. 188, 107840 (2021)
[10] Fitni, Q.R.S.; Ramli, K.: Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In: 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and ommunications Technology (IAICT), pp. 118–124. (2020)
[11] Fitni, Q.R.S.; Ramli, K.: Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and ommunications Technology (IAICT), pp. 118–124. (2020)
[12] Krishnaveni, 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, Singapore (2020)
[13] Liang, C.; Shanmugam, B.; Azam, S.; Jonkman, M.; Boer, F.; Narayansamy, G.: Intrusion detection system for internet of things based on a machine learning approach (2019)
[14] Yang, L.; Cai, M.; Duan, Y.; Yang, X.: Intrusion detection based on information entropy for random forest classification. ICBDC 2019. New York, NY, USA: Association for Computing Machinery, p. 125-129 (2019)
[15] Kachavimath, A.V.; Nazare, S.V.; Akki, S.S.: Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics, 2020.
[16] Verma, A.; Ranga, V.: Machine learning based intrusion detection systems for IOT applications. Wireless Pers. Commun. 111(4), 2287–2310 (2020)
[17] Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X.: Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)' (2020)
[18] Sah, G.; Banerjee, S.: Feature reduction and classifications techniques for intrusion detection system. Int. Conf. Commun. Signal Process. 2020, 1543–1547 (2020)
[19] Abdulrahman, A.; Ibrahem, M.K.: Evaluation of ddos attacks detection in a new intrusion dataset based on classification algorithms. Iraqi J. Inf. Commun. Technol. 1, 49–55 (2019)
[20] Latah, M.; Toker, L.: An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. CCF Trans. Netw. 3(3), 261–271 (2020)
[21] Pangsuban, P.; Wannapiroon, P.: A real-time risk assessment for information system with cicids2017 dataset using machine learning. Int. J. Machine Learn. Comput. 10(3), 465–470 (2020)