DEEP LEARNING FOR INTRUSION DETECTION IN IOT BASED SMART HOMES

By

NAZIA BUTT

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

August, 2022

Deep Learning for Intrusion Detection in IOT based Smart Homes

By

NAZIA BUTT

MCS, Fatima Jinnah Women University, Rawalpindi 2014

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

Computer Science

To

FACULTY OF ENGINEERING & COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

NATIONAL UNIUVERSITY OF MODERN LANGUAGES FACULTY OF ENGINEERIING & COMPUTER SCIENCE

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Title: Deep Learning for Intrusion Detection in IOT based Smart Homes

Submitted By: Nazia Butt **Registration #:** 34 MS/CS/S20

Master of Science in Computer Science (MSCS)
Title of the Degree

Computer Science
Name of Discipline

Dr. Sajjad Haider
Name of Research Supervisor

Signature of Research Supervisor

Dr. Basit Shahzad
Name of Dean (FE&CS)

Signature of Dean (FE&CS)

Prof. Dr. Muhammad Safeer Awan
Name of Pro-Rector Academics

Signature of Pro-Rector Academics

August 29, 2022

AUTHOR'S DECLARATION

I Nazia Butt

Daughter of Dilawar Butt

Registration # 35 MS/CS/S20

Discipline Computer Science

Candidate of **Master of Science in Computer Science (MSCS)** at the National University of Modern Languages do hereby declare that the thesis **Deep Learning for Intrusion Detection in IOT based Smart Homes** submitted by me in partial fulfillment of MSCS degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be cancelled and the degree revoked.

Signature of Candidate

Nazia Butt

Name of Candidate

29thAugust, 2022

Date

ABSTRACT

Title: Deep Learning for Intrusion Detection in IOT based Smart Homes

Scurrying growth in IOT has been alleviating the different fields like Health Care Units, Industrial Units, Smart Homes or Military and so is trending topic for research. However, with the emergence of IOT, there is also high risk of security violations. Security breach involved the different categories of attack, illegitimate access and other privacy risks in IOT systems. Therefore, different researches had been conducted to palliate Cyber-attacks by configuring Intrusion Detection in different scenarios but as attacks are also growing with the same rate therefore, more work is still demanded or expected. In the proposed study, the comparative analysis of different Anomaly Based Intrusion detection system is conducted concerning existing state-of-the-art studies with respect to datasets, Machine Learning and Deep learning models. To overcome the limitations highlighted in existing work, the research proposed a novel solution for anomaly based intrusion detection in IOT with increased performance, lessen overfitting/underfitting issues and generalizable in nature. To ensure high performance w.r.t. different evaluation metrics, hybridization of Machine learning and Deep Learning models LSTM, KNN and DT was done and implemented on real time dataset CIC-IDS-IOT2022. To avoid underfitting/overfitting issues, feature selection and hyperparameter tuning was implemented. To check its impact, same solution was tested on benchmark dataset UNSW-NB15. Google Colab and python were used as a platform and language. Experiment results showed significant increase in performance while minimizing misclassification and other limitations in comparison with state-of-the-art solutions. Involvement of more datasets and hybridization of other ML/DL algorithms inspired by the proposed solution in real time IOT-IDS network is a future research goal.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IOT	-	Internet of Things
DOS	-	Denial of Service
DDOS	-	Distributed Denial of Service
TCP	-	Transmission Control Protocol
IDS	-	Intrusion Detection System
FSM	-	Finite State Machine
PCA	-	Principal Component Analysis
KNN	-	K Nearest Neighbor
ML	-	Machine Learning
DL	-	Deep Learning
IDS-IOT	-	Intrusion Detection in Internet of Things
NIDS	-	Network Intrusion Detection System
SLR	-	Systematic Literature Review
SVM	-	Support Vector Machine
DT	-	Decision Tree
GA	-	Genetic Algorithm
NB	-	Naïve Bayes
LSTM	-	Long Short-Term Memory
CNN	-	Convolutional Neural Network
MLP	-	Multilayer Perceptron
RBM	-	Restricted Boltzmann Machine
RF	-	Random Forest
FFNN	-	Feed Forward Neural Network
DBN	-	Deep Belief Network
DAE	-	Deep Auto Encoder
DNN	-	Deep Neural Network
RNN	-	Recurrent Neural Network
GRU	-	Gated Recurrent Unit
TP	-	True Positive
TN	-	True Negative
FP	-	False Positive

FN	-	False Negative
P	-	Precision
R	-	Recall
BiRNN	-	Bidirectional Recurrent Neural Network

ACKNOWLEDGMENT

Foremost, I want to offer this endeavor to Almighty Allah, for the wisdom he bestowed on me, the strength, sound health and mind that helped me to complete the research. This study would not be fulfilled without encouragement and honest espousal that was extended from various sources for which I would like to express my sincere gratitude. There were continued supporters and significant contributors for my success, especially my research supervisor Asst. Professor Dr. Sajjad Haider, who left no stones to guide me throughout. Therefore, I would like to say a special thanks to my supervisor for his support, guidance and deep insights in the research.

Lastly, I shall not forget the extended help from the administrations of Department of Computer Sciences for their continuous support and guidance in coping up the challenges I faced. Thanks to all the people I did not mention but who kept me going and made an effort that I shall not ignore, many thanks for all.

DEDICATION

I dedicate this thesis work to my parents and siblings whose love n affection, support and role modeling throughout the years laid the foundations to complete any task with hard work. I also dedicate this work to my mentors and teachers who equipped me with pearls of knowledge and strive for excellence.

CHAPTER 1

INTRODUCTION

1.1 Overview

Internet-of-Things (IOT) is interconnection of sensors, machines, objects or other computing devices over internet in order to communicate with least human interference. IOT technology automates task and connect devices with internet. Specific types of sensor are involved to get information from physical entities and after analyzing it is stored into local storage which is then sent to cloud storage where appropriate action is taken according to the information. IOT usually came up with six elements identification, sensing, communication, computation, services and semantics [1]. These elements are capable of naming, addressing, data collection from sensors or actuators, exchange of messages, decision making according to the received information and facilitate users acting like human brains. The IOT devices needs to be in communication range so that instant communication can happen [2].

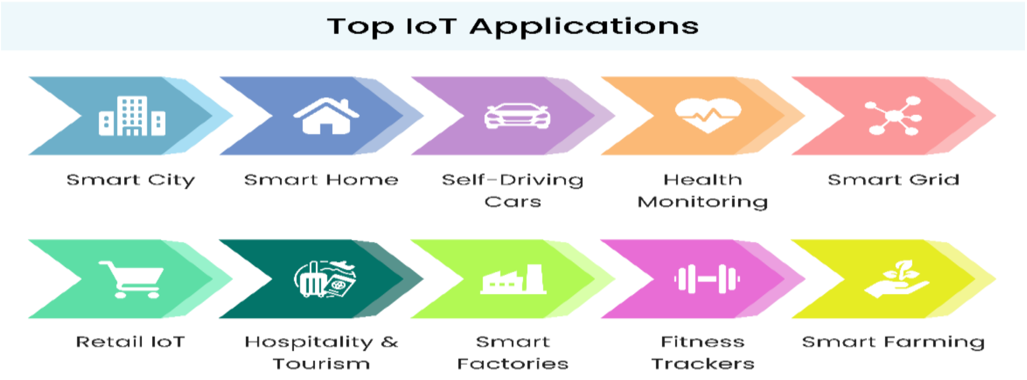


Figure 1.1 : Applications of IOT

IOT is everywhere from smart homes, smart business, smart healthcare, agriculture or in predicting natural disasters. However, IOT is facing hurdles in the way of implementing as global technology due to multiple challenges like identification, scalability, energy efficiency, Data Management, interoperability, self-organizing capabilities and security. Among all challenges, security and privacy is the key necessity that ensures wide adoption of IOT. Moreover, to make communication secure and safe, security has already become a significant issue for many firms and Military as well. Involvement of IOT applications for daily chores, its security has also become a concern for a normal person. Security breach could range from information control to action control which could be a nightmare for all as it could affect lifestyle, health data, information theft and other serious implications [3].

There are multiple attacks that could compromise the like DOS, DDOS, Bruteforce, Ransomware, HeartBleed, synflood, TCP or UDPflood or hacking. To mitigate these threats, traffic in an IOT network needed to be monitored and analyzed. This brings focus to fields of Datamining, Bigdata and Intrusion Detection Systems. Intrusion detection is the concept of monitoring traffic and classifying it into benign or malign. Intrusion detection in IOT network could be signature based, Anomaly based or Specification based. There is further classification of each type of Intrusion detection system (IDS), each ensures the security and prevention of a network in one way or another [4]. However, one responsible for prevention is known as Intrusion Prevention system (IPS).

In anomaly based intrusion detection system, normal behavior is recorded and stored as patterns and then used to compare it with traffic patterns to see if noise and other probabilities of intrusions are anomalous or normal [4]. There are multiple techniques for anomaly based intrusion detection systems like Data Mining, Machine Learning, Statistical Model, Rule Model, payload Model, Protocol Model and signal processing model each with its pros and cons.

Machine Learning and Deep learning techniques for anomaly detection had been implemented in various fields to tackle attacks in a network with significant performance [5]. It usually consists of two phases Learning phase (Training), Testing phase and have various algorithms which are categorized as supervised, unsupervised, semi-supervised or reinforcement learning. Each category has multiple algorithms that could be used according to the need and scenario. However, there are also consequences of implementing these techniques in which requirement of training data and training time are the significant ones along with other evaluating factors which shows performance and resource requirements.

Among other challenges, availability of good data is a big challenge. There are wide range of public datasets available for research purposes for intrusion detection in a network. However, most of them are not comprised of real time data and some of them are also getting outdated for providing enough mechanism for mitigating attacks in an IOT environment. Thus, finding a sound solution for anomaly based intrusion detection in an IOT environment with suitable Machine Learning or Deep Learning techniques on updated datasets with high performing metrics like accuracy, precision and low error rate, able to mitigate number of attacks is still ongoing quest.

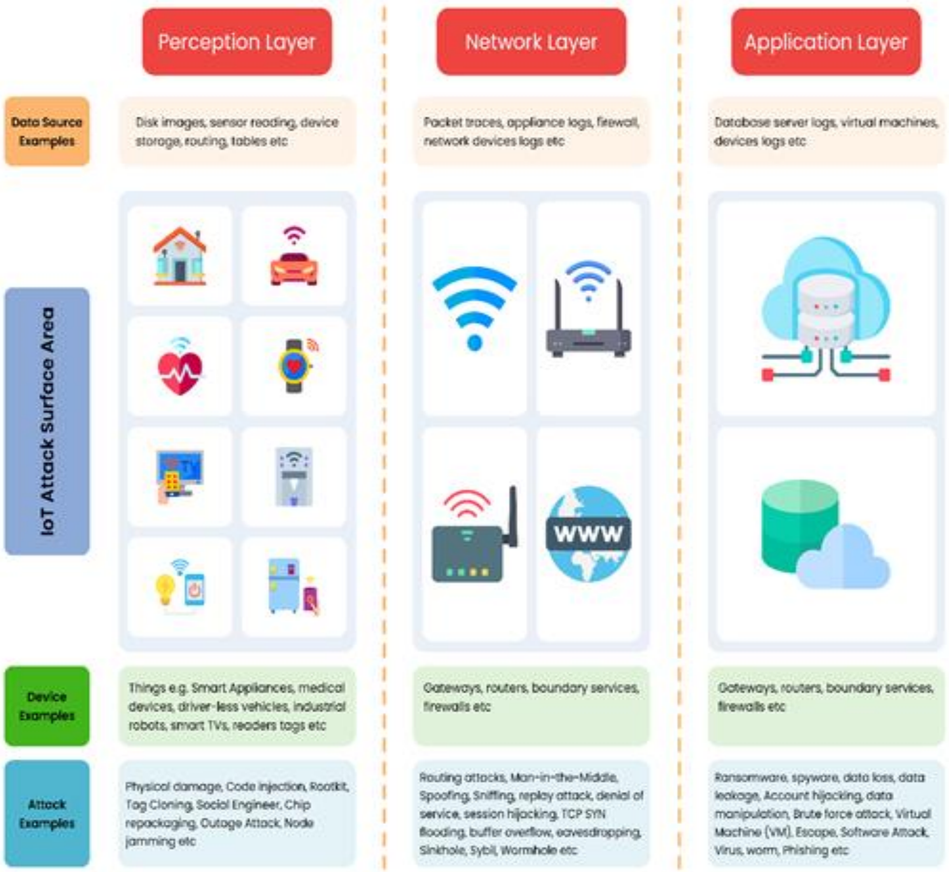
1.2 Motivation

The widespread of IOT devices is smart environments came up with security challenges. Several attacks pointed towards IOT network have become a motivation for implementing attack countermeasures [1]. Attacks could be an intrusion or anomaly with consequences. Hence, providing an anomaly based intrusion detection system, to prevent confidential resources from intruders, practicing Machine learning and deep learning models with remarkable performance has been pursuit of many researchers since years. However, by reviewing state-of-the art schemes, loop holes in them pulled researchers in the field for the sake of improvement. Many of the schemes lacked intrusion detection in IOT environment whereas showing results for NIDS. Moreover, using old datasets with no real time data is also an issue. Number of attacks and their patterns are continually emerging, therefore, system must be able to detect zero day attacks while keeping in mind the resource constraints. Machine Learning and Deep learning techniques for anomaly based detection showed valuable results while hybridization of machine learning and deep learning techniques is still needed to be explored along with hyperparameter tuning to enhance the performance in terms of accuracy, precision, recall and F1-score and minimizing the issues caused by noise in the data. Due to high volume network data, Identification and selection of features to enhance the performance is also needed to be done. This research is dedicated to find an anomaly based machine learning technique for IDS-IOT with improved performance in comparison with state-of-the-art schemes.

1.3 Architecture of IOT and IOT-IDS

There is no consensus for IOT architecture. They are usually divided into 3 layers, 4 layers or 5-layer architectures. The name of three layers are Application, Network and Perception while in four or five layer other layers could be named as support layer, processing layer or business layer while these three layers remain constant [1].

In three-layer architecture, Perception layer consists of sensors, RFIDs or WSNs. It senses environment and collect information therefore it is a main target of intruders. Eavesdropping, Node capture, fake node, Timing and Replay attacks are the threats attached to it. Network layer is known as transmission layer works as a bridge between perception and application layers. Transmission source could be wired or wireless whereas, it is also used to connect to other networks or smart devices and therefore high range of threats or attacks are attached to it. Denial of service (DOS), Man in the middle (MiTM), storage attacks or exploits attacks are the main attacks that could compromise network performance and confidentiality. Application layer consists of all application areas implementing IOT technology like Smart Homes, Smart Agriculture, Industry and many more. This layer works like service provider to all applications depending upon the information collected by sensors. Frequent vulnerabilities could be Cross Site scripting, Malicious Code Attack and Mass Data.



	Perception Layer	Network Layer	Application Layer
Data Source Examples	Disk images, sensor reading, device storage, routing, tables etc	Packet traces, appliance logs, firewall, network devices logs etc	Database server logs, virtual machines, devices logs etc
Device Examples	Things e.g. Smart Appliances, medical devices, driver-less vehicles, industrial robots, smart TVs, readers tags etc	Gateways, routers, boundary services, firewalls etc	Gateways, routers, boundary services, firewalls etc
Attack Examples	Physical damage, Code injection, Rootkit, Tag Cloning, Social Engineer, Chip repackaging, Outage Attack, Node jamming etc	Routing attacks, Man-in-the-Middle, Spoofing, Sniffing, replay attack, denial of service, session hijacking, TCP SYN flooding, buffer overflow, eavesdropping, Sinkhole, Sybil, Wormhole etc	Ransomware, spyware, data loss, data leakage, Account hijacking, data manipulation, Brute force attack, Virtual Machine (VM), Escape, Software Attack, Virus, worm, Phishing etc

Figure 1.2: Architecture of IOT

Intrusion detection systems to mitigate network attacks is a necessity for a chaos free and private communication demanded by all application areas ranging from smart homes to Smart Agencies [4] [6]. Hence, there are three classes of IDS named as Signature Based, Anomaly Based and Specification based or Hybrid. Among these classes, Anomaly Based detection is able to detect unknown attacks as well therefore, gaining limelight of researchers. Anomaly based detection is further classified into FSM, Statistical, Data mining and Machine Learning. Due to diversity of Machine Learning methods and their abilities to detect anomalies, they are frequently implemented.

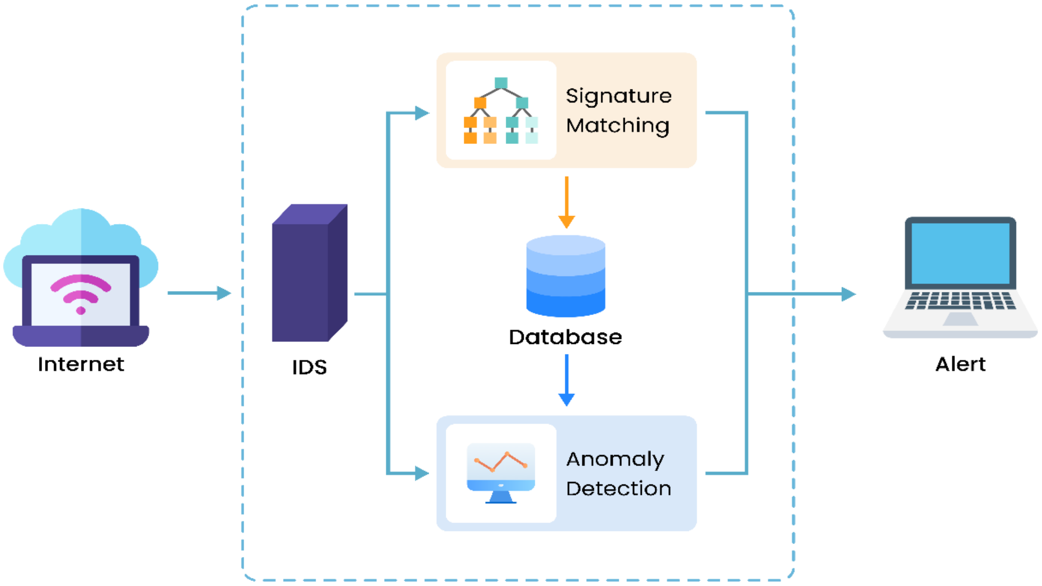


Figure 1.3: Architecture of IDS

Architecture of Anomaly based IDS consists of Detector that monitors the environment and analyzes the events according to baseline profile/models [7]. If observed activity matches the baseline profile than it is classified as normal or if it does not match but still in a threshold range, then the profile is updated but if it is not in threshold range then it is classified as anomaly and action is taken according to the designed system.

Number and types of attack depends upon the Data source. Anomaly based intrusion detection could be implemented while capturing network traffic and process it as dataset or on publicly available datasets that serves the desired purpose.

1.4 Applications of IDS-IOT

Applications of intrusion detection system in a network could be categorized as Network traffic processing, Anomaly Detection, Threat Classification, Threat reporting, Prevention System and signature matching. It is also used in areas where fewer security incidents, selective logging, privacy protection, reputation protection, multiple or dynamic threat protection are goals to attain like in smart industries, homes or healthcare [4] [5] [8]. It monitors or analyzes threats in order to cope up with intruders in real time, prevent numbers of attack like DOS/DDOS, detect unknown attacks or other malicious activities. This makes it an important research paradigm.

1.5 Constraints in Anomaly Based IDS-IOT

Despite of several methods for anomaly based intrusion detection in network, there are still numerous constraints that are needed to be handled or cannot be overlooked:

1.5.1 False Alarm Rates

False Alarm Rate is also known as False Positive Rate which means that the system declares something as true while it is actually false. Minimizing False Alarm Rate is one of the biggest challenges for secure IDS. However, minimizing it to zero is impossible but reducing it to maximum range to enhance performance is a research challenge. Hence, to increase the performance of network intrusion detection system, it is vital to reduce the False Alarm Rate [9] [10].

1.5.2 Low Detection Rate

Ratio between correctly predicted tasks and number of attacks is called DR or sensitivity. It should be high as low DR represents that there are large number of attacks that are not classified or predicted [10] [11]. Hence low detection rate is a constraint that is needed to be remove in order to implement a secure IDS.

1.5.3 Imbalanced Datasets

Imbalanced data is caused by unfair distribution of classes. Resampling for minority classes is necessary so that correct representation of data can be done [12]. However, in real time intrusion detection, data is usually imbalanced so there are multiple methods to deal with the data in order to classify minority classes correctly as well [13].

1.5.4 Noise

Noise is caused by the data that is not necessary or useful and may be redundant [14]. Big noise in the data cause low performing IDS. Hence, it is necessary to remove noise in the data which could be done by implementing correct methods to reduce noise [15].

1.5.5 Bias

Bias arises due to oversampling. It is also known as prediction error. Low bias is the challenge to attain with correct methods and techniques [16] [17].

1.5.6 Variance

If the classifier performs good with train set but could not perform well with test data, then Variance occurs. Variance must be low to avoid underlying issues cause degradation in performance [16] [17].

1.5.7 Overfitting/underfitting

Overfitting occurs when there is low bias and high variance [17] [18]. It reduces the efficiency and performance of IDS as this system tries to cover values which could be a noise only. Underfitting is caused by high bias and low variance [17] [18]. It means that system is unable to learn enough from train data and hence cannot predict well which results in low accuracy. Goodness of fit is a term that must be achieved to avoid these limitations.

1.5.8 Reduce Dimensionality

To convert dataset from high dimensional to low dimension with less features with appropriate algorithms like PCA, KNN etc., is usually required as data is usually high dimensional [19] [20] [21]. It is necessary to handle high dimensional data.

1.5.9 Response Time

Timely response is a must option for an optimal intrusion detection system. Intrusion detection system with delay in response cannot be called as good solution even if they are able to predict and detect attacks [16].

1.5.10 Generalizable Model

When a model is able to give good results even for unseen data then the model is call Generalizable. Model is not generalizable usually due to no updating in the datasets for new attacks. Moreover, it is not easy to attain and depends upon Machine Learning algorithms, hyperparameters and Regularization techniques [22] [23].

1.5.11 Computational Complexity

Computational complexities are termed as Time and Memory constraints. In IDS, preprocessing in the training stage and deployment in testing phase could cause high computational complexity which is needed to be taken care of [7] [10].

1.5.12 Updating patterns vs performance

Updating profiles in the database during data capture or preprocessing should be in a way that is not causing low performance [24]. Retraining could be used for updating patterns.

1.5.13 Attribute Selection

Feature selection that are representing a specific category or attack is another challenge that could be attained with different methods or techniques [25] [26]. Chi square, Random Forest Importance and Recursive Feature Elimination are common methods for Attribute selection.

1.5.14 High Accuracy

Accuracy is measured in percentage and represents the ability of a system to classify benign or malign classes of attacks correctly. Achieving high accuracy is the basic goal to achieve for the good IDS [27] [28]. It could be achieving by accurate detection of attacks. Choice of good classifier according to the requirement ensures it.

1.5.15 Cross Validation

Cross Validation is a method to assess the performance of Machine Learning algorithms by testing it on subsets of the data. There are many types of cross validation techniques to test the accuracy of the model. Kfold, Rolling, Monte Carlo etc., Hence, choosing good validation mechanism is also important [29] [30].

1.5.16 *Real Time Datasets*

Intrusion detection system in IOT is possible by using real time datasets, however, there are no such real time datasets available publicly and hence it is also a challenge to get a good real time dataset [31].

1.5.17 Classifier

Choosing a best classifier for a better performing IDS-IOT is a big challenge which is achieved by choosing unbiased and non-associated algorithm for a desired problem as suggested by [32].

1.5.18 Protection Against New Attacks

Protection against unknown attacks is required to secure an IOT environment. Therefore, continuous updating of profiles is required which could be attain by using AE or network flow patterns [33].

1.6 Problem Background

IOT is the concept of ubiquitous connectivity where all physical and virtual objects are embedded with Internet Protocol suite which enables them to connect with internet. The availability on the Internet makes it vulnerable to several security threats which require security mechanism to cope up with these threats. Security is a term that consists of multiple techniques with the objective of preserve, restore and guarantee the protection of information in computer systems from attacks [34]. For this very reason, Intrusion Detection system for IOT to minimize the effects of attack and to make it secure has been studied since years. IDS is in the form of hardware or software that is used to monitor and analyze traffic in order to detect attacks. IDS is further classified into Signature Based and Anomaly Based techniques [8]. In signature Based techniques, detection is done by matching signatures stored in the database with the signatures of the traffic flow. In Anomaly based detection, there is no such need of signatures and hence it is also able to detect zero day attacks. There are multiple methods of Anomaly detection in which Machine Learning algorithms are most frequently adapted. There are multiple studies on Anomaly based IDS with Machine Learning techniques for attacks detection in IOT environment.

The main focus of state-of-the-art studies were to make use of different Machine learning and Deep learning techniques for intrusion detection. However, most of them lack in using real time dataset or updated dataset which leads to degradation in performance as attacks usually change their patterns and ways. Public available datasets in SLR lacked in having IOT traces and mostly performed well for NIDS. Along with overhead on monitored system, some of them also increased time complexity. Moreover, there were many underlying issues like noise, overfitting, underfitting, complexity and dimensionality which were caused due to carelessness in data cleaning, feature extraction, selection and normalization techniques. The good performance of algorithms on unseen data make it a general model which is another important factor. Accuracy is an important parameter to measure the performance of the model which needs to be maximized whereas there was difference in test and train accuracy of many existing models which depicts overfitting or underfitting. To countermeasure these issues, appropriate methods were selected for Data cleaning, hyperparameter selection and hybrid ML/DL based classifiers were used on recent real time data set for intrusion detection in an IOT environment which is IOT based smart home.

1.7 Problem Statement

Vulnerabilities in an IOT environment in the forms of network attacks make it less secure. On the contrary, IOT is everywhere from homes, education sector, healthcare, businesses to highly authoritative agencies and is getting its way in taking over almost all the activities [2], therefore, intrusion detection system to cope up the attacks were presented by many researchers which shows the unavailability of latest real time datasets with a need to improve the performance of classifiers by adapting correct techniques and making it a generalized model.

1.8 Research Questions

The study accomplishes the answers to these questions.

i. How to implement hybrid ML/DL model for intrusion detection in IOT based Smart Homes.

ii. How to improve the performance of the IDS by using hybrid ML/DL model for classification in order to make it generalizable in nature as well by using correct evaluation techniques.

1.9 Aim of the Research

The aim of the research is to provide ways of securing an IOT smart environment through hybridization of Machine Learning and Deep Learning techniques. It is also inclined towards mitigating different network attacks through IDS-IOT. Improving performance through correct selection of methods and techniques is also a goal.

1.10 Research Objectives

The objectives of the research are:

i. To implement hybrid ML/DL model for intrusion detection in IOT based smart Homes.

ii. To find ways for improving the performance of IDS by using hybrid ML/DL models by using appropriate evaluation techniques.

1.11 Scope of the Research

The intrusion detection system for IDS through hybrid ML/DL techniques was implemented for Smart Home on real time IOT dataset. However, it could also be used for NIDS. It is able to mitigate attacks: BruteForce, UDPflood, TCPflood and HTTPflood attacks. It is also able to classify normal behavior. It could also be used for smart environments by updating profiles in the dataset.

1.12 Thesis Organization

Thesis is structured as:

Chapter 2 provides a review on different Machine Learning and Deep Learning Methods for intrusion detection. Taxonomy of these ML-based IDS-IOT is also presented. Based on this taxonomy, different schemes have been reviewed in the literature and tabular comparison of these schemes in terms of classifiers, datasets and attacks is provided. By reviewing these schemes, limitations of the state-of-the-art schemes and gaps in them leads towards this research work.

Chapter 3 provides a complete methodology and details of steps for solving an identified problem. It includes research design and experimental setup. It also includes architectural diagram of the proposed solution along with pseudo code having complete details. Benefits of the proposed model is also explained along with important equations and Mathematics.

Chapter 4 gives results and analysis of the proposed scheme in comparison with different benchmark schemes. It also gives analytical details of the proposed solution by applying same on benchmark dataset. Performance is evaluated on the basis of different metrics and represented in the forms of graphs.

Chapter 5 concludes the research in simpler words and highlights the potential future directions for research community.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter comprises overview of different Machine Learning and Deep Learning techniques for mitigating various attacks and thus securing Internet of Things (IOT) network as stated in existing studies. It also includes taxonomy of intrusion detection in IOT using Machine Learning, Deep Learning and Hybrid algorithms. Diversity of IOT applications make its more prone towards security breach and considerable growth in ML/DL techniques in the recent era could be used to address the issue. Thus, previous ML/DL schemes for detecting malicious behaviors in IOT network along with their benefits and limitations are inspected which provides different research challenges.

2.2 Architecture of IDS

IOT architecture is usually explained as three layers [35],[36]. The functionality of these layers boosts up the security risk [35]. To secure an IOT network, intrusion detection techniques are vital which are categorically called as Signature Based and Anomaly Based [37]. Anomaly Based Detection is used to distinguish between normal and abnormal actions by continuously noting down and maintaining the normal behavior of the system [37]. Hence, Different Machine Learning, Deep Learning Models are used to detect anomaly [35],[3].

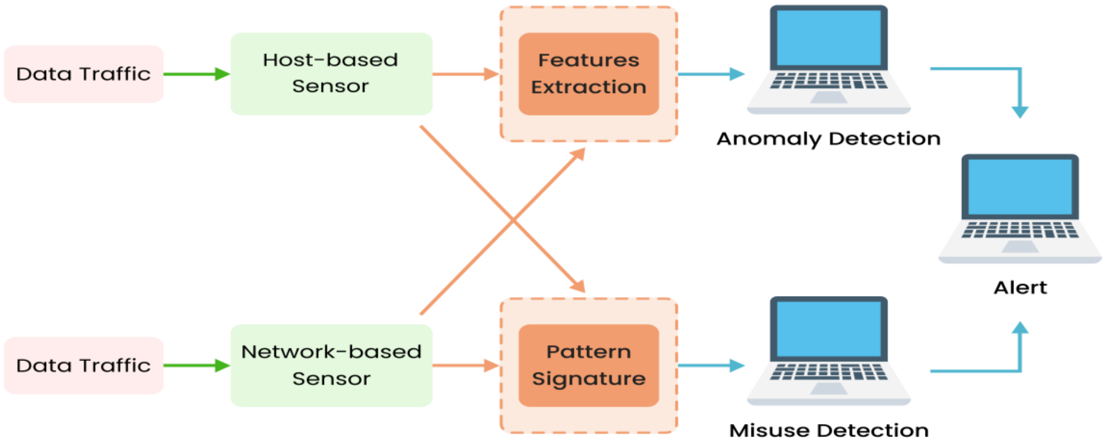
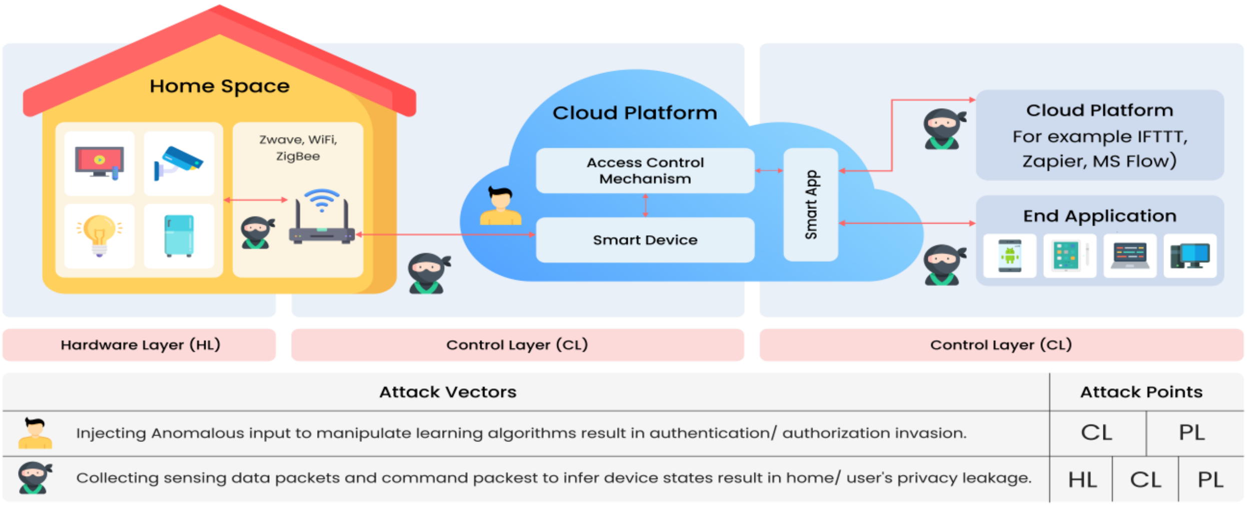


Figure 2.1: Working of IDS

Machine and Deep learning based intrusion detection was done by many researchers in different fields like smart home, smart healthcare, smart grid or other networks. Applications of ML algorithms in smart home is shown in Fig 2.2 influenced by the research.



Attack Vectors	Attack Points		
Injecting Anomalous input to manipulate learning algorithms result in authentication/ authorization invasion.	CL	PL	
Collecting sensing data packets and command packest to infer device states result in home/ user's privacy leakage.	HL	CL	PL

Figure 2.2: Intrusion detection in Smart Home

2.3 Schemes/Models for Intrusion Detection in IOT

The classification of different types of attack is extensively handled by Machine Learning and Deep learning models. The Machine learning models or methods for anomaly detection are categorized as labelled learning, unlabeled learning, labelled + unlabeled learning and trial/error learning. There are multiple algorithms and models in these categories which are able to deal with different type of data samples and feature sets. Based on the specifications of model and scenario, different types of attacks could be handled and performance could be measured with performance metrics. Moreover, to validate these models for IDS in IOT, there are wide range of public datasets on which these models could be trained. Most common are KDD99, NSL KDD, BOT-IOT, TON_IOT, Android Malware and IOT 23 etc. Taxonomy explaining categorization of different ML/DL schemes is given in FIG 2.2

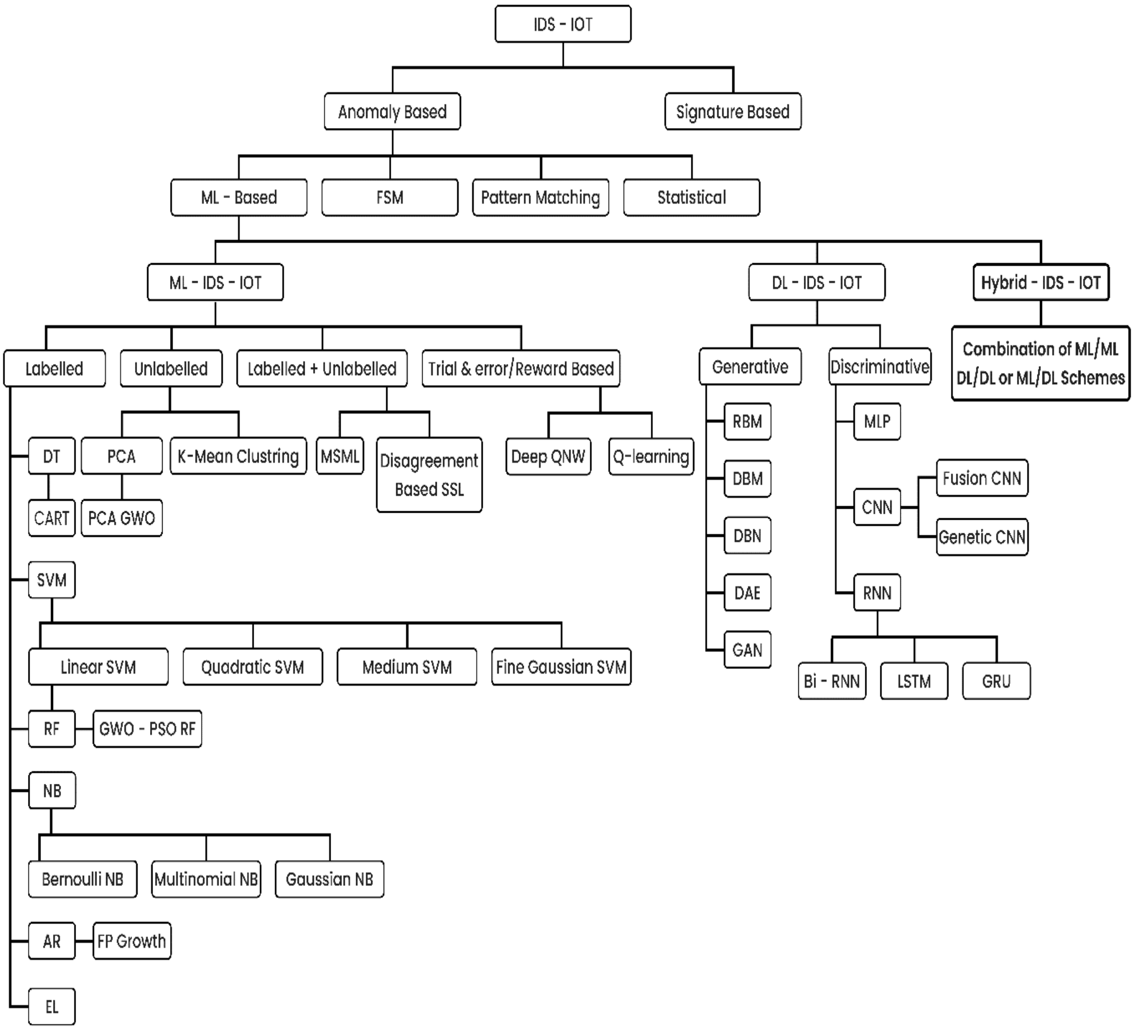


Figure 2.3: Taxonomy of IDS-IOT w.r.t. ML/DL in SLR

2.3.1 ML Models for IDS-IOT

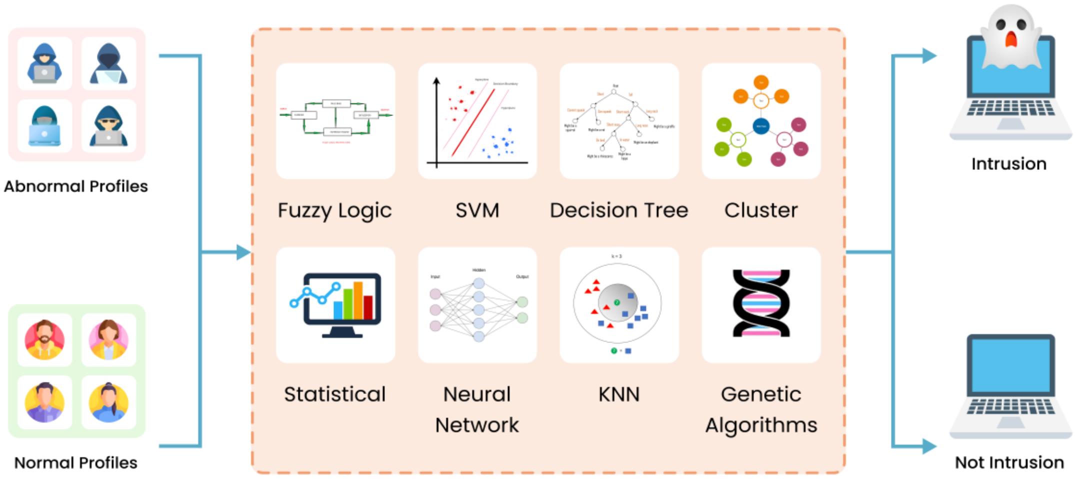


Figure 2.4: Anomaly Based IDS using ML

Decision Tree is most frequently used Machine Learning model. It uses top down strategy. In [38] CART algorithm based on Decision Trees was used to split the parent and child nodes based on Gini index criterion. Ensemble classifiers utilized the results of multiple Decision Trees through voting. It means that multiple classifiers were used for the selection of sample classes through voting rather than a single model [38]. CART Decision Tree is famous for Classification and Regression. Combinations of three decision tree was used in [38]along with NSL KDD dataset simulated in MATLAB which resulted in improved performance and accuracy while detecting variety of attacks like DOS, R26 and Probe etc. However, the time of Modeling was increased due to combination of Trees which could be ignored or manageable after further research and testing on it. Moreover, the architecture of Decision Tree needs high storage and could be understood easily if numbers of trees are not high [35]. Decision Tree and other Machine Learning algorithms were used to classify normal and malicious traffic in IOT network and the evaluating Dataset was Sensor 480 which resulted in high performing Decision Tree in comparison to SVM, Adaboost and Naive Bayes [39]. The building of Training Dataset with high volume of Data to mitigate intercepting activities in IOT network while developing ML model is a challenging task. Security Features are also vital to be considered in an advance level for dealing with network attacks [39].

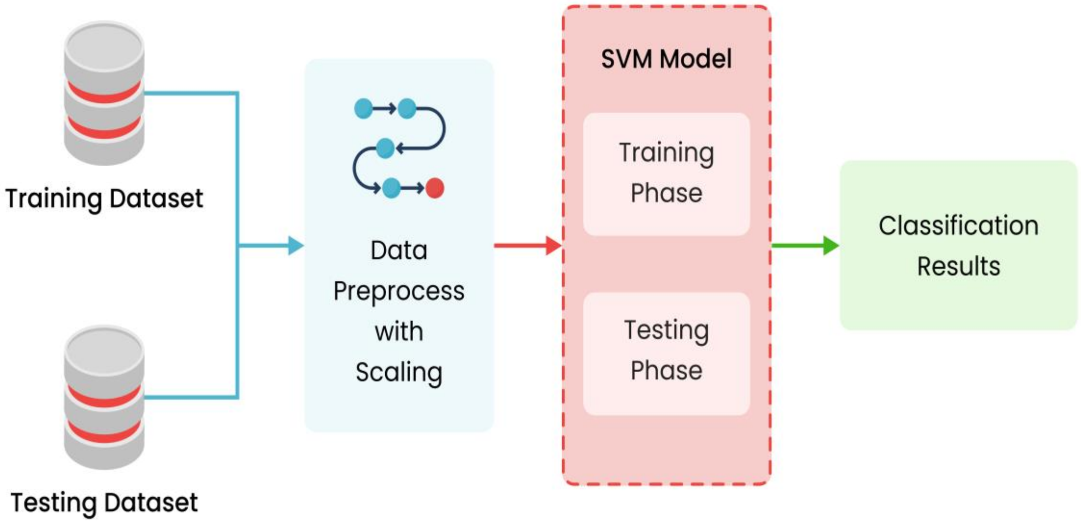


Figure 2.5: Experimental setup for IDS using SVM

Support Vector Machines are used for intrusion detection through hyperplane [40],[41],[42]. In [7] nonlinear SVM model was proposed for intrusion detection as it was more suitable for UNSW-NB15 dataset with difference in values. Non-linearity model made it independent of the data values which was the requirement of the UNSW NB15 Dataset. The proposed model showed effective performance both in Binary class and Multiclass Classification. In [6] different SVM techniques named as Linear SVM, Quadratic SVM, Fine Gaussian SVM and Medium Gaussian SVM on NSL-KDD dataset were simulated through MATLAB. Linear SVM involves linear kernel and is used when data is linearly separable. If z_s and z_t are data points, then kernel in this scenario is:

$$k(z_s, z_t) = z_s z_t \qquad \textbf{(2.1)}$$

In Quadratic SVM, kernel is

$$k(z_s, z_t) = (1 + z_s z_t)^2 \qquad \textbf{(2.2)}$$

Fine Gaussian SVM showed clear difference between classes with kernel sqrt(P)/4 (P are the predicators). Medium Gaussian showed less differences between classes where Kernel is sqrt(P). Analysis was done through ROC and confusion Matrix. Fine Gaussian SVM was high performing among other SVM techniques with minimal error rate. Real time Dataset could be involved in future to see more of these SVM techniques for IOT security in terms of intrusion detection. The widespread of IOT makes it vulnerable to attacks mainly DOS attack so lightweight IDS was developed based on Genetic Algorithm-SVM [42]. In [8], number of experiments were done and packet arrival rate attribute was used with proposed SVM algorithm which provided good performance in detecting DDOS attacks. GA-SVM minimized the computational time as it involved feature selection through GA in training phase. Moreover, there were other values that varied according to the need and optimal value is selected through hit and trial. Time window and kernel parameters were controlled. If number of dimensions were increased, then it could be challenging for classifier. More number of attacks is indicated as future implication in research.

In [43]KNN and LSTM were used for protection against illegitimate user in IOT network. It took three phases. In preprocessing, normalization of data was done in R [0,1] through min-max function. After preprocessing, feature selection was done through which best features for intrusion detection were selected. Finally, KNN and LSTM were implemented to detect intrusion if any. The functionality of KNN resembles clustering [43]. Grouping of instances was done according to the value of K and distance measured. LSTM was used as it is able to minimize the error rate by calculating difference between expected outcome and original outcome and then adjust these calculations through varying values of weights and bias accordingly. Simulation was done through MATLAB and BOT-IOT dataset was used. Mean, Detection time and Kappa stats were evaluating parameters for performance check. Detection time was needed to recognize attack whereas mean value was used to set and balance TPR and TNR. Accuracy was shown through kappa stats. The comparison of KNN and LSTM was also drawn which determined that LSTM had no under fitting and over fitting flaws so it was better performing algorithm in the scenario. More attacks and high number of instances in real time IOT scenario needed to be extend from this work.

In [44] Naive Bayes Model was utilized in which the probability of intrusion for a specific attribute was calculated and repeated for each attribute during training phase. In testing phase, time taken to calculate probabilities is proportional to n as worst case. Intrusion detection over KDD 99 cup Dataset through NB was done which shows good performance as classifier using Multiclass Classification. However, as NB has two layers with no interference between information nodes which causes limited paradigm to this work. It could be minimized using event based classification and active environment where samples have dependent features. There are three more types of Naïve Bayes algorithm [45] named as Gaussian NB, Multinomial NB and Bernoulli NB. In Gaussian NB, probability is calculated as

$$P(a|x) = [P(x|a)P(a)]/P(x) \qquad \textbf{(2.3)}$$

Where P (a | x) is the posterior probability, P(a) is the prior probability of attack, P (x | a) is the likelihood which is the probability of predictor given class and P(x) is the prior probability of predictor.

In Multinomial NB, continuous dataset with discrete count is used whereas in Bernoulli NB, both discrete and categorical data could be used but feature vector should be binary [45]. The experimental setup in [45] used Gaussian NB as it was aimed to cope up with more than two groups of attacks. Moreover, sklearn library of python was used to evaluate all parameters over KDD dataset. PCA was also used to reduce the attributes and execution time of KDD dataset over KDD dataset which exhibits better performance than traditional Naïve Bayes. However, if number of components are increased then it will affect accuracy which could be a challenge to overcome in future.

Grey Wolf optimization(GWO) and Particle Swarm Optimization (PSO) were used for feature extraction and selection and Random Forest was used as classifier for intrusion detection through simulation in Python language on KDD 99, NSL-KDD and CIC IDS 2019 datasets [46]. Random Forest is a bagging classifier as it includes multiple DTs. RF has multiple DTs in which each DT is involved in voting and RF collects the predictions and selects highly voted features for classification, hence, RF is robust to overfitting and has no noise-sensitivity [47]. RF has high variance and low bias but with the GWO-PSO-RF problem, biasing problem was solved [46]. Hence, it showed optimal results but it could also be implemented in real learning environment depicting IOT security to ensure its performance and distillation technique in IOT-NIDS could also be incorporated to enhance performance. Moreover, new datasets could also be developed for IOT-IDS as in [48], new Dataset ToN_IoT was proposed as a representation of Normal and malicious activities in IOT network and ML and DL based Classifiers were used to see its effectiveness. RF and CART worked best with the proposed dataset however, hyperparameter optimization could be done with genetic and Bayesian algorithms.

Security breach is always a concern for Routing Protocol in Low power and Lossy Networks [47]. For this the study [13] explained the initials of developing IOT-NIDS using ML and DL techniques for detecting routing attacks against RPL. For this, binary class classification and multiclass classification datasets were generated by simulating routing attacks and processing the captured traffic. Then different ML and DL classifiers were trained to build IDS. 5-fold cross validation strategy was used which depicts RF with lowest fitting time and accuracy equals to other ML models like KNN and DT whereas DL models like MLP, NB and LR could not perform very well in the scenario.

To overcome Routing attacks, four Ensemble Learning ML models were implemented on RPL-NIDIDS17 dataset [49]. Ensemble learning models were Boosted Trees, Bagged Trees, Discriminant and RUSBoosted Trees and dataset contained packet traces of Sybil, Clone ID, Black Hole and Hello Flooding etc. Preprocessing of data was done through cleaning, one hot encoding and scaling methods. Missing values were handled through cleaning, one hot encoding converted nominal data in numerical form and scaling was used to scale it between 0-1. After, preprocessing data was converted into train and test samples and four Ensemble Learning Models were trained on train sets and then tested to see the expected outcomes in terms of attack detection as normal or attack class [49]which depicted the good performance of EL ML models. However, lightweight solution for securing smart nodes in IOT network would be the target for future.

To improve the feature sets, Association rule mining techniques such as FP Growth algorithm could be used as in [50], after improvement of feature sets through FP growth algorithm, the CNN model was implemented for detection of Botnet attacks which gave higher accuracy then existing features. However, number of attacks, sample size and more ML/DL models with tested threshold value could be a challenge for future work.

K-Means clustering model is unlabeled ML model used for clustering by calculating Euclidean Distance [26], [51]. To enhance the performance of the intrusion detection, Classification and detection phase was completed through hybridization of K-Mean and SVM algorithm [26]. Clustering of normal traffic and malicious traffic was done through K-Mean and then SVM was implemented for classification of normal and abnormal instances with 70% training and 30% testing data. K-Mean+SVM gave low false rate and high accuracy versus K-

Mean and SVM model as a separate model. More models and attacks could be incorporated in it to see effectiveness of other models in the scenario.

For feature extraction and reduction of dimensionality in feature set, Principal Component Analysis ML technique is used [52], [19]. In [52], PCA was used to reduce dimensionality and then its performance was tested through IOT testbed scenario through KNN model. Results declared the effectiveness of PCA+KNN after comparing it with the results without PCA. In [19], hybridization of PCA with GWO algorithm was done to reduce the dimensionality of feature set to more extent and then it was further hybridized with different algorithms like RF, SVM and DNN for DDOS detection in IOMT. Evaluating parameters like accuracy, sensitivity and specificity declared PCA-GWO-DNN model the best among others. However, it could be used for multiclass classification in future.

In the real network scenario, it is impossible to have labelled data in all scenarios, there are higher chances of unlabeled data as well therefore semi supervised machine learning works best in the scenarios like these [24],[53]. To address class imbalance problem and non-identical distribution problem, Multi-level semi supervised ML (MSML) was proposed [24]. The limitation of the study was that hyper-parameters were not flexible enough. Moreover, distributed environment to speed up the training of model is also a challenge for researchers. In [53], Disagreement based SSL was used for IDS on DARPA dataset and in a IOT company which depicted it performs very well for detecting abnormal activities. In Disagreement based SSL works on the basis of disagreement between base learners. Through comparison, it was evident that this technique not only improves detection rate but also minimized false alarm rate. However, large number of samples is still a dispute to see same level of effectiveness.

In Reinforcement Learning model, the agent has no prior information and learns through trial and error about the actions with higher number of awards and thus is called as reward based model [35]. To predict the cyberattacks, Q learning model was used and the problem of QOS control was managed by RL learning algorithm [54]. The RL based model was also compared in terms of accuracy and precision with other DL models with significant performance and AUC was also improved. However, the increased epoch caused decrease in precision. More DL models with different calculation should be trained to make an effective IDS.

2.3.2 DL Models for IDS-IOT

To make cost effective solution for real time intrusion detection, Multi-Layer Perceptron (MLP) was trained on NLS-KDD dataset which was updated version of NSL-KDD dataset by [55]. MLP is fully connected Feed forward neural network with input layer, one or more hidden layers and output layer [55] [56]. The MLP is Discriminative Learning Deep Learning model in which derivatives of weight are calculated by Back propagation algorithm and weights are updated by the Gradient Descent Algorithm [55]. Intrusion detection in IOT system was inspected by training of two MLP models with 26 input layer neurons, 9 hidden layer neurons and 2 output layer neurons on NSL KDD dataset and KDD 99 version. Performance was calculated through different parameters while implementing these models in the Arduino and training of model was done in Python due to availability of large number of libraries in the language. The result not only showed increased performance but also ensured low power consumption due to Arduino. In future, more attacks and upgraded versions of datasets with significant feature sets could be used to secure IOT network. Moreover, unsupervised learning could also be implemented to overcome new attacks.

Another supervised Deep learning model for attack detection in IOT systems is Convolutional Neural Network (CNN) used by [57]. Feature set was encoded into a digital matrix which was further used to detect intruding activities using CNN model. During training session, the input was digital matrix and the output was weight array in neuron links. While in detection phase, preprocessed data was extracted and significant feature set was gone through calculation model on the obtained weight set in training phase to detect anomaly. To overcome the problems in classical neural networks that arises due to connectionless nodes in layers, CNN emerged as promising one due to availability of convolutional layers, polling layers, fully connected layers and dropout layers [58]. Data clustering was used as novel method for intrusion detection by [58] in which feature data was divided into four parts and clustering helped to learn the high level relationships between global features. The input was converted into images as CNN was used as classifier. Single CNN structure was used to train and test the data in four parts and then to obtain better results the results of for single CNN model was merged through multi-CNN fusion method on KDD dataset which gave 86.95% and 76.67% accuracy for binary class classification and 81.33% and 64.81% accuracy for multiclass classification. Fusion of deep learning models in online learning environment would be the goal for future to secure industrial IOT systems. The combination of IFS, CNN extractor and BG classifier known as Genetic CNN was implemented in [59] to mitigate dangerous network attacks. It gave high performance and ignored overfitting problems which is aimed to use in practical environments to secure them against illegitimate activities. To handle zombie attacks, a novel Adaptive Swarm Optimization CNN(APSO-CNN) algorithm was proposed by [60] for IOT intrusion detection. PSO algorithm with varying inertia weight was used to optimize parameters of one dimensional CNN, the cross-entropy loss function value of validation set was obtained from the first training of CNN which was considered as fitness value of PSO. The comparison was also done through new evaluation method between prediction probability and prediction label. Finally, the comparison of APSO-CNN with other models show the effectiveness of APSO-CNN for intrusion detection. The limitation of the work was time complexity of heuristic search algorithm which needs to be work upon in future. Selection of effective features to detect new attacks is also a challenge.

Recurrent Neural Network is artificial neural network with reflexive feedback connection from neurons. RNN is supervised learning model which is able to see abnormalities in data and thus recommended for intrusion detection [61]. In [61], prediction about next packet that whether its normal or abnormal is done through analyzing industrial IOT data through distance calculation by training Long short term Model (LSTM). Predicted data was compared with actual packet to see whether its normal or not. Cosine similarity was used for anomaly detection by setting boundaries which gives higher intrusion detection performance than many other data mining techniques. However, due to complex structure of IOT systems, more research is needed to find the optimal solution to even get protected from hackers. Protocol Based DID dataset was used in [62] which reduced number of features by comparing it with UNSW-NB15 and BoT-IoT datasets and LSTM was used as classifier with promising results but misclassification of DOS and DDOS occurred due to similarities between their features which could be mitigate in future.

Smart environment is getting into limelight since last decade but with the evolving smart environment and IOT sensors, also comes a security threat more commonly DDOS attack [63]. Therefore, A. Elsaeidy et al. represented intrusion detection system where Restricted Boltzmann Machine (RBM) was used to select best features as RBM was able to handle unlabeled data as well and then multiple algorithms like FFNN, RF and SVM were used as classifiers to see and compare their effectiveness in anomaly detection. FFNN was best performing classifier for attack detection in terms of accuracy. More types of attacks with new generated datasets could be done to improve the results.

To detect different types of attack, intrusion detection model was presented in which Genetic algorithm(GA) and Deep Belief Network(DBN) were combined to obtain desired results [64]. Optimal Network structure was produced through iterations by GA and then this network structure was used for IDS to classify different attacks. DBN is combination of multiple RBMs and usage of GA-DBN helped to develop a generalized model for IDS with less complexity and increased accuracy. However, other parameters should be considered that help to reduce training time with high accuracy in classifying attacks for IOT.

DAE is an AE with more than one hidden layer which allows DAE to learn complex patterns of data more easily. The encoder E1 encodes input Y, the second encoder E2 encodes the output of E1 and the third encoder E3 encodes the output from E2. The encoding on the middle layer could be stated as Z=E3(E2(E1(Y))) [65]. To extract best features from raw data and present it as low dimensional features, Deep Auto-encoder (DAE) was used in order to improve the efficiency and effectiveness of attack detection and classification (Binary or Multiclass) on NSL-KDD and CIC-IDS2018 datasets. Hyperparameter tuning was done through Random Search after the DAE extraction and then different DL models were trained to classify attacks with increased accuracy. The working of more DL models and feature extraction methods needed to be study for future.

To deal with imbalanced data, Generative Adversarial Networks(GAN) was used to resample the data and then Random Forest(RF) was used as classify different attacks which showed that minority classes were also classified correctly as normal classes [12]. The comparison of GAN with SMOTE resampling data was also done which shows high accuracy of GAN-RF method in classifying minority classes by avoiding overlapping of classes and noise problems. The incorporation of an Auto Encoder (AE) before resampling with GAN to enhance the performance is a challenge.

Deep Neural Network (DNN) was used by initializing four-layer neural network [66]. Network was trained with stochastic gradient descent through Back-propagation. The input data was propagated through hidden layers and transformed to final output. The weights were updated for each epoch. ReLu, sigmoid were used as activation function. DNN model was using early stopping algorithm to attain best validation accuracy. LSTM was also used along with DNN to get time related non-linear dynamics in the data. Four layered LSTM was used and connected dense layer was stacked to process the outputs of LSTM model. LSTM also used early stopping algorithm to attain best validation accuracy's value. Ensemble learning was used to utilize stacked approached of DL models with combinations of DNN and LSTM followed by a Meta-Classifier to detect anomalies with increased performance on heterogeneous datasets like IOT23, LITNET2020, NetML2020. The implementation can be further extended if more datasets are incorporated and additional computational methods like Apache Spark are utilized in the future to speed up the processing speed.

An intelligent network IDS model was proposed in [67] that was comprised of two parallel Auto Encoders, both were having three successive layers of convolutional filters. The decoder part of the APAE was different from encoder with eight successive convolutional layers. Positional self-attention and channel self-attention were used to enhance features. This helped to let APAE detect minority classes as well. The evaluation of model was done through UNSW-NB15, CICIDS2017 and KDDcup99 datasets which clearly showed that APAE as good performing model. However, the difference between accuracy is not very high but as it was lightweight algorithm due to usage of less parameters and required less resources therefore, it is still considered as optimal solution for IDS-IOT. Further, more DL/Models could be used as stacked or ensemble approach to monitor their performance.

To minimize False Detection Rate in network, HCRNN- based model was developed in [22] which utilized CNN and RNN structures. Feature extractor in CNN had convolution and pooling layers. The feature map as extracted output became input of classifier. However, it missed the temporal dependency of features for which RNN layers after CNN layers were used in HCRNN. The model was evaluated by implementing on CSE-CIC-IDS 2018 which shows high accuracy and good detection rate. Real datasets could be used to test the effectiveness of HCRNN has become a challenge for researchers.

2.3.3 Hybridization of ML/DL Models for IDS-IOT

There were many approaches based on Machine learning models and deep learning models to cope up with different types of malicious activities in existing studies but there were few limitations to them which could be avoided by exploiting Hybrid Machine Learning or Deep Learning model as a new model/approach [68],[69],[70]. In [37], Deep Learning model CNN was used to learn features from IOT network which were given to LSTM as input. In this way, LSTM classified the malicious activities. IOT 23 was the dataset used and hybrid CNN-LSTM model trained on IOT 23 undergoes no underfitting or overfitting problems. CNN-LSTM model gives 96% accuracy which could be further improved by considering different scenarios and DL models. In [69], CNN along with Gated Recurrent Neural Network (GRU) were trained on six datasets to prevent IOT network from various kinds of attack while ensuring high accuracy and evaluation on the basis of precision, Recall and F1 score both in binary class classification and multi class classification. Recursive feature elimination was also used to extract important features and overall model worked fit for IOT-IDS but there is still a room to explore more Deep Learning techniques and their comparison with CNN-GRU. Another hybrid model named as hybrid deep random neural network (HDRaNN) was presented to palliate network attacks in IIOT [70]. HDRaNN combined DRNN with MLP with dropout regularization over DS2OS and UNSW-NB15 datasets with 98% and 99% accuracy.

Feature extraction and selection are significant steps for good performing attack detecting system in a network, hence in [25] feature selection utilizing Information Gain (IG) and Gain Ratio (GR) was done. It used insertion and union of subsets on top 50% IG and GR features. The result obtained showed high accuracy and thus feature selection plays a vital role in enhancing performance of IDS. More algorithms could be used for feature selection and their influence on IDS in the future.

M. A. Khan et al. represented hybrid DL model incorporating CNN+LSTM on ICSX-UNB Dataset where ML Spark was used for normal behavior classification whereas CNN+LSTM were used as Classifiers with significant accuracy and detection rate however it did not involve recent datasets which could be done by researchers and attacks usually tend to change their patterns as well [71].

PCA+KNN was used by R. Wazirali et al. in which feature extraction was done with PCA and after hyperparameter tuning, KNN was used as classifier [29]. It required low energy while attaining high detection rate whereas it is not generalizable because altering data caused degradation in performance so work is needed to make it as generalized solution.

AAE and GAN were used to avoid noise in the data and latent representation of the data whereas KNN was used as Classifier on IOT 23 Dataset [72]. Both techniques showed good performance but GAN outer performed in terms of accuracy. Instances of Minority classes could have been increased and Features Resemblance Method could have used for detection of new attacks.

To secure IOT based smart environments, multiple studies had been conducted as in [73], three-layer architecture was presented in which multiple supervised ML algorithms were used to classify normal behavior, malicious behavior and types of attacks. Nine Classifiers were used for three-layer structure in which J48 Tree structure performed the best in terms of accuracy and F measure. Deep learning algorithms could be implemented in the same manner, to compare their effectiveness with ML based algorithms, should be targeted research paradigm.

2.4 Comparison of ML/DL based IDS-IOT

This section comprised of Tabular representation for comparative analysis of different Machine Learning, Deep Learning and Hybrid algorithms caters as sensing various attacks along with some other parameters/ factors which lead towards various research ideas.

Table 2.1 : Comparison of ML/DL schemes for IDS

Algorithm/Model	**Dataset**	**Domain**	**Classification**	**Attacks**	**Performance Metrics**	**Procedure**	**Merits**	**Demerits**
Hybrid DT [38]	NSL-KDD	IOT based SG	Binary	DOS, Probe, U2R, R26	ACC, Precision, Recall, F1-Score	Combination of three DTs were used for classification while CART algorithm was used to structure tree nodes.	Improved Accuracy.	Value of Recall was lower when compared to benchmark schemes.

Linear SVM, Quadratic SVM, Fine Gaussian SVM, Medium Gaussian SVM [5]	NSL-KDD	NIDS	Multi-Class	DOS, Probe, U2R, R26	ACC, Error, ROC, Confusion Matrix	Multiple SVM based techniques were used for classifying different attacks in a network.	Fine Gaussian SVM performed really well with least error rate.	Optimization of SVM could have been done.
Non Linear SVM [6]	UNSW NB 15	NIDS	Binary, Multi-class	Analysis, Backdoor, DOS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, worms	ACC, DR, FPR	Non-Linear SVM was used because of high dimensionality of Data.	Performed well for classification of attacks.	Difference in train and test accuracy depicts Low bias and High variance which may depict overfitting.
KNN, LSTM [43]	Bot-IOT	IOT Network	Binary	DDoS, DoS, OS, Service Scan, Keylogging, Data exfiltration.	ACC, DR, Kappa Stats, Geometric Mean	Normalization was done using MinMax and then preprocessing was done through GR and IG. KNN and LSTM were classifiers.	Caters Overfitting and underfitting issues and faster learning rate with LSTM.	Difficult to select suitable value of K while using KNN and LSTM takes relatively more time and power to train.
PCA+NB [45]	NSL-KDD	NIDS	Multi-Class	Probe, DOS, U2R, R2L	ACC, Confusion Matrix	PCA was used to reduce dimensionality of data while NB was used as classifier.	Decreases the execution time by minimizing number of components.	Degradation of Accuracy if number of components are increased.
GWO-PSO-RF [46]	KDDCUP99, NSLKDD99, CIC IDS 2017	IOT network	Binary Class, Multi-Class	DOS, DDOS, Heartbleed, Botnet, Infiltration etc.,	ACC, Precision, Recall, F1 score, Support, Confusion Matrix	GWO-PSO was used to select relevant features and RF was used to classify attacks.	Balanced GWO-PSO-RF reduced biasing problem and DR of minority classes.	Real Time Dataset could have been used.
EL [49]	RPL-NIDDS17	NIDS	Binary	Sinkhole, Blackhole, Sybil, Clone ID, Selective Forwarding, Hello Flooding and Local Repair	ACC, AUC	Bagged Trees, Boosted Trees, Discriminant Trees and RUS boosted Trees were used as classifiers and ensemble through voting.	Ensembled Learning showed good performance in mitigating Routing Attacks.	Accuracy could have been improved.
FP Growth Algorithm, CNN [50]	N-BaIOT	IOT	Binary	IOT Botnet Attacks	ACC, Precision, Recall, F1-score	Associatin Rule technique FP growth was used to improve the feature set and CNN was used for classification.	Method for Improvement of original feature set was given and ACC was also good.	Number of classes and data size was small. Threshold value could have been tested. Only one attack class was catered.

SVM+K-Mean [26]	Proposed Real Dataset	IOT, WSN	Binary	DOS, Probe, U2R, R26	ACC, DTR, FPR	Clustering of normal and malicious traffic was done through K mean clustering whereas SVM was used for classification.	Updating of attack and normal traffic was done in real dataset.	More attacks could have been handled with the incorporation of more ML/DL models.
PCA-GWO+DNN [19]	Kaggle Dataset	IOMT	Binary	R2L, Phishing, Probe, DoS and U2R	ACC, Sensitivity, Specificity	PCA+GWO were used to reduce high dimensional data into low dimensional and DNN was used as classifier.	ACC was increased and time complexity was decreased.	Multi-Class problem were missing.
MSML [24]	KDD99CUP	NIDS	Multi-Class	DOS, Probe, U2R, R26	ACC, Precision, Recall, F1-Score, Confusion Matrix	Imbalance data and unknown pattern recognition was handled through MSML.	ACC was increased.	Hyperparameter optimization was not done.
Disagreement-SSL [53]	KDD99CUP	IOT	Threat Alarm	DOS, Probe, U2R, R26	Error Rate, Hit Rate	Disagreement Based SSL was used to reduce Error rate while detection of attacks.	Used for unlabeled data and labeled data with reduction in error.	Recent Dataset could have been used and more Algorithms could have been used for investigating it further.
MDP [54]	NSL-KDD	IOT	Multi-Class	DDOS, DOS	ACC, Precision, Sensitivity, AUC	Reinforcement Based MDP was used for IDS.	Gave the best precision and AUC curve.	Low Accuracy. High epoch could cause overfitting.
MLP [56]	CSC-CIC IDS 2017, CSC-CIC IDS 2017,	NIDS	Binary, Multi-Class	Bot, DDOS, Portscan, Heartbleed, Infiltration etc.	ACC, Precision, Recall, FPR	Preprocessed data was fed for hyperparameter tuning and MLP was used for classification.	Model was generalizable.	Minority class was not correctly classified which highlighted underfitting.
Multi-CNN [58]	NSL-KDD	IIOT	Binary, Multi-Class	DOS, Probe, U2R, R26	ACC, Precision, Recall, F1-Score	Multiple CNNs were used and SoftMax was used to obtain the final prediction result.	High Accuracy, Low complexity.	Online Learning tool could be used for IDS.
GA-CNN [59]	KDD version	NIDS	Binary, Multi-Class	Probe, DOS, U2R, R2L	ACC, FPR, TPR, ROC	The input of IFS was passed to CNN which was selected by GA and then DFS was generate	Avoided Overfitting.	Time Consuming.

						and BG was classifier.		
APSO-CNN [60]	N-BaIOT	IOT	Multi-Class	Ack, COMBO, Junk, Scan, Syn, TCP, UDP, UDPplain	ACC, Precision, Kappa stats, Hamming Loss	PSO algorithm was used to optimize parameters for CNN and cross entropy value by first structure of CNN was given as fitness value to PSO.	Able to detect multiple zombie attacks effectively and reliably.	Time Complexity needed to be reduced.
LSTM [61]	Generated Dataset	NIDS	Binary	Anomaly Detection	ACC	Distance measure and score was used to learn packet by LSTM.	Effective for anomaly detection with high accuracy.	Cost issue.
LSTM [62]	UNSW-NB15, Bot-IoT	IOT	Multi-Class	Dos, DDos	ACC, Confusion Matrix	Protocol Based IDS was used with LSTM for attack detection.	Imbalanced data issue and overfitting were countered.	More methods for noise removal in data could have checked.
RBM [63]	Generated Dataset	Smart Cities	Binary, Multi-Class	DDOS	F-Measure	Series of experiment were conducted by varying RBM layers to learn features from raw data and benchmark classifier for IDS.	Worked well for IDS as unsupervised learning.	Methods to investigate rich features was missing.
GA-DBN [64]	NSL-KDD	IOT	Binary, Multi-Class	Probe, DOS, U2R, R2L	ACC, DR, FAR, Precision, Recall	GA generated optimal number of neurons and hidden layer to improve DR by DBN.	Reduce Complexity, and improved DR.	Training time of DBN was high.
DAE [65]	NSL-KDD, CSC-CICIDS2018	NIDS	Multi-Class	Dos, Botnet, Brute Force, Infiltration, Web Attacks etc.	ACC, Precision, Recall, TPR, FPR, Prediction Time, Training Time	HPO was used for hyperparameter optimization and DAE was classifier.	Improved performance.	Imbalanced data was not used to check the effectiveness of technique.
GAN [12]	CSC-CICIDS2017	NIDS	Multi-Class	Dos, Heartbleed, Infiltration etc.	ACC, Precision, Recall, F1-Score	GAN was used to resample minority classes while RF was classifier.	Handled Imbalanced data effectively. Avoided underfitting/overfitting.	Compression of data characteristic by AE was required for better results.
EL [66]	IOT23, Litnet2020,	NIDS	Binary	Dos, Smurf, HTBot etc.	ACC, Precision, Recall, F1-Score, MCC	DNN, LSTM and LR were	Improved Performance.	Multi-class problem was missing.

	NETML2020					stacked to detect attacks.		Processing speed was slow.
APAE [67]	UNSW-NB15, NSLKDD, CSCCICIDS2017	IOT	Binary, Multi-Class	Dos, U2R, Smurf etc.	ACC, Precision, Recall, F-Score	APAE was used with convolutional filters to extract short and long-range information from feature vector.	Light weight solution, Good for minority class classification. Limited processing requirement	Very less difference in ACC in comparison to benchmark schemes.
HCRNNIDS [22]	CSC-CIC-IDS-2018	NIDS	Binary	Brute-force, DOS attacks, DDOS attacks, Brute-force SSH, Infiltration, Heartbleed, Web attacks, Botnet.	ACC, Precision, Recall, F-Score, DR, FAR	CNN layers after RNN layers were used.	Reduced Computational Complexity, Increased DR, ACC.	More recent dataset was needed as signatures of attack traffic keeps on changing.
CNN+LSTM [37]	Dataset from Raspberry pi infected devices	IOT	Binary	CC, HeartBeat, FileDownload, Torii, DDOS, Mirai, PortScan	ACC, Precision, Specificity, Recall, F-Measure, FNR, FPR	CNN was used for feature extraction-representation and LSTM classified data.	Does not suffer from Overfitting/Under fitting. Scalable for adding CNN Module if network is added.	Little Computational overhead for LSTM classifier if sub-network is added.
CNN+GRU [69]	IOT-DS2	IOT	Binary, Multi-Class	Dos, DDos, Mirai, MQTT Brute Force etc.	ACC, Precision, Recall, F1-Score	CNN layers were used for Normalization and feature mapping and GRU layers were used to further flatten data and avoid overfitting.	Increased performance w.r.t. Evaluation metrics.	80:20 ratio was used for training testing which could be 70:30 for more transparency.
HDRaNN [70]	DS2OS, UNSW-NB15	IIOT	Multi-Class	Dos, Fuzzers, Backdoor, Reconnaissance etc.	ACC, Precision, Recall, F1-Score, log loss, ROC, AUC	DRNN layers and MLP layers were used as hit and trial method.	Overcame Overfitting with dropout regularization.	High performance devices were required.
PCA+KNN [29]	NSL-KDD	NIDS	Binary	Dos, Probe, U2R, R2L	ACC, Precision, Recall, F-Score	PCA was used to identify critical area of data and KNN was classifier.	High DR.	Altering data caused difficulty in detection. New attacks could not be handled.
BiGAN+KNN [72]	IOT23	IOT	Multi-Class	Dos, Botnet Attacks	ACC, Precision, Recall, F-Score	Bi-GANN were used with KNN to detect unknown attacks.	Efficient detection of Zero Day Attack.	Shared Features in the data were not analyzed and required more instances of minority class.

2.5 Potential Research Directions

After reviewing state-of-the-art schemes for intrusion/attack detection in IOT based environments, following loop holes came into limelight.

- Availability of good data is foremost important in any ML/DL based IOT-IDS which was missing in most of the cases reviewed.
- Latest datasets are also a quest of researches in the said field, whereas many of them has used NSL-KDD and older datasets which could be problematic as attacks are also changing their patterns and signatures.
- Feature Engineering, Feature Selection and Hyperparameter tuning plays a significant role in enhancing the performance of ML/DL based IOT-IDS, whereas there was limited amount of work on it.
- Correct selection of method to find out rich features that could represent characteristics in best ways and assist in classifying attacks by ML/DL models was still not a part of many studies.
- There must be ways to reduce dimensionality of data from high to low, many studies were conducted but some of them could not do it efficiently.
- Noise in the data caused overfitting/underfitting issues and in imbalanced data, it is the built-in case therefore there must be ways to handle them. There were many ways to cope up noise in existing studies but they were still facing overfitting/underfitting issues due to imbalanced bias and variance.
- Time and budget constraints were also noticed.
- Generalizable ML/DL models were also less in state-of-the-art studies.
- Performance of solutions w.r.t. evaluating parameters like Acc, Precision etc., must be enhanced by splitting data fairly into train and test whereas many studies could not do justice with it.
- Hybridization of ML schemes with DL schemes was done by many researchers for NIDS security however, it must be explored on real time dataset to ensure promising solution for IOT security.

By keeping in view all these points, the proposed study is inclined towards exploration of recent real time dataset with efficient methods of finding rich features, representing them in low dimension and detection of attacks in IOT based smart environment by ensuring high performance w.r.t. performance metrics through hybridization of Machine learning and Deep learning algorithms as generalizable model.

2.6 Summary

This chapter represented deep inspection of various IDS solutions concerning datasets, feature engineering, hyperparameter tuning, classification, ML/DL models and other evaluating parameters like Acc, Precision, Recall, etc., Tables 2.1 highlighted important points of the ML/DL schemes reviewed which emphasized the importance of finding new dataset for IOT-IDS and training of hybrid ML/DL model on the dataset in an efficient manner.

CHAPTER 3

HYBRIDIZATION OF ML/DL FOR IDS-IOT IN SMART HOMES

3.1 Overview

This chapter explains the research methodology, proposed novel hybrid ML/DL scheme for intrusion detection for real time dataset mainly designed for Smart Homes. The theme of this scheme is to provide detecting mechanism for attacks in IOT environment with improved efficiency by utilizing recent dataset. Utilization of recent dataset for IOT environment after comparing the existing intrusion detection with ML/DL schemes in order to improve them by choosing correct methods of hyperparameter tuning and data preprocessing is the main idea of research. The proposed mechanism shows the complete mechanism for hybridization of ML/DL schemes on CIC-IDS-IOT 2022 dataset with objectives of improved accuracy, detection of attacks, detection of normal class and making it generalizable in nature by testing its performance on unseen data. To explain these, experimental setup, architecture of ML/DL schemes, pseudocode and equations for techniques used are also elaborated in detail.

3.2 Research Methodology

The research methodology consists of four steps. In first step, systematic Literature Review is done by reviewing mechanism, pros and cons of existing Machine Learning and Deep Learning algorithms for intrusion detection in network or IOT domain. By keeping in view these details, gaps are identified in the existing schemes by highlighting limitations and ways to overcome them mostly in terms of performance and data. It leads to different research ideas, through which Problem Statement for the proposed research is identified and stated which emphasized the need of Anomaly Based Intrusion Detection in IOT with correct ML/DL techniques to enhance performance. Research objectives were also well stated to validate the research in the end. Generalized model with improved accuracy and enough preprocessing is demanded.

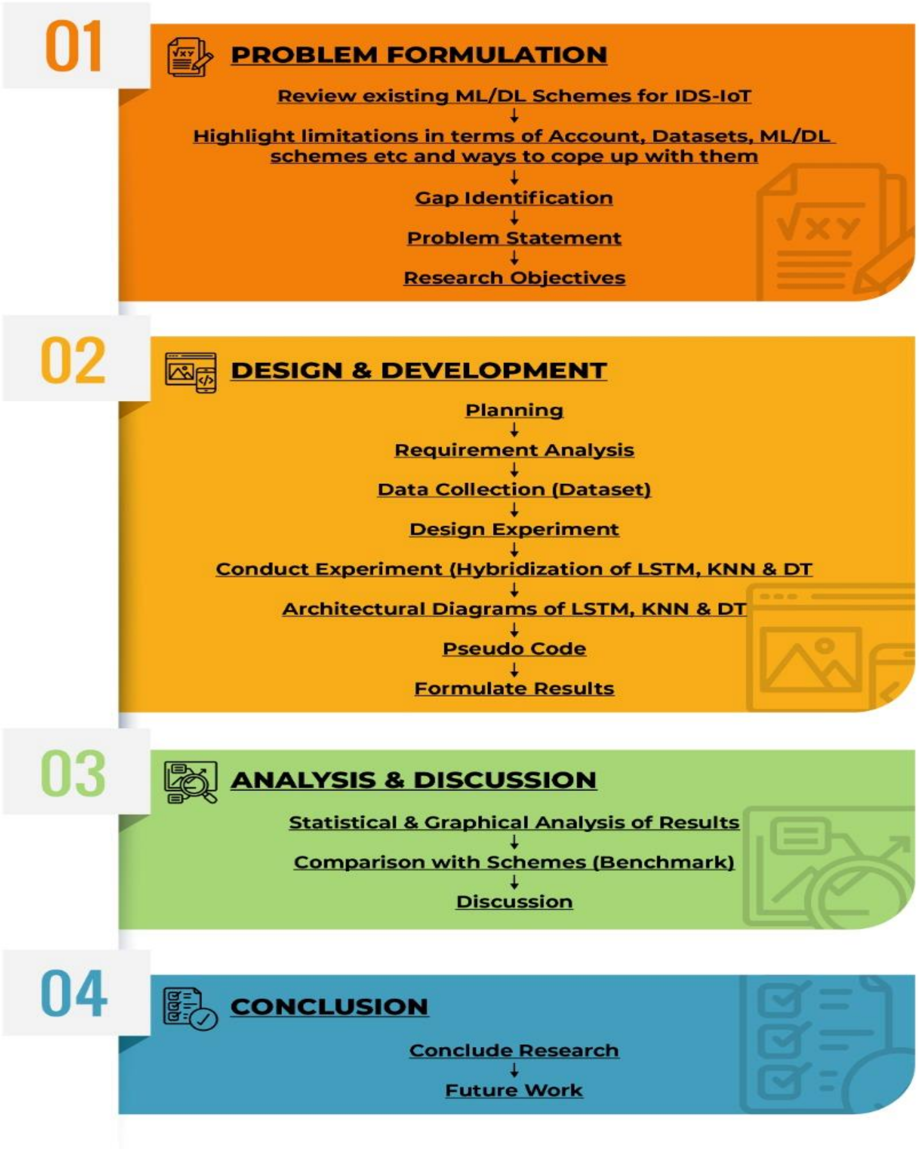

01
PROBLEM FORMULATION
Review existing ML/DL Schemes for IDS-IoT
Highlight limitations in terms of Account, Datasets, ML/DL schemes etc and ways to cope up with them
Gap Identification
Problem Statement
Research Objectives
02
DESIGN & DEVELOPMENT
Planning
Requirement Analysis
Data Collection (Dataset)
Design Experiment
Conduct Experiment (Hybridization of LSTM, KNN & DT
Architectural Diagrams of LSTM, KNN & DT
Pseudo Code
Formulate Results
03
ANALYSIS & DISCUSSION
Statistical & Graphical Analysis of Results
Comparison with Schemes (Benchmark)
Discussion
04
CONCLUSION
Conclude Research
Future Work

Figure 3.1: Research Methodology

In the second step, planning for implementing experimental setup in order to prove the objectives is done. For this, requirement analysis is done in which required resources are well mentioned. After planning, Data collection was a major problem. Most of the existing approaches used datasets of 1999 like KDD99 whereas others have no IOT traces like CIC-

IDS2017. These deficiencies lead to other problems like less reliable solution as attacks are also changing patterns which requires continuous updating of profiles in database or csv files. To cope up this, real time IOT dataset published in 2022 by Canadian website is used in the proposed research. Then Google colab is used to train and test the proposed hybrid model using LSTM, KNN, DT and Adaboost on the Dataset. Details of experiment along with architectural diagrams, mathematics and pseudo code is also given to elaborate the mechanism.

In Analysis and Discussion, performance metrics of the proposed schemes are statistically and Graphically compared with benchmark schemes by implementing them as well. Experiment is repeated with different epoch and train test split to ensure reliability and validity. Proposed Hybrid Model is also tested on benchmark dataset UNSW-NB15 to measure its effectiveness. Values of Accuracy, Precision, Recall, F1 score and confusion matrix are used as performance metrics to compare the models. The important criterion and other consequences have also been discussed.

At last, whole research is concluded in compact and precise manner with future scope of the research.

3.3 Requirement Analysis

Foremost in design and development is planning, after which resources are needed to be stated according to the requirement. For this research real data depicting IOT smart Home network scenario was needed. Further, system with GPU was required to conduct experiment.

3.4 Dataset

Anomaly Based intrusion detection relies on the datasets to increase accuracy and reduce false positive rates. However, it is comprehensive task to generate a new dataset for detection of network attacks. There are many datasets available to cater the said purpose but most of them are simulated and not representing real time data. Moreover, most of the datasets are not updated as well which cause validity of research to be doubtful. For the proposed research, different public platforms have been searched for real time data. According to the characteristics and citations, two datasets have been selected. CIC-IOT 2022 was used as the part of novel scheme whereas UNSW-NB15 was used later to see the effectiveness of proposed hybrid model on that as well.

CIC-IOT 2022 dataset is a public dataset generated for profiling, behavioral analysis and vulnerability testing of different IOT devices with IEEE 802.11, Zigbee and Z-wave. Network configuration of dataset is shown in figure. 64-bit window machine with two NICs, one was connected to network gateway while other was connected to unmanaged network switch. Wireshark was used to capture and saves output packet captured (pcap) files. IOT devices that needed Ethernet connection were connected to switch. An additional smart hub, Vera plus was connected to unmanaged switch to serve IOT devices which were compatible to Wi-Fi, ZigBee, Z-wave and Bluetooth. Network traffic was captured through six different type of experiments. Power, idle, interactions, scenarios, active and attacks states were used to capture traffic. RTSP Brute force and Flood were the types of attack launched.

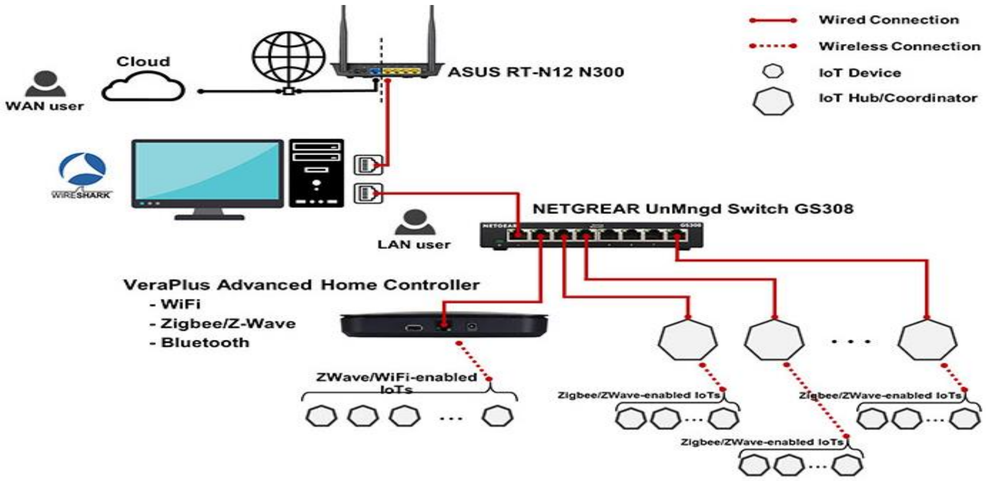


Figure 3.2: CIC-IOT 2022 dataset configuration

The dataset contains about 202,266 data instances used for training and testing purpose.

3.5 Experimental phases and setup

Experimental setup is divided into three phases which are responsible of Data conversion, Data Preprocessing, Train-Test split, Model training, Classification by Model, Evaluation as shown in the diagram.

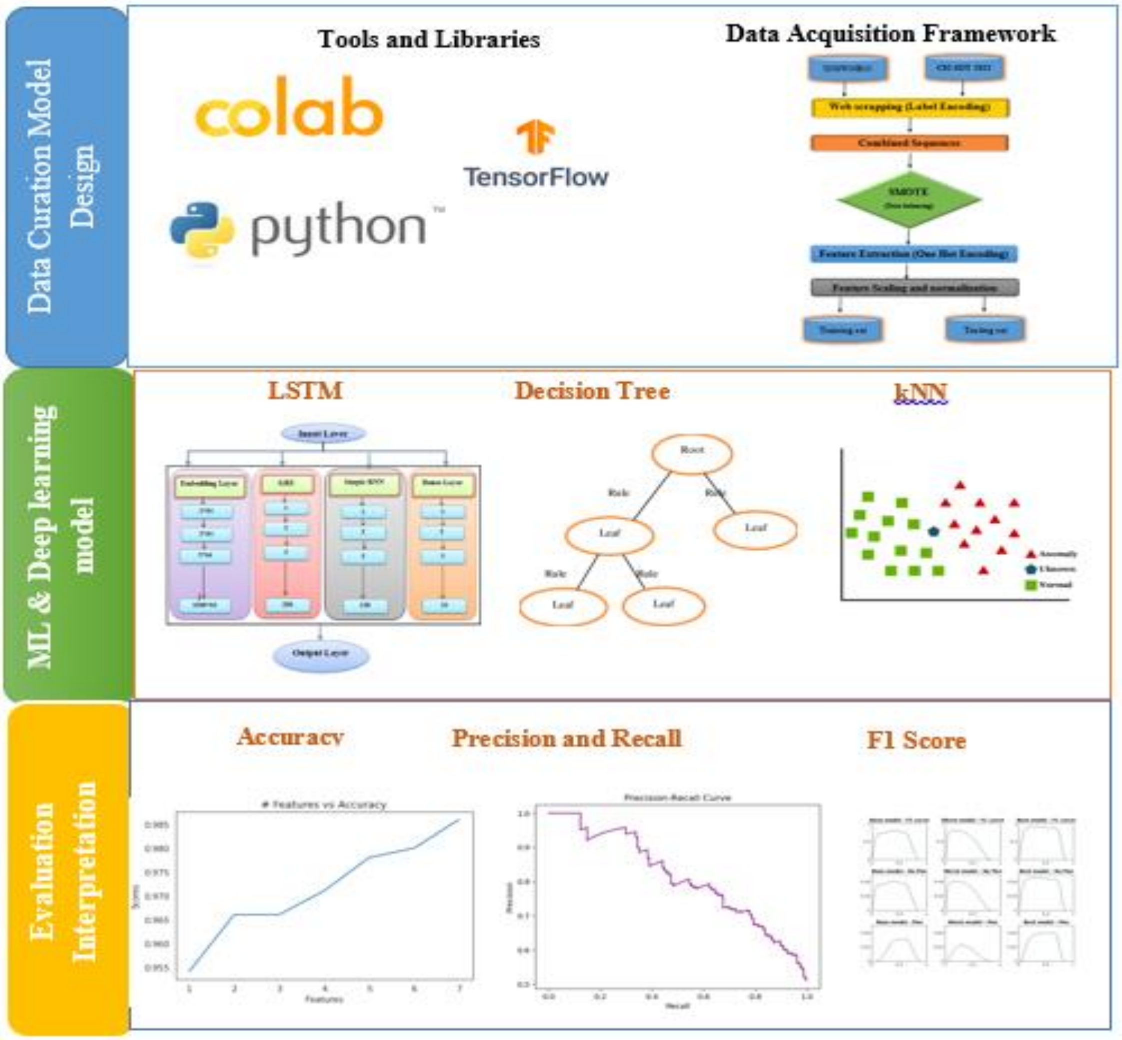


Figure 3.3: Experimental Phases of IDS-IOT

3.5.1 Data Conversion

Network captured files are by default in pcap format. However, Machine Learning models usually work with csv files. Therefore, following steps are followed for Data conversion.

i. Download pcap files.

ii. Download and install wireshark.
iii. Open downloaded pcap files with wireshark.
iv. Convert pcap files into csv files by choosing csv conversion from File Menu.

3.5.2 Data Preprocessing

Data preprocessing is an important and critical task that enhances the quality of data to promote the meaningful extractions from the data. In Machine learning, it mainly refers to cleaning and organizing the raw data to make it suitable for training Machine and Deep Learning models. There are multiple ways of Data preprocessing phase. In this research, Data Cleaning, Normalization, one hot encoding and Data reduction were used. For Data cleaning, “preprocessing.Labelencoder()” was used for transforming labels into numerical forms. It makes it better understandable by computers.

Data preprocessing the process of balancing and extracting the main features of raw data from big data for the enhancement of data features [74]. Data preprocessing is an important and critical task that enhances the quality of data to promote the meaningful extractions from the data. In Machine learning, it mainly refers to cleaning and organizing the raw data to make it suitable for training Machine and Deep Learning models. There are multiple ways of Data preprocessing phase. In this research, Data Cleaning, Normalization, one hot encoding and Data reduction were used [75]. For Data cleaning, “preprocessing.Labelencoder ()” was used for transforming labels into numerical forms. It makes it better understandable by computers. Figure 3.4 illustrates the data preprocessing for the proposed model.

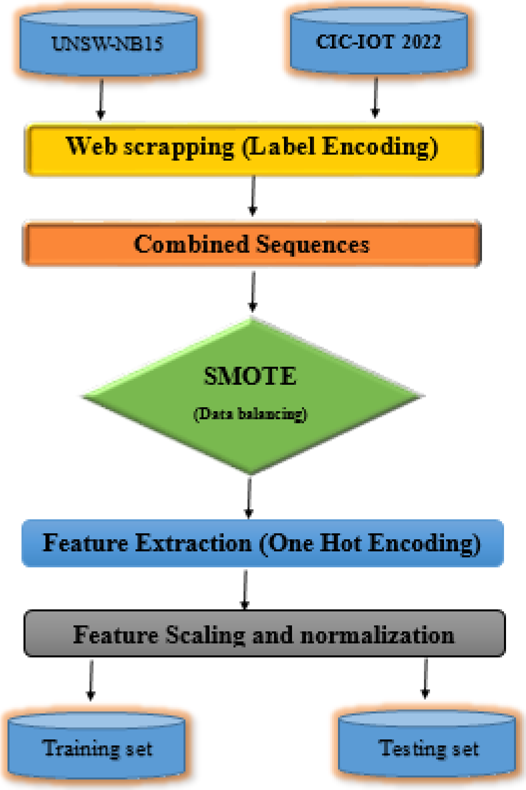

UNSW-NB15
CIC-IOT 2022
Web scrapping (Label Encoding)
Combined Sequences
SMOTE
(Data balancing)
Feature Extraction (One Hot Encoding)
Feature Scaling and normalization
Training set
Testing set

Figure Error! No text of specified style in document.**.4: Data Acquisition Framework**

3.5.3 SMOTE

The dataset use for the study is unbalanced so for balancing that dataset Synthetic Minority Over-Sampling Technique (SMOTE) technique is used. There are two common techniques used to balance dataset over sampling and under sampling.

- In under sampling technique the number of majority classes is reduced to balance the dataset. The overall data records are reduced.
- In over sampling technique the number of minority classes are increased[76].

If the dataset is imbalanced then the classification will not be equally distributed so SMOTE use for this study to balance the normal and mutated data sequences. In SMOTE technique minority sample is oversampled by creating synthetic examples [77]. In SMOTE technique the total amount of oversampling O is set up, after that an iterative process is occurs with, several steps. Firstly, random instance is taken from minority class training set. Next, its N nearest neighbor's instances are obtained. Finally, O of these N instances are randomly chosen to compute the new instances by interpolation [78]. For this the difference between feature vector under consideration and each of the selected neighbors is taken. This difference is multiplied by any random selected number drawn between 0 and 1, and then it is added to the previous feature vector. This causes the selection of a random point along the "line segment" between the features. In case of nominal attributes, one of the two values are selected at random Figure 3.5 explains how to create synthetic data points in SMOTE.

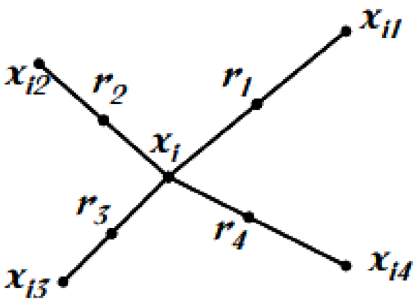


Figure 4 Error! No text of specified style in document.**.5: Creation of Synthetic data points in SMOTE**

The algorithm for SMOTE is

- Mark majority and minority classes from the dataset.
- Create the percentage of oversampling for calculating instances.
- Identify k instance in the minority class and also find its N Neighbor.
- Calculate the distance between N and K
- Multiply the answer with any number exist between 0 and 1 and add this distance in k.
- Repeat the process till required instances.

The benchmark dataset for the purposed study is denoted by D, which is defined as

D=D^{+} UD^{-} **(3.1)**

Here D^{+} considered as positive data sequences while D^{-} **is** considered data sequences and U is the union for both sequences.

3.5.4 ONE Hot Encoding

Feature is a dimension reduction process in which the data sequences are represented in such a way that interesting part represent more effectively and it reduce the calculation time for algorithm. In Machine learning the most important part of data feature extraction and pattern recognition because on the basis of these sequences training and testing process is performed [79]. For the purposed model One Hot Encoding with panda's technique is used for feature extraction from the dataset. One hot encoding allows representing data in categorical features using $log2(D)$ vectors. Here D is the dimensions that are associated with one hot encoding [80]. In this type of encoding technique each categorical value is assigned with a binary value and converted into a new column.

The feature vector of one hot encoding is represented by equation 2.

v $\in \{\mathbf{0},\mathbf{1}\}$ $\sum_{i=1}^{m} \boldsymbol{v1} = \mathbf{1}$ **(3.2)**

Here v is vector one hot encoder , m is the length of the vector [81].

v $\in \{\mathbf{0},\mathbf{1}\}$ $\sum_{i=1}^{m} \boldsymbol{v1} = \mathbf{1}$ **(3.3)**

Here v is vector one hot encoder , m is the length of the vector [81]. For the proposed study there are five classes of data as Power, idle, interactions, scenarios, active from the data sequences which is needed to be convert into which type of attack the system is facing. One Hot encoding method is used to convert the classes into one hot encoding vectors. After completing the data balancing and feature extraction the data is split into training and testing. The splitting of data saves the model from under fitting and over fitting. The dataset inputs features are in numerical format Neural network has the problem while processing this type of data due to gradient exploding and vanishing problem [82] that cause poor model performance and low accuracies. To overcome this problem the data features are scaled in the form of 0 and 1 as maximum number will be 1 and minimum number will be 0.

3.5.5 Train Test-split

Train-Test split method is utilized to measure the estimated performance of ML/DL algorithms. 70:30 is most commonly used ratio of train and test split. The train set of a dataset is used to train the model while test set is used to evaluate the performance of the model on the basis of different parameters.

3.5.6 Prediction Algorithm

For the proposed study Machine learning and deep learning algorithm are applied for intrusion detection. Machine learning and deep learning algorithms play a vital role in the detection and prediction scenarios. This study is using three deep learning and machine learning algorithms including Long term short term memory network (LSTM), K nearest neighbor and Decision tress algorithm. LSTM model of deep learning consists of multiple layers. Each layer is inspired by the human neuron and process the input. The input passes from various hidden layers and generates output. At the mean time the back-propagation algorithms take back the errors with them and learn from these errors. For every iteration of feed forward and backward pass the accuracy, precision and recall is calculated.

These learning features inside machine learning algorithms learn by itself using different learning procedures [83]. The machine learning algorithms used in the study is explained in the below section.

3.6 Architecture of LSTM, KNN, DT and Adaboost

LSTM, KNN and DT was hybridized with Adaboost for detecting RTSP Bruteforce Attack and UDP flood Attack on CIC-IDS2022. The mathematics and logic behind the architecture of these methods is explained in below section.

3.6.1 Long Short-Term Memory Network (LSTM)

LSTM is one of the most commonly used algorithms in artificial intelligence and deep learning methods. The algorithm is mostly used in the field of speech recognition, robotics, text recognition, handwriting recognition etc. LSTM is the combination of cell. Each cell contains input gate, forget gate and an output gate [84]. LSTM used for IDS-IOT consists of following parts.

Input gate: This gate passes the information to the cell layers. It determines the extent of the information that is to be passed inside the cell. This gate obtains the information coming from the previous cell.

Forget gate: Forget gate in LSTM is responsible for carrying the information. This gate decides which information pass to the next layer and discard the information that are not much necessary for the cell.

Output gate: It generates the output and pass the information to the next LSTM cell.

The algorithm of LSTM is developed for tackling the vanishing gradient problem [82]. Vanishing gradient problem occurs in deep learning model due to a greater number of layers. When there are a greater number of layers inside the deep learning model the product of derivation decreases and the value of loss function reaches to zero. LSTM tackle this problem with the help of gates ,by increasing the space of RNN model[85].

The gate in LSTM is responsible for the regulation of information from one cell to another cell. Different activation functions are applied in each gate [86][87]. Figure 3.6 explains the LSTM architecture used in the proposed scheme.

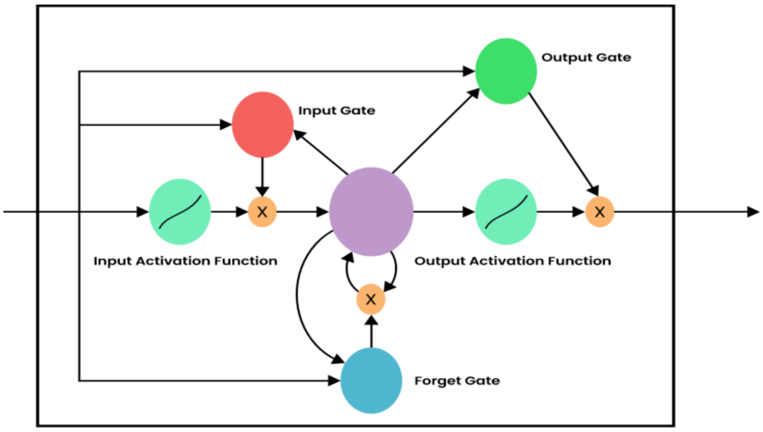


Figure 3.6: LSTM Architecture for IDS

In the figure x_t is the input at specific time and y_t is the output at specific time t. f_t represent forget gate, i_t *and* o_t represent input gate and output gate respectively. Every cell of LSTM has three inputs x_t, A_{t-1}, B_{t-1} and has two output as b_t and h_t. Equation 4, 5, 6, 7, 8, 9 explain LSTM

$$i_t = \sigma\left(y_t U^i + A_{t-1} W^i\right) \qquad \textbf{(3.4)}$$

$$f_t = \sigma\,(y_t U^f + A_{t-1} W^f) \qquad \textbf{(3.5)}$$

$$o_t = \sigma\,(x_t U^o + A_t W^o) \qquad \textbf{(3.6)}$$

$$B_t{}' = tanh(x_t U^c + A_{t-1} W^c) \qquad \textbf{(3.7)}$$

$$B_t = \sigma\,(f_t * B_{t-1} + i_t * B_t{}' \,) \qquad \textbf{(3.8)}$$

$$y_t = \tanh\,(B_t) * o_t \qquad \textbf{(3.9)}$$

In the equations x_t is the input, A_{t-1} is the previous data cell output, B_{t-1} is the previous cell memory, B_t is the current cell memory, W and U are the weights for the forget, input and output gate. Different activation functions are applied inside the gates. Tanh and sigmoid are the most commonly used activation functions inside the gates.

Tanh Function

Tanh function is used to regulate the flow of network. It maintain the value of the network between -1 and 1 [88]. Figure 7 explains the curve of Tanh function

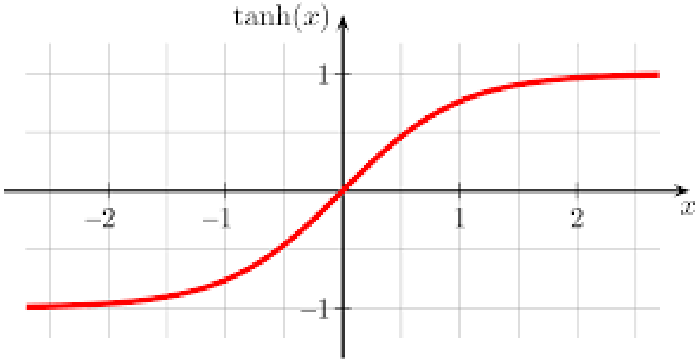


Figure 3.7: Working of Tanh Function

Tanh function allows the values of the gates to remain inside the boundaries. When the values passed from the network it undergoes changed due to a lot of mathematical functions implemented inside the LSTM cell. Tanh function ensure that the values will remain in the boundaries and thus regulate the output.

Sigmoid Function

Sigmoid activation function is also most commonly used activation function in feed forward neural network that works just like the tanh function but it regulates the values between 0 and 1 instead of -1 to 1 [89]. In LSTM cell sigmoid function is used to update or forget the

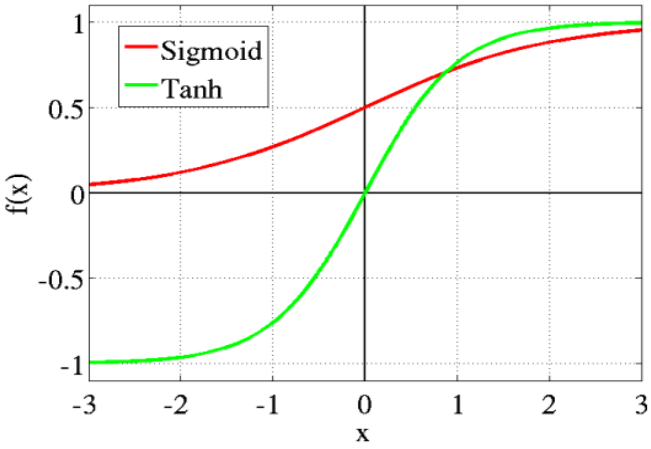


Figure 3.8: Comparison between Tanh and sigmoid functions

data. In the cell if the value of the information is 0 that indicate to forget and 1 indicates to forward the information. Figure 3.8 explains the comparison curve between sigmoid and tanh.

In the proposed study LSTM is used for the intrusion detection from the real time dataset. Figure 9 explains the LSTM layers for the proposed model.

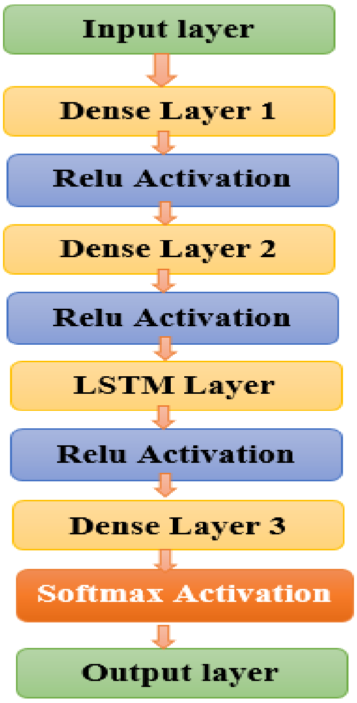


Figure 3.9: LSTM layers for Proposed Scheme

The sequential deep learning model is used in the current scenario. In this study one input layer along with 3 dense layers, one LSTM layer and one output layer is used. In each of the dense layer Relu activation function is used. Relu is a rectified linear unit that works on min, max principle. The working method of Relu is explained in equation 10 and 10 a

$$\text{Relu} = \text{-ev } (\text{-ev}, 0) = 0 \qquad \textbf{(3.10)}$$

$$\text{Relu} = \text{+ev}(\text{+ev}, 0) = \text{+ev value} \qquad \textbf{(3.10a)}$$

The Relu activation function turns the classification into 1 if the value is greater than zero and makes the classification to zero if the value is below zero. The dense layer from the model receives the input from all the previous layers and classify the output based on the output from the convolutional layers. The LSTM layer helps in the gradient flow. The LSTM layer is responsible for taking the data from the input layer and the dense layers, calculates the parameterized vector from these layers and apply activation functions for element wise on each gate. After applying the two dense layers on the dataset the LSTM layer with RELU activation is applied.

SoftMax activation function is applied in the last dense layer. The SoftMax activation function is used for determining the probability of the class from which the input data belongs [90]. The output of the SoftMax activation is equal to the number of classes from where the data belongs. This is also known as probability distribution. And the sum of all the classes is equal to one. In our model we have five output classes as Power, idle, interactions, scenarios, active. The formula for calculating the SoftMax activation function is explained in equation 10

$$P\left(y = j \setminus \theta^{(i)}\right) = \frac{e^{\theta^{(i)}}}{\sum_{j=0}^{k} e^{\theta^{(i)}}} \qquad \textbf{(3.11)}$$

In the equation θ represents the one hot encoding matrix, and j is the set of weights. Figure 10 represents the probability of each class in the proposed model.

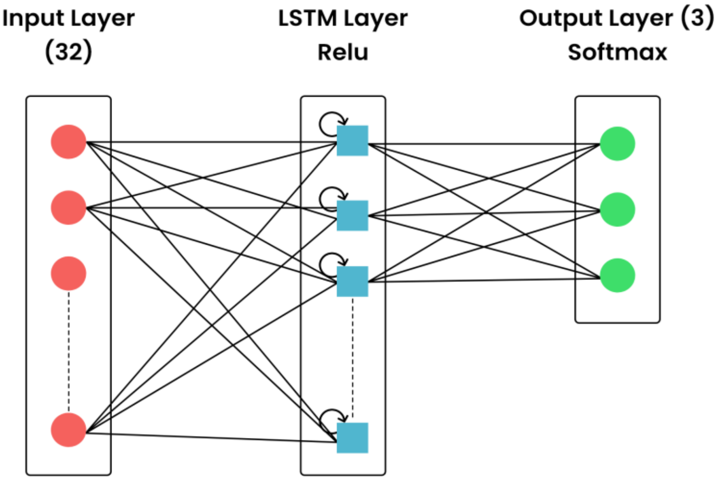


Figure 3.10: The probability of the instances of each class in LSTM (proposed)

3.6.2 K Nearest Neighbor Algorithm

KNN is a machine learning problem used for both classification and regression problems. This algorithm stores the data and classify the new data according to the similarity with the data classes. The algorithm for KNN is as follows [91]

1. Input different classes of a sample data S. e.g. S(x), S(y)
2. Select a parameter k for the data.
3. Give a new data sample x.
4. Determine the k-nearest neighbor of sample x by calculating the distance. It can be determined by Euclidean distance. The mathematical formula for finding Euclidean distance is

$$\mathbf{d\ (x,y)=}\ \sqrt{\sum_{K=1}^{MN}(x^k - y^k)^2} \qquad \mathbf{(3.12)}$$

5. Combine the classes of sample y in one class.
6. Find the output.

The value of k will be different according to the data. Figure 3.11 illustrate the working of KNN for intrusion detection [92]

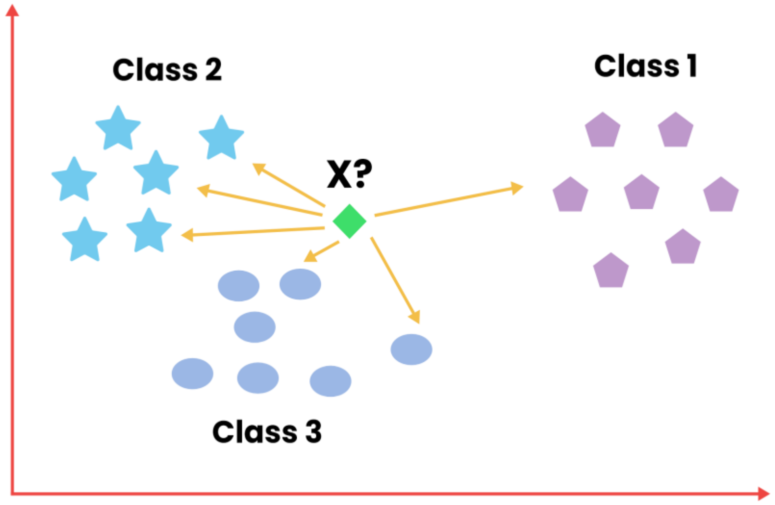


Figure 3.11: KNN for Intrusion detection

In machine learning KNN is widely used because it easily classifies the data. There are a number of studies that use KNN for the detection of different problems[93] [94][91].

3.6.3 Decision Tree

Decision tree is a supervised machine learning technique used for classification and regression problems. In decision tree root nodes can be used as input these nodes are filtered through decision nodes and leaf nodes used for getting desired output [95]. Entropy is used to control how data will be split in decision tree and information gain tells how much information a feature gives about the respective class. Equation 3.13 explains the formula for calculating Entropy and information gain in decision tree [96].

$$\textbf{Entropy} = -\sum_{i} \wp \log_{2} \wp \qquad \textbf{(3.13)}$$

In decision tree the data flow in nodes**.** Figure 3.12 explain the working of decision tree algorithm [97].

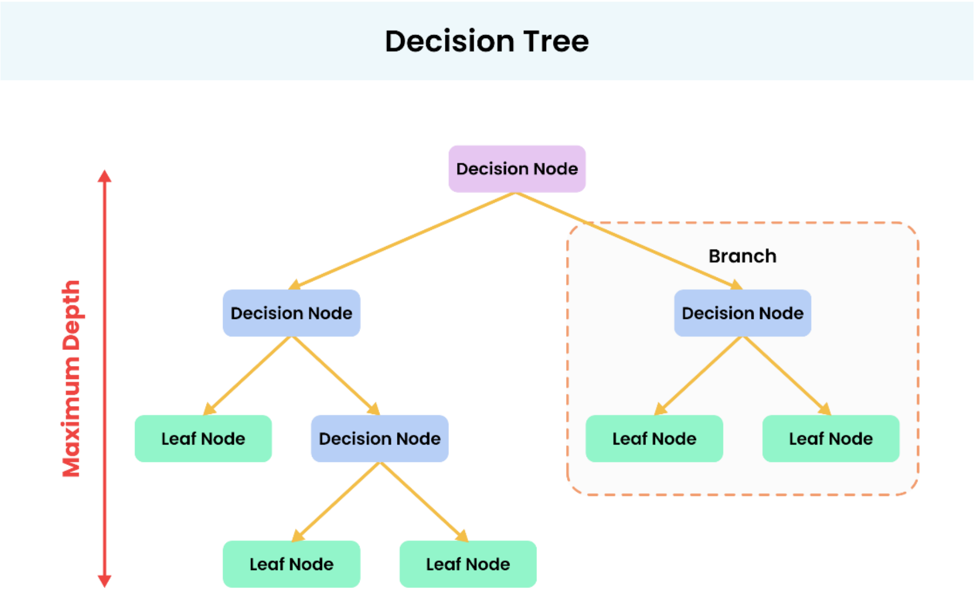


Figure 3.12: Architecture of Decision Tree

The overall model summary is explained in Figure 3.13

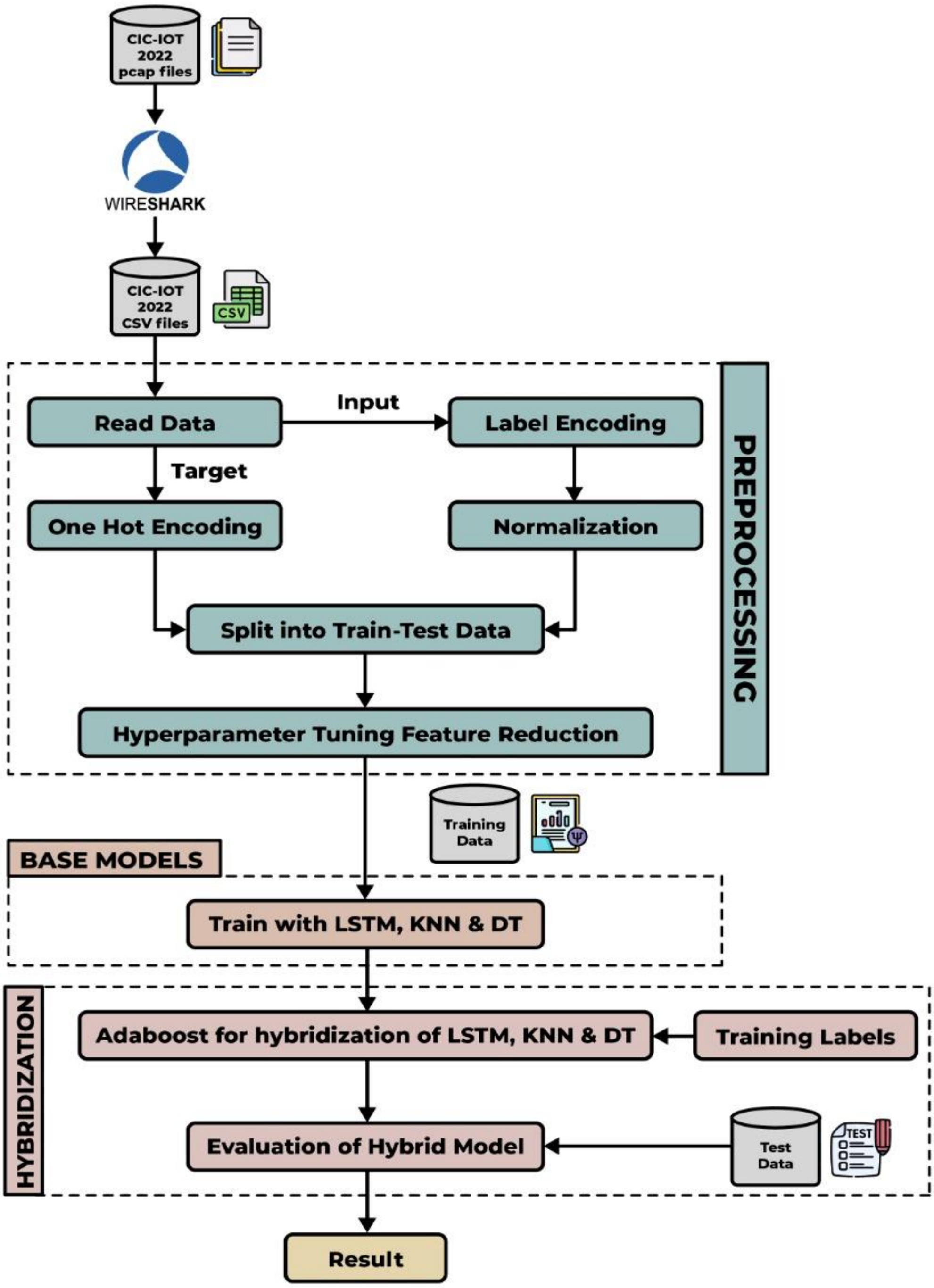


Figure 3.13: Experimental setup for Hybrid KNN, LSTM and DT

Each phase of experiment is shown in the Figure 3.13. Hyperparameter Tuning with epoch is also explained in Figure 3.14.

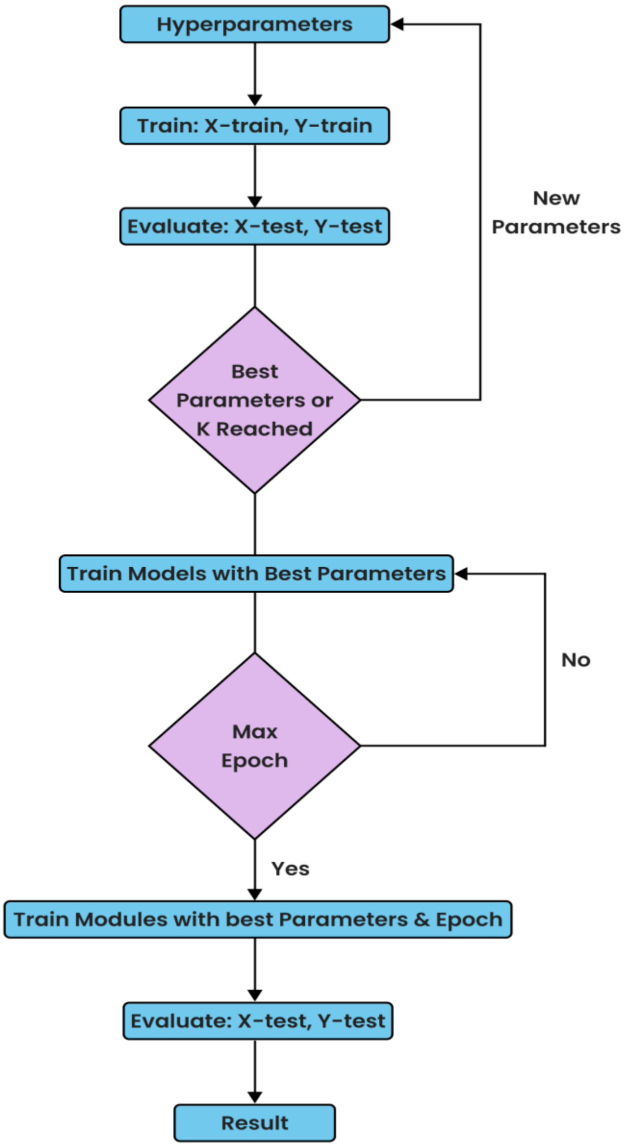


Figure 3.14: Hyperparameter Tuning

3.6.4 Adaboost

Adaboost also called Adaptive Boosting is a machine learning technique used as Ensemble Method. Just as people learn from their mistakes and try not to repeat them later in life, the Boosting algorithm tries to build a strong learner (predictive model) from the mistakes of several weaker models. Predictions are made by computing the weighted average of the weak classifiers.

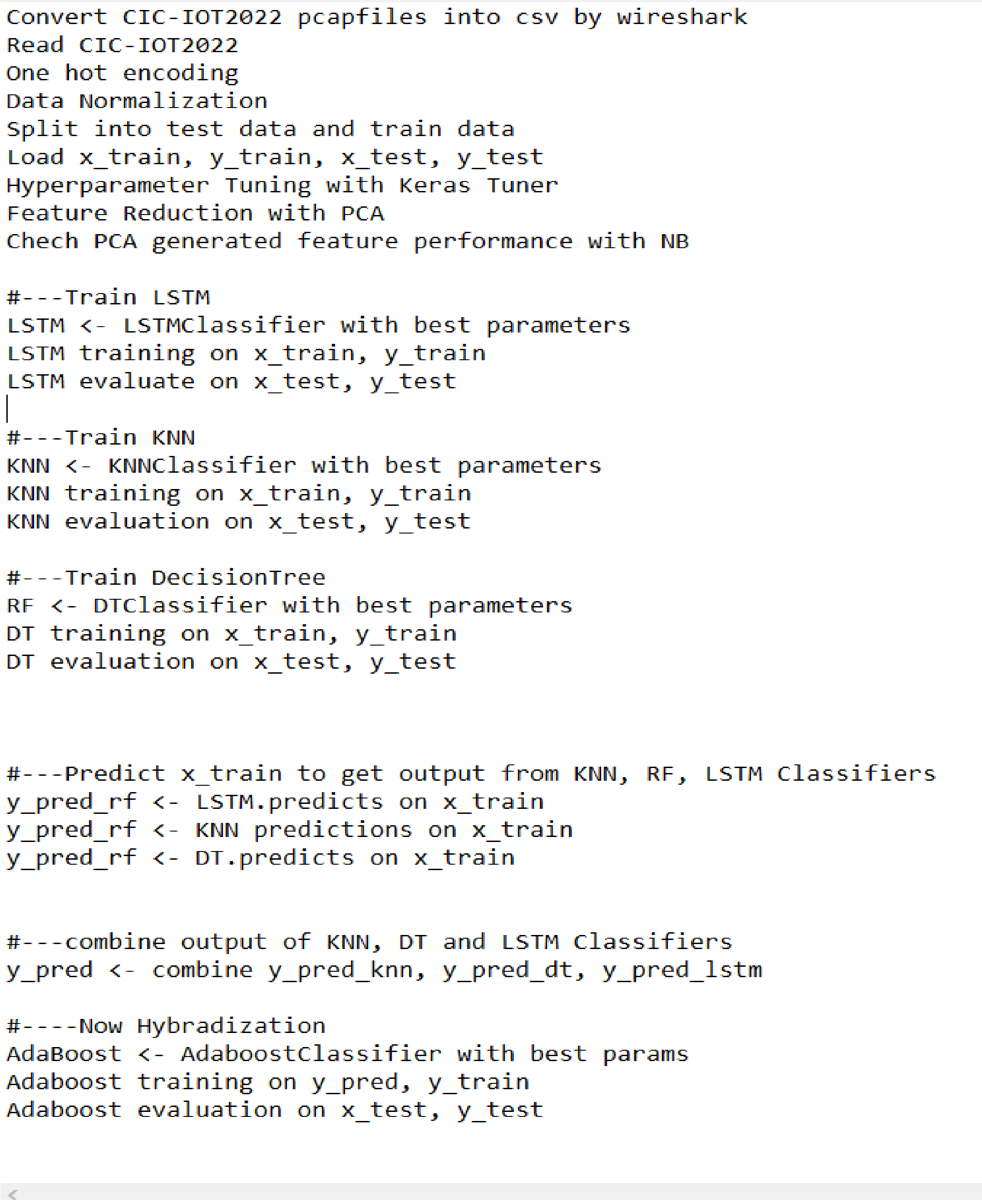
For a new input instance, each weak learner computes the predicted value as +1.0 or -1.0. Predicted values are weighted by the value of each weak learner. The prediction for the ensemble model is taken as the sum of the weighted predictions. In case of positive sum, the first class is predicted else the second class is predicted.

For example, 5 weak classifiers can predict the values 1.0, 1.0, -1.0, 1.0, -1.0. From the majority vote, it looks like the model will predict a value of 1.0 or first class. These 5 weak classifiers can have degree values of 0.2, 0.5, 0.8, 0.2, and 0.9, respectively. Calculating the weighted sum of these predictions results in an output of -0.8, which would be an aggregate prediction of -1.0 or the second class. The pseudocode for Adaboost is given as:

```
Initialize weights
for Each base learner do:
    Train base learner with a weighted sample.
    Test base learner on all data.
    Set learner weight with a weighted error.
    [$\alpha = \frac{1}{2} ln \frac{(1-total\ error)}{Total\ error}$]
    Update weights based on ensemble predictions.
end for
```

In the proposed study outputs of LSTM, KNN and DT are combined and given to Adaboost as single input to classify intrusion, benign and attacks classes. The pseudo code is given as:

3.7 Pseudo Code for Hybrid KNN, DT and LSTM with Adaboost



```
Convert CIC-IOT2022 pcapfiles into csv by wireshark
Read CIC-IOT2022
One hot encoding
Data Normalization
Split into test data and train data
Load x_train, y_train, x_test, y_test
Hyperparameter Tuning with Keras Tuner
Feature Reduction with PCA
Chech PCA generated feature performance with NB

#---Train LSTM
LSTM <- LSTMClassifier with best parameters
LSTM training on x_train, y_train
LSTM evaluate on x_test, y_test

#---Train KNN
KNN <- KNNClassifier with best parameters
KNN training on x_train, y_train
KNN evaluation on x_test, y_test

#---Train DecisionTree
RF <- DTClassifier with best parameters
DT training on x_train, y_train
DT evaluation on x_test, y_test



#---Predict x_train to get output from KNN, RF, LSTM Classifiers
y_pred_rf <- LSTM.predicts on x_train
y_pred_rf <- KNN predictions on x_train
y_pred_rf <- DT.predicts on x_train


#---combine output of KNN, DT and LSTM Classifiers
y_pred <- combine y_pred_knn, y_pred_dt, y_pred_lstm

#----Now Hybradization
AdaBoost <- AdaboostClassifier with best params
Adaboost training on y_pred, y_train
Adaboost evaluation on x_test, y_test
```

3.8 Objectives of Research

The main objective of research is fulfilled through hybridization of KNN, LSTM and DT for intrusion detection in IOT based smart Home which also improved performance which is further discussed in next chapter.

3.9 Assumptions and Limitations

Real time dataset was used for IDS-IOT in smart homes but could not use in real scenario due to time and resource limitation. However, results are assumed to be almost same in real scenario as well.

3.10 Summary

This chapter includes the explanation of proposed methodology for intrusion detection in IOT based Smart Homes. CIC-IOT2022 dataset was trained and tested through Hybrid KNN, DT and KNN with Adaboost. It also exhibits complete explanation of the experimental phases diagrammatically and with pseudocode as well.

Chapter 4

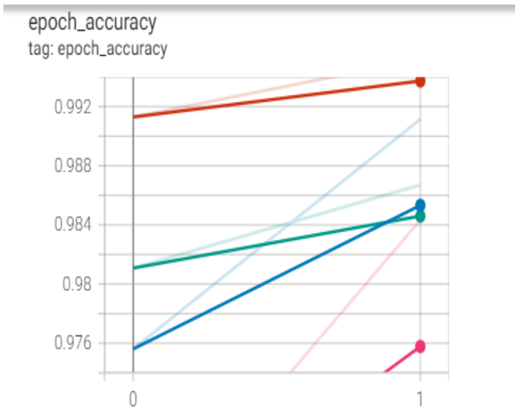
PERFORMANCE EVALUATION

4.1 Overview

This chapter comprises of evaluation of the proposed hybrid ML/DL scheme in terms with respect to different performance metrics like Accuracy, Precision, Recall and F1-score. Comparative Analysis of the proposed scheme with other ML and DL schemes is also presented on two datasets named as CICIDS2022 and UNSW-NB15. It shows that proposed scheme outer performed other benchmark ML/DL schemes. PCA was used to generate features concerning improved accuracy and hybrid KNN,DT and LSTM improved other performance metrics when used in hybrid combination because of overcoming single technique's faults.

4.2 Result and Analysis

The performance metrics were Accuracy, Precision, Recall and F1-score to evaluate the hybrid ML/DL based IDS-IOT. The implementation of hybrid model was done by utilizing the Keras Tuner to select best hyperparameters so that accuracy could be improved while decreasing the loss.



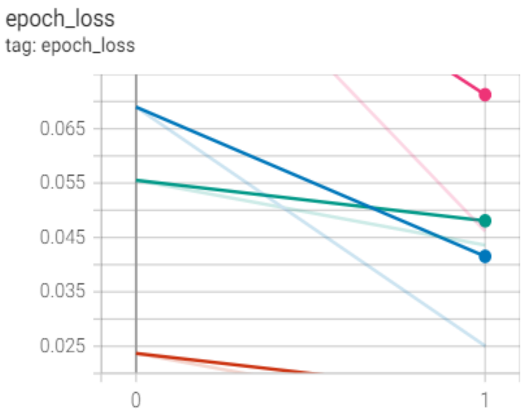


Figure 4.1: Epoch vs Accuracy **Figure 4.2: Epoch vs Loss**

Important features with scores are represented in graph as:

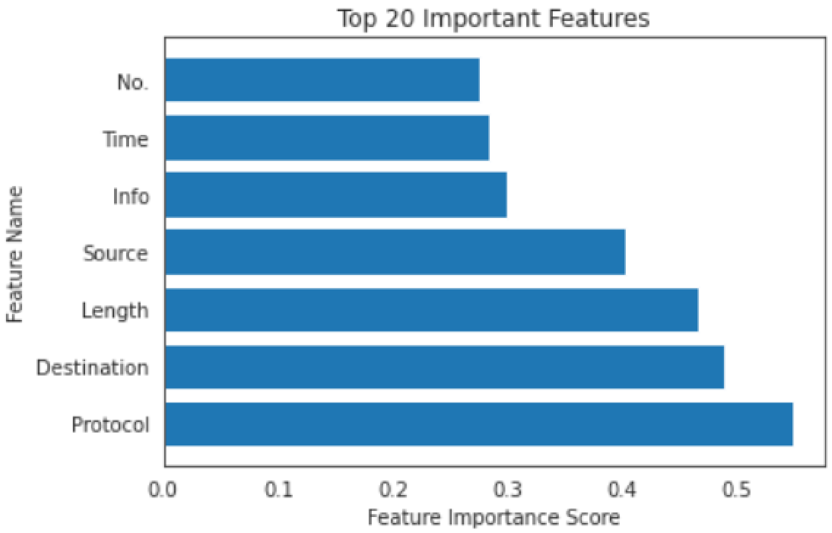


Figure 4.3: TOP 20 Features

The correlation matrix between ten important features is also represented as

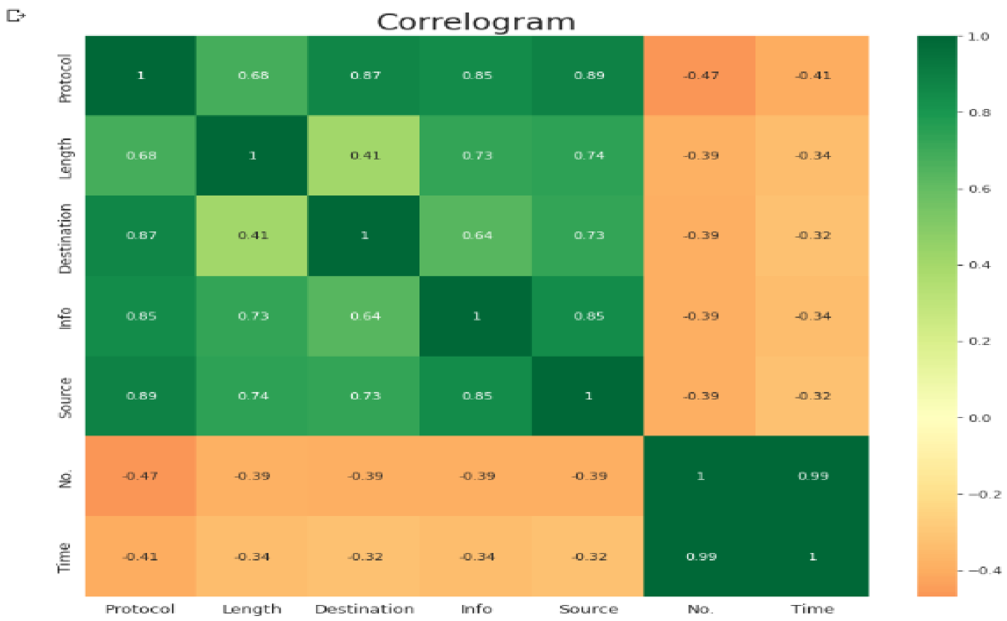


Figure 4.4: Correlogram

PCA was used to generate auto features in order to improve performance which is then tested using Naive Bayes algorithm. The performance of these features on accuracy is shown as

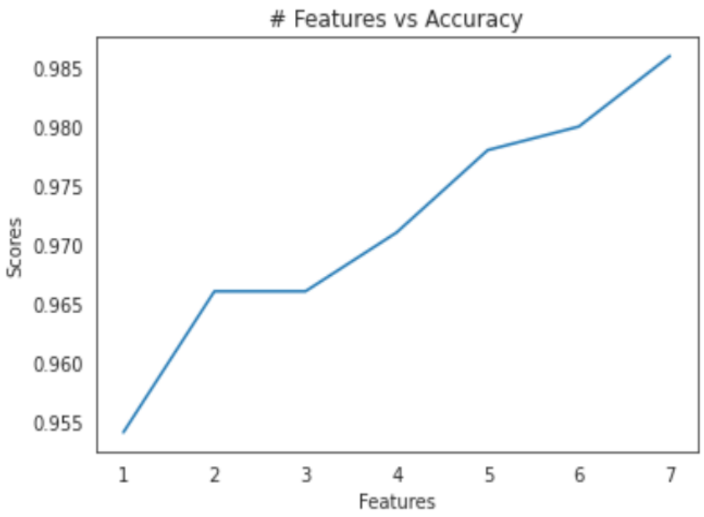


Figure 4.5: Features vs Accuracy

Confusion Matrix is used to show the performance of the classification ML/DL models.

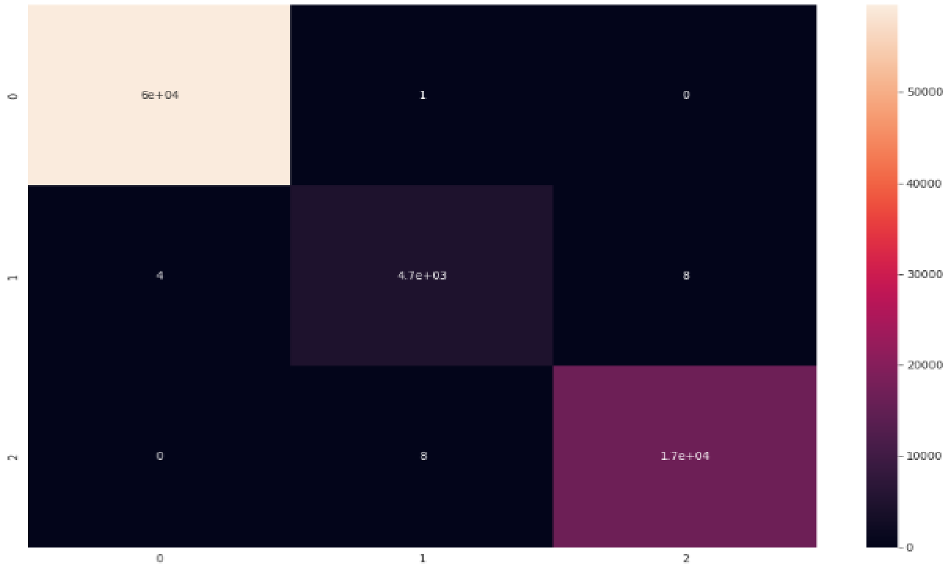


Figure 4.6: Confusion Matrix

4.3 Proposed Hybrid Scheme Vs Benchmark Schemes

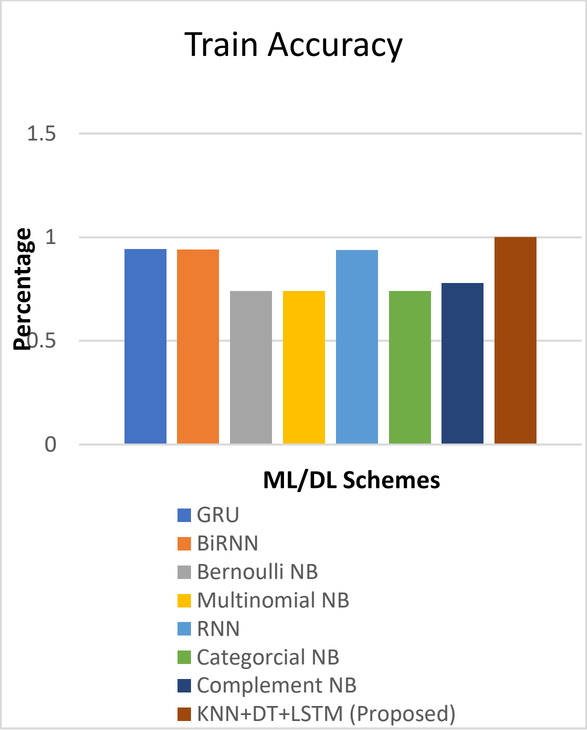
The proposed Hybrid KNN, DT and LSTM model was compared with other ML/DL schemes named as GRU, BiRNN, Bernoulli NB, Multinomial NB, RNN, Categorical NB and Complement NB.

4.3.1 Accuracy

Accuracy is the measure of correct classification of Machine Learning or Deep Learning Algorithm. It could be represented as

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{(4.1)}$$

For Multiclass classification, it means the correct classification of an instance for each class. In the proposed solution, accuracy means the correct classification of an instance for Benign and two attacks classes as well. The comparison of the proposed scheme and other schemes in terms of accuracy is represented in graphs.



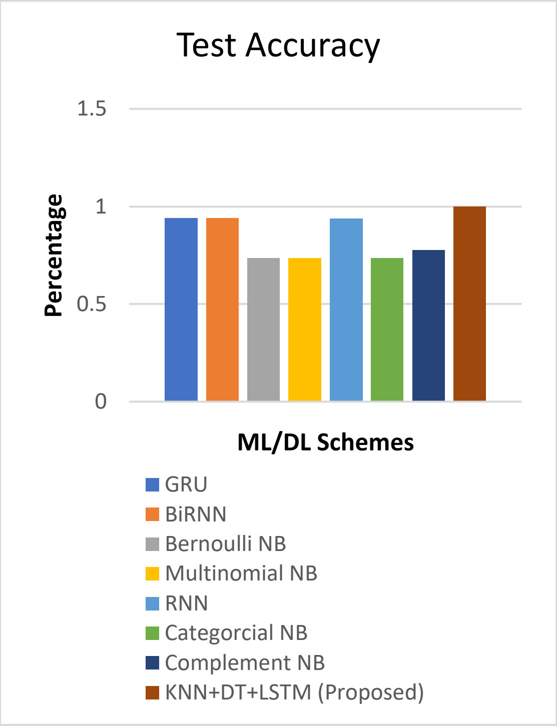


Figure 4.7: Train Accuracy on Different Models, Figure 4.8: Test Accuracy on Different Models

It clearly depicts that proposed KNN+DT+LSTM outer performs the other schemes. The gates involved in LSTM architecture makes it better for long term dependencies and results in improve accuracy when hybrid with DT and KNN for attack detection in the proposed smart home scenario.

4.3.2 Precision

Precision is the measure of reliability of Machine Learning or Deep Learning Model. It measures model's accuracy in classifying an instance as Positive.

Its Mathematical Equation is

$$Precision = \frac{TP}{TP+FP} \qquad \textbf{(4.2)}$$

For the proposed Hybrid Model, if it is able to classify Benign Traffic correctly and does not mistaken in classifying attacks as Benign then the Precision will become high.

KNN is the slow for real time detection of attacks, therefore DT and LSTM were hybridized to improve precision and performance of IDS-IOT.

Graphical representation of Precision while comparing with other schemes is given below. It shows that the proposed hybrid KNN, DT and LSTM performed well.

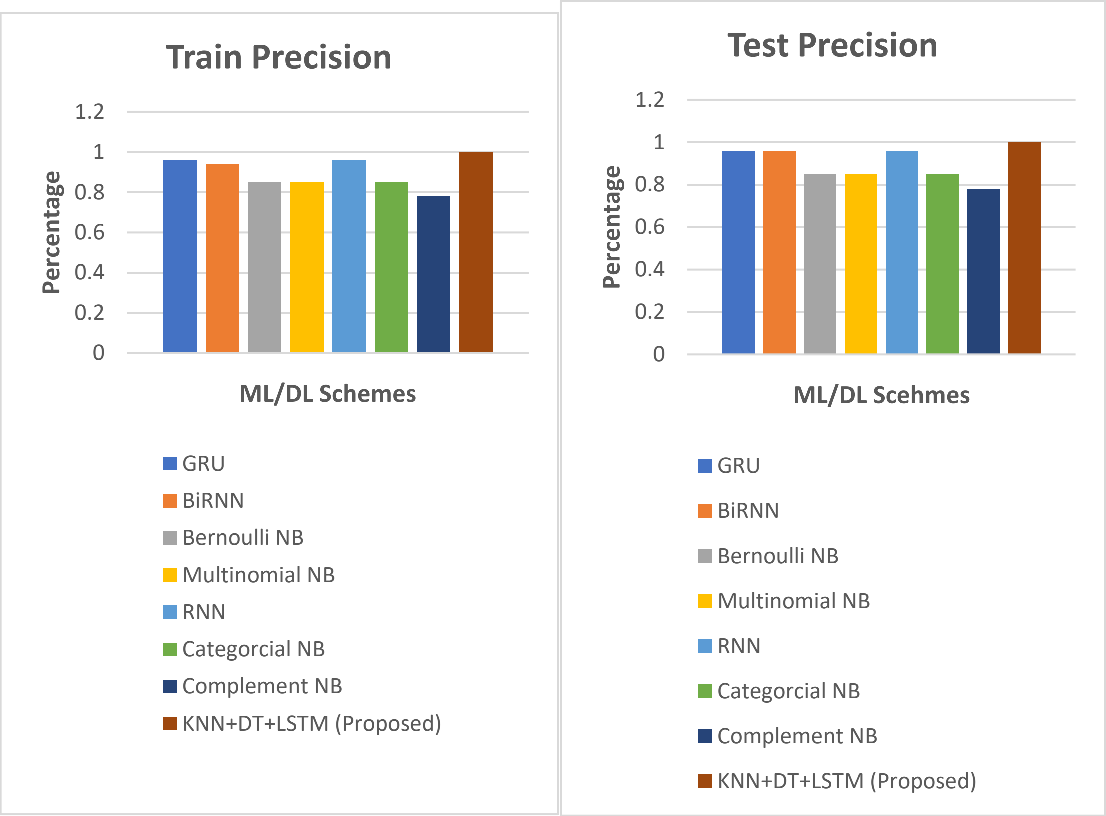


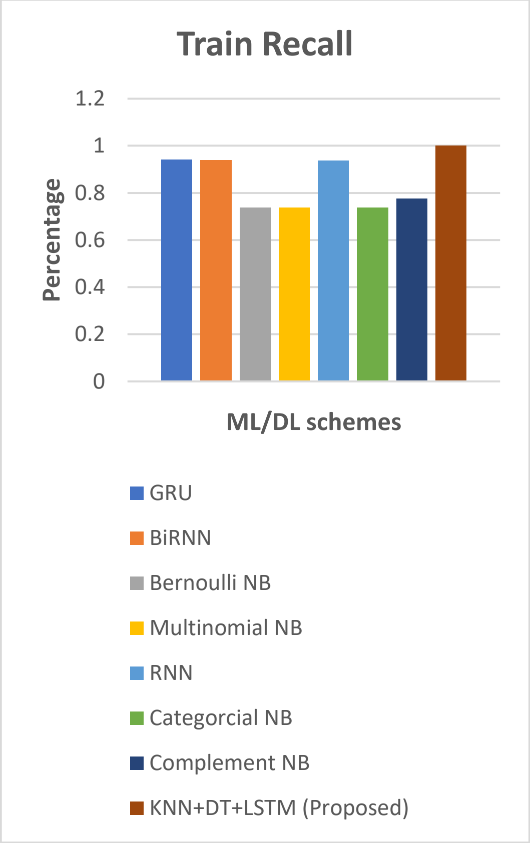
Figure 4.9: Train Precision on Different Models, Figure 4.10: Test Precision on Different Models

4.3.3 Recall

Recall is the capability of classifying positive samples to the total number of positive samples. In the current scenario, it is ability to detect benign samples. Its mathematical representation is given as.

$$Recall = \frac{TP}{TP+FN} \tag{4.3}$$

Graphical Comparison is as follows.



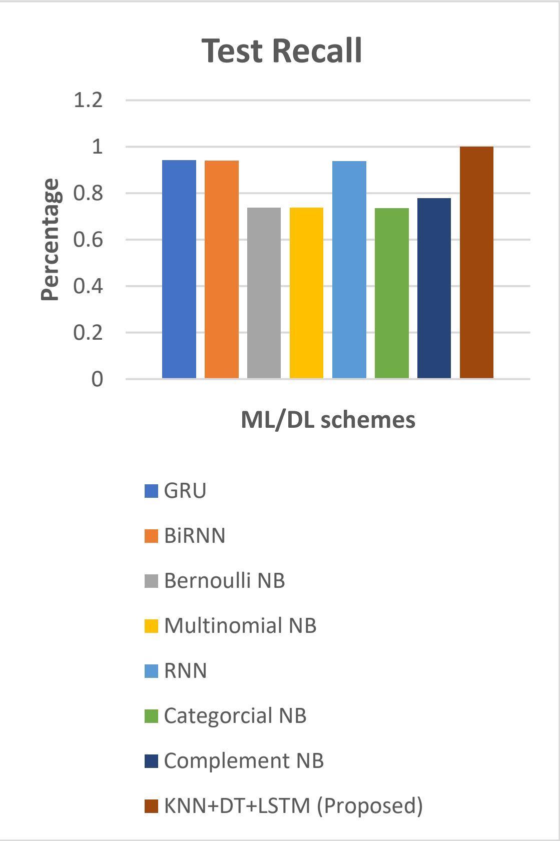


Figure 4.11: Train Recall on Different Models, Figure 4.12: Train Recall on Different Models

4.3.4 F1-Score

F1-score is the harmonic mean of precision and recall. It is basically used to combine classifiers with different precision and recall. Mathematical expression of F1-Score is

$$F1 - Score = \frac{2*P*R}{P+R} \tag{4.4}$$

The comparison of F1-score is given in a graph.

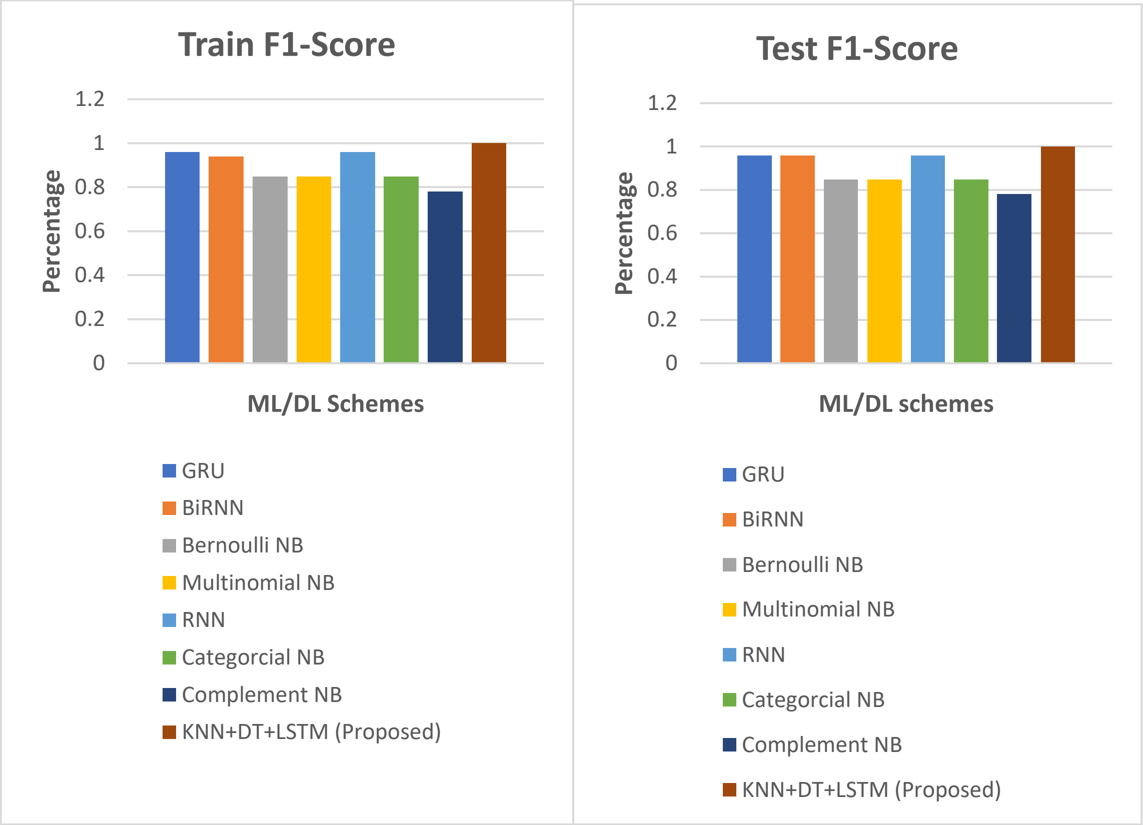


Figure 4.13:Train F1-Score on Different Models, Figure 4.14: Test F1-Score on Different Models

Table 4.1 : Performance Metrics of Different ML/DL Models (Train)

Techniques	**Accuracy**	**Precision**	**Recall**	**F1-Score**
	Training			
GRU	0.94165	0.98387	0.94165	0.9591
BiRNN	0.9401	0.98334	0.9401	0.94006
Bernoulli NB	0.73807	0.99985	0.73807	0.84919
Multinomial NB	0.73805	0.99995	0.73805	0.84924
RNN	0.93833	0.98586	0.93833	0.95891
Categorcial NB	0.73801	1	0.73801	0.84926
Complement NB	0.7768	0.84577	0.7768	0.7801
KNN+DT+LSTM (Proposed)	1	1	1	1

Table 4.2 : Performance Metrics of Different ML/DL Schemes(Test)

Techniques	Accuracy	Precision	Recall	F1-Score
	Testing			
GRU	0.94161	0.98292	0.94161	0.95853
BiRNN	0.94006	0.98246	0.94006	0.95766
Bernoulli NB	0.7359	0.99977	0.7359	0.84772
Multinomial NB	0.73588	0.99998	0.73588	0.84783
RNN	0.9379	0.98547	0.9379	0.9585
Categorical NB	0.73586	1	0.73586	0.84783
Complement NB	0.77767	0.84608	0.77767	0.78095
KNN+DT+LSTM (Proposed)	0.99974	0.99974	0.99974	0.99974

4.4 Performance of Hybrid Approach on UNSW-NB15

Hybrid ML/DL was also implemented on benchmark dataset UNSW-NB15 which shows good performance as well which depicts that proposed algorithm could also be used for NIDS and intrusion detection in other IOT environments with slight modifications if needed.

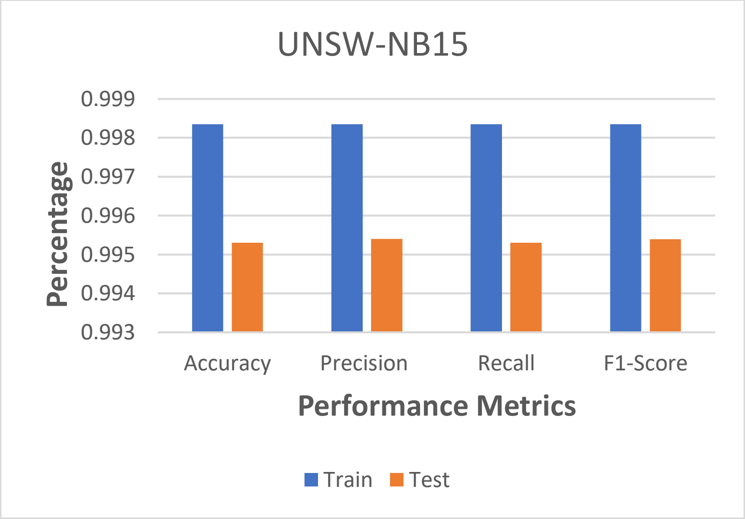


Figure 4.15: Performance Metrics on UNSW-NB15 Dataset

4.5 Summary

The comparative analysis of the proposed scheme and other Machine Learning/Deep Learning scheme is presented which shows that the proposed solution improves the performance in detecting of attacks and benign classes specialized in an IOT-Smart Home. It also overcomes underfitting/overfitting issues and is generalizable in nature which makes it more promising.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Overview

The proposed IDS-IOT in smart Homes is implemented in Google Colab environment using Python. Performance parameters like Accuracy, Precision, Recall and F1- score of proposed mechanism is compared with other benchmark ML/DL schemes. It shows significant improvement in the performance of proposed solution in comparison with the other schemes. Future scope of the proposed research is also discussed in the chapter.

5.2 Conclusion

IOT network is ubiquitous in nature and spreading all over in different fields which makes it more vulnerable to network attacks. Hence, intrusion detection is one of the most important factors to ensure smooth and secure working environment. Machine Learning is one of the most promising techniques for anomaly based intrusion detection in networks recently. Therefore, ML and DL based techniques for attack detection in network have been investigated in Literature which highlighted benefits, issues and gaps lying in them. To overcome some of issues, fill loop holes and increase performance hybrid ML/DL based scheme has been proposed which is also influenced from the literature. Hybridization of KNN, DT and LSTM was done and implemented on latest CIC-IDS2022 dataset with appropriate methods and steps for dimensionality reduction and classification, using Tensor Flow in Google Colab which results in increased accuracy, precision, Recall and F1-score when compared with other techniques.

The proposed scheme is also tested on an old dataset UNSW-NB15 to see its effectiveness in longer spectrum.

5.3 Future Work

The proposed scheme should be implemented in real environment to see its effectiveness and limitations. Moreover, it could also be implemented on newer versions of public and private datasets to involve more number of potential attacks and zero day attacks as well.

[References]

[1] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, “IoT elements, layered architectures and security issues: A comprehensive survey,” *Sensors (Switzerland)*, vol. 18, no. 9, pp. 1–37, 2018, doi: 10.3390/s18092796.

[2] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar, “CRAIoT: Concept, Review and Application(s) of IoT,” *Proc. - 2019 4th Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2019*, pp. 1–4, 2019, doi: 10.1109/IoT-SIU.2019.8777467.

[3] O. Georgiana Dorobantu and S. Halunga, “Security threats in IoT,” *2020 14th Int. Symp. Electron. Telecommun. ISETC 2020 - Conf. Proc.*, pp. 17–20, 2020, doi: 10.1109/ISETC50328.2020.9301127.

[4] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, “Intrusion detection systems for IoT-based smart environments: a survey,” *J. Cloud Comput.*, vol. 7, no. 1, pp. 1–20, 2018, doi: 10.1186/s13677-018-0123-6.

[5] M. A. Alsoufi *et al.*, “Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review,” *Appl. Sci.*, vol. 11, no. 18, 2021, doi: 10.3390/app11188383.

[6] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.

[7] D. Mudzingwa and R. Agrawal, “A study of methodologies used in intrusion detection and prevention systems (IDPS),” *Conf. Proc. - IEEE SOUTHEASTCON*, no. September, 2012, doi: 10.1109/SECon.2012.6197080.

[8] H. U. & H. B. P. TARIQAHMAD SHERASIYA, “a Survey: Intrusion Detection System for Internet of Things,” *Int. J. Comput. Sci. Eng.* , vol. 5, no. 2, pp. 91–98, 2016, [Online]. Available: http://www.iaset.us/view_archives.php?year=2016&id=14&jtype=2&page=2

[9] M. Darkaie and R. Tavoli, “Providing a method to reduce the false alarm rate in network intrusion detection systems using the multilayer perceptron technique and backpropagation algorithm,” *2019 IEEE 5th Conf. Knowl. Based Eng. Innov. KBEI 2019*, pp. 1–6, 2019, doi: 10.1109/KBEI.2019.8735024.

[10] S. Dwivedi, M. Vardhan, and S. Tripathi, “Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection,” *Cluster Comput.*, vol. 24, no. 3, pp. 1881–1900, 2021, doi: 10.1007/s10586-020-03229-5.

[11] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, “A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks,” *Electron.*, vol. 8, no. 11, 2019, doi: 10.3390/electronics8111210.

[12] J. H. Lee and K. H. Park, “GAN-based imbalanced data intrusion detection system,” *Pers. Ubiquitous Comput.*, vol. 25, no. 1, pp. 121–128, 2021, doi: 10.1007/s00779-019-01332-y.

[13] M. Panda, A. A. A. Mousa, and A. E. Hassanien, “Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks,” *IEEE Access*, vol. 9, pp. 91038–91052, 2021, doi: 10.1109/ACCESS.2021.3092054.

[14] W. Fang, X. Tan, and D. Wilbur, “Application of intrusion detection technology in network safety based on machine learning,” *Saf. Sci.*, vol. 124, no. December 2019, p. 104604, 2020, doi: 10.1016/j.ssci.2020.104604.

[15] Y. Wu, W. W. Lee, Z. Xu, and M. Ni, “Large-scale and robust intrusion detection model combining improved deep belief network with feature-weighted svm,” *IEEE Access*, vol. 8, pp. 98600–98611, 2020, doi: 10.1109/ACCESS.2020.2994947.

[16] A. Verma and V. Ranga, “Machine Learning Based Intrusion Detection Systems for IoT Applications,” *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020, doi: 10.1007/s11277-019-06986-8.

[17] D. Rani, N. S. Gill, P. Gulia, and J. M. Chatterjee, “An Ensemble-Based Multiclass Classifier for Intrusion Detection Using Internet of Things,” vol. 2022, 2022.

[18] A. M. Aleesa, M. Younis, A. A. Mohammed, and N. M. Sahar, “Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques,” *J. Eng. Sci. Technol.*, vol. 16, no. 1, pp. 711–727, 2021.

[19] R. M. Swarna Priya *et al.*, “An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture,” *Comput. Commun.*, vol. 160, no. February, pp. 139–149, 2020, doi: 10.1016/j.comcom.2020.05.048.

[20] S. Zhao, W. Li, T. Zia, and A. Y. Zomaya, “A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things,” *Proc. - 2017 IEEE 15th Int. Conf. Dependable, Auton. Secur. Comput. 2017 IEEE 15th Int. Conf. Pervasive Intell. Comput. 2017 IEEE 3rd Int. Conf. Big Data Intell. Compu*, vol. 2018-Janua, pp. 836–843, 2018, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.141.

[21] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K. K. R. Choo, “A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks,” *IEEE Trans. Emerg. Top. Comput.*, vol. 7, no. 2, pp. 314–323, 2019, doi: 10.1109/TETC.2016.2633228.

[22] M. A. Khan, “HCRNNIDS : Hybrid Convolutional Recurrent Neural,” 2021.

[23] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, “A hierarchical hybrid intrusion detection approach in IoT scenarios,” *2020 IEEE Glob. Commun. Conf. GLOBECOM 2020 - Proc.*, vol. 2020-Janua, 2020, doi: 10.1109/GLOBECOM42002.2020.9348167.

[24] H. Yao, D. Fu, P. Zhang, M. Li, and Y. Liu, “MSML: A novel multilevel semi-supervised machine learning framework for intrusion detection system,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1949–1959, 2019, doi: 10.1109/JIOT.2018.2873125.

[25] P. Nimbalkar and D. Kshirsagar, “Feature selection for intrusion detection system in Internet-of-Things (IoT),” *ICT Express*, vol. 7, no. 2, pp. 177–181, 2021, doi: 10.1016/j.icte.2021.04.012.

[26] M. Nivaashini and P. Thangaraj, “A framework of novel feature set extraction based intrusion detection system for internet of things using hybrid machine learning algorithms,” *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 44–49, 2019, doi: 10.1109/GUCON.2018.8674952.

[27] B. W. Masduki and K. Ramli, “Study on Implementation of Machine Learning

Methods Combination for Improving Attacks Detection Accuracy on Intrusion Detection System (IDS),” pp. 56–64, 2015.

[28] “Analysis of Intelligent Classifiers.pdf.”

[29] R. Wazirali, “An Improved Intrusion Detection System Based on KNN Hyperparameter Tuning and Cross-Validation,” *Arab. J. Sci. Eng.*, vol. 45, no. 12, pp. 10859–10873, 2020, doi: 10.1007/s13369-020-04907-7.

[30] M. Artur, “Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features,” *Procedia Comput. Sci.*, vol. 190, no. 2019, pp. 564–570, 2021, doi: 10.1016/j.procs.2021.06.066.

[31] Y. Al-Hadhrami and F. K. Hussain, “Real time dataset generation framework for intrusion detection systems in IoT,” *Futur. Gener. Comput. Syst.*, vol. 108, pp. 414–423, 2020, doi: 10.1016/j.future.2020.02.051.

[32] L. Dhanabal and S. P. Shantharajah, “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015, doi: 10.17148/IJARCCE.2015.4696.

[33] S. Zavrak and M. Iskefiyeli, “Anomaly-Based Intrusion Detection from Network Flow Features Using Variational Autoencoder,” *IEEE Access*, vol. 8, pp. 108346–108358, 2020, doi: 10.1109/ACCESS.2020.3001350.

[34] M. C. Computing, *IoT Security*. 2020. doi: 10.1002/9781119527978.

[35] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, “A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/COMST.2020.2988293.

[36] G. Moukarzel, M. A. Lemay, and A. J. Spence, “and Analyses,” *Biomed Signal Process Control*, pp. 1–20, 2021.

[37] K. S. Bhosale, M. Nenova, and G. Iliev, “Intrusion detection in communication networks using different classifiers,” *Techno-Societal 2018 - Proc. 2nd Int. Conf. Adv. Technol. Soc. Appl.*, vol. 2, pp. 19–28, 2020, doi: 10.1007/978-3-030-16962-6_3.

[38] S. M. Taghavinejad, M. Taghavinejad, L. Shahmiri, M. Zavvar, and M. H. Zavvar, “Intrusion Detection in IoT-Based Smart Grid Using Hybrid Decision Tree,” *2020 6th Int. Conf. Web Res. ICWR 2020*, pp. 152–156, 2020, doi: 10.1109/ICWR49608.2020.9122320.

[39] K. V. V. N. L. Sai Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, “Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2372–2379, 2020, doi: 10.1016/j.procs.2020.04.257.

[40] B. S. Bhati and C. S. Rai, “Analysis of Support Vector Machine-based Intrusion Detection Techniques,” *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 2371–2383, 2020, doi: 10.1007/s13369-019-03970-z.

[41] D. Jing and H. B. Chen, “SVM based network intrusion detection for the UNSW-NB15 dataset,” *Proc. Int. Conf. ASIC*, pp. 1–4, 2019, doi:

10.1109/ASICON47005.2019.8983598.

[42] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019, doi: 10.1109/ACCESS.2019.2907965.

[43] S. S. Swarna Sugi and S. R. Ratna, "Investigation of machine learning techniques in intrusion detection system for IoT network," *Proc. 3rd Int. Conf. Intell. Sustain. Syst. ICISS 2020*, pp. 1164–1167, 2020, doi: 10.1109/ICISS49785.2020.9315900.

[44] M. Panda and M. R. Patra, "Network Intrusion Detection Using Naïve Bayes," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 12, pp. 258–263, 2007.

[45] B. S. Sharmila and R. Nagapadma, "Intrusion detection system using naive bayes algorithm," *2019 5th IEEE Int. WIE Conf. Electr. Comput. Eng. WIECON-ECE 2019 - Proc.*, pp. 8–11, 2019, doi: 10.1109/WIECON-ECE48653.2019.9019921.

[46] P. K. Keserwani, M. C. Govil, E. S. Pilli, and P. Govil, "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model," *J. Reliab. Intell. Environ.*, vol. 7, no. 1, pp. 3–21, 2021, doi: 10.1007/s40860-020-00126-x.

[47] F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, "Fault-tolerant AI-driven Intrusion Detection System for the Internet of Things," *Int. J. Crit. Infrastruct. Prot.*, vol. 34, p. 100436, 2021, doi: 10.1016/j.ijcip.2021.100436.

[48] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[49] A. Verma and V. Ranga, "ELNIDS: Ensemble Learning based Network Intrusion Detection System for RPL based Internet of Things," *Proc. - 2019 4th Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2019*, pp. 2–7, 2019, doi: 10.1109/IoT-SIU.2019.8777504.

[50] L. T. Hong Van, P. Van Huong, L. D. Thuan, and N. Hieu Minh, "Improving the feature set in IoT intrusion detection problem based on FP-growth algorithm," *Int. Conf. Adv. Technol. Commun.*, vol. 2020-Octob, pp. 18–23, 2020, doi: 10.1109/ATC50776.2020.9255431.

[51] S. Alhaidari and M. Zohdy, "Hybrid learning approach of combining cluster-based partitioning and hidden Markov model for IoT intrusion detection," *ACM Int. Conf. Proceeding Ser.*, pp. 27–31, 2019, doi: 10.1145/3325917.3325939.

[52] Sharipuddin *et al.*, "Features extraction on iot intrusion detection system using principal components analysis (Pca)," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2020-Octob, pp. 114–118, 2020, doi: 10.23919/EECSI50503.2020.9251292.

[53] W. Li, W. Meng, and M. H. Au, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments," *J. Netw. Comput. Appl.*, vol. 161, no. March, 2020, doi: 10.1016/j.jnca.2020.102631.

[54] G. Kalnoor and S. Gowrishankar, "Markov decision process based model for performance analysis an intrusion detection system in IoT networks," *J. Telecommun. Inf. Technol.*, vol. 2021, no. 3, pp. 42–49, 2021, doi: 10.26636/JTIT.2021.151221.

[55] F. De Almeida Florencio, E. D. Moreno, H. T. MacEdo, R. J. P. De Britto Salgueiro, F. B. Do Nascimento, and F. A. O. Santos, "Intrusion detection via MLP neural network using an arduino embedded system," *Brazilian Symp. Comput. Syst. Eng. SBESC*, vol. 2018-Novem, pp. 190–195, 2018, doi: 10.1109/SBESC.2018.00036.

[56] A. Rosay, K. Riou, F. Carlier, and P. Leroux, "Multi-layer perceptron for network intrusion detection: From a study on two recent data sets to deployment on automotive processor," *Ann. des Telecommun. Telecommun.*, 2021, doi: 10.1007/s12243-021-00852-0.

[57] P. Van Huong, L. D. Thuan, L. T. Hong Van, and D. V. Hung, "Intrusion detection in IoT systems based on deep learning using convolutional neural network," *Proc. - 2019 6th NAFOSTED Conf. Inf. Comput. Sci. NICS 2019*, pp. 448–453, 2019, doi: 10.1109/NICS48868.2019.9023871.

[58] Y. Li *et al.*, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Meas. J. Int. Meas. Confed.*, vol. 154, p. 107450, 2020, doi: 10.1016/j.measurement.2019.107450.

[59] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Futur. Gener. Comput. Syst.*, vol. 113, pp. 418–427, 2020, doi: 10.1016/j.future.2020.07.042.

[60] X. Kan *et al.*, "A novel IoT network intrusion detection approach based on Adaptive Particle Swarm Optimization Convolutional Neural Network," *Inf. Sci. (Ny).*, vol. 568, pp. 147–162, 2021, doi: 10.1016/j.ins.2021.03.060.

[61] S. H. Park, H. J. Park, and Y. J. Choi, "RNN-based Prediction for Network Intrusion Detection," *2020 Int. Conf. Artif. Intell. Inf. Commun. ICAIIC 2020*, pp. 572–574, 2020, doi: 10.1109/ICAIIC48513.2020.9065249.

[62] M. Zeeshan *et al.*, "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022, doi: 10.1109/ACCESS.2021.3137201.

[63] A. Elsaeidy, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "Intrusion detection in smart cities using Restricted Boltzmann Machines," *J. Netw. Comput. Appl.*, vol. 135, no. March, pp. 76–83, 2019, doi: 10.1016/j.jnca.2019.02.026.

[64] Y. Zhang, P. Li, and X. Wang, "Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network," *IEEE Access*, vol. 7, no. c, pp. 31711–31722, 2019, doi: 10.1109/ACCESS.2019.2903723.

[65] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *J. Inf. Secur. Appl.*, vol. 58, no. March, p. 102804, 2021, doi: 10.1016/j.jisa.2021.102804.

[66] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, "A deep learning ensemble for network anomaly and cyber-attack detection," *Sensors (Switzerland)*, vol. 20, no. 16, pp. 1–20, 2020, doi: 10.3390/s20164583.

[67] A. Basati and M. M. Faghih, "APAE: an IoT intrusion detection system using asymmetric parallel auto-encoder," *Neural Comput. Appl.*, vol. 9, 2021, doi: 10.1007/s00521-021-06011-9.

[68] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection

using hybrid Deep Learning Model," *Comput. Commun.*, vol. 176, no. December 2020, pp. 146–154, 2021, doi: 10.1016/j.comcom.2021.05.024.

[69] I. Ullah, A. Ullah, and M. Sajjad, "Towards a Hybrid Deep Learning Model for Anomalous Activities Detection in Internet of Things Networks," *IoT*, vol. 2, no. 3, pp. 428–448, 2021, doi: 10.3390/iot2030022.

[70] Z. E. Huma *et al.*, "A Hybrid Deep Random Neural Network for Cyberattack Detection in the Industrial Internet of Things," *IEEE Access*, vol. 9, pp. 55595–55605, 2021, doi: 10.1109/ACCESS.2021.3071766.

[71] M. A. Khan, M. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry (Basel).*, vol. 11, no. 4, 2019, doi: 10.3390/sym11040583.

[72] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022, doi: 10.1109/ACCESS.2021.3140015.

[73] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, 2019, doi: 10.1109/JIOT.2019.2926365.

[74] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Anal.*, vol. 1, no. 1, pp. 1–22, 2016, doi: 10.1186/s41044-016-0014-0.

[75] L. Li, "Data quality and data cleaning in database applications," no. September, p. 242, 2012.

[76] P. Kaur and A. Gosain, "Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise," *Adv. Intell. Syst. Comput.*, vol. 653, no. January, pp. 23–30, 2018, doi: 10.1007/978-981-10-6602-3_3.

[77] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. February 2017, pp. 321–357, 2002, doi: 10.1613/jair.953.

[78] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.

[79] "Feature extraction: A survey | IEEE Journals & Magazine | IEEE Xplore."

[80] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing," *Degree Proj. Technol.*, p. 41, 2018.

[81] "categorical encoding - Compact notation for one-hot indicator vectors? - Cross Validated."

[82] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowlege-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998, doi: 10.1142/S0218488598000094.

[83] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," 2015, doi: 10.1038/nature14539.

[84] D. Rengasamy, M. Jafari, B. Rothwell, X. Chen, and G. P. Figueredo, “Deep learning with dynamically weighted loss function for sensor-based prognostics and health management,” *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030723.

[85] M. Sundermeyer, R. Schlüter, and H. Ney, “LSTM neural networks for language processing,” *Interspeech 2012*, pp. 194–197, 2012.

[86] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/J.PATCOG.2017.10.013.

[87] G. Lin and W. Shen, “Research on convolutional neural network based on improved Relu piecewise activation function,” *Procedia Comput. Sci.*, vol. 131, pp. 977–984, 2018, doi: 10.1016/j.procs.2018.04.239.

[88] S. A. Elwakil, S. K. El-Labany, M. A. Zahran, and R. Sabry, “Modified extended tanh-function method and its applications to nonlinear equations,” *Appl. Math. Comput.*, vol. 161, no. 2, pp. 403–412, 2005, doi: 10.1016/j.amc.2003.12.035.

[89] J. Han and C. Moraga, “The influence of the sigmoid function parameters on the speed of backpropagation learning,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 930, pp. 195–201, 1995, doi: 10.1007/3-540-59497-3_175.

[90] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU).” 2018.

[91] A. Al Bataineh, “A comparative analysis of nonlinear machine learning algorithms for breast cancer detection,” *Int. J. Mach. Learn. Comput.*, vol. 9, no. 3, pp. 248–254, 2019, doi: 10.18178/ijmlc.2019.9.3.794.

[92] M. A. H. A. Bakr, H. M. Al-Attar, N. K. Mahra, and S. S. Abu-Naser, “Breast Cancer Prediction Using JNN,” *Int. J. Acad. Inf. Syst. Res.*, vol. 4, no. 10, pp. 1–8, 2020.

[93] A. P. Pawlovsky and M. Nagahashi, “A method to select a good setting for the kNN algorithm when using it for breast cancer prognosis,” *2014 IEEE-EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2014*, pp. 189–192, 2014, doi: 10.1109/BHI.2014.6864336.

[94] Ö. Günaydin, M. Günay, and Ö. Şengel, “Comparison of lung cancer detection algorithms,” *2019 Sci. Meet. Electr. Biomed. Eng. Comput. Sci. EBBT 2019*, 2019, doi: 10.1109/EBBT.2019.8741826.

[95] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, “Overview of use of decision tree algorithms in machine learning,” *Proc. - 2011 IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2011*, pp. 37–42, 2011, doi: 10.1109/ICSGRC.2011.5991826.

[96] Linda Shapiro (University of Washington), “Information Gain Which test is more informative?,” 2015.

[97] “Decision Tree Algorithm, Explained - KDnuggets.”