National College of Ireland

Robust Intrusion Detection Model for Internet of Things

MSc Research Project
CYBERSECURITY

AYOBAMI JAIYEOLA
Student ID: X19220511

School of Computing
National College of Ireland

Supervisor: MICHAEL PANTRIDGE

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: JAIYEOLA AYOBAMI ADELEKE……

Student ID: X19220511……..……

Programme: CYBERSECURITY…… **Year:** 2021……..

Module: ACADEMIC INTERNSHIP……..……

Supervisor: MICHAEL PANTRIDGE……..……

Submission Due Date:

……16/08/2021……..……

Project Title: **Robust Intrusion Detection Model for Internet of Things**

Word Count: …9708…… **Page Count**……22……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: JAIYEOLA AYOBAMI……

Date: ……15/8/21……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Robust Intrusion Detection Model for Internet of Things

Ayobami Jaiyeola
x19220511

Abstract

As the development and usage of Internet of Things (IoT) continues to rise globally, the need for security of data and devices has risen simultaneously. Currently there are intrusion tolerance systems which have been provided tackle these issues; sadly, it remains a difficult task for these systems to develop a trusted interconnection between many IoT nodes due to the extremely massive number and vigorous make-up of devices. To pull off a better performance in intrusion detection, collaborative IDSs are often deployed in practical scenarios, where intrusion tolerance systems nodes are allowed to distribute essential information between them. In this research, a hybrid intrusion detection system has been proposed. It comprises of two stages of detection. Firstly, anomaly detection is carried out by using Spark Machine Learning; the next step involves deploying Convolutional LSTM network for misuse detection. After carrying out an experiment where the system will be fed with an up-to-date and suitable dataset; analysis will be carried out by using four performance metrics to create a confusion matrix that will be used evaluate the system's performance. It is expected at the end of this study that the proposed hybrid architecture will show higher accuracy when compared to similar approaches to intrusion detection.

Keywords – Intrusion Detection System (IDS), Internet of Things (IoT)

1. INTRODUCTION

With the continuous surge in development and usage of Internet of Things (IoT) globally, the need for security of data and devices has also been on the rise. It remains a difficult task for IDS to develop a trusted interconnection between numerous IoT devices due to the large number and dynamic nature of devices [1]. As a means of improving security, IDSs have been found to be very significant security tool designed to improve defence in computer networks against malicious attacks. To achieve better intrusion detection performance, collaborative IDSs are often deployed in practical scenarios where information can be shared between the different IDS nodes [2]. In his research, [1] also showed by using a case study, that the importance of sampling and filtering network traffic in determining privacy between IoT devices cannot be over emphasized.

Recently, minimizing the consumption of power by electrical appliances has helped to usher in recent era of smart devices. Electronic devices regularly upgrade physical objects over the Internet to help develop local intelligence and initiate connection with other devices. The devices in IoT are usually connected via the internet directly, which is not safe; this has led the attackers to easily access these devices. This public usage of the unsafe internet has caused devices to become more prone to the intrusions. Due to unhealthy access to public internet, the significance of IDSs in the IoT is undeniable [3]. As seen again in [4] and [5] intrusion detection is becoming more and more important, the total number and varieties of security challenges in IoT has seen an extraordinary surge recently.

The Internet plays a vital role to private individuals and organisations at large which means that every form of smart device, PCs, systems connected to it are very prone to various attacks and these are becoming more prevalent every day. To confront the present circumstance, conventional security strategies are inadequate and adaptation to non-critical failure methods are getting progressively cost-effective [6]. IoT usually combines numerous devices into networks to provide advanced and intelligent services; because of this, it is very important to protect privacy of users and, tackle threats like spoofing, DoS attacks, eavesdropping and jamming [7]. Many IoT systems operate on tiny, connected devices that compulsorily deals with serious energy and processor restrictions. These restricted processing resources do not permit standard security mechanisms to be utilized on the nodes often which makes IoT devices and applications easily prone to various threats.

Security challenges have been increasing just as the application of IoT is rapidly increasing. Intrusion attacks keep increasing despite the availability of many forms of security protocols created to combat these challenges. Intrusion has become a very common form of attacks carried out against computer networks today. Intrusion detection techniques are security measures that are deployed in IoT to boost its security strength. To ensure the security of a system, detecting intrusions is the first step; this has been made possible by using various intrusion detection techniques. The best way of defending against attacks is by preventing them from happening in the first place.

Intrusion tolerance involves using techniques to handle intrusions - a group of faults which includes intentional and malicious faults. It also includes counteraction and recovery from issues caused by intrusion. Without taking appropriate actions, intrusions are capable of crippling the complete security of a system. In intrusion tolerance, triggers are put in place to prevent intrusion from causing a system failure. An IDS is capable of analysing network traffic, servers' state and to report suspected intrusions [6].

As already discussed, the regular security practices and systems are not good enough to handle intrusions because of the resource constrains on IoT devices. To safeguard IoT networks, other more reliable methods must be employed; hence our decision to analyse the use of IDSs for intrusion tolerance in IoT. IDSs detect policy violations and malicious actions by monitoring network traffic and system actions. Also, they are usually software or hardware deployed in existing systems, this makes them appropriate for IoT nodes (which are resource constrained) [4]. A research to analyse how IDSs are used in IoT and determine their efficacy in intrusion detection is a very significant study, as it will expose different pros and possible cons of using this noble technique in the field of IoT; the findings in this study are expected to provide more insight to researchers and industry professionals regarding handling intrusion tolerance in IoT.
Firstly, a review on security issues currently being encountered in IoT will be carried out. A review will also be carried out on IDSs to analyse them and how they can be used to solve security issues in IoT. After the reviews have been carried out, IDSs will be implemented in IoT scenarios. The performance of these techniques will be evaluated, this will help us understand their significance in tackling security challenges like intrusion in IoT. From this analysis we can derive the benefits and disadvantages associated with using IDSs and suggest improvements where necessary by using performance metrics such as performance overheads and detection accuracy.

1.2. RESEARCH QUESTION

IoT and its applications is rapidly becoming predominant. In recent times IoT consists of numerous devices and protocols. There are different kinds of devices that come with different forms of data usage and vulnerabilities, this has led to the increase in concerns related to data security and privacy issues. As a result of these concerns, researchers and organizations are increasingly concerned about security issues in IoT. Many security practices being used currently are not good enough, this is because they are resource intensive in nature. Due to this obvious shortcoming, it has become very mandatory for second line defences to be incorporated into IoT networks. It is also important that these systems should be properly evaluated for their efficiency in different types of networks and protocols [8]. In this study, we are going to be using Intrusion Detection Systems (IDSs) for intrusion detection and tolerance in IoT. IDSs are usually considered to be the second defence line when other security mechanisms (such as access control and authorization) are unable to detect attacks. The IDS is used to detect ongoing attacks which are usually characterized by unusual activities. To accurately understand the effectiveness of using intrusion detection systems to tackle security challenges in IoT which leads to my research question "**How can Intrusion Detection Systems act as an enabler in solving rapidly growing privacy issues in IoT?**"

In order to tackle the research question of this research, the following objectives will be followed:

i. How effective can intrusion detection systems be in tackling intrusion issues in IoT?
ii. Considering that resource constrains are general issues in IoT and IDS, how can this issue be addressed?
iii. How can these intrusion tolerance systems be made more robust?

Firstly, a review on security issues currently being encountered in IoT will be carried out. A review will also be carried out on IDSs to analyse them and how they can be used to solve security issues in IoT. After the reviews have been carried out, IDS will be implemented in IoT scenarios. The performance of these techniques will be evaluated, this will help us understand their significance in tackling security challenges like intrusion in IoT. The aim of this project is to propose an IDS model using performance metrics such as performance overheads to improve detection accuracy in solving these challenges in IoT today.

2. LITERATURE REVIEW

The use of IoT devices over the years has significantly increased, as of today, IoT devices are being used in homes, offices, cars, schools, hospitals, IoT devices have become a very important part of our lives and because of that is very important to take seriously the security issues that can be encountered in IoT devices. Because of the need for IoT devices to be lightweight and compact, the authors have to limit their processing powers and energy supply [9]. This makes the already available security tools too heavy weighted to be applied on IoT devices. This means the IoT devices are sometimes released without appropriate security and are vulnerable to different types of attacks.

A. Types of Attacks

There are various ways IoT devices can be targeted, they can be attacked, or they can also be used to carry out a coordinated attack against other systems. The types of attacks are:

i. Denial of Service (DoS) - The attacker maximizes the nodes of a network with tasks in a way that the IoT devices, thereby rendering the device unusable. These types of attack are directed towards the resources of IoT devices, they either disable or slow down the resources of the device [10].

ii. Distributed Denial of Service (DDos) - Linux.Hydra, Tsunami, PsybOt, Chuck Norris, Aidra, and Spike are all examples of DDoS attacks. Mirai malware [11], can be used to exploit many devices to build a larger malicious network which can be referred to as botnet [12].

iii. Spoofing in [13] is described as a process in which an unauthorised user floods IP packets by copying the source address of approved users. In a spoofing scenario, the destination does not have a way of knowing who the authorized user is and they will not receive the expected data when response packets are sent back to the source IP address because the data would have been compromised by the attacker.

B. Intrusion Detection System

Considering the threats to IoT it is important to come up with safety measures for IoT devices, it is imperative to protect IoT device data sources from unauthorized access from malicious users. Because of this need, IDSs are very necessary as they detect the intruders in an IoT network and prevent unauthorized users from gaining access. Incorporating IDS is a difficult process due to the low energy requirements of IoT devices. To overcome this difficulty, a centralized IDS can be used to monitor the network and identify intruders. An alert is then triggered and sent to the network administrator when an anomaly is been detected. Figure 1 illustrates how an IDS works.

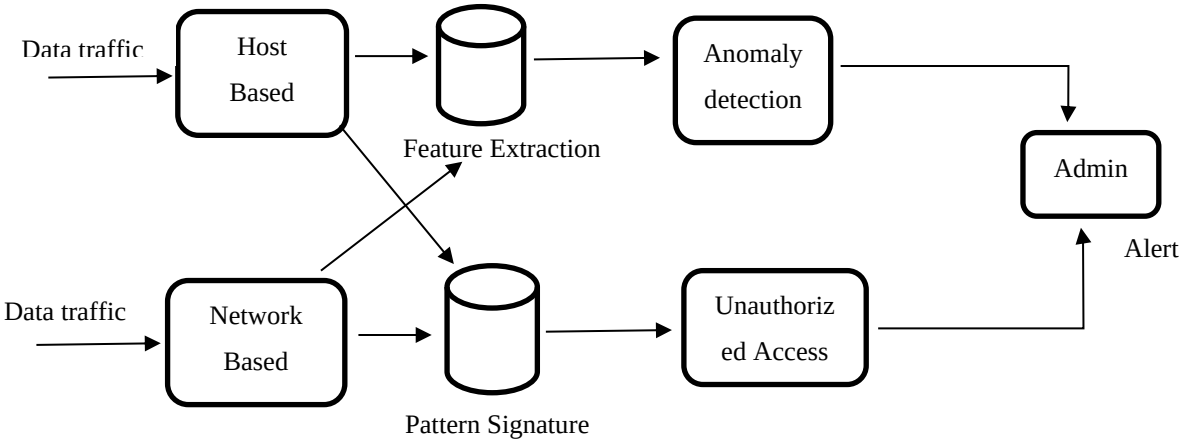

Figure 1: Intrusion Detection System

The operation of an IDS can be categorized into three: Monitoring phase, Analysis phase and Detection phase. The first phase is monitoring phase, which is primarily focused on host or network sensors. The Analysis phase carries out pattern signature identification and feature extraction. Lastly, the detection phase detects malicious or abnormal behaviours and alerts the admin.

Traditional IDS infrastructure is primarily intended to provide security to internet management features, but it falls short in terms of real-time huge volume data stream security. Essentially, regular Intrusion Detection Systems are grouped into three classes which are validation strategy, detection strategy and placement strategy. The detection strategy is being considered more between these three classes and most of existing systems are built primarily on only detection strategies. Anomaly-based IDS, Signature-based IDS, Hybrid IDS and Specification-based IDS are the sub-classes in detection strategies.

Anomaly based IDS - This is a first-stage IDS that collects data and can detect anomalies in the system. Regular and unusual practices are distinguished based on a given threshold, and the network administrator is alerted about the irregularities. It identifies the obscure assaults proficiently; however, huge memory is required, and computational costs are the constraints of this type of intrusion detection system.

Signature based IDS – This identifies and exposes both patterns and attacks in the network. The signature-based discovery framework issues a warning about the suspicious activity and performs design coordination when a network attack is detected. Based on the similarities and differences, the user's access or caution is granted, and the attacks are successfully identified.

Hybrid IDS – This is a model that combines signature and anomaly-based IDSs to provide competitive advantages between processing costs and capacity with fewer false alerts. Lately, a large portion of systems depend on Hybrid IDS because of its improved operations and powerful detections.

Specification based IDS – This is a set of thresholds and rules that define how network components such as routing tables, protocols and nodes should behave. It detects intrusions when network behaviours differ from the defined specifications. This is therefore very similar to the anomaly-based IDS but the only difference is that a human expert is expected tpo manually define each rules for the required specifications. This compared to anomaly-based IDS usually provides lower false positive alerts.

C. Related Works

Due to the growing concerns of privacy and security in IoT, a lot of researchers keep carrying out studies (as already seen in previous sections), either to explore improvements to the current intrusion detection techniques or to bring up new ideas. Irrespective of the enormous studies by many researchers, it is still not efficient enough to use IDSs to combat security issues as demonstrated in [14]. In [15] a study which explores how IDSs are used in IoT and how they can be improved was conducted. A classification scheme that was provided in [16] was used in the paper to rate the current IDSs used in IoT. There were rated using certain criteria like Types of attack, attack responses, detection techniques and implementation strategies.

In Eskandai et al [17], an IDS called Passban was introduced. Passban was described as an intelligent IDS which provides protection for any device that is connected to it directly. Passban IDS is composed of two main phases of operation – learning and training. It relies on machine learning algorithms to create the model of a system after learning the normal behaviour of the system. It uses the created model to detect abnormal behaviours in the system. The IDS can be used directly by very cheap IoT platforms, this means that the system has the capability of taking advantage of every available resource close to data sources to detect threats. A comprehensive study has been carried out in [18]. It proposed a mechanism (Kalis) that configures IDS depending on the configurations and parameters of a network to trigger detection techniques suitable for that specific network. It makes use of anomaly and signature based detection methods. Disadvantage of using Kalis is it causes complexity for the network because it requires detection modules to be integrated to detect each attack type which leads to weak detection rate.

An IDS architecture using Constrained Application Protocol (CoAP) was proposed in [19]. A couple of routing attacks such as insertion, drop and relay attacks were simulated and scenarios including byte exchanges and bit flips for evaluation. They had a lightweight approach, but the execution had poor performance because it was only designed to mitigate routing attacks. They suggested hybrid IDS comprising of both anomaly and signature based detection in their future works which is what would be addressed in this thesis. Benkhelifa et al. [20] however discussed the improvements in intrusion detection systems for the IoT devices. On the other hand, they failed to explore solutions of intrusion detection for MANET, CPS and WSN systems, these systems can also implement for IoT networks. Likewise, in contrast to this paper, concisely they also failed to discuss about implementation issues of intrusion systems for networks in IoT devices.

Restuccia et al. [21] carried out a survey on IoT security research by looking into consideration the application of machine learning and software-defined networking only. Ud Din et al. [22] on the other hand only discussed a survey on trust management techniques for the IoT networks, they failed to consider the evolvements in other fields of IoT security. Further, the authors of the article failed to discuss the challenges faced when deploying intrusion detection systems on real IoT devices. In this research, we are going to look at other intrusion techniques, which will make the research broader and wider.

In [23] a study was carried out where techniques involving supervised, unsupervised, and reinforced learning were deployed. They were implemented for both host-based and network-based security solutions in the IoT environment. According to their research network-based machine learning techniques enables implementations of approaches to be carried out without restriction. The only dent to detection of security issues is the extent to which communications can be encrypted within the network. In the host-based techniques, the implementation of machine learning defence algorithms on IoT devices enables the detection of attacks that may be hard to identify from network traffic alone. Also, they explored some challenges that affect the effective deployment of machine learning techniques. A similar study was carried out in [24] where two deep learning techniques of detection were proposed. In the first model a disjoint training is employed, and data is tested for a deep belief network (DBN) and corresponding artificial neural network (ANN). In the other proposed detection model, DBN is trained using new unlabelled data to provide it with more knowledge pertaining the changes in the malicious attack patterns. After testing and analysis, the experiments showed that these techniques performed better than standard detection algorithms in terms of accuracy. In [25] a similar approach was taken, and it was also confirmed that machine learning based detection algorithms perform better than others, not only in terms of accuracy but also, a higher detection rate and lower false positive rate.

In [26] a project called SORRIR with the aim of developing an IoT platform which is self-organizing for reliable and secure service execution was presented. One of the main aims of the project was to create an abstraction in IoT by separating the application development from other properties of the system. This means that it becomes possible for developers in IoT to carry out the application development without properly knowing the make-up of the underlying technical architecture of the system. A rather different but effective approach was taken in [27] where the blockchain technology was considered a missing key part of the puzzle if a truly decentralized, and secured domain for the IoT can be built. Their main study was to find out how blockchains can effectively be utilized to create a decentralized, secure medium for IoT.

Kouicem, et al. [28] caried out a thorough research of the security and privacy methods used in IoT. They specifically studied the advantages of new approaches for security like block-chain and Software Defined Networking (SDN); they can bring a significant improvement in security and minimize privacy concerns in IoT components, they can also help to improve flexibility and scalability. In the end, they provided a public categorization of current solutions and compared them using key parameters. However, the process of selecting articles is ambiguous. Yang, et al. [29] also carried out a survey on security methods but also considering the resources being use by the IoT devices. They established the constraints in IoT gadgets in terms of battery and computing resources. They suggested that extending the battery life of gadgets and lightweight computing is important moving forward. They have researched on existing security and privacy methods. Even though the authors claim, to have surveyed very recent works, the process of selecting the articles are ambiguous and is not mentioned.

3. Research Methods & Specification

This section gives a detailed explanation on the research methodology, approach, data collection, data pre-processing, data analysis, architecture of the proposed model, implementation, and research limitations. The traditional IDS systems that currently exists fail to detect known security threats and cannot identify most attack traffic which becomes latent over a given period. This research paper proposes a scalable Intrusion Detection system based on CNN-LSTM network to address these issues and improve scalability and accuracy. The methodology used in this research is the CRISP-DM methodology [30]. This is a well proven and robust approach and provides a systematic approach to plan a data mining project. This methodology discusses project stages, their associated roles and tasks, and the connections between them. Trying to identify all connections is not achievable based on the user's goals, interests, context, most importantly, on the data, connections or relationships may exist between any data mining tasks.

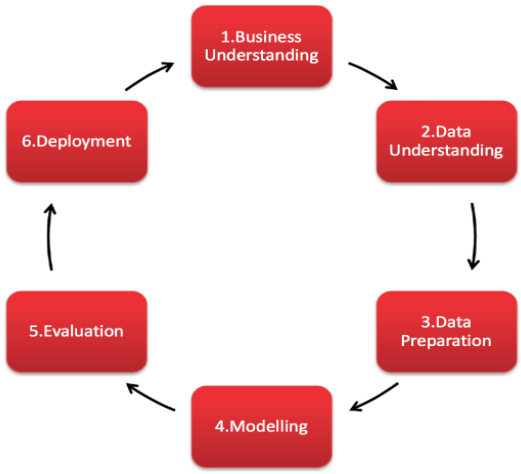

Fig 1: CRISP-DM Methodology

This methodology consists of six stages and at each stage, there may be need to go back and forth between the different stages. The end result at each stage determines the next subsequent tasks to be carried out. Most times, it is important to return to previous tasks and repeat certain actions.

Business Understanding: In this stage, the objectives, and requirements of the project from a business point of view is prioritised, thereafter the knowledge is integrated into a data mining problem and a provisional strategy is made to accomplishing the targets.
Data Understanding: This stage begins with data collection and then processes that assist you in getting to know and analyse the data so that you can identify data quality issues.
Data Preparation: The data preparation stage includes all activities required to construct the final dataset from the original dataset. Activities carried out during Data preparation are likely to be repeated several times and in no particular order.
Modelling: A specific type of modelling technique or different types can be implemented in this stage and their variables tuned to optimal values. If multiple techniques are implemented, the task should be carried out separately for each technique. There are different techniques to tackle the same type of data mining problem.
Evaluation: The quality of the model is analysed and evaluated to determine if it meets business requirements at this stage before proceeding to the deployment phase.
Deployment: In this stage, the results from the evaluation are gathered and a strategy is done to implement them in this phase. If a general process has been created to develop the chosen model, the process is described here to be deployed later.
Companies make use of supervised learning algorithms like Naïve Bayesian, Logistic Regression, Random Forest, and Decision Tree classifiers in detecting network attacks in IDS. In this research four different classifiers are used.

3.1. Naive Bayesian

The Naïve Bayesian algorithm is based on the Bayes theorem with a presumption of independency between predictors. Naive Bayesian methodologies is a set of supervised learning algorithms that use Bayes theory assuming implicit independencies between all sets of features given the value of the class variable. It is a classification technique that presumes that the existence of a class feature is not related to the existence of another feature. As a result, the interdependencies within data are conveniently ignored. Naive Bayes is considered to perform more than even the most advanced classification algorithms due to its simplicity. The Naive Bayes algorithm is simple to develop and is mainly effective for large datasets. Naive Bayes variants are mostly used in improving baseline performance depending on the model of the variant, dataset or task and features. The Naive Bayes classifier requires less training data and outperforms other algorithms such as logistic regression when the assumption of independence is met [31]. Predicting the test dataset class is faster and easier. It also has a good performance when it comes to multi-class prediction. This means the probabilities of multiple classes of target variables can be predicted. It can also be used to make predictions in real time. Compared to other algorithms, Naïve Bayes classifiers have a higher success rate when used in text classification. Therefore, it is widely used in Sentimental analysis (analysis in social media to pinpoint negative and positive user comments).

3.2. K-Nearest Neighbor

The K-Nearest Neighbor algorithm is based on Supervised learning technique and is one of the simplest machine learning algorithms. The K-NN algorithm makes an assumption that the new data and existing datasets are similar and places the new data in a group most identical to the existing groups. It saves all available data and based on their similarity, it classifies a new data point. Therefore, K-NN algorithm can be used to easily classify new data into a well-suited group when it appears. It can be used for both Classification and Regression problems but most often it is used to resolve Classification problems. Instead of learning immediately from the training set, K-NN stores the dataset and performs an action on it at the stage of Classification. At the training phase, it just stores the dataset and when a new data is gotten it is classified into a group more identical to the new data. K-NN classifies new records using a variation of K's most recent major previous records. This is a popular statistical methodology that has been extensively studied in pattern recognition over the years. K-NN is robust to noisy training data and can be more effective when training large a dataset. KNN can be beneficial in resolving problems whose solutions rely on finding similar objects if you have enough computational resources to rapidly handle the data that is being utilized to generate predictions [32].

3.3. Logistic Regression

Logistic regression is an algorithm used to resolve classification problems. Logistic regression assigns observations to a discrete set of classes. It can provide probabilities and makes use of discrete and continuous datasets to classify new data. It makes use of the sigmoid function to convert its output to return a probability value. The sigmoid function is used to map values that have been predicted to probabilities. Based on input variables, it can give the probability of an event to occur or not (in term of 0 and 1). Logistic Regression can also have multinomial effects, such as predicting the preferred type of food menu: Jamaican, Spanish, Irish, and so on. Logistic Regression is used to estimate a categorical target variable. Logistic Regression is a common data mining and statistical technique used by researchers and statisticians to identify, classify, and measure proportional and binary respons**e** datasets [33]. Some of the advantages of Logistic Regression are no scaling is needed with input features, it is very efficient and easy to train, it is not complex to implement, it is easy to regularize, Logistic regression can naturally produce probabilities and implement it in multiclass classification problems. Most often, Logistic regression can be used to resolve issues at an industry-scale level. The same principles of Linear regression is employed in majority of approaches used to study Logistic Regression models. Since a probability score is the result of a Logistic Regression, defining a personalised performance metrics used in obtaining a cutoff point is very vital. It can be used in identifying the target to implement it in resolving business issues. As a result, multicollinearity and tiny data noise has no effect on Logistic regression.

3.4. CNN-LSTM

A Convolutional Neural Network (CNN) is a deep Learning algorithm that takes in an image input, allocate importance like biases and learnable weights to different objects/aspects in the image and be able to distinguish them from each other. When compared to other classification algorithms, the amount of pre-processing needed by a Convolutional Neural Network is significantly less. CNN consists of two different layers called the Convolution and Pooling layers. These layers are multiple building blocks in CNN which helps it adaptively

learn spatial features hierarchies automatically . The convolution layer carries out tasks like feature extraction for the input images. By learning image features with small squares of input data, the relationship between pixels in convolution can be preserved [34]. The pooling layer's primary function is to sub-sample the feature maps. The convolutional operations are used to generate these maps. This method condenses large feature maps into smaller ones and at the same time, it retains the vast bulk of the features in every stage of the pooling process. This process helps CNN learn feature well.

The Long Short Term Memory network is a type of Recurrent Neural Network (RNN) that can learn order dependence in pattern prediction issues. LSTM networks were created to address the vanishing gradient problem that occurs when a typical RNN is trained [35]. Because there can be lags of unknown duration between important events in a time series, LSTM networks are suited well to classify, process, and make predictions based on time series data. In this research paper, these two neural networks are merged to help improve accuracy for network attack detection for the IDS model.

3.5. Performance Metrics

Following the implementation of a machine learning algorithm, the next process is to determine the efficacy of the model based on some metrics making use of test datasets. Like performance metrics, machine learning tasks can be segmented to either Classification or Regression. Different performance metrics can be used to test different machine learning algorithms. It is important to evaluate the efficiency of an algorithm. In this research, after the models were trained, it was evaluated using performance metrics and then computed. In many cases, different algorithms are evaluated to decide which is best ideal for the application. The main aim of a machine learning model is to interpret well on previously unseen data. Performance metrics helps to determine how well the model generalizes on new data [36]. Sometimes people use classification accuracy to calculate the efficiency of the model but it is not enough so other metrics are used to properly evaluate it. In this research, our model is evaluated on the following performance metrics.

Confusion Matrix

It is the most concise and simplest way to determine the performance of a classification issue where the output can be of two or more types of classes. Confusion matrix is one of the easiest metrics used to determine the accuracy and reliability of a model. It describes the results expected from two classes (binary classification problem) or more than two classes (multi-class classification problem). In this research paper, four essential scenarios were considered in computing the confusion matrix.

- False Positive (FP) – It represents an incorrect forecast of a positive attack when, otherwise, the attack that was detected is normal.
- False Negative (FN) – It particularly identifies malicious attack that the model predicted incorrectly as normal. It represents incorrect predictions.
- True Positive (TP) – It determines the number of times of actual positives that attacks were identified correctly.
- True Negative (TN) – It determines the number of times of actual negatives that attacks were identified correctly.

Table 1: IDS Classification Model

		Predicted	
Actual	Normal	TP	FN
	Anomaly	FP	TN

Classification Accuracy

It is defined as the number of correct predictions made divided by the total number of predictions made. It is known as one of the simplest performance metrics for classification algorithms. The accuracy will be high if the dataset is significantly imbalanced, and the model classifies all of the data points as majority. As a result, accuracy should not be the only performance metric to evaluate efficiency of a model.

Precision

This is the number of positives predicted correctly divided by the total number of positives predicted by the classifier. It is used in calculating the ability of the model to correctly classify positive values.

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

Recall

This is the number of true positives divided by the number of true positives in ground truth. It is used in calculating the ability of the model to predict positive values.

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

F1 Score

This is the weighted average of both Recall and Precision. Its range is (0, 1) so the best F1 value is 1 while the worst is 0. It determines how robust and precise the classifier is.

$$F1 = 2 * (precision * recall) / (precision + recall)$$

ROC Curve

Receiver Operating characteristic curve is explicitly used for binary classification, but it can also be used for multiclass classification. It is determined by plotting False Positive against True Positive set at different thresholds.

4. DESIGN SPECIFICATION

In this section, the process flow and design specification of our proposed model is presented and discussed in detail. The diagram below shows the steps taken or activities carried out in the development to achieve the objective of the model for IoT network attacks and threat detection. Choosing the right dataset to test the IDS system is very vital. Dataset is collected containing both normal and malicious traffic, the dumps from the packet traffic was passed into the data preparation and pre-processing stage. Data preparation was done to get rid of incomplete, inconsistent, and irrelevant values. The next phase is the Feature selection phase where optimal features that needed to be trained and tested were extracted. Different attack states in the data such

as U2R (Illegal Access from Remote Machines), DDoS were detected. The next phase shows the four classifiers that were used in training the model, a performance evaluation was then performed on each of them to produce the best classifier. To simulate an IoT infrastructure, the best model is then deployed in a real-time server which would receive packet streams from simulated IoT client devices. The IoT server would classify in real-time the different connection types from the multiple classes of connection and would share the statistics of detection of each device on a dashboard in real-time.

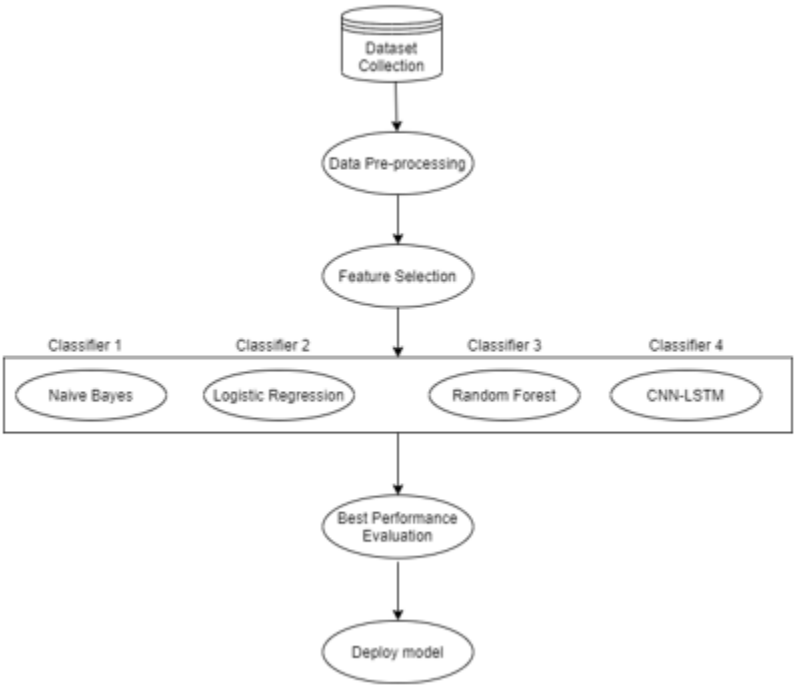

Fig 2: Proposed architectural model design

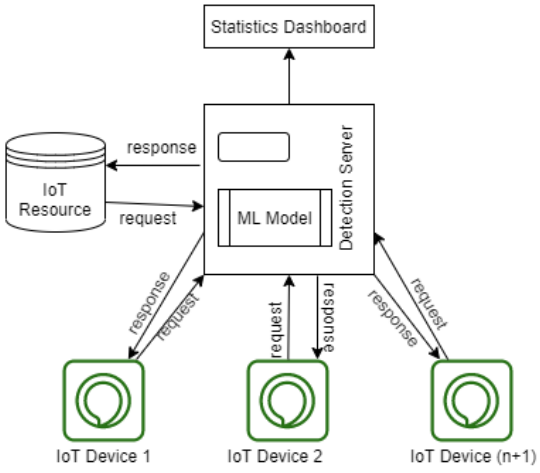

Fig 3: Implementation of model in real-time

4.1. Dataset Collection

Even at the availability of various publicly available datasets for IDS, many of them have inaccurate and inconsistent performance evolutions, unreliable and out of date. Some lack metadata and feature set, they do not show recent attacks and have inadequate traffic volumes and diversity. The CIC-IDS2017 dataset contains more recent and updated network attacks and benign which resembles PCAPS (real world data). This dataset is produced and gotten from the Canadian Institute for Cybersecurity [37]**.** It also includes the results of the network traffic analysis with labelled flows performed with CICFlowMeter depending on protocols and attack (CSV files), source and destination port, source and IP addresses, stamp, and source. The abstract behavioural patterns of users for this dataset was built depending on email, SSH, FTP, HTTPS and HTTP protocols. DDoS, web attack, DoS, brute-force SSH, brute force FTP and infiltration were among the attacks implemented. Also, eleven criterias were used to build this dataset and some of them are complete network configuration and traffic, attack diversity, available protocols, feature set, heterogeneity and metadata.

4.2. Data Pre-Processing

In machine learning, data pre-processing is essential because any useful information such as the data quality gotten from it can directly impact the model's learning ability. Cleaning and preparing your data for training is done during pre-processing. This includes formatting, arranging, standardizing and management of missing information. Therefore, pre-processing data before sending it to the model is really important to avoid errors that would further lead to inaccurate or low accuracy results. In a typical machine learning process, the data is encoded or converted until it gets to a point it can be parsed easily by the machine.

4.2.1. Data Analysis

Data is stored in eight different CSV files, each of which contains different attack data at different times. So firstly, all of the data from the files were combined into a single pandas DataFrame. The dataset contains roughly 2.5 million records across 79 columns. Dataset mostly consists of float64 and int64 types asides from three attributes of 'object' type. Dataset contains of network traffic data during different attacks, represented with values such as packet size, SYN/ACK/FIN/.. flag counts, packet length, IP addresses and port numbers. A closer look reveals that the data comprises 15 labels. Labels represent web or network attacks and the BENIGN state which is the normal business day network traffic.

4.2.2. Data Dimensionality Reduction

Dimensionality reduction is a data preparation technique done after data scaling and cleaning before the model is being trained. Having too many features requires a more complex model meaning more computing power to train the model and more training data is needed. Dimensionality reduction gets rid of any features that affect the performance or not contributing to the accuracy of the model. Therefore, the dataset redundancy is reduced and learning performance of the IDS is improved.

4.3. Feature Extraction

Feature extraction refers to techniques for selecting and combining variables into features to reduce the data amount needed to be processed while still describing the original dataset completely and accurately. The dataset has 78 features and is split into 14 categories (1 normal state and 13 attacks). The dataset was then visualized to see what it looks like in feature space. For this, principal component analysis (PCA) was used

to reduce dimensionality and then pass the reduced dataset to t-SNE (t - Distributed Stohastic Neighbor Entities) for visual representation in 2D space. Principal Component Analysis is an effective tool used to explore information like facial recognition and image compression. It is also a technique used to search for patterns in large amounts of data. The main aim of PCA is to reduce the dimensionality of data while retaining any likelihood of recent changes in the original dataset.

4.4. Modelling

Python libraries such as TensorFlow, Scikit-learn, Keras were used in developing the proposed algorithms used in building the model. The algorithms used are Naïve Bayesian, K-Nearest Neighbor, Logistic Regression and CNN-LSTM. Machine Learning was utilized to execute classifiers such as Logistic regression and Naïve Bayesian to categorize incoming traffic (as either normal or malicious traffic). The training set is then split into two parts; 20% was used for testing the system and 80% for training the system. The classifiers were developed to be able to differentiate between malicious and normal incoming traffic in a binary classification setting after which we will be able to test it if it works accurately [38]. Grid search and cross validation was performed while these algorithms were being trained for tuning the hyperparameters. For every scenario, the model with the best performance will be chosen to assess the test dataset. CNN-LSTM was utilized to detect any form of malicious threats. Deep Learning is a detection technique used in LSTM to develop a module which is being used as a benchmark profile for malicious threats. Convolutional Neural Networks comprises of two segments naturally named Classifier and Feature extractor. The feature extractor is made up of two layers known as pooling and convolutional layers [39]. The result gotten from the extractor is then forwarded to the classifier which is the second segment. The LSTM layer is placed after the Convolutional layer to ensure an extensive dependency and efficiently learn from variable sequences [40].

5. IMPLEMENTATION

In this section, all the tools and technologies used in the implementation methodology to develop the model are discussed and explained.

5.1. JavaScript

JavaScript is an object-oriented, text-based programming language that is used to make web pages interactive on both server-side and client-side. JavaScript is used mainly for web browsers and web-based applications but can also be used for other purposes such as machine learning as in the paper. JavaScript is primarily supported by all PC and mobile browsers which means ML applications are sure to run on most mobile and PC devices. JavaScript was used to build the monitoring dashboard.

5.2. Python

Python is also an object-oriented, high-level programming language used for various purposes such as data analysis, data visualization, programming applications, artificial intelligence, and machine learning. Humans easily understand python codes which makes it very easy to build machine learning models. It gives a precise and readable code. It takes a lot of time to implement AI and ML algorithms. To help developers come up with best coding solutions, it is important to have an environment that is well structured and tested. With Python's rich stack of technology, it has wide libraries set for machine learning and artificial intelligence.

5.3. Libraries

A software library is a collection of pre-written codes that is used by developers to resolve common programming tasks. Python frameworks and libraries are used by programmers to reduce development time. Python has an extensive set of in-built libraries such as Seaborn, Matplotlib, TensorFlow, Pandas, NumPy and so on. Some of the ML libraries used to implement the proposed model are TensorFlow, Keras, SciKit-Learn, Pickle.

TensorFlow

This is an open-source machine learning framework used to carry out high performance numerical computations. It offers great architectural support, allowing for simple computation deployment over a wide range of platforms, from edge devices to mobiles, servers, and desktops. It carries out high level tasks needed to build advanced neural network models and provides flexibility such that functionalities for your model can be defined. TensorFlow was used to build the CNN-LSTM model.

Keras

Keras is a Python-based deep learning API that runs on top of the TensorFlow machine learning platform. The main aim of developing Keras is to enable fast experimentation. It reduces the number of necessary user actions for typical use cases. It offers simple and consistent APIs. Keras offers a simple way to run neural networks and makes deep learning accessible which helps developers sequentially learn complex features from input data. Keras was used as an API for TensorFlow to build the CNN-LSTM model.

Scikit-Learn

This is a python library used in implementing machine learning models for clustering, classification, regression as well as statistical tools to analyse them. It offers functionalities for inbuilt datasets, ensembling techniques, feature extraction, feature selection and dimensionality reduction. This was used to build K-NN, Logistic Regression and Naïve Bayes classifier models.

Pickle

Pickle is a python library used to serialise and deserialise a python object structure. In Python, any object can be pickled and saved to disk. To reuse the trained models, it must be saved in a file and restore them to compare the models with each other and test the models. After performing label encoding on our models, Pickle was used to save the encoders.

5.4. Twisted Framework

This is a framework used to write event-driven asynchronous network programs in python. It provides support for GUI frameworks as well as for network protocols like SMTP and HTTP. It was used to simulate IoT server and devices in this research**.** This tool was used to build the deployment server for the IoT devices.

5.5. Jupyter Notebook

This is an open source web application that helps in developing and sharing informative text documents, visualisations, equations and live codes. It is used for machine learning, mathematical modelling, numerical

simulation, data cleaning and conversion. This tool was used for data preparation, data analysis and building of the machine learning models.

6. EVALUATION

The efficiency of the proposed model is evaluated in this chapter. We carried out experiments on the CIC-IDS2017 dataset to analyse and compare the performance of all the algorithms to our proposed model in detecting IoT Network intrusions and DDoS attacks. After the experiments were done, the classifier with the best performance evaluation was deployed**.**

6.1. Experiment 1: Naïve Bayes Classifier

The first algorithm is the Naïve Bayesian classifier and the results gotten are shown below in the diagram. The algorithm got an accuracy, precision, recall and F1-Score of 74%, 92%, 74% respectively F1-score and ROC of 80%.

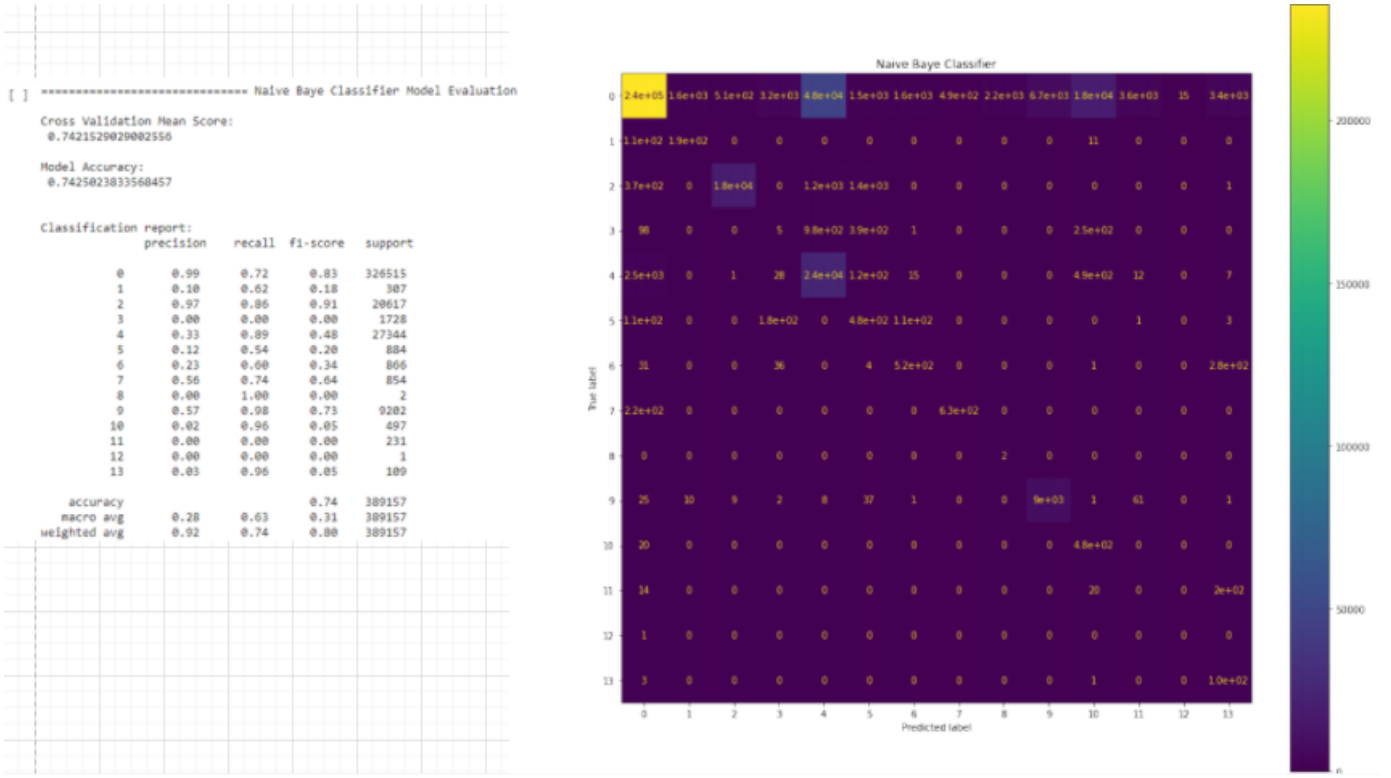

Fig 3: Naïve Bayes Classifier Precision, Accuracy, Recall and F1-Score

6.2. Experiment 2

The second algorithm that was evaluated is the Logistic Regression classifier and the results gotten are shown below in the diagram. The algorithm got an accuracy, precision, recall and F1-Score of 81%, 81%, 81% respectively, F1-score and ROC of 80%.

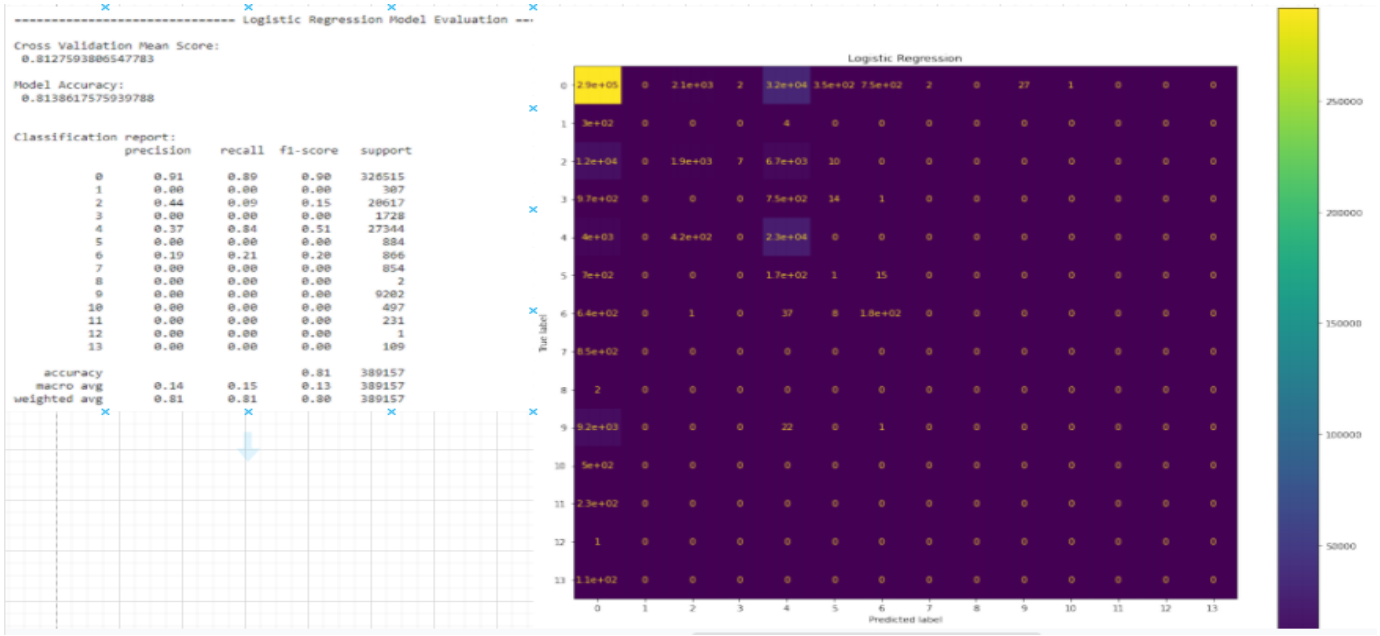

Fig 4: Logistic Regression Classifier Precision, Accuracy, Recall and F1-Score

6.3. Experiment 3

The third algorithm that was evaluated is the K-Nearest Neighbor classifier and the results gotten are shown below in the diagram. The algorithm got an accuracy, precision, recall and F1-Score of 98.7%, 99%, 99% respectively, F1-score and ROC of 99%.

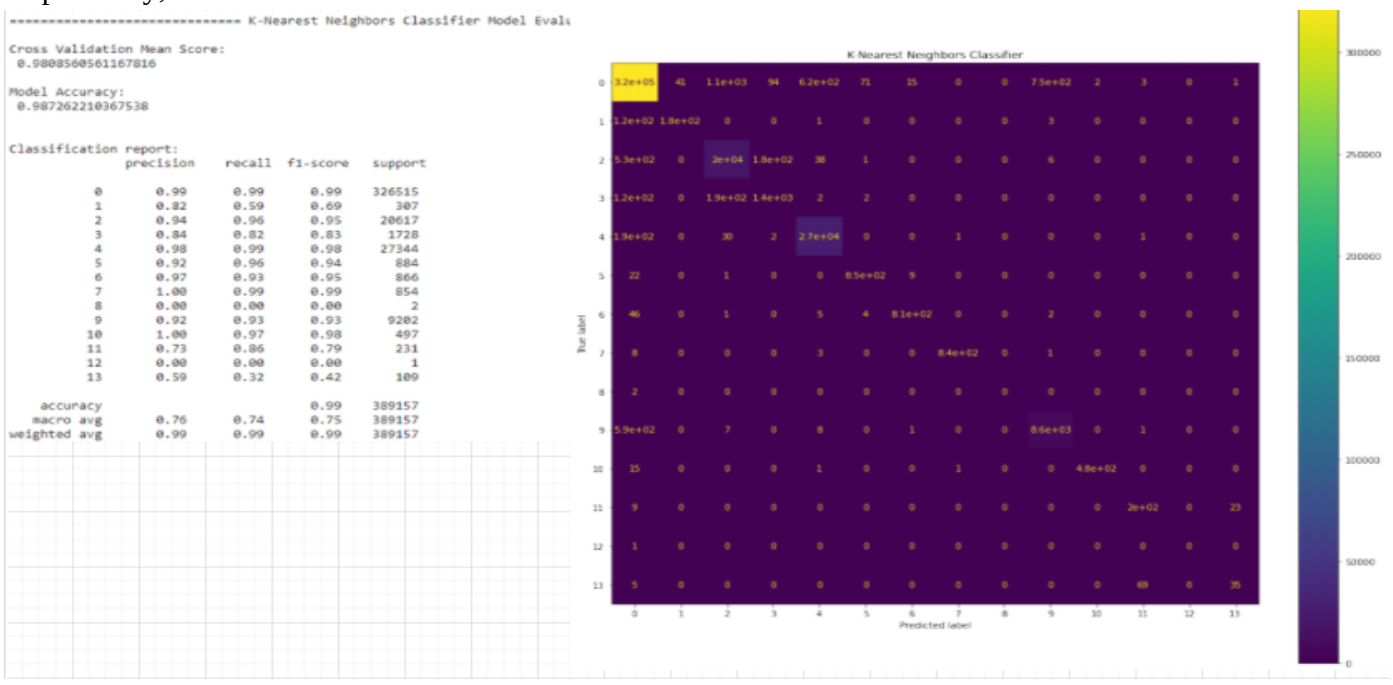

Fig 5: K-Nearest Neighbor Classifier Precision, Accuracy, Recall and F1-Score

6.4. Experiment 4

The last algorithm that was evaluated is the CNN-LSTM classifier and the results gotten are shown below in the diagram. The algorithm got an accuracy, precision, recall of 99.82%, 100%, 100% respectively, F1-score and ROC of 100%.

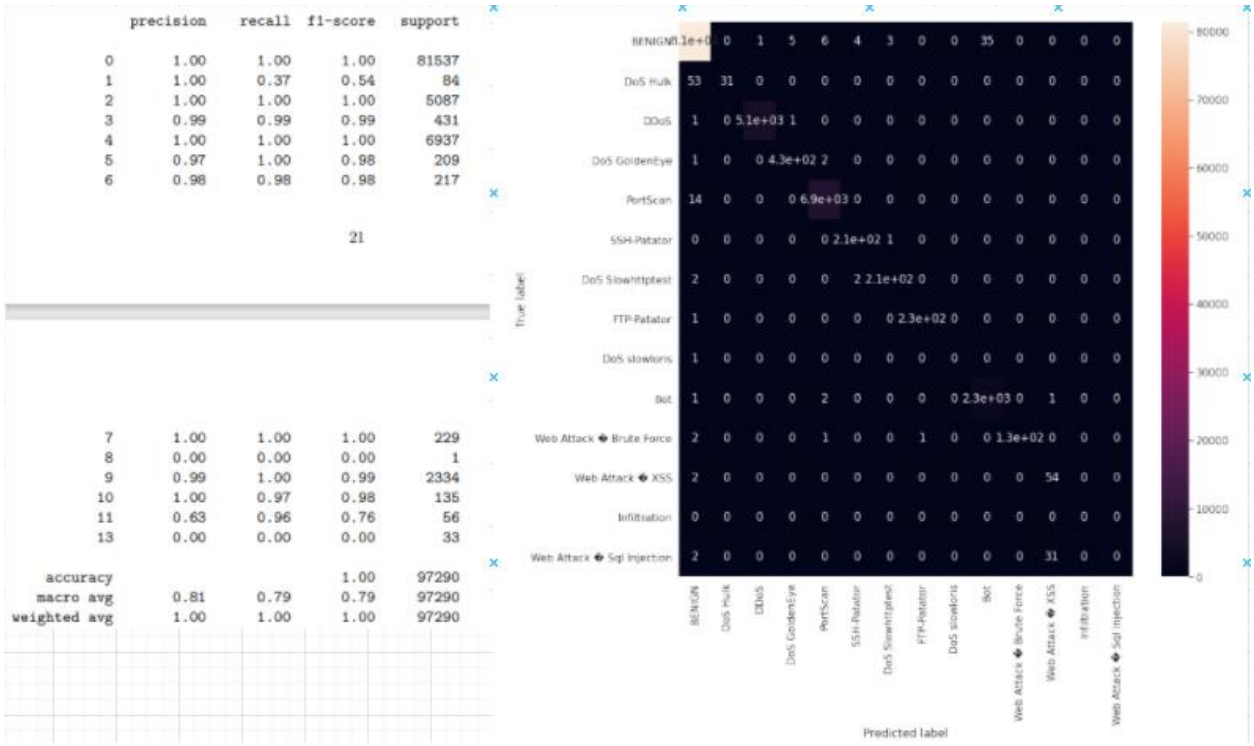

Fig 6: CNN-LSTM Classifier Precision, Accuracy, Recall and F1-Score

Algorithms	Accuracy	Precision	Recall	F1 Score	ROC Curve
Naïve Bayes	74.25	92.0	74.0	80.0	80.0
K-Nearest Neighbor	98.72	99.0	99.0	99.0	99.0
Logistic Regression	81.38	81.0	81.0	80.0	80.0
CNN-LSTM	99.82	100	100	100	100

Table 2: Evaluation Comparison of Algorithms

6.5. Discussion

In this research, experiments were carried out to prove that the CNN-LSTM model can be used to achieve higher detection accuracy for IoT network intrusions and anomalies compared to the other algorithms for multi-classification problems which is important for research done in Cybersecurity. After the performance evaluation of all algorithms, we can see that the proposed model can have a good security level against intrusions using this IDS model which is vigorous, fast, simple and can be used in real time detections too. CNN has effective features such as locality and pooling which helps in improving identification of network anomalies. The extraction features in the convolution layers are potent to noises and the locality feature prevents noisy effects on the data. While LSTM was implemented after the CNN layer to improve the ability to efficiently learn from variable extent sequences and ensure longer dependencies. This helped the proposed

model achieve better results compared to the other algorithms because LSTM can also be used as a Signature-based Detection technique because it is able to store attack signatures whenever intrusions occur. LSTM can keep track records of data in a period of time which makes increases its ability to accurately predict attacks. They are useful in time-series prediction because they recall previous inputs. Also, the hyperparameters (learning rate) was tuned to improve accuracies for the testing and training phase. However, a fallback in this research is that experiments were carried out on just one dataset. It is possible other classifiers might achieve a better result if other datasets were to be used. Also, experiments were carried out with three other algorithms. Other algorithms such as SVM, Random Forest may achieve competitive results with the proposed model. These algorithms could not be implemented due to some technical issues that was faced during the implementation. Nonetheless, we have seen that it is possible to a great security level using this model. Some of the limitations of this project is that the model can only detect anomalies found in the dataset. Also, in the simulation, it can only work with IoT devices that has been added to the monitoring dashboard.

7. Conclusion

Integrating both networks together has given a predictive accuracy of 99.82% after which the model was deployed in real-time to identify IoT network intrusions which helps in answering the research question "How can Intrusion Detection Systems act as an enabler in solving rapidly growing privacy issues in IoT?" The proposed IDS is based on the CNN-LSTM network and ML and was evaluated on the CIC-IDS2017 dataset. It was compared to the other algorithms Naives Bayes, K-Nearest Neighbor and Logistic Regression and had the highest accuracy of 99.82% after performance evaluations were carried out in each of the experiments done. It combines both networks which can be an effective approach for Anomaly-based detection. To simulate an IoT infrastructure, the best model is then deployed in a real-time server which would receive packet streams from simulated IoT client devices. One of the most attractive features asides from accuracy is that it also reduces computing complexity. Intrusion Detection Systems play a very important role in the network security field. It helps to learn and identify triggers and behavioural patterns to be efficient in detecting intrusions. As it provides greater visibility across the entire network, it aids companies in meeting security regulations and maintaining regulatory compliance**.** For future works, firstly, we would like to implement other algorithms to compare with the CNN-LSTM model and see if they achieve better results. Also, as stated in the previous section just one dataset was used. Carrying out experiments using other datasets would also be important in analysing our model and compare its performance with other algorithms.

4. REFERENCES

[1] W. Meng, "Intrusion Detection in the Era of IoT: Building Trust via Traffic Filtering and Sampling," *Computers,* vol. 51, no. 7, pp. 36-43, 2018.

[2] W. T. S. M. W. &. W. Y. Li, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Future Generation Computer Systems,* vol. 96, pp. 481-489, 2019.

[3] S. W. K. B. M. &. N. N. J. Hajiheidari, "Intrusion detection systems in the Internet of things: A comprehensive investigation," *Computer Networks,* vol. 160, pp. 165-191, 2019.

[4] R. &. N. M. Gaddam, "AN ANALYTICAL APPROACH TO ENHANCE THE INTRUSION DETECTION IN INTERNET OF THINGS NETWORK," *International Journal of Latest Trends in Engineering and Technology,* vol. 9, no. 3, pp. 258-267, 2018.

[5] J. A. ,. M. A. A. R. S. K. A. M. &. I. R. Arshad, "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT," *Electronics,* vol. 9, no. 4, 2020.

[6] G. S. R. T. N. &. K. D. N. Reddy, "An Intrusion Tolerance Approach for Internet Security," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 1, no. 8, pp. 536-543, 2017.

[7] L. W. X. L. X. Z. Y. &. W. D. Xiao, "IoT Security Techniques Based on Machine Learning," 2018.

[8] E. W. ,. T. &. H. W. Benkhelifa, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 4, pp. 3496 - 3509, 2018.

[9] P. H. Zeeshan Ali Khan, "Recent Advancements in Intrusion Detection Systems for the Internet of Things," *Security and Communication Networks,* 2019.

[10] A. S. M. A. a. B. S. U. Javaid, "Mitigating IoT Device based DDoS Attacks using Blockchain," *CryBlock'18,* pp. 71-76, 2018.

[11] N. D. A. G. A. S. Michele De Donno, "DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation," *Security and Communication Networks,* 2018.

[12] A. G. a. M. M. N. Dragoni, "The internet of hackable things," in *5th International Conference in Software Engineering for Defense Applications (SEDA16)*, 2016.

[13] K. S. S. a. P. G. S. S. Rajashree, "Security with IP Address Assignment and Spoofing for Smart IOT Devices," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India, 2018.

[14] H. U. a. H. b. P. T. Sherasiya, "A SURVEY: INTRUSION DETECTION SYSTEM FOR INTERNET OF THINGS," *International Journal of Computer Science and Engineering,* vol. 1, no. 5, pp. 81-90, 2016.

[15] Z. A. Khan and P. Herrmann, "Recent Advancements in Intrusion Detection Systems for the Internet of Things," *Security and Communication Networks,* 2019.

[16] "A survey on intrusion detection in mobile ad hoc networks," in *Wireless Network Security*, Springer-Verlag, 2007, p. 159–180.

[17] . E. Mojtaba, J. H. Zaffar , V. Massimo and A. Fabio , "Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices," *IEEE INTERNET OF THINGS JOURNAL,* vol. 7, p. 6882, 2020.

[18] D. Midi, A. Mudgerikar, E. Bertino and A. Rullo, "Kalis — A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[19] S. Peter and J. Krimmling, "Integration and evaluation of intrusion detection for CoAP in smart city applications," in *2016 IEEE Conference on Communications and Network Security*, 2016.

[20] T. W. a. W. H. E. Benkhelifa, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 4, pp. 3496-3509, 2018.

[21] S. D. a. T. M. F. Restuccia, "Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking," *IEEE Internet of Things Journal,* vol. 5, no. 6, 2018.

[22] M. G. B. K. S. H. a. M. K. K. I. Ud Din, "Trust Management Techniques for the Internet of Things: A Survey," *IEEE Access*, vol. 7, pp. 29763-29787, 2019.

[23] S. Zeadally and M. Tsikerdekis, "Securing Internet of Things (IoT) with machine learning," *Zeadally, Sherali; Tsikerdekis, Michail (2019). Securing InterInternational Journal of Communication Systems* , 2019.

[24] S. Huda, S. Miah, J. Yearwood, S. Alyahya, H. Al-Dossari and R. Doss, "A malicious threat detection model for cloud assisted internet of things (CoT) based industrial control system (ICS) networks using deep belief network," *Journal of Parallel and Distributed Computing*, 2018.

[25] M. AL-Hawawreh, N. Moustafa and E. Sitnikova, "Identification of malicious activities in industrial internet of things," *Journal of Information Security and Applications*, 2018.

[26] P. Eichhammer, C. Berger, H. P. Reiser, J. Domaschka, F. J. Hauck, G. Habiger, F. Griesinger and J. Pietron, "Towards a Robust, Self-Organizing IoT Platform for Secure and Dependable Service Execution," in *Tagungsband des FB-SYS Herbsttreffens 2019*, Bonn, Gesellschaft für Informatik e.V., 2019.

[27] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui and F. Antonelli, "Applications of Blockchains in the Internet of Things: A Comprehensive Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 21, no. 2, 2019.

[28] A. B. a. H. L. D. E. Kouicem, "Internet of things security: A top-down survey," *Computer Networks*, vol. 141, pp. 199-221, 2018.

[29] L. W. G. Y. L. L. a. H. Z. Y. Yang, ""A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250-1258, 2017.

[30] S. Huber, H. Wiemer, D. Schneider and S. Ihlenfeld, "DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model," *Procedia CIRP*, vol. 79, pp. 403-408, 2019.

[31] S. Ernawati, E. R. Yulia, F. and S. , "Implementation of The Naïve Bayes Algorithm with Feature Selection using Genetic Algorithm for Sentiment Review Analysis of Fashion Online Companies," in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, 2018.

[32] S. Zhang, X. Li, M. Zong, X. Zhu and D. Cheng, "Learning k for kNN Classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 3, p. 1–19, 2017.

[33] D. Xu, S. Yuan and X. Wu, "Achieving Differential Privacy and Fairness in Logistic Regression," in *WWW '19: Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

[34] Z. Wang and Z. Qu, "Research on Web text classification algorithm based on improved CNN and SVM," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, 2017.

[35] A. Diro and N. Chilamkurti, "Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 124 - 130, 2018.

[36] M. Almseidin, M. Alzubi, S. Kovacs and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 2017.

[37] "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," Canadian Institute for Cybersecurity, [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. [Accessed 5 June 2021].

[38] H. Zenati, M. Romain, C.-S. Foo and B. Lecouat, "Adversarially Learned Anomaly Detection," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018.

[39] S. Kido, Y. Hirano and N. Hashimoto, "Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN)," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018.

[40] Y. Yu , X. Si, . C. Hu and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation (2019) 31 (7): 1235–1270.*, vol. 31, no. 7, 2019.

[41] Q. L. Nguyen and A. Sood, "Comparison of Intrusion Tolerant System Architectures," InfoQ, 2021. [Online]. Available: https://www.infoq.com/articles/intrusion-tolerant-system-architectures/.

[42] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 18, 2021.

[43] D. K. L. &. B. N. Djenouri, "A survey of security issues in mobile ad hoc and sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 4, pp. 2-28, 2005.

[44] G. R. a. B. C. M. Ammar, "Internet of Things: a survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, p. 8–27, 2018.