Acknowledgement

We are extremely thankful to Prof. M.R Dhage for her expert guidance and continuous encouragement throughout to see that adequate research has been conducted to approve the project. Collectively, we would also like to thank our project committee members Prof. E. Jayanthi and Prof. A.S. Kalaskar for their time, suggestions, and for graciously agreeing to be on our committee, and always making themselves available. We would like to express deepest appreciation towards Prof. M. P. Wankhade, Head of Department of Computer Engineering, Dr. S. D. Lokhande, Principal, Sinhgad College Of Engineering.

ABSTRACT

IoT devices are becoming popular day-by-day. Several vulnerabilities in IoT present the need for IoT security. The number of attacks on these devices keep increasing and most of them are slight variations of the previously known attacks, which can bypass the conventional firewall systems.

The existing systems are not suitable for IOT devices as IOT devices have low computational power. Those that use signature-based intrusion detection. It works only on known patterns and attacks, hence they cannot recognize newer attacks with unknown pattern. Also, many systems use cloud computing, which has a downfall that it needs access to internet at all times, also the cloud services are most often paid.

In the proposed system, we are planning to use anomaly-based detection system. The anomaly-based intrusion detection system comes into effect when detecting newer attacks, that are not filtered by the firewall. It is capable of handling newer/unknown attacks, which signature based cannot. Also we are setting up the IDS on a local higher powered device rather than on cloud. Machine learning ensemble model Random forest is used. The model will be trained on the DS2OS traffic traces dataset.

INDEX

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Background and Basics

Internet of things, or IoT, is a system of interrelated ”things ” that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. These ”things” could be computing devices, mechanical and digital machines such as build in sensors, monitors etc.

A good example of a network of IoT devices is the security system implemented by various locations which include, CCTV cameras, motion sensors, automatic locks, smoke detectors, temperature sensors etc.

IOT devices are becoming pervasive. They are used extensively in a lot of fields and their utility is just going to keep increasing. IOT devices help to automate things, reduce labour costs and facilitate smart living.

Hence, it is important to build an optimal system that can provide proper safety and security measures for IOT devices.

An intrusion detection system is one such system that can be a building block in making the IOT network as secure as possible. An intrusion detection system is a system that passively monitors the data exchange in the network and of the network with externals entities and looks for malicious activities that can be classified as an intrusion or attack. It then notifies the user or sends a notification to some other system which may or may not take action against the detected intruder. Simply put, if the network of devices is a home, an intrusion detection system is a CCTV camera.

Also, a step up from just a intrusion detection system is an anomaly based intrusion detection system. This type of system creates a profile of normal behaviour, and any activity that falls outside of the normal category is marked as anomalous. Anomaly based system is better suited to defend the system against zero-day attacks.

There are certain challenges while building this system which are specific to IOT devices. For example, the low computational powers and limited amount of resources (such as space/memory). The IDS system should be build keeping in mind all these challenges and they should be overcome in the most efficient manner.

1.2 Literature Survey

Table 1.1: Literature Survey

Title	Year	Publication	Data-set	HIDS /NIDS	Anomaly /Signature	Model used	Drawbacks
Anomaly based intrusion detection system through feature selection analysis and building hybrid efficient model	2018	Journal of computational science 25(2018) 152-160	NSL-KDD	NIDS	Anomaly	Hybrid	Not implemented for iot Devices
Towards Machine Learning Based IoT Intrusion Detection Service	2018	Springer International Publishing AG, part of Springer Nature 2018	UNSW-NB15	NIDS, cloud	Anomaly	Neural network, random forest	uses cloud computing, which increases the resource requirements
Distributed attack detection scheme using deep learning approach for Internet of Things	2017	Future Generation Computer Systems (2017)	NSL-KDD	NIDS, fog	Signature	Deep learning	signature based, new attacks cannot be recognized

Table 1.2: Literature Survey

Host based intrusion detection system with combined CNN/RNN model	2019	Springer Nature Switzerland AG 2019	ADFA-LD	HIDS	Anomaly	RNN with GRU	HIDS, and not NIDS Performance doesnt match ensemble models
Anomaly-based intrusion detection of iot device sensor data using provenance graphs	2018	1st International Workshop on Security and Privacy for the Internet-of-Things (IoTSec)	longitudinal data on the thermal	HIDS	Anomaly	Provenance graphs	Works with offline data, doesnt work with real time data

1.3 Project Undertaken

1.3.1 Problem Definition

Design and implement an anomaly Based Intrusion Detection System in IoT Networks using Random Forest which will passively monitor the IoT network traffic and alerts the user for any intrusion detected.

1.3.2 Scope Statement

Product can be used in any IOT network for the purpose of the security. Since use of IOT devices is increasing, day by day, in all fields and sector, there is an increasing need for security mechanisms designed specifically for IOT devices or networks. The main function of the IDS system is to detect anomalous behavior that can be caused by the attacks. So it can be deployed anywhere where we need security measures in place to protect IOT devices.

1.4 Organization Of the Project Report

The project report is organized as follows:-

1.4.1 Chapter 1

Chapter 1 is Introduction. It gives the background and basics of the project. It is followed by a detailed literature survey of similar works in the past. Problem statement and scope of the project are defined as well.

1.4.2 Chapter 2

Chapter 2 is Project Planning and Management. It has details of the system requirement specifications which include functional and non-functional requirements, system overview, deployment environment environment, external interface and other requirements. The project process model applicable to this project is also mentioned. Cost estimate analysis and time line scheduling is done as well.

1.4.3 Chapter 3

Chapter 3 is Analysis and Design. It consists of idea matrix, mathematical model and feasibility analysis. All the analytical and design diagrams are also included in this chapter. These diagrams include use case diagram, activity diagram, architecture diagram, class diagram, ER diagram, sequence diagram, state transition diagram and deployment diagram.

1.4.4 Chapter 4

Chapter 4 is Testing. In this chapter, test cases regarding different types of testing are given. The types of testing included are unit testing, integration testing and acceptance testing.

Chapter 2

PROJECT PLANNING AND MANAGEMENT

2.1 Detail System Requirement Specification

2.1.1 System Overview

Product Perspective

The system has been inspired by the already existing intrusion detection systems which attempt to protect the home-network against attacks and intrusions. We are taking these pre-existing models and combining them to get additional advantages and reduce the downsides as much as possible. Till date, there have been very few attempts at making an IDS for IOT devices. Most of the IDS present in the market cater to non-IDS networks. We are taking the principles of these IDS systems and modifying them as per requirement of an IOT network.

Product Functions

1) User authentication
2) Data connection establishment
3) Intrusion detection analysis
4) Notification

User Classes and Characteristics

Factories and companies employing a network of IOT devices. Usually the IOT networks for these users are larger and require higher level security systems as economic factors are involved. Residential habitants employing network of IOT devices. Includes home-IOT networks (smart homes), or residential society IOT networks like CCTVs, smoke detectors, fire alarms etc. These networks are usually smaller than the previous.

Operating Environment

The system will be implemented on Windows. Python is used for programming the backend using Django Framework. Languages such as Python and MySQL and platform like XAMPP are used for database and connection. Libraries like sklearn, pandas etc. will be used. For real life implementation, IOT data will be connected from Wireshark.

Design and Implementation Constraints

- The user will be notified through a web app, which should be installed on his system.
- All the nodes of the IOT network should be connected to the main server where the processing is to take place.
- The device to be notified also has to be part of the given network.
- Processing time should be as low as possible.
- Backend should be connected to database.
- Zigbee(IEEE 802.15.4) supported on all the IOT devices.
- Server has an Zigbee adapter.

User Documentation

- User manual.

Assumptions and Dependencies

- We are assuming that the machine has the required resources (memory and processing power etc.) and capabilities to run the system.
- The user has updated system.

We are assuming that the system has the required packages and dependencies (such as Django and Xampp or SQL) to run the system.

2.1.2 External Interface Requirements

User Interfaces

1)User login page
Username
Password
Login(Button)
2)User Notification page Notifies when the network is under attack or some anomalous activity is detected.

Hardware Interfaces

Hardware interfaces for a web application will run only on windows. Other applications can be created on other operating system like Apples IOS, Linux, Android, etc. For connection of IOT nodes to the main server will require a network. For a larger model, the network and servers will increase. IOT nodes are embedded with Arduino microcontroller which uses Zigbee protocol (IEEE 802.15.4). The main servers needs to be fitted with a Zigbee adapter to enable data transfer from IOT nodes.

Software Interfaces

1)Operating system-We have chosen Windows operating system because of its user friendly features, benefits of security and control without complexity and unrealistic costs.
2)Database-Mysql database using Xampp.
3)Python-To program and implement the backend. The web application is connected to the backend which is on the network server. The server is connected to all the IOT nodes of that networks. These nodes transfer data to this server. The backend then processes this data to detect abnormal behavior.

Communications Interfaces

1)Web application runs on a web browser
2)HTTP is used for notification system
3)Zigbee (IEEE 802.15.4) is used for data transfer from IOT nodes to the server.

2.1.3 System Features

Sensing of the data using IOT sensor

Description and Priority

First, sensors or devices collect data from their environment. The benefit of this feature is much larger and the cost of this feature is reasonably low. Only risk of this feature is that it sometimes may not work as intended.

Stimulus/Response Sequences

Once the IOT device is started it generates a constant stream of data which is sent to the network server. Based on this data, an intrusion or attack can be detected.

Functional Requirements

IOT sensor nodes equipped with Arduino.

Connectivity

Description and Priority

The sensors/devices is be connected to the network through Zigbee protocol. This feature is of high priority because the sensed data needs to be processed further. The data will be processed only if it gets proper way to reach to the server. Only risk of this feature is that it can sometimes not send the data to the server.

Stimulus/Response Sequences

Data sensed is sent to the server.

Functional Requirements

Main server of the network along with ZigBee adapter.

Data Processing

Description and Priority

Once the data gets to the server, software performs processing on it. Here IDS software is used to determine the category of the attack. This feature is of high priority because this is the main objective behind selecting this project. Only risk of this feature is that it can sometimes not categorize the attack perfectly.

Stimulus/Response Sequences

In response, it will determine whether the attack was normal or malicious.

Functional Requirements

IDS software.

User interface

Description and Priority

Next, the information is made useful to the end-user in some way. This is an alert to the user (text, notification, etc.). This feature is of medium priority. Only risk of this feature is that it can sometimes not work as intended.

Stimulus/Response Sequences

In response it will give notification of the attack on the GUI.

Functional Requirements

GUI

2.1.4 Non- Functional Requirements

Performance Requirements

The data transfer from the IOT nodes to the main server should take as little time as possible for proper real time functioning of the system. Also, the processing of the data should be fast enough so as to give proper result as soon as the attack happens, i.e. the time between the occurrence and detection of the attack should be minimum.

Safety Requirements

The backend could crash resulting in failure of the whole system. If the system takes up all the processing power, that machine could crash. Regular maintenance of the networking components, such as checking the proper functioning of the Arduino and Zigbee adapter should be done.

Security Requirements

User need to login to get access to the system, in this case users data should be protected. Systems should be secure against unauthorized access to any of their data, unauthorized use of them or any of their components. Details regarding IOT devices and their data should also be protected.

Software Quality Attributes

Our software has many quality attribute that are given below: -

- Availability: This software is freely available to all users. The availability of the software is easy for everyone.
- Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
- Reliability: The performance of the software is better which will increase the reliability of the Software.
- User Friendly: Since, the software is a GUI application, the output generated is much user friendly in its behavior.
- Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
- Security: Users are authenticated using many security phases so reliable security is provided.

2.1.5 Other Requirements

1)Configuration- -Systems should be configurable for detection and reaction, as feasible for alerts, updates, protocol and port coverage, and detection-threshold levels -Systems should be configurable treat identified IP or other network addresses exceptionally; for example, configurable to never block or shun network activity from one or more IP addresses.
2)System Updates-These capabilities should be engineered in such a manner as to allow updates during operational use of the products without disruption.
3)Ease of use-Operator console should not require undue expertise to operate; experts may reside external to operator staff, experts may reside external to operator stuff.

2.2 Project Process Modeling

Iterative waterfall model is the best suited for this project. Iterative waterfall model can be thought of as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost same as the classical waterfall model except some changes are made to increase the efficiency of the software development.

The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.In our project, every phase is well defined and to be executed one after the other sequentially, like the waterfall model. But if need be, there is space for going back to previous stages making changes. Hence, iterative waterfall is to be used for this project.

2.3 Cost Efforts Estimates

As per the basic COCOMO cost estimation formula projected cost for our product,
Development Effort = a1 x (KLOC)a2PM
= 2.4 * (4)1.05 = 10 Person Months(approx)

Nominal development time=b1 x (Effort)b2 Months
= 2.5 * (10)0.38 = 6 months(approx)

Cost required to develop the product = NDT x Average salary per month x members

=6 * 5000 * 4

=Rs 120,000/-

Where, KLOC is the estimated size of the software product expressed in Kilo Lines of Code

a1, a2, b1, b2 are constants for each category of software products

2.4 Project Scheduling

Time Line Chart

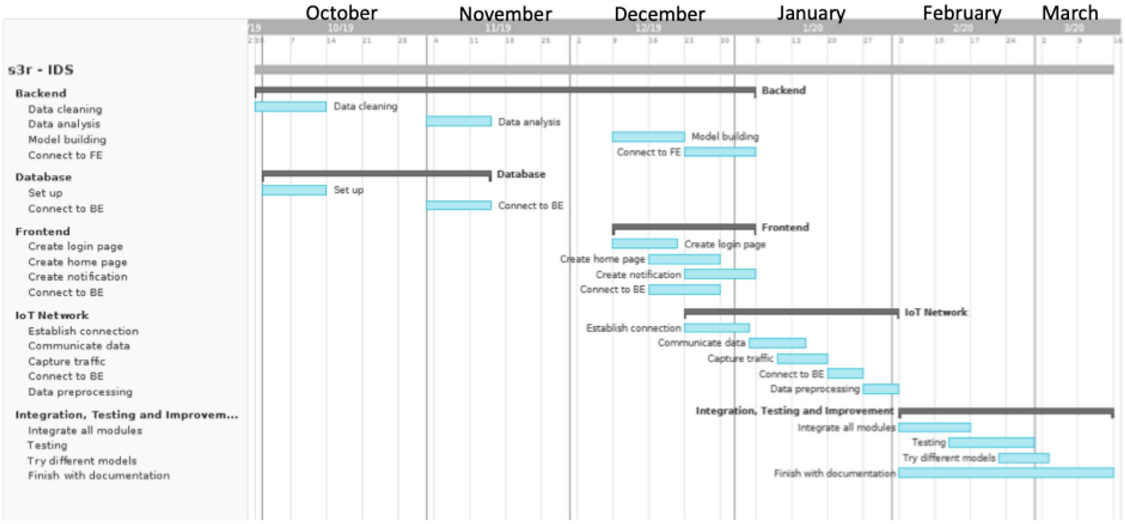

Figure 2.1: Time Line Chart

Chapter 3

ANALYSIS DESIGN

3.1 IDEA matrix

Table 3.1: Idea Matrix

Idea Matrix		
Increase	1)It increases Network security 2)It increases network monitoring	1)Network security 2)Network monitoring
Improve	1)It improves zero day attack detection 2)Improves IoT security 3) Improves awareness of user about network intrusions	1)Unknown attacks 2)IoT security 3) User awareness
Ignore	1)It ignores the normal behavior in the network	1)Normal behavior
Deliver	1)It delivers efficient intrusion detection system for IoT devices	1)IoT security
Document	1)Documentation of the project for researchers to understand and improve on the project in the future.	1)Project documentation
Decrease	1)It decreases efforts taken to create signatures	1)Effort

Educate	1) Educate project members with security concepts, machine learning, MySQL, Wireshark 2)Educate user	1)Project members 2)User
Evaluate	1)Continuously monitor network traffic for intrusions 2)Evaluate packets to determine if there is an intrusion	1)Network Traffic 2)Packets
Experiment	1)Experiment on different types of attacks 2)Experiment on different types of ML models	1)Attacks 2)ML Models
Accelerate	1)It increases the processing speed by using a separate device instead of the IoT device itself	1)Processing speed
Analysis	1)It analyses detected anomalies to determine what type of attack it is	1)Category of attack
Advertise	1)Advertise about IoT security and need of an IDS 2)Create a GitHub page, which will include code, documentation and process	1)Security awareness 2)GitHub

3.2 Mathematical Model

Let S be the solution set for the given problem statement.
S={Input,Function,Output, Terminate,Success,Failure}.
Where,Input =Input to the System.
Function =Functions of the system.
Output =Output of the system.
Terminate= Terminating Condition of the System.
Success =Success cases for the System.
Failure =Failure cases for the system.

1. Input ={UserName,Password,Data Packets}

a. UserName :-use_rid

b. Password : user_passsword

c. Data Packets:- A data packet is a unit of data made into a single package that travels along a given network path.

2. Function={Login_auth,Network_connection,train_test,intrusion_detection,Notification}
a. Login_auth: This function will take Username and Password as the input and gives the authorization rights to the user accordingly.

IF Y=F(X) is the login_auth function then

X:- Username and Password taken as an input.

Y:- Authorization rights of a user.

b. Network_connection =This function will take port number, ip address as an input to give the status of the connection network.

IF Y=F(X) is the Network_Connection Function, then

X:- Inputs to setup a network, connecting to a device ,routing the incoming data and device configuration.

Y:- Status of the network connection.

c. train_test: This function will take features as an input to give the accuracy of the model and further it will be turned to improve the accuracy of the model. Random Forest has been used as a Machine learning algorithm for building the model.

IF Y=F(X) is the function to test and train the model then,

X:- Features of the DS2OS dataset as an input.

Y:- Accuracy of the model

d. intrusion_detection : This function will take features like sourceID, sourceAddress, sourceType, sourceLocation, destinationServiceAddress, destinationServiceType, destinationLocation, accessedNodeAddress, accessedNodeType, operation, value, timestamp as an input and notifies the user the system for the intrusion.

- Random Forest Classifier has been used for the building and implementing the model.

Basically, a random forest is an average of tree estimators. As with nonparametric regression, simple and interpretable classifiers can be derived by partitioning the range of X. Let n = A1, . . . , AN be a partition of X. Let Aj be the partition element that contains x. Then h(x) = 1 if XiAj Yi XiAj (1 Yi) and h(x) = 0 otherwise.

We conclude that the corresponding classification risk satisfies R(h) R(h) = O(n1/(d+2)).

Trees are useful for their simplicity and interpretability. But the prediction error can be reduced by combining many trees.

These are bagged trees except that we also choose random subsets of features for each tree. The estimator can be written as

$$\widehat{m}(x) = \frac{1}{M} \sum_{j} \widehat{m}_j(x)$$

where mj is a tree estimator based on a subsample (or bootstrap) of size a using p randomly selected features. The trees are usually required to have some number k of observations in the leaves. There are three tuning parameters: a, p and k. You could also think of M as a tuning parameter but generally we can think of M as tending to . For each tree, we can estimate the prediction error on the un-used data. (The tree is built on a subsample.) Averaging these prediction errors gives an estimate called the out-of-bag error estimate.

IF Y=F(X) is the function then,

X: sourceID, sourceAddress, sourceType, sourceLocation, destinationServiceAddress, destinationServiceType, destinationLocation, accessedNodeAddress, accessedNodeType, operation, value, timestamp as an input

Y: Notifies the system.

e. Notification: This function takes the input which is provided by the function (intrusion_detection) and notifies the user.

IF Y=F(X) is the function then,

X: Output of the function(intrusion_detection).
Y: Intrusion message to the user.

3. Output = {display_intrusionmsg }

a. display_intrusionmsg : display error message if any intrusion occurs.

4. Intermediate Results

   a. Successful working of module.

   b. Successful Working of Network.

   c. Successful User authentication.

   5. Terminate= {Invalid_details, Network_failure, Timeout

   a. Invalid User Authentication.

   b. Network failure

   c. timeout

6. Success

   a. Successful user login.

   b. Successful connection establishment of nodes and ids.

   c. Successful detection of intrusion.

   d. Displaying the results.

   e. Appropriate error messages in case of invalid input.

7. Failure

   a. Web app Failure.

b. Hardware faults.

c. Network establishment failure.

d. Not displaying required results.

3.3 Feasibility Analysis (NP Completeness Analysis)

The problem is to detect an intrusion in a Iot network. The main objective of our project is to provide a real-time solution for any possible intrusions.

In the proposed system, we are planning to use anomaly-based detection system. The anomaly-based intrusion detection system comes into effect when detecting newer attacks, that are not filtered by the firewall. Random Forest is an ensemble model of decision trees.

Training Complexity: O((n$\hat{2}$)pntrees)
Prediction Complexity: O(pntrees)

Calling n the number of training sample, p the number of features, ntrees the number of trees (for methods based on various trees),
where,
n= 21000 (Approx)
p = 17
ntress= 4

Hence O(pntrees) runs in polynomial time complexity.
The algorithm falls in P. Hence, the given problem is NP.

3.4 Use-Case Diagrams

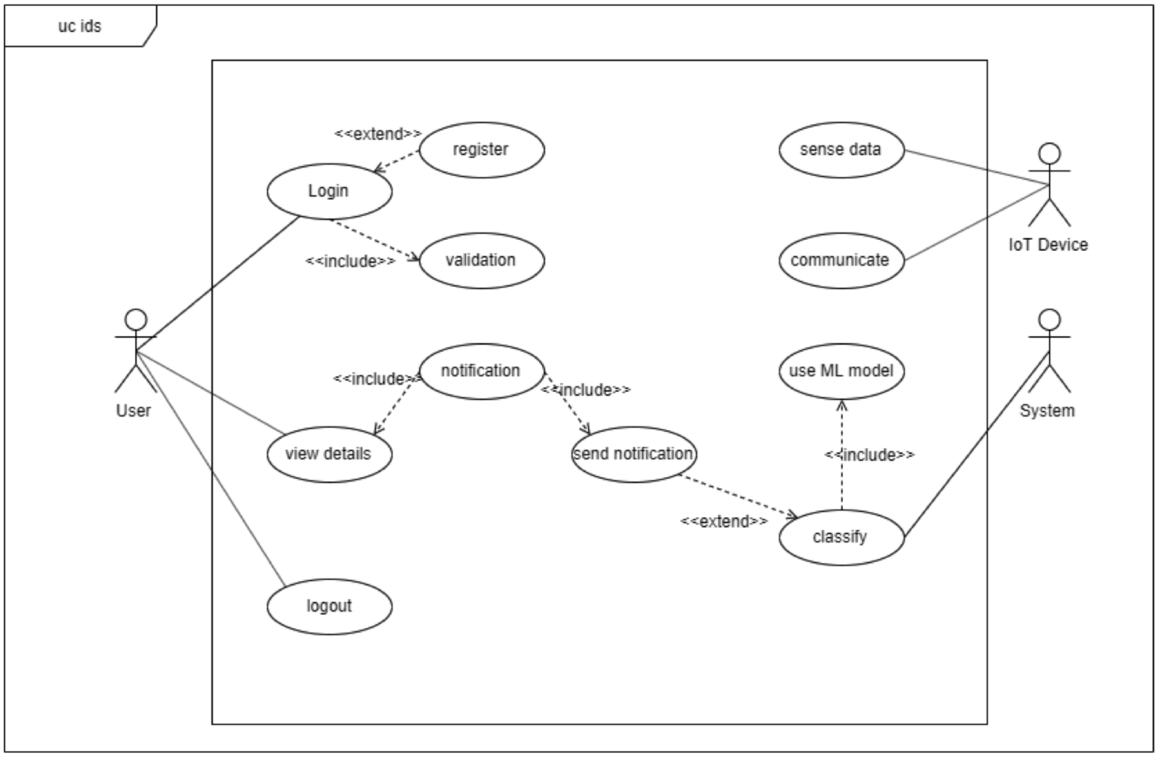

Figure 3.1: Use-case Diagram

3.5 Activity Diagram

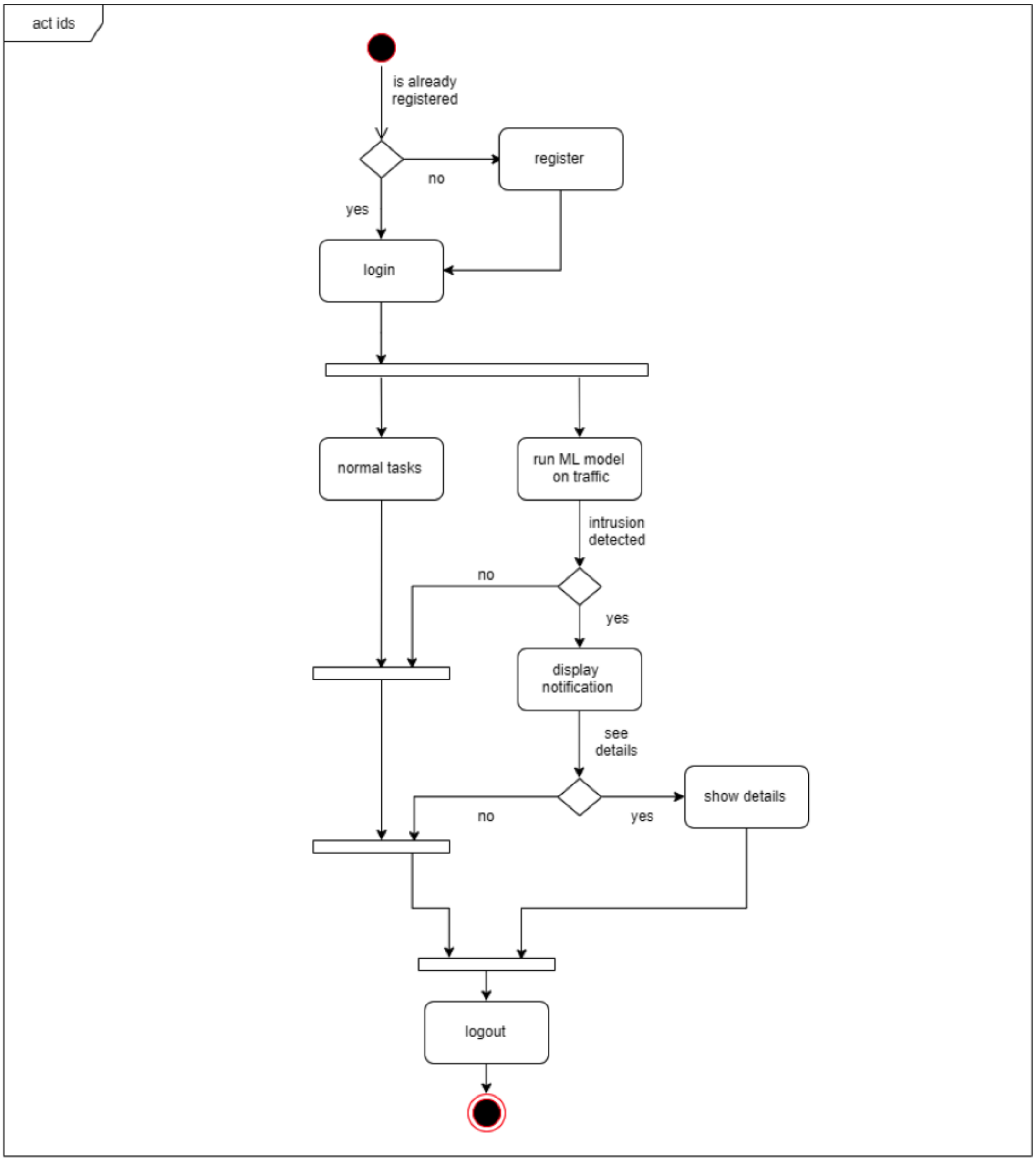

Figure 3.2: Activity Diagram

3.6 Architecture Diagram

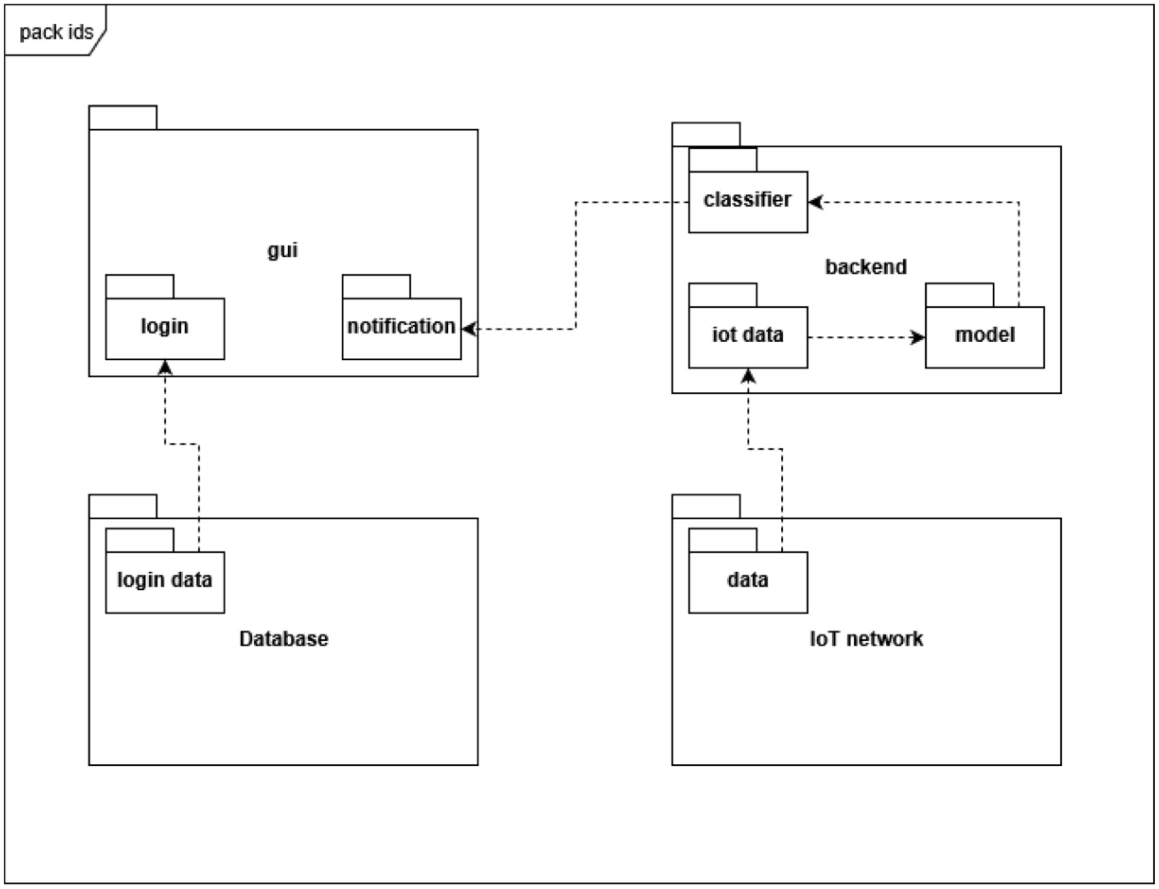

Figure 3.3: Package Diagram

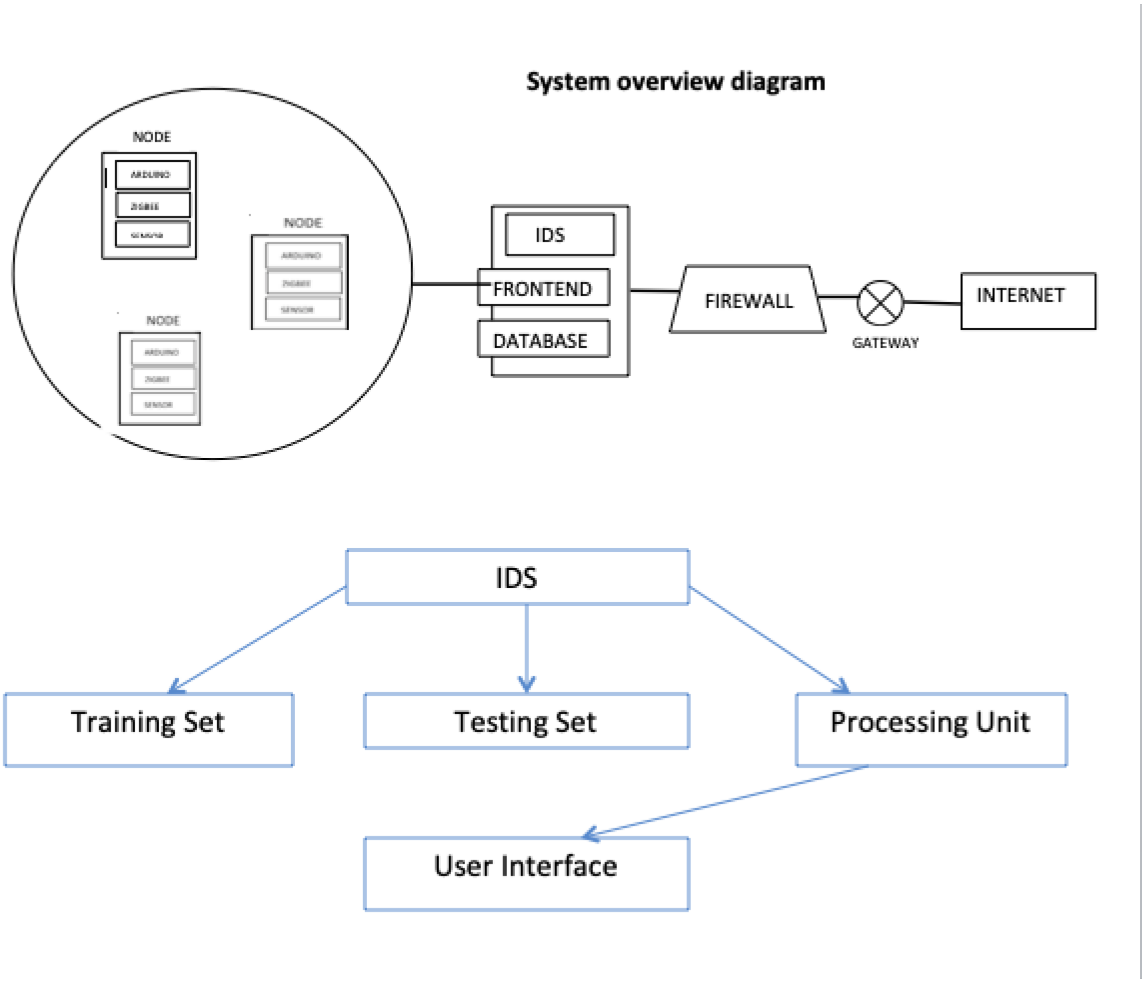

Figure 3.4: System Overview Diagram

3.7 Class Diagrams

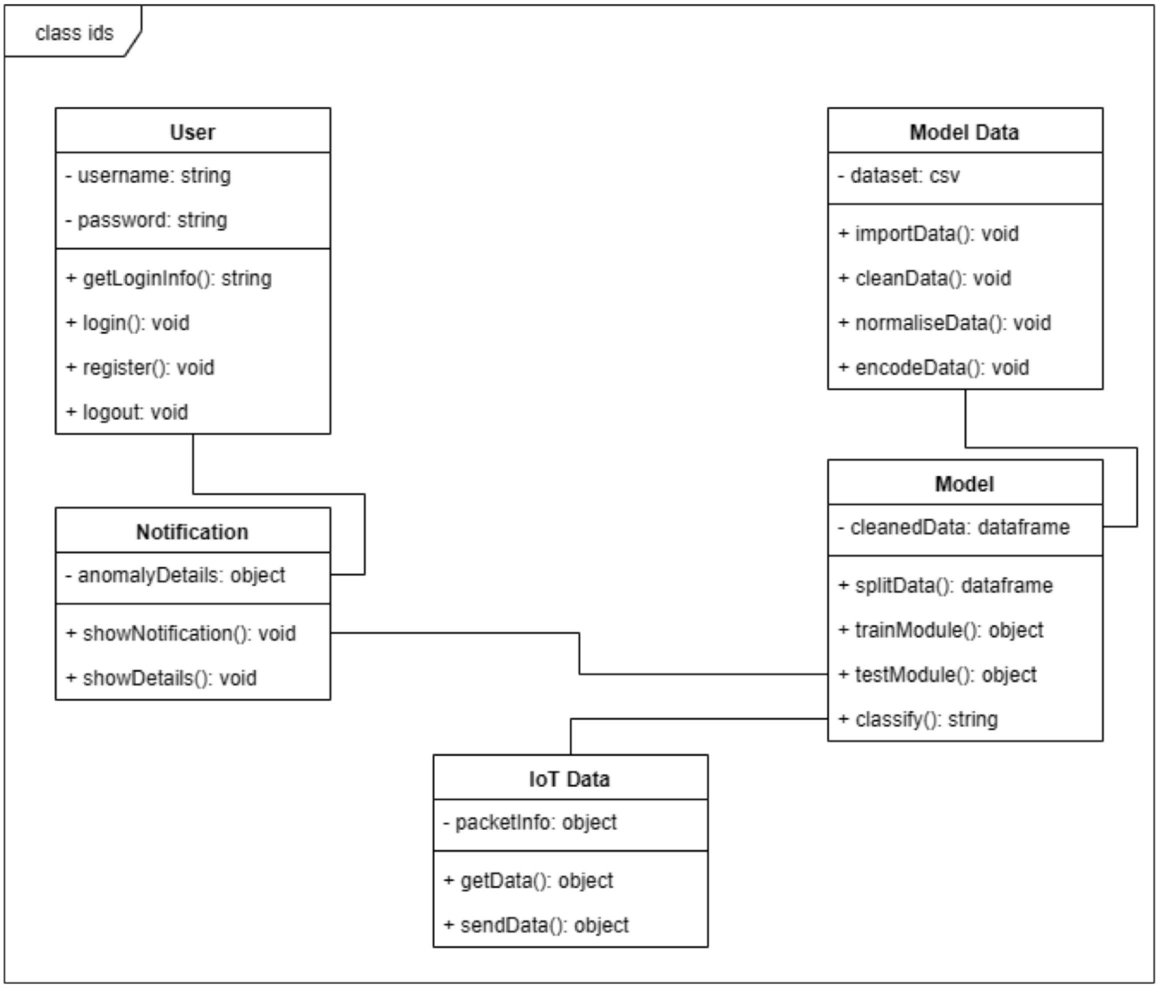

Figure 3.5: Class Diagram

3.8 Sequence Diagrams

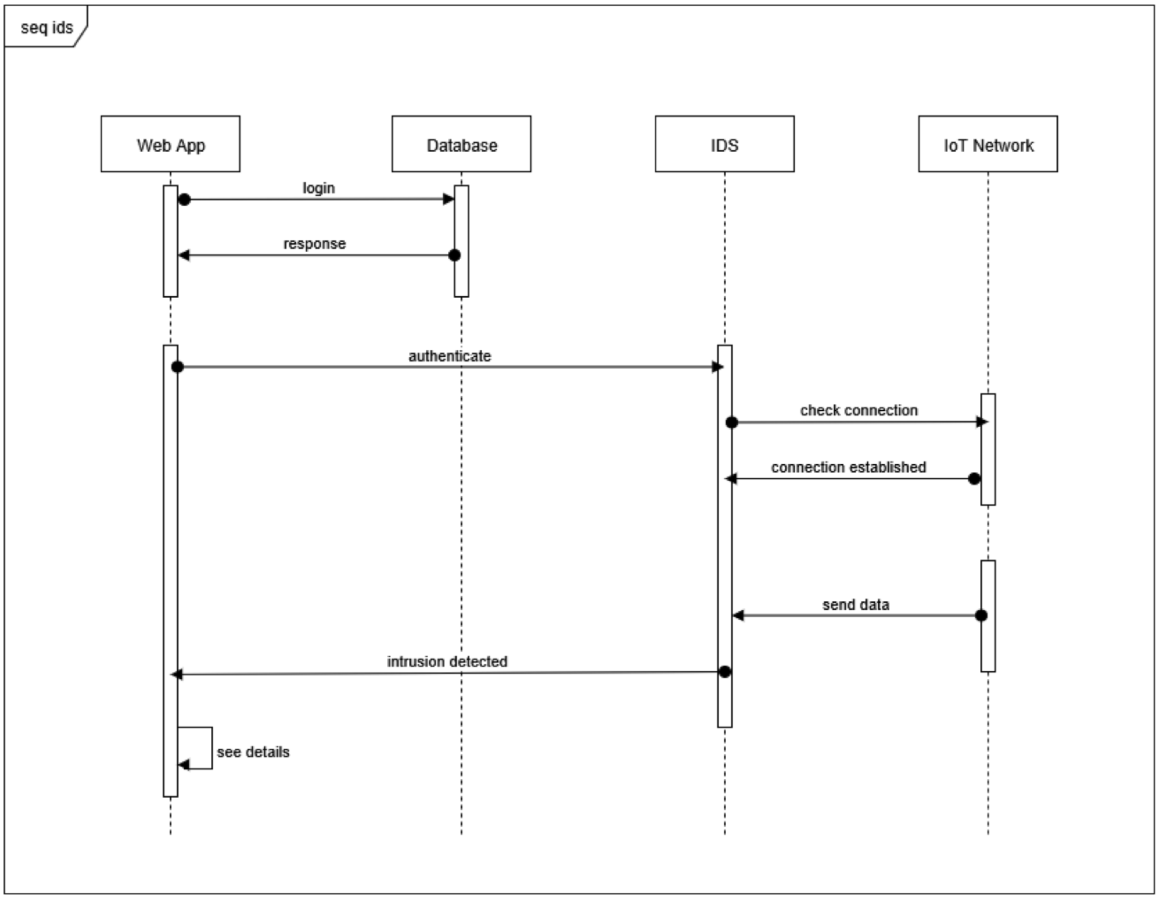

Figure 3.6: Sequence Diagram

3.9 State Transition Diagrams

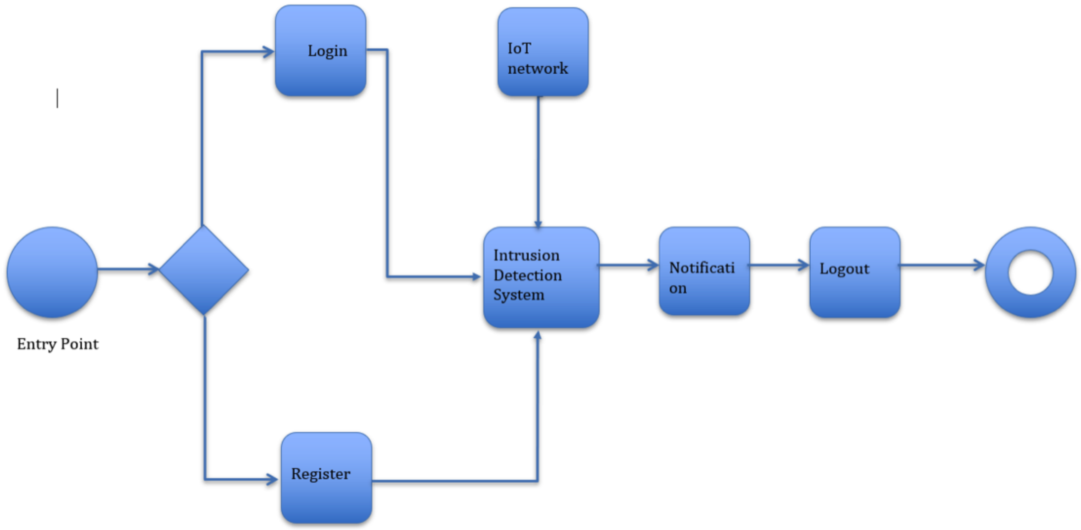

Figure 3.7: State Transition Diagram

3.10 Deployment Diagrams

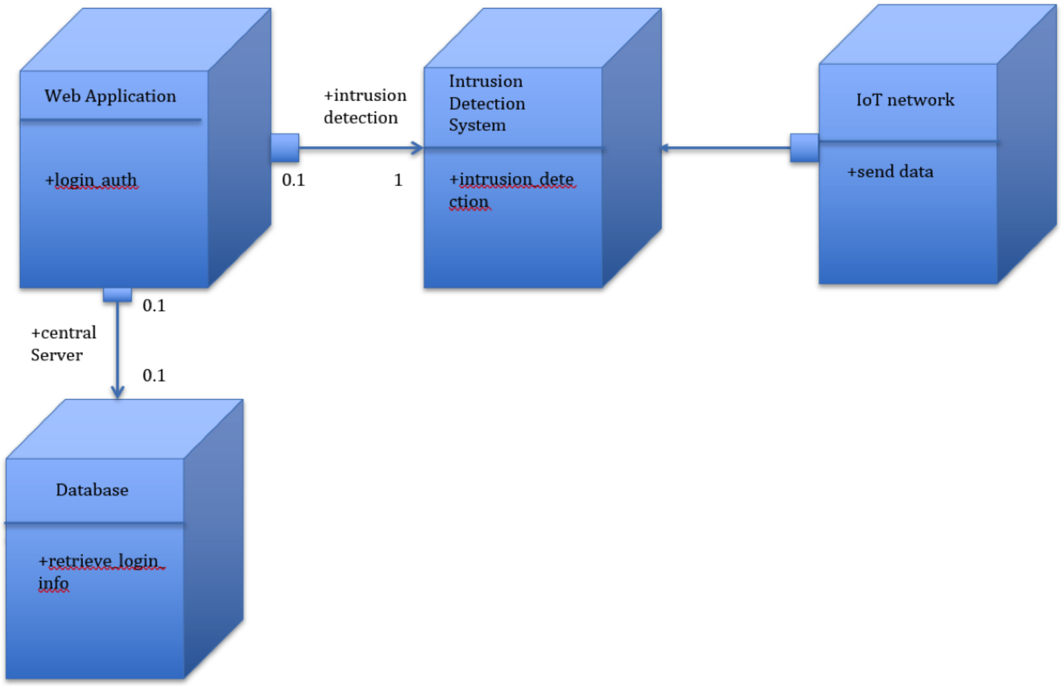

Figure 3.8: Deployment Diagram

Chapter 4

TESTING

4.1 Unit Testing

Table 4.1: Unit Testing 1

Title	Login and Authentication check
Test Item	Username, Password,Data packets
Input Specification	Test Item containing alphanumeric characters
Description	Specified details of Login/Register are entered and stored in database
Expected Results	Entry for Test Item is created in Database.

Table 4.2: Unit Testing 2

Title	Network Connection Check
Test Item	IOT devices, zigbee adapter
Input Specification	Test Item containing alphanumeric characters
Description	Zigbee protocol is used for connection,communication and sending the sensed data to base station
Expected Results	Data sensed is sent to the server.

Table 4.3: Unit Testing 3

Title	Intrusion Detection System processing check
Test Item	Sensed based data of specified sensor
Input Specification	Test item containing various ongoing intrusions on IOT network
Description	Intrusion detection data is used to detect malicious attacks on the IOT network and ignore normal behavior of the system
Expected Results	Category of the attack is determined

Table 4.4: Unit Testing 4

Title	Notification Check
Test Item	Notification page of web application(GUI)
Input Specification	Test Item containing processed data
Description	Alert only for abnormal behavior of the system
Expected Results	Alert to the user on the notification page if malicious attacks are detected

4.2 Integration Testing

Table 4.5: Integration Testing 1

Title	Intrusion Detection System
Description	Web Application is used to provide an interface between the network connection and the frontend
Test Steps	Run the program for detection of any intrusion on the IOT network
Expected Results	Detect the category of an attack

Table 4.6: Integration Testing 2

Title	Web Application
Description	Web Application is used to provide an interface between the frontend and the user
Test Steps	Run the application program
Expected Results	Notification on the web page of if any abnormal behavior of the attack is detected

4.3 Acceptance Testing

Table 4.7: Acceptance Testing 1

Title	Sensed Availability data
Description	Ultrasonic sensor and temperature sensor are used to sense data and is processed to gather sensed availability data in the database
Expected Output	Sensed Availability data is sent to the base station using the zigbee protocol

Table 4.8: Acceptance Testing 2

Title	Performance
Description	The time required to transfer data from IOT devices to base station should be as little as possible and the time between the occurrence and detection of the attack should be minimum
Expected Output	Real time functioning of the system is achieved

References

- Journal Article/ IEEE Paper

[1] Abebe A. Diro., Naveen C., ”Distributed attack detection scheme using deep learning approach for Internet of Things”, Future Generation Computer Systems, 82, 2017, 761-768.

[2] Ashima C., Brian L., Sheila F., Paul J., ”Host based Intrusion Detection System with Combined CNN/RNN Model”, IWAISe 2018- ECML PKDD Conference, 2018.

[3] Ebelechukwu N., Andre C., Gedare B., ”Anomaly-based Intrusion Detection of IoT Device Sensor Data using Provenance Graphs”, 1st International Workshop on Security and Privacy for the Internet-of-Things (IoTSec), 2018.

[4] Shadi A., Monther A., Muneer B. Y., ”Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model”, Journal of Computational Science, 25, 2018, 152-160.

[5] TagyAldeen M., Takanobu O., Takayuki I., ”Towards Machine Learning Based IoT Intrusion Detection Service”, Recent Trends and Future technology in Applied Intellignece, 2018, 580-585.