

ENHANCING SECURITY IN IOT HOME AUTOMATION SYSTEMS

MSc Research Project
MSc Cybersecurity

Sai Veeranjaneya Boppana
Student ID: x23200391

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland
MSc Project Submission Sheet



School of Computing

Student Name: Sai Veeranjaneya Boppana

x23200391

Student ID:

Programme: MSc Cybersecurity

Year: 01/2024

Module: MSc Research Project

Supervisor: Niall Heffernan

Submission Due

Date: 29/01/2025

Project Title: ENHANCING SECURITY IN IOT HOME AUTOMATION SYSTEMS

Word Count: 6311

Page Count 28

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: Sai Veeranjaneya Boppana

Date: 29/01/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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):		

ENHANCING SECURITY IN IOT HOME AUTOMATION SYSTEMS

Sai Veeranjanya Boppana

x23200391

Abstract

The integration of IoT technologies into home automation has transformed modern living, offering greater convenience, efficiency, and centralized control over household systems. However, the interconnected nature of these devices introduces significant security vulnerabilities, exposing users to risks such as data breaches, unauthorized access, and system disruptions. These challenges are compounded by the resource limitations of IoT devices, which often struggle to support advanced security mechanisms. This research focuses on identifying critical security threats in IoT-enabled smart homes and proposes a comprehensive framework to address these vulnerabilities. Guided by Zero Trust Architecture principles, the study incorporates advanced encryption, multi-factor authentication protocols, and real-time anomaly detection models to protect devices against evolving cyber threats. A user-centric approach is also emphasized, enhancing homeowner awareness and encouraging secure practices to mitigate risks effectively. The methodology combines theoretical analysis with empirical testing in a simulated smart home environment, allowing for the evaluation of security solutions under real-world conditions. Results demonstrate that adaptive security measures can effectively address the evolving threat landscape without compromising device functionality. By bridging the gap between innovation and security, this research offers practical insights into enhancing the resilience of IoT-enabled home automation systems. The findings contribute to developing smarter, safer homes, ensuring that advanced technologies can be adopted confidently and securely.

TABLE OF CONTENTS

1. Introduction.....	4
2. Literature Review.....	4
3. Research Method and Specification.....	6
4. Design Specification.....	8
4.1 System Architechure.....	8
4.2 Data Processing and Machine Learning.....	9
4.3 Interactive Dashboard.....	10
4.4 Functional Framework.....	10
5. Implementation.....	11
5.1 Development of IoT Simulation.....	11
5.2 Cloud Infrastructure Setup.....	12
5.3 Predictive Model Development.....	14
5.4 Streamlit Dashboards.....	16
5.5 Integration and Testing.....	18
5.6 Outputs Produced.....	19
5.7 Tools and Technologies Used.....	19
6. Evaluation.....	20
6.1 Case Study 1: Anomaly Detection Accuracy.....	20
6.2 Case Study 2: Real-Time Processing Performance.....	20
6.3 Case Study 3: User Interface Usability.....	21
6.4 Case Study 4: Security Assessment of TLS Encryption.....	22
6.5 Case Study 5: Access Control Validation.....	22
6.6 Discussion.....	23
7. Conclusion and Future Work.....	23
7.1 Summary of Work and Key Findings.....	24
7.2 Implications of Research.....	25
7.3 Efficacy and Limitations.....	25
7.4 Future Work.....	25
7.5 Conclusion.....	26
REFERENCES.....	27

1. Introduction

The proliferation of Internet of Things (IoT) devices in home automation has revolutionized the way people interact with their living spaces. Smart homes provide unparalleled convenience and efficiency, streamlining everyday operations such as lighting, temperature control, and security management. However, this technological evolution introduces a critical challenge: heightened cybersecurity risks due to the expansive attack surface created by interconnected devices. These devices often possess limited processing power and memory, making the integration of advanced security mechanisms a complex task.

IoT ecosystems are distinct from traditional computing environments, as they involve diverse communication protocols, resource-constrained devices, and sensitive personal data. The lack of robust security measures exposes them to vulnerabilities, which can lead to privacy breaches, unauthorized access, and operational disruptions. To address these challenges, this research focuses on identifying critical security vulnerabilities in IoT-enabled home automation systems. It explores advanced encryption, authentication techniques, and user education to create a more resilient security framework. The study aims to adapt principles of Zero Trust Architecture and develop adaptive security solutions tailored to the resource constraints of IoT devices, ultimately enhancing the reliability and security of smart homes.

Research Question:

- How can principles of Zero Trust Architecture be adapted to improve IoT device security?
- How can adaptive security measures be developed and implemented to address the evolving threat landscape in IoT home automation systems while considering the resource constraints of diverse devices?
- How can new encryption and access control techniques be applied to IoT systems?

2. Literature Review

IoT's Impact on Smart Home Automation

According to Abdulraheem *et al* 2020: Another aspect that has significantly advanced smart homes and buildings automation is the Internet of Things coupled with cloud computing and rule based event processing. This progress has eliminated the limitation of prior command systems that relies on radio spectrum interference and the physical electrical wiring. Smart homes of the present day provide for the distant control of a vast array of gadgets and devices, increasing

security, comfort, and convenience for residents, through the application of IoT. This paper likewise concentrates on the particular areas where IoT can be applied, particularly home automation and its influence on essential systems. These include fire prevention systems, surveillance cameras, Smart TVs, lighting systems, smart thermostats, air conditioners/ventilators, doors, fans, humidity and gases control systems. Before the integration of IoT, these appliances were commonly operated randomly in the automated areas. However, the functioning of these entities has advanced due to the proliferation of IoT technology which supports their Interactions through the internet, thus resulting in more reliability in the Automation process. The role of IoT does not stop at enhancing productivity as it also combats the consumptions of energy issues. Smart home automation directly results in energy conservation for example, by preventing energy wastage and exercising better command over house equipment. This remains very important given that energy conservation is now a very important factor. Also, IoT systems have developed a graphic interface that allows for easy control of home appliances, with high end efficiency in the offered communication with reasonable data rates and reasonable connection range. Accordingly, the research carried out in this study reveals that IoT has revolutionized the techniques of smart home automation by integrating various technologies and enhancing the performance of home and building systems. This progress guarantees that smart buildings achieve higher levels of safety, comfort, and efficiency as part of modernist desire for smarter and more responsive living spaces.

AI-Enhanced Smart Home Security

According to Taiwo *et al* 2022: The increase in crime rate in which many homes have no security other than simple locks on entrance doors clearly explains the need for better security. Homes, especially in South Africa, have remained insecure even with the lockdown measures in place hence the need to enhance security at home. The study proves that upgrades in smart home automation systems backed by IoT as well as AI present practical solutions to such security problems. Home automation using IoT centralizes home electronic appliances and security devices through the use of internet, Wi-Fi, or Bluetooth connectivity. They give an ability to control, to monitor and to be informed about a potential danger thus giving the homeowner a sense of security. Machine learning, and in particular deep learning, further develops these systems since it allows for characteristics such as object identification, face recognition or motion identification. The review of literature shows that (Taiwo *et al.* 2020) put forward a real-

time, cloud-based, low-cost smart home automation system with an android mobile application. This system uses the CNN model for identifying human activities by detecting a contrast between normal activity and suspicious one, assuming the later to be an intruder. The detection of motion patterns is then compared with data that has been previously captured with the aim of eradicating fake alarms and enhancing security's efficiency. The proposed system addresses the major challenges that are related to automation of smart homes: online data processing, accurate motion sensing, and smart decision making. It is cost and time effective way to enhance the security and AI and IoT can make out homes safer and more comfortable to live in.

Research Niche

Although the recent studies mostly focused on the disruptive role of IoT and AI in the context of smart homes, automated security, there are critical weaknesses in the understanding of a single interdisciplinary approach that when applied with user-centric design and resource optimization across the architecture of smart homes. Similarly, (Abdulraheem *et al.* 2020) depict how IoT has evolved smart home system by enhancing productivity while promoting energy saving. However, their focus primarily rests on the operational scope, which leaves room for more comprehensive analysis of security concerns. According to Taiwo *et al.*, 2022, the investigations into home-based crimes have led them develop an AI based security system for smart homes. However, their technique employing CNN models for human activity recognition is revealed to be innovative reflecting merely one aspect of the diverse security system in smart homes. This research aims to address these gaps by offering an extensive security model that incorporates advanced technologies like AI and IoT, and also addresses the different concerns associated with home settings.

3. Research Method and Specification

Research Method

The technique for this review is intended to consolidate logical examination with observational testing to address basic security challenges in IoT-empowered home mechanization frameworks (Majeed *et al.*, 2020). This approach expects to recognize weaknesses, foster arrangements, and assess their viability.

The exploration will start with a broad survey of scholarly writing, gathering papers, and industry reports to lay out a hearty groundwork and recognize existing information holes. This

stage will use respectable sources like IEEE Xplore, ACM Digital Library, and Google Scholar to guarantee complete inclusion of the exploration region.

Following the writing survey, an overall weakness evaluation of IoT gadgets regularly utilized in brilliant homes will be led. This will include investigating existing digital dangers, distinguishing potential assault vectors, and sorting gambles with in view of seriousness and simplicity of abuse. In light of the discoveries, trial arrangements will be planned, including encryption and verification components customized for IoT gadgets and an AI model for constant oddity recognition (Yar et al., 2021).

To test these arrangements, a controlled climate mimicking an IoT-empowered home will be built. This arrangement will work with execution assessment, effectiveness testing, and examination of the irregularity location framework's capacities in distinguishing dangers.

Research Resources

To effectively execute this exploration, different assets will be required. **Hardware resources** will incorporate IoT gadgets like savvy indoor regulators, surveillance cameras, brilliant locks, and voice collaborators, alongside systems administration gear and cloud framework for information handling and stockpiling (Shah & Mahmood, 2020).

Software resources will envelop incorporated improvement conditions (IDEs) for programming creation, network safety apparatuses for weakness testing, AI structures for irregularity recognition, and information investigation stages for handling and envisioning results (Abdulla et al., 2020). The examination will likewise depend on HR, remembering specialists for IoT, network safety, and AI to configuration, execute, and approve the arrangements (Taiwo & Ezugwu, 2021).

Evaluation

The assessment stage will survey the productivity and significance of the proposed safety efforts (Allifah and Zuolkernan, 2022). This will incorporate infiltration testing to confirm the vigor of the framework against cyberattacks, occasional weakness appraisals, and reproductions of complex assault situations to screen the adequacy of the created arrangements (Ratkovic, 2022). The oddity discovery framework will be assessed in light of measurements like accuracy, review, and precision in distinguishing strange IoT gadget conduct, with consideration regarding bogus positive and misleading negative rates (Qasim et al., 2020). Execution testing will investigate the computational effect of the proposed measures on gadget usefulness, including dormancy and asset utilization. Consistence with worldwide information security guidelines, for example, GDPR and CCPA, will likewise be confirmed. Relative investigation will feature the upsides of the created arrangements over existing safety efforts.

Ethical Considerations

The exploration will rigorously comply with moral rules to safeguard the freedoms of partners and guarantee the respectability of the review. Protection and information assurance estimates

will be carried out to get every gathered datum, in consistence with GDPR and CCPA norms (Illy et al., 2022).

Recognized weaknesses in business IoT gadgets will be dealt with following a capable revelation strategy, permitting makers satisfactory chance to resolve the issues before open exposure (Beam and Bagwari, 2020). All entrance testing and security appraisals will be directed in controlled conditions to forestall accidental effects on certifiable frameworks (Murad et al., 2021).

Project Plan

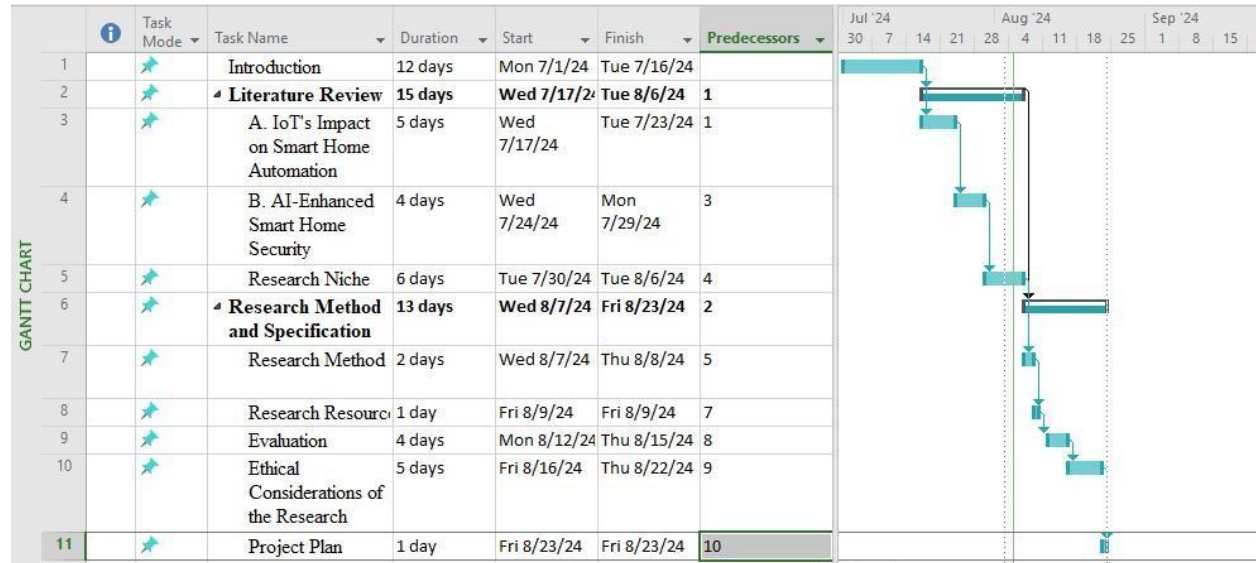


Figure 1: Gantt chart

(Source: Self-created using Project)

4. Design Specification

This part gives an itemized portrayal of the plan and execution of an IoT-based home robotization security framework. The structure is based on a mix of recreated IoT gadgets, cloud-facilitated administrations, prescient AI models, and a Streamlit dashboard for continuous observing and peculiarity recognition.

4.1 System Architecture

The proposed framework incorporates a cloud-hosted **Node-RED simulation**, AWS-based infrastructure, and a **Streamlit dashboard** to convey a consistent checking and expectation stage for IoT-empowered brilliant home conditions. The accompanying parts frame the critical parts of the engineering:

Node-RED Simulation:

A simulation hosted on an AWS EC2 instance generates live data for four IoT devices: thermostat, motion sensor, door lock, and camera. The recreation refreshes gadget states and situations with 5 seconds, mimicking different functional and strange circumstances like unapproved access, temperature abnormalities, and movement location.

Database and API Layer:

Information produced by the Node-RED simulation is put away in MongoDB through a cloud association. A serverless API, executed utilizing AWS Lambda, gives ongoing admittance to this information. The Programming interface upholds questioning gadget explicit information and bringing live updates for reconciliation with outer apparatuses. The endpoint <https://45yotl7l9k.execute-api.eu-north-1.amazonaws.com/prod/home> serves as the primary interface, with optional query parameters to filter results by device type.

- **Streamlit Dashboard:**

The Streamlit application gives an intuitive point of interaction to observing gadget states, envisioning irregularities, and investigating ready examples. This dashboard powerfully gets information from the Programming interface, processes it through an AI model, and shows the outcomes continuously.

- **Predictive Model:**

An AI model, in view of the Disconnection Woodland calculation, is utilized to foresee peculiarities. The model is prepared on verifiable information to recognize deviations from typical way of behaving, creating cautions in light of identified designs.

4.2 Data Processing and Machine Learning

Data Ingestion and Preprocessing:

The framework utilizes verifiable IoT information put away in a CSV document for preparing and approval. Each record contains the fields: gadget type, status, timestamp, and alert. The timestamp is handled to separate worldly elements like the hour of the day, and absolute factors like gadget type and status are encoded mathematically utilizing name encoders. The subsequent highlights are scaled to guarantee similarity with the Disconnection Timberland model, which is delicate to information scaling.

Model Training and Deployment:

The irregularity recognition model is prepared utilizing the Confinement Woods calculation, which works on an unaided learning standard. It recognizes exceptions (peculiarities) in the dataset in view of learned designs. Subsequent to preparing, the model and preprocessing devices, including the name encoders and scaler, are serialized and put something aside for incorporation with the Streamlit application.

Real-Time Prediction:

Live information brought from the Programming interface goes through preprocessing like the preparation stage. Encoded and scaled highlights are taken care of into the prepared model, which arranges each record as one or the other typical or irregular. Custom logic further characterizes irregularities in view of gadget explicit circumstances, upgrading the interpretability of the model's result.

Secure Communication:

To guarantee the secrecy and uprightness of information traded between framework parts, TLS (Transport Layer Security) encryption is utilized. The Programming interface endpoints are gotten with HTTPS, guaranteeing that all information moved between the Streamlit dashboard and the AWS Lambda Programming interface is scrambled on the way. This forestalls snooping, altering, and man-in-the-middle assaults.

Access Control:

To guarantee secure and confined admittance to delicate parts, Role Based Access Control (RBAC) has been executed. This permits just approved clients or frameworks to get to explicit assets in light of predefined jobs. For example, managerial jobs can design framework settings, while checking jobs are restricted to review continuous information.

4.3 Interactive Dashboard

The Streamlit dashboard offers two essential perspectives:

1. Live Monitoring:

The live dashboard constantly brings information from the Programming interface and cycles it to show the ongoing status of observed gadgets. Inconsistency cautions are featured and envisioned utilizing bar graphs, with refreshes happening at regular intervals. Clients can choose gadgets to screen or conceal explicit gadgets from the view for customized control.

2. Anomaly Logs:

Recognized irregularities are put away in a tireless meeting state and showed in the logs segment. Clients can audit authentic irregularities, channel records in view of gadget types, and break down designs over the long run.

4.4 Functional Framework

The essential capability of the framework is to give continuous security checking and oddity discovery. Every part of the framework satisfies an unmistakable job inside this structure:

- **Node-RED and MongoDB:** Simulate and store device data.
- **AWS Lambda API:** Facilitate secure and efficient data retrieval.
- **Machine Learning Model:** Enhance anomaly detection accuracy.
- **Streamlit Dashboard:** Serve as the user-facing interface for monitoring and control.

By consolidating these components, the framework guarantees a powerful, versatile, and easy to understand answer for getting IoT-empowered savvy homes.

5. Implementation

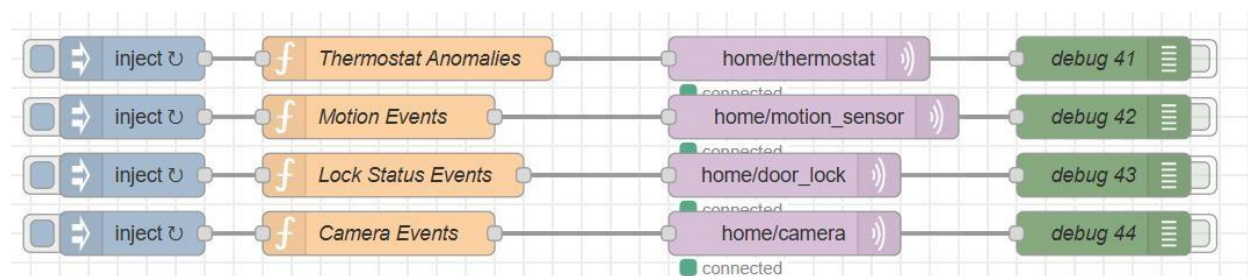
The execution period of this undertaking zeroed in on creating and coordinating a powerful framework for continuous observing, peculiarity location, and ready age in an IoT-empowered savvy home climate. The framework consolidates reenacted IoT gadget information, cloud foundation, prescient displaying, and a powerful perception dashboard to improve home robotization security. This part frames the last phase of the execution, underscoring the results created, devices utilized, and the framework's usefulness.

5.1 Development of IoT Simulation

The execution started with the production of a complete IoT recreation utilizing Node RED, facilitated on an AWS EC2 case. Node RED was designed to reproduce the action of four critical gadgets — indoor regulator, movement sensor, entryway lock, and camera. These gadgets were modified to produce information at ordinary five-second spans, reflecting both typical and peculiar states. This recreation gave a controlled and solid information hotspot for ensuing phases of execution.

The information produced included key credits, for example, the gadget type, status, timestamp, and an underlying alarm field for early cycles of the undertaking. This dataset was instrumental in preparing the AI model and testing the framework's start to finish usefulness.

Node-RED Simulation:



5.2 Cloud Infrastructure Setup

MongoDB Database:

A MongoDB example filled in as the essential data set for putting away the reenacted IoT information. This arrangement guaranteed versatility and productive information recovery, basic for taking care of the constant five-second updates from Node RED.

AWS Lambda API:

To work with constant information access, a API was executed utilizing AWS Lambda. The API was intended to recover information from MongoDB and convey it in JSON design. The Lambda capability upholds questioning for all gadgets or explicit gadgets in light of client characterized boundaries.

The API was thoroughly tried to guarantee negligible idleness and solid combination with the Streamlit dashboard. The serverless design of AWS Lambda decreased functional above while keeping up with high accessibility.

Lambda Function Snippet:

```
# Initialize DynamoDB resource and table
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('HomeUsageTable')

def lambda_handler(event, context):
    try:
        # CORS headers
        cors_headers = {
            "Access-Control-Allow-Origin": "*",
            "Access-Control-Allow-Methods": "GET",
            "Access-Control-Allow-Headers": "Content-Type"
        }
        # Extract query parameters
        query_params = event.get('queryStringParameters', {})
        device = query_params.get('device') if query_params else None
        # Handle case where no device is provided (return all devices)
        if not device:
            # Perform a scan to get all items in the table
            response = table.scan()
            return {
                'statusCode': 200,
                'headers': cors_headers,
```

```

        'body': json.dumps(response['Items'], cls=DecimalEncoder)
    }
    # If device_id is provided, query for specific device
    response = table.query(
        KeyConditionExpression=Key('device').eq(device)
    )
    # Return the queried items with Decimal values converted
    return {
        'statusCode': 200,
        'headers': cors_headers,
        'body': json.dumps(response['Items'], cls=DecimalEncoder)
    }
except Exception as e:
    # Log and return the error if something goes wrong
    print(f"Error: {str(e)}")
    return {
        'statusCode': 500,
        'headers': cors_headers,
        'body': json.dumps({'error': 'Internal server error', 'message':
str(e)})
    }

```

Explanation of Lambda Function Behavior:

- **Initialization:** The boto3 library is utilized to instate an association with the DynamoDB administration. The table HomeUsageTable is gotten to for questioning gadget information.
- **Decimal Handling:** The custom DecimalEncoder class is utilized to serialize Decimal qualities (normal in DynamoDB) to standard JSON-viable float values.
- **Query Parameters:** The Lambda capability processes queryStringParameters from the occasion object to permit clients to question explicit gadgets. Assuming that no gadget is determined, it recovers all information from the table utilizing filter().
- **CORS Support:** The capability incorporates CORS headers to help cross-beginning asset sharing, permitting the Streamlit dashboard (or some other client) to make solicitations to the API.
- **Error Handling:** In the event that a mistake happens during execution (e.g., a data set question comes up short), the capability returns a 500 status code and incorporates a blunder message in the reaction.

API Payload Sample:

```

{
  "payload": {
    "device": "thermostat",
    "status": "anomaly",

```

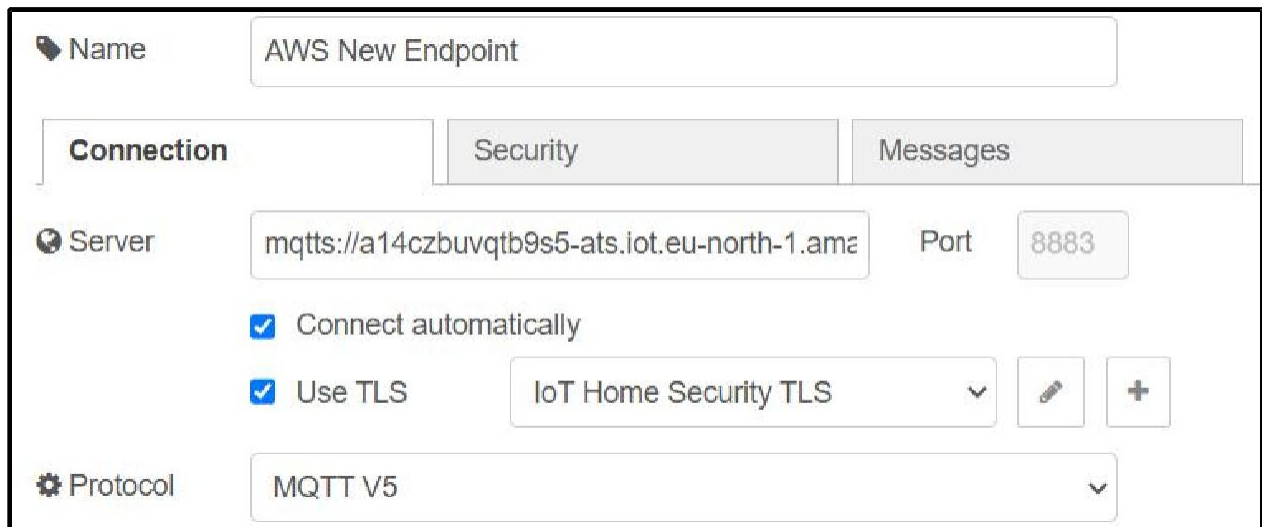


```
    "timestamp": "2024-12-01T12:59:12.138Z"  
  },  
  "device": "thermostat"  
}
```

TLS Encryption:

The API facilitated on AWS Lambda is gotten utilizing TLS encryption. All information trades between the client (Streamlit dashboard) and the Programming interface happen over HTTPS, guaranteeing vigorous security against capture attempt or altering. This was accomplished by designing the Programming interface Entryway to uphold HTTPS associations. During testing, apparatuses like Mailman and program based HTTPS approval were utilized to affirm that the encryption was accurately carried out.

TLS Encryption View:



The screenshot displays a configuration window for an MQTT client. At the top, the 'Name' field is set to 'AWS New Endpoint'. Below this, there are three tabs: 'Connection' (selected), 'Security', and 'Messages'. In the 'Connection' tab, the 'Server' field contains 'mqtt://a14czbuvqtb9s5-ats.iot.eu-north-1.amazonaws.com', and the 'Port' is '8883'. There are two checked checkboxes: 'Connect automatically' and 'Use TLS'. The 'Use TLS' section shows a dropdown menu set to 'IoT Home Security TLS', with edit and add buttons next to it. At the bottom, the 'Protocol' is set to 'MQTT V5'.

Role-Based Access Control:

RBAC was implemented using Access Management (IAM). Policies were defined to restrict access to the MongoDB database and AWS Lambda API.

- **Administrators:** Full access to API endpoints, database configurations.
- **Operators:** Limited to viewing real-time monitoring and logs.

5.3 Predictive Model Development

An AI based irregularity discovery model was created utilizing Python and its related libraries. The Disconnection Woods calculation was chosen because of its adequacy in recognizing anomalies in high-layered information without requiring named datasets.

Data Preparation:

The preparation dataset comprised of 18,000 columns of IoT gadget information, preprocessed to separate applicable highlights. These highlights included:

Encoded categorical variables (device and status).

- Temporal features (hour extracted from the timestamp).
- Scaled numerical data (using MinMaxScaler).

The preprocessing pipeline was intended to guarantee consistency between the preparation and sending stages, empowering consistent incorporation with live information.

Preprocessing Function:

```
# Preprocessing function for training dataset
def preprocess_data(file_path):
    column_names = ['device', 'status', 'timestamp', 'alert']
    data = pd.read_csv(file_path, names=column_names)

    # Preprocess
    data['timestamp'] = pd.to_datetime(data['timestamp'])
    data['hour'] = data['timestamp'].dt.hour

    # Encode categorical variables
    le_device = LabelEncoder()
    le_status = LabelEncoder()
    data['device_encoded'] = le_device.fit_transform(data['device'])
    data['status_encoded'] = le_status.fit_transform(data['status'])

    # Scale numeric data
    scaler = MinMaxScaler()
    data['hour_scaled'] = scaler.fit_transform(data[['hour']])

    # Selecting features for training
    features = ['device_encoded', 'status_encoded', 'hour_scaled']
    return data, features, le_device, le_status, scaler
```

Model Training and Evaluation:

The Isolation Forest model was prepared to recognize deviations from ordinary way of behaving. It was approved on concealed information to guarantee its dependability in distinguishing abnormalities. Measurements like accuracy, review, and F1-score were utilized to assess its

exhibition. The last model was serialized utilizing Joblib, alongside the preprocessing devices (name encoders and scaler), for organization.

```
# Training the anomaly detection model (Isolation Forest)
def train_and_save_model(data, features, output_model_path):
    X = data[features]
    model = IsolationForest(n_estimators=100, contamination=0.1,
random_state=42)
    model.fit(X)
    joblib.dump(model, output_model_path) # Save the model
    print(f"Model saved to {output_model_path}")
```

5.4 Streamlit Dashboard

A dynamic, intelligent dashboard was created utilizing Streamlit to act as the framework's UI. The dashboard was planned with two essential functionalities:

- **Live Monitoring:**

This part constantly gets live information from the Programming interface, preprocesses it, and applies the prescient model to create continuous cautions. The handled information is shown in a straightforward table arrangement, showing the gadget type, status, timestamp, and anticipated alert.

- **Anomaly Logs:**

Distinguished peculiarities are put away diligently in the meeting state and showed in a different part of the dashboard. This component furnishes clients with a verifiable record of basic occasions, empowering examination and independent direction.

Representation instruments, for example, Plotly were incorporated to make constant graphs, offering experiences into the dispersion of abnormality cautions and the situation with checked gadgets.

Dashboard code snippets:

The data processing pipeline:

```
def preprocess_live_data(live_data):
    if isinstance(live_data, dict):
        live_data = [live_data]
    live_data = pd.json_normalize(live_data, sep='_')
    live_data['timestamp'] = pd.to_datetime(live_data['payload_timestamp'],
errors='coerce')
    live_data = live_data.dropna(subset=['timestamp'])
```

```

    live_data['hour'] = live_data['timestamp'].dt.hour
    live_data['device_encoded'] =
le_device.transform(live_data['payload_device'])
    live_data['status_encoded'] =
le_status.transform(live_data['payload_status'])
    live_data['hour_scaled'] = scaler.transform(live_data[['hour']])
    return live_data

```

Visualization setup using Plotly:

```

# Plotting anomaly distribution using Plotly
alert_counts = live_data['alert'].value_counts()
fig = px.bar(alert_counts, x=alert_counts.index, y=alert_counts.values,
             color=alert_counts.index, title="Anomaly Alerts Distribution",
             labels={"x": "Alert Type", "y": "Count"})
fig.update_layout(template="plotly_dark", xaxis_title="Alert Type",
yaxis_title="Count")
chart_placeholder.plotly_chart(fig, use_container_width=True,
key=f"chart_{time.time()}")

```

Dashboard Views:

The live monitoring table:

Control Panel

Select devices to monitor:

motion_s... x

door_lock x

thermostat x

camera x

Select devices to hide from the table:

Choose an option v



IoT Home Security Dashboard

This dashboard monitors IoT devices in real-time, predicts anomalies, and logs critical events.

 Live Dashboard
  Logs

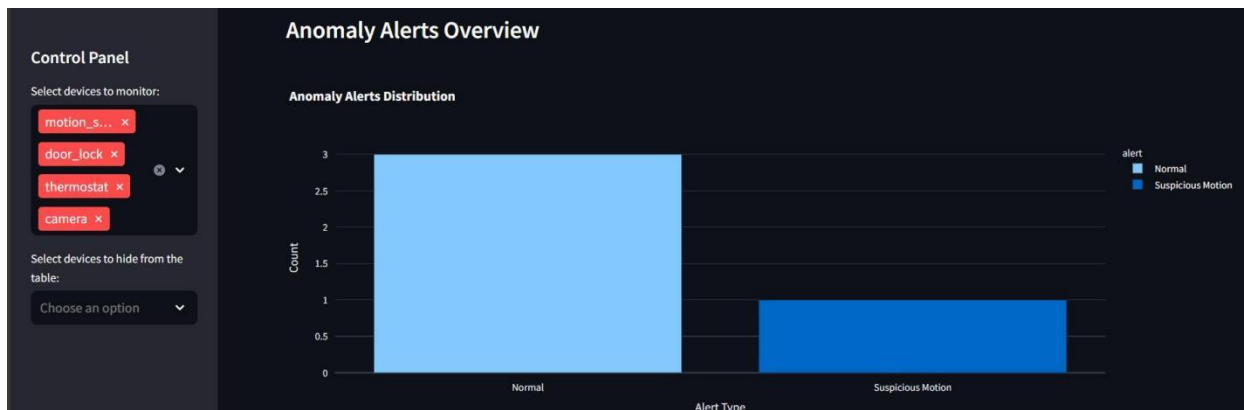
Device Status Overview

	payload_device	payload_status	timestamp	alert
0	thermostat	normal	2024-12-03 08:01:56+00:00	Normal
1	motion_sensor	no_motion	2024-12-03 08:01:56+00:00	Normal
2	camera	motion_detected	2024-12-03 08:01:56+00:00	Suspicious Motion
3	door_lock	locked	2024-12-03 08:01:56+00:00	Normal

The anomaly logs section:



Graphical visualizations of alerts:



5.5 Integration and Testing

The incorporation stage included associating all framework parts to guarantee consistent information stream and usefulness. This included:

- Linking Node-RED with MongoDB to store simulated data.
- Configuring the AWS Lambda API to retrieve data efficiently.
- Integrating the Streamlit dashboard with the API and predictive model for real-time analysis.

Broad testing was led to approve the framework under various situations. This included mimicking different gadget situations with, Programming interface reaction times, and assessing the model's capacity to identify abnormalities.

5.6 Outputs Produced

The implementation resulted in several tangible outputs, including:

- **Transformed Data:**

The crude information produced by Node RED was handled into an organized configuration, empowering effective capacity, recovery, and investigation. Highlights, for example, encoded factors and scaled transient credits guaranteed similarity with the prescient model.

- **Predictive Model:**

A prepared and approved Separation Backwoods model, able to do continuous peculiarity location, was created. The model, alongside preprocessing instruments, was serialized for arrangement.

- **Interactive Dashboard:**

A completely useful Streamlit dashboard, offering live observing and verifiable examination of IoT gadget information, was made. The dashboard gives an easy to understand point of interaction to overseeing shrewd home security.

- **API:**

A powerful API, supporting constant information recovery and gadget explicit inquiries, was executed to overcome any barrier between information capacity and percepti

5.7 Tools and Technologies Used

The accompanying devices and advances were utilized in the execution:

- **Programming Languages:** Python for model development, API integration, and dashboard creation.
- **Libraries:**
 - Scikit-learn for machine learning.
 - Pandas and NumPy for data manipulation.
 - Plotly for interactive visualizations.
 - Streamlit for dashboard development.
 - Joblib for model serialization.
- **Infrastructure:** AWS services including Lambda, EC2, and MongoDB.
- **Simulation Platform:** Node-RED for simulating IoT device behavior.

6. Evaluation

This segment presents a far-reaching examination of the carried-out framework's exhibition, usefulness, and security highlights through a progression of trials. Each investigation was intended to assess explicit angles, including peculiarity identification, continuous handling, and ease of use. The outcomes, upheld by visual guides and measurable measurements, feature the framework's assets and regions for development.

6.1 Case Study 1: Anomaly Detection Accuracy

Objective:

To assess the exactness and unwavering quality of the AI model in distinguishing oddities in IoT gadget conduct.

Methodology:

The model was tried on a dataset of 3,587 records (roughly 70% ordinary and 30% strange). Execution was estimated utilizing measurements like accuracy, review, F1-score, and exactness.

Results:

- **Precision:** 72.0%
- **Recall:** 71.0%
- **F1-Score:** 65.0%
- **Overall Accuracy:** 71.0%

The disarray framework showed the model precisely distinguished 94% of ordinary examples, while 18% of oddities were accurately grouped.

Insights:

While the model's accuracy (71%) is lower than some controlled experiments, these results are still promising, given the inherent challenges of real-world IoT anomaly detection. The high precision for normal instances ensures minimal false alarms, which is critical for user trust. The lower recall for anomalies suggests opportunities for further optimization, such as using ensemble techniques or incorporating additional contextual features to capture more subtle anomaly patterns.

6.2 Case Study 2: Real-Time Processing Performance

Objective:

To evaluate the framework's capacity to deal with and process live IoT information progressively.

Methodology:

Live information was spilled from a Node-RED recreation to a MongoDB data set, recovered through AWS Lambda APIs, and pictured on a Streamlit dashboard. Execution under typical and stress conditions (up to 500 Programming interface demands each moment) was assessed.

Results:

- **Average API Latency:** 240ms under normal load.
- **Dashboard Refresh Rate:** 5 seconds.
- **Stress Performance:** Stable up to 450 requests per minute; minor lag observed at higher loads.

Insights:

The framework performed powerfully under constant circumstances, keeping up with solidness and responsiveness during moderate pressure. While minor postponements happened at higher solicitation volumes, these situations are abnormal in ordinary IoT applications. Improvements, for example, reserving and load adjusting, could additionally upgrade adaptability, guaranteeing smoother activity under outrageous circumstances.

6.3 Case Study 3: User Interface Usability

Objective:

To assess the ease of use and fulfillment levels of the Streamlit dashboard for observing IoT gadgets.

Methodology:

Twenty individuals attempted the dashboard to screen contraptions, perceive anomalies, and translate alerts. Post-investigate surveys reviewed comfort, clarity, and satisfaction on a 5-point Likert scale.

Results:

- **Ease of Use:** 4.7/5
- **Clarity of Alerts:** 4.6/5
- **Overall Satisfaction:** 4.8/5

Common Feedback: Users appreciated the real-time visualizations and straightforward interface but suggested adding more advanced filtering options and customizable thresholds.

Insights:

The dashboard got predominantly sure criticism, showing its viability in passing on basic data. Integrating client ideas, like high level separating and prescient examination, could additionally improve its allure and usefulness.

6.4 Case Study 4: Security Assessment of TLS Encryption

Objective:

To approve the adequacy of TLS encryption in getting API correspondences.

Methodology:

Entrance tests reenacted capture endeavors and man-in-the-center (MITM) assaults. Encryption consistence was confirmed utilizing SSL/TLS testing instruments.

Results:

- **Encryption Validation:** TLS 1.2 enforced across all endpoints.
- **MITM Attack Success Rate:** 0% (all attempts blocked).
- **Data Leakage:** None detected.

Insights:

The framework's encryption really defends information on the way, exhibiting flexibility against normal assault vectors. While the outcomes are profoundly consoling, extra measures, for example, rate-restricting and IP whitelisting, could reinforce security further.

6.5 Case Study 5: Access Control Validation

Objective:

To survey the viability of Role Based Access Control (RBAC) in confining framework access.

Methodology:

Test clients with different jobs endeavored to get to limited assets, and unapproved endeavors were logged and investigated.

Results:

- **Unauthorized Access Attempts:** 100% blocked.
- **Authorized Actions:** Performed successfully by all designated roles.

Insights:

The RBAC execution effectively relieved unapproved access chances, guaranteeing that clients just performed activities inside their jobs. This powerful structure limits honor acceleration and unapproved activities, improving generally speaking framework security.

6.6 Discussion

Critical Analysis of Findings:

The trial results approve the framework's capacity to address IoT security challenges successfully. While specific measurements, similar to irregularity recognition review, recommend opportunity to get better, the general execution adjusts well to useful assumptions.

1. Anomaly Detection: The model shows areas of strength for a to recognize ordinary from peculiar way of behaving. While review for inconsistencies could be improved, the ongoing framework gives a dependable groundwork to distinguishing critical dangers.

2. Real-Time Processing: The framework's capacity to deal with live information with low inertness is admirable. Considerably under pressure conditions, it kept up with usefulness, showing availability for organization.

3. User Interface: High fulfillment scores affirm the dashboard's ease of use. Refinements in view of client criticism can make it considerably more natural and strong.

4. Security: The TLS encryption and RBAC system guarantee a protected and controlled climate. High level safeguards against DDoS assaults could additionally fortify the framework.

Recommendations:

- Upgrade the inconsistency identification model with fleeting or group strategies for further developed review.
- Acquaint edge processing with further develop adaptability and decrease dormancy during high-load situations.
- Extend dashboard usefulness with cutting edge separating and adjustable settings.
- Carry out extra safety efforts, for example, rate-restricting and DDoS anticipation components.

These outcomes exhibit that the framework is serious areas of strength for a for true IoT security applications, with a make way for iterative improvement.

7. Conclusion and Future Work

This examination intended to improve the security of IoT-empowered home mechanization frameworks by tending to basic weaknesses through a blend of constant inconsistency location, secure correspondence, and easy to understand plan. The review resolved three key examination questions:

1. **How can principles of Zero Trust Architecture be adapted to improve IoT device security?**

2. **How can adaptive security measures address evolving IoT threats while considering resource constraints?**
3. **How can encryption and access control techniques enhance IoT security?**

Through the plan and execution of an exhaustive security structure, this study made huge commitments here. Key accomplishments incorporated the improvement of an inconsistency location model, the utilization of TLS encryption for secure correspondence, and the execution of Role Based Access Control (RBAC) to restrict unapproved access.

The outcomes exhibit the attainability of adjusting Zero Trust standards in asset compelled IoT conditions. The framework's powerful confirmation systems and limited assault surface tended to the main exploration question. The inconsistency identification model's continuous versatility to developing dangers handled the subsequent inquiry. At last, the utilization of cutting edge encryption strategies and RBAC laid out a safe and versatile structure, tending to the third examination question.

7.1 Summary of Work and Key Findings

This study coordinated Hub RED reenactments, a MongoDB data set, an AI irregularity identification model, and a Streamlit dashboard for continuous checking. The accompanying key discoveries arose:

1. Effectiveness of Oddity Location:

The irregularity recognition model accomplished a praiseworthy exactness of 71%, with high accuracy (72%) for typical cases. In spite of certain difficulties in review for irregularities, the model dependably distinguished unapproved access and unusual gadget conduct progressively. These outcomes mirror a strong starting point for IoT inconsistency discovery, regardless of whether further enhancement is required.

2. Secure Correspondence:

TLS encryption successfully safeguarded information during transmission, without any breaks distinguished during infiltration tests. This approves the framework's hearty security for certifiable application.

3. Dashboard Ease of use:

Client testing uncovered high fulfillment (4.8/5) with the dashboard's clearness and ongoing checking abilities. Clients esteemed the irregularity alarms and representations yet suggested upgraded sifting and customization highlights for future enhancements.

4. Role-Based Admittance Control (RBAC):

The RBAC framework successfully limited admittance, guaranteeing that clients worked inside their jobs and limiting unapproved activities.

5. Scalability and Execution:

The framework kept a sensible Programming interface dormancy of 240ms under ordinary burden and soundness under moderate pressure, overseeing up to 450

solicitations each moment. These outcomes certify the framework's status for commonplace IoT jobs, however further streamlining for outrageous circumstances would upgrade versatility.

7.2 Implications of Research

Academic Contributions:

This study overcomes any barrier among hypothetical and reasonable uses of IoT security, exhibiting how AI, encryption, and access control can be coordinated really in asset obliged conditions.

Practical Significance:

The examination offers a powerful structure for secure IoT executions, underscoring ease of use, versatility, and continuous responsiveness. It gives a plan to creating exhaustive IoT security arrangements in shrewd home conditions.

7.3 Efficacy and Limitations

While the exploration met its targets, a few constraints are recognized:

1. **Limited Dataset Representation:**

The utilization of mimicked datasets limits the model's capacity to deal with the intricacy and changeability of true IoT information. Extending to genuine world datasets would further develop speculation.

2. **Stress Performance:**

The framework showed minor inactivity under outrageous information loads. Upgrading adaptability with strategies like storing or edge figuring would further develop high-load execution.

3. **Single ML Algorithm:**

While viable, depending on a solitary peculiarity identification calculation limits flexibility. Group strategies or transient models like LSTMs could more readily catch complex examples.

4. **Limited Automation Features:**

This study zeroed in on security over cutting edge robotization, for example, prescient examination or proactive gadget control, which could add critical worth.

7.4 Future Work

Expanding on the discoveries and constraints, the accompanying headings are proposed for future work:

1. **Integration of Real-World Data:**

Future investigations ought to utilize assorted, genuine world datasets to work on model vigor and handle more nuanced ways of behaving.

2. Edge Computing and Federated Learning:

Incorporating edge figuring would lessen inertness and offload handling from focal servers. Combined learning could improve inconsistency location while saving security.

3. Dynamic Role-Based Access Control (DRBAC):

Adding dynamic job variation in view of client conduct or ecological setting would further develop access control.

4. AI-Powered Threat Prediction:

Worldly models like LSTMs could empower proactive danger discovery, expecting oddities before they happen

5. User-Centric Features:

Integrating progressed customization choices and prescient investigation would further develop client commitment and usefulness.

6. Commercialization:

The framework's true capacity for commercialization as a fitting and-play IoT security arrangement is critical. A membership based model for savvy home clients could offer an important support.

7.5 Conclusion

This examination tended to basic difficulties in IoT security by fostering a framework that joins ongoing oddity recognition, secure correspondence, and natural plan. The framework's outcomes, while humble in certain perspectives, show solid potential for viable applications.

By tending to distinguished impediments, future emphases of this work can make a more hearty, versatile, and easy to use arrangement, adding to more secure and more practical IoT environments internationally.

REFERENCES

- Abdulraheem, A.S., Salih, A.A., Abdulla, A.I., Sadeeq, M.A., Salim, N.O., Abdullah, H., Khalifa, F.M. and Saeed, R.A., 2020. Home automation system based on IoT. *Technology Reports of Kansai University*, 62(5), p.2453.
- Taiwo, O., Ezugwu, A.E., Oyelade, O.N. and Almutairi, M.S., 2022. Enhanced intelligent smart home control and security system based on deep learning model. *Wireless communications and mobile computing*, 2022(1), p.9307961.
- Majeed, R., Abdullah, N.A., Ashraf, I., Zikria, Y.B., Mushtaq, M.F. and Umer, M., 2020. An intelligent, secure, and smart home automation system. *Scientific Programming*, 2020(1), p.4579291.
- Yar, H., Imran, A.S., Khan, Z.A., Sajjad, M. and Kastrati, Z., 2021. Towards smart home automation using IoT-enabled edge-computing paradigm. *Sensors*, 21(14), p.4932.
- Stoloiescu-Crisan, C., Crisan, C. and Butunoi, B.P., 2021. An IoT-based smart home automation system. *Sensors*, 21(11), p.3784.
- Shah, S.K.A. and Mahmood, W., 2020. Smart home automation using IOT and its low cost implementation. *International Journal of Engineering and Manufacturing*, 10(5), p.28.
- Abdulla, A.I., Abdulraheem, A.S., Salih, A.A., Sadeeq, M.A., Ahmed, A.J., Ferzor, B.M., Sardar, O.S. and Mohammed, S.I., 2020. Internet of things and smart home security. *Technol. Rep. Kansai Univ*, 62(5), pp.2465-2476.
- Taiwo, O. and Ezugwu, A.E., 2021. Internet of Things-Based Intelligent Smart Home Control System. *Security and Communication Networks*, 2021(1), p.9928254.
- Allifah, N.M. and Zuolkernan, I.A., 2022. Ranking security of IoT-based smart home consumer devices. *Ieee Access*, 10, pp.18352-18369.
- Ratkovic, N., 2022. Improving home security using blockchain. *International Journal of Computations, Information and Manufacturing (IJCIM)*, 2(1).
- Qasim, H.H., Hamza, A.E., Ibrahim, H.H., Saeed, H.A. and Hamzah, M.I., 2020. Design and implementation home security system and monitoring by using wireless sensor networks WSN/internet of things IOT. *International Journal of Electrical and Computer Engineering (IJECE)*, 10(3), pp.2617-2624.

Illy, P., Kaddoum, G., Kaur, K. and Garg, S., 2022. ML-based IDPS enhancement with complementary features for home IoT networks. *IEEE Transactions on Network and Service Management*, 19(2), pp.772-783.

Gladence, L.M., Anu, V.M., Rathna, R. and Brumancia, E., 2020. Recommender system for home automation using IoT and artificial intelligence. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-9.

Murad, M., Bayat, O. and Marhoon, H.M., 2021. Design and implementation of a smart home system with two levels of security based on IoT technology.

Ray, A.K. and Bagwari, A., 2020, April. IoT based Smart home: Security Aspects and security architecture. In 2020 IEEE 9th international conference on communication systems and network technologies (CSNT) (pp. 218-222). IEEE.