WiFi Based Authentication Scheme for Smart Lock IoT Devices

A Project submitted in fulfilment of the
requirements for the award of the degree
of

BACHELORS OF TECHNOLOGY

by

Dikshant Sharma (21bcs019)
Piyush (21bcs107)
Sajal Bajaj (21dcs002)
Tushar Thakur (21dcs013)

Under the guidance of

Dr. Preeti Soni

Department of Computer Science & Engineering
National Institute of Technology Hamirpur
Hamirpur, India-177005

Candidate's Declaration

We hereby declare that the report presented for this project, titled **"WiFi-Based Authentication Scheme for Smart Lock IoT Devices"**, is submitted in fulfilment of the requirements for the award of the degree of **Bachelor of Technology**. It has been prepared under the Department of Computer Science and Engineering, National Institute of Technology Hamirpur, as an authentic record of our own work conducted between January 2025 and May 2025 under the guidance of **Dr. Preeti Soni**, Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology Hamirpur.

The matter presented in this report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Sajal Bajaj (21dcs002)	**Dikshant Sharma (21bcs019)**
Tushar Thakur (21dcs013)	**Piyush (21bcs107)**

This is to certify that the above statement made by the candidates is true to the best of my knowledge and belief.

Dr. Preeti Soni
Date: **Assistant Professor**

The B.Tech Viva-Voce exam has been successfully held on ....

Convener, DBPC **Head, DoCSE**

ACKNOWLEDGMENTS

We would like to express our heartfelt gratitude to our supervisor, **Dr. Preeti Soni**, for her invaluable guidance, constructive criticism, and continuous encouragement throughout this project. Her expertise and unwavering support played a pivotal role in shaping this work, greatly enhancing the quality and depth of our research.

We extend our sincere appreciation to **Dr. Siddhartha Chauhan**, Head of the Department of Computer Science and Engineering, National Institute of Technology, Hamirpur, for his leadership and support throughout the course of this work. His efforts in fostering a collaborative and resource-rich environment were vital to the successful completion of this project.

We are deeply grateful to the faculty members and support staff of the Department of Computer Science and Engineering for their assistance, valuable resources, and technical support, which were instrumental in the progression of our research. Their contributions ensured a smooth and enriching research journey.

Our heartfelt thanks also go to our peers and classmates, who provided helpful discussions, feedback, and collaboration during the challenging phases of this research. Their camaraderie and encouragement made this journey both productive and enjoyable.

We are especially indebted to our parents and family for their unwavering support, patience, and belief in our abilities. Their love and encouragement provided us with the strength and motivation to overcome obstacles and focus on our goals.

Finally, we express our profound gratitude to the Almighty for bestowing us with the resilience, health, and determination required to bring this project to fruition.

ABSTRACT

This project presents an enhanced Wi-Fi Radio-Based Authentication System aimed at improving the security, usability, and scalability of smart lock IoT devices. Traditional authentication approaches, such as PIN codes or passwords, often involve manual effort and are prone to human error and vulnerabilities like credential theft. To address these issues, this work introduces a zero-effort proximity-based authentication mechanism that utilizes ambient Wi-Fi signals combined with machine learning (ML) models.

The system identifies users based on key Wi-Fi signal features, such as beacon frame data and RSSI values from nearby access points (APs). Two Raspberry Pi 3 Model B boards emulate the smart lock and the user's mobile device, enabling effective data collection across diverse network conditions. A curated dataset was used to train and evaluate ML models including Decision Tree, Random Forest, and K-Nearest Neighbors, achieving a maximum accuracy of 93%.

To transition from a prototype to a real-time solution, the system was extended to include a server-based architecture and a custom web application. The Web interface allows users to log in securely, monitor their registered smart locks, and remotely manage access. Real-time authentication decisions are made on the server and communicated back to the lock system, enabling seamless and interactive operation.

This research offers a flexible and scalable framework suitable for a wide range of smart lock applications, from residential automation to enterprise-level security. Future enhancements may include larger-scale deployment, integration of motion or biometric sensing, and optimization for multi-user environments to further improve system reliability and user convenience.

Table of Contents

List of Figures

List of Acronyms/Abbreviations

2FA	Two-Factor Authentication
API	Application Programming Interface
BSSID	Basic Service Set Identifier (MAC address of Wi-Fi access point)
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DT	Decision Tree
ECG	Electrocardiogram
EMG	Electromyogram
GNB	Gaussian Naive Bayes
GPU	Graphics Processing Unit
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IoT	Internet of Things
IR	Infrared
KNN	K-Nearest Neighbors
ML	Machine Learning
NFC	Near Field Communication
OTP	One-Time Password
RFID	Radio Frequency Identification
RF	Random Forest
RSSI	Received Signal Strength Indicator
SSID	Service Set Identifier (Wi-Fi Network Name)
SVM	Support Vector Machine
SSL	Secure Sockets Layer
TLS	Transport Layer Security

Chapter 1

Introduction

1.1 Overview

The integration of Internet of Things (IoT) devices in the day-to-day environment has revolutionized user interaction with technology to make it more accessible and secure. Among these technologies, smart locks are now an inseparable part of today's access control technologies that offer digital, remote, and automatic locking of physical places. Conventional methods of authentication like PIN numbers, passwords, and biometric systems have their limitations. These are cumbersome systems requiring expensive hardware and are still vulnerable to numerous attacks such as password theft, social engineering, and device spoofing.

Also, these systems rely on custom sensors or biometric modules that not only add to the development expense but also add to the power requirements, a particular significant limitation for battery-powered IoT devices. Therefore, there exists a growing need for lightweight, secure, and simple-to-use authentication schemes that address these issues without sacrificing usability or scalability.

1.2 Motivation

With increasing IoT solutions being deployed in commercial and residential settings, more affordable and secure access control systems have gained greater popularity. Traditional practices are marred by the following problems:

- Passwords and PINs are easily guessable, phishable, or forgettable by the user and do not usually have good fallbacks.

- Biometric logon, while being more secure, uses specialized and usually expensive hardware like fingerprint or facial recognition sensors.

  In contrast, Wi-Fi technology is already embedded in most IoT devices, making it a low-cost, ubiquitous channel to implement proximity-based authentication. Based on the characteristics of Wi-Fi signals in some beacon frames and the Received Signal Strength Indicator (RSSI) values, we can construct a passive, zero-effort authentication system with no additional hardware components. The solution not only avoids the limitation of traditional systems but also meets the low-power, high-efficiency needs of modern IoT environments.

1.3 Objective

The general purpose of this project is to design and implement an intelligent **Wi-Fi Radio-Based Authentication Scheme for Smart Lock IoT Devices** with the following general goals:

- **Effortless Authentication:** Offer effortless, proximity-based access control without the need for user action.

- **Cost-Effective Deployment:** Leverage existing Wi-Fi infrastructure to reduce hardware dependency and deployment expenses.

- **Energy Efficiency:** Optimize to execute on low-power, battery-operated IoT devices.

- **High Accuracy and Security:** Utilize machine learning models for high-accuracy classification of authorized and unauthorized users.

- **Scalable and Flexible Design:** Support multiple real-world environments and facilitate easy management of multiple devices and users.

In the last phase of this project, the system was augmented with a real-time server backend and web interface, with remote lock management, as well as monitoring facilities.

1.4 Scope of Work

This project involves the development of an end-to-end Wi-Fi signal analysis and machine learning-based proximity-based authentication system. The system includes:

- Retrieval of data through Raspberry Pi devices simulating the user device and smart lock.
- Validation of the model and learning from RSSI data collected from various Wi-Fi settings.
- A central server for performing backend authentication.
- An application where one can register, log in, and control their smart locks remotely in real time.

The project is focused on practical deployment, scalability, and flexibility to real-world environments, with the goal of closing the gap between experimental prototypes and deployable IoT security solutions.

1.5 Structure of the Report

The rest of the report follows this organization:

- **Chapter 2** summarizes related work and current technologies in authentication for smart locks.
- **Chapter 3** deals with experimental setup, hardware components, and data acquisition process.
- **Chapter 4** outlines the intended methodology, such as data preprocessing, machine learning model selection, as well as server integration.
- **Chapter 5** illustrates the evaluation outcome, such as performance measures and real-time system verification.
- **Chapter 6** concludes the report with the key findings and sketches out possible areas for future research and development.

Chapter 2

Literature Review

Limited work exists on applying these methods specifically to IoT smart locks, leaving room for this project's contributions.

S. AlQahtani et al. proposed a two-factor authentication system leveraging Wi-Fi and machine learning in two studies. The first study, published in 2023, introduces a *zero-effort two-factor authentication (2FA) system* using Wi-Fi radio wave transmission and ML to analyze beacon frame characteristics and RSSI values, providing a secure yet user-friendly method of authentication [1]. The second study, published in 2024, extends this concept with *environmental continuous two-factor authentication*, incorporating continuous verification and robust security measures against cyberattacks, but notes challenges with scalability and signal interference [2].

Reem Alrawili et al. (2024) do a detailed survey on *biometric characteristics for user authentication*, with regard to security, convenience, scalability, and cost [3]. The limitation, however, is that generalizability to non-English contexts is very limited.

H. F. Atlam et al. (2020) designed a framework for IoT data collection and forensic analysis with its focus on smart home appliances. It supports data management, scalability, and collection of evidence but lacks the necessary handling and analysis of large-scale data in IoT [4].

D. R. Bhuva et al. (2023) presented an IoT device's mechanism based on *continuous authentication employing biometric signals* such as ECG and EMG signals. The technique emphasizes precision with low power consumption, yet suffers due to high power usage for compute-intensive CNN models on EMG for IoT [5].

Aswini D et al. (2021) proposed a *smart door locking system* using Raspberry Pi, RFID, OTP, and a mobile app for remote locking. This system gives ease of remote access and has good security through time-limited OTPs but is dependent on consistent cloud connectivity and heavy user dependency on mobile devices [6].

Kaustubh Dhondge et al. (2016) proposed an *infrared-based smart lock system* using smartphone IR light and OTPs with cloud servers. The system was able to unlock correctly at a distance of 20m but has scalability issues and is line-of-sight dependent [7].

Sandeep Gupta et al. (2019) developed *SmartHandle*, a behavioral biometric-based authentication system for smart locks. The system uses hand-movement data captured by accelerometers, gyroscopes, and magnetometers with ML. Despite its robust authentication capabilities, the approach faces challenges due to limited training data and usability issues in varied environments [8].

Naresh Kumar Masurkar et al. (2019) proposed an *NFC-based dual authentication system* that integrated biometrics with access control protocols. This system enhances security through biometric verification but lacks scalability for integration with modern IoT systems [9].

Sarah Delgado Rodriguez et al. (2024) proposed *Act2Auth*: an authentication mechanism using capacitive sensing via desk interactions. The method prioritizes usability and memorability of secrets but is limited to desk setups and sensitive to object positioning [10].

S. Saloni et al. (2023) studied *WiFi-Aware as a connectivity technology* for IoT, enabling proximity-based services with low power consumption and interoperability, but struggling in large IoT networks due to scaling issues [11].

Aditya Saroha et al. (2022) proposed an *IoT-based smart lock system* using facial recognition and passcode authentication. Though face recognition is efficient, its usage is impacted by lighting dependency and increased power consumption for constant monitoring [12].

Muhammad Shahzad et al. (2017) proposed a *continuous authentication system* for IoT devices using Wi-Fi signal processing. While it achieves good precision and real-time performance, it's limited to about six meters range and lower accuracy for security-critical applications [13].

Chapter 3

Experiment Setup

The experiment was designed to evaluate the performance of a real-time, server-integrated **Wi-Fi Radio-Based Authentication System for Smart Lock IoT Devices**. The system utilizes ambient Wi-Fi signal characteristics, such as beacon frame data and RSSI values, to authenticate users based on proximity. It integrates hardware components with cloud-based machine learning inference and a web interface for real-time monitoring and control.

3.1 System Configuration

The authentication infrastructure is built as a client-server model involving smart hardware nodes, a central decision-making server, and a web application for user interaction. The key components of the setup are:

- **Smart Lock Node:** A Raspberry Pi 3 Model B installed at the access point acts as the lock device, continuously scanning for nearby Wi-Fi access points and collecting signal data.
- **User Device Simulator:** Another Raspberry Pi 3 Model B mimics the user's mobile device, emitting identifiable Wi-Fi traffic for proximity detection.
- **Authentication Server:** A cloud-hosted or local server receives Wi-Fi data via API, applies trained ML models for proximity classification, and makes authentication decisions.
- **Web Dashboard:** A user-facing web interface allows registered users to log in, monitor the state of their smart locks, and review access history in real time.

This configuration enables continuous two-way communication between the smart lock node and the server for seamless and responsive access control.

3.2 Real-Time Workflow

The real-time operation of the system follows this workflow:

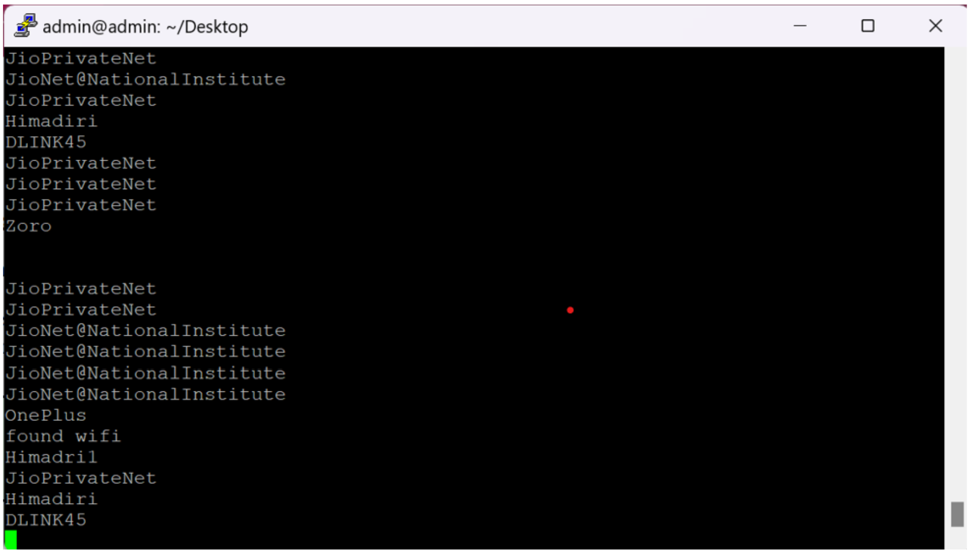
1. The smart lock device scans surrounding Wi-Fi access points and captures beacon frame metadata including SSID, BSSID, frequency, and RSSI.
2. The scanned data is sent to the central server via a secure RESTful API.
3. The server processes the data using a pre-trained machine learning model to determine whether the user's device is within an acceptable proximity threshold.
4. Based on the classification result, the server grants or denies access, and the decision is logged to the user's account on the web dashboard.
5. Users can log into the dashboard to view lock status, recent authentication attempts, and manage associated smart locks.

This architecture supports scalable multi-user scenarios and ensures that lock decisions are made dynamically with minimal delay.

3.3 Data Collection and Environment

Wi-Fi signal data was collected in real-world conditions to simulate practical usage scenarios:

- Over **250 labeled samples** were captured, representing both authorized (within 7 feet) and unauthorized (beyond 7 feet) proximity scenarios.
- Data was gathered across multiple rooms, corridors, and residential hostel areas to reflect environmental variation and test model generalization.
- Key features recorded included SSID, BSSID, RSSI, and frequency from multiple Wi-Fi access points.
- Figure 3.4 shows the Raspberry Pi smart lock detecting multiple APs during scanning.



```
JioPrivateNet
JioNet@NationalInstitute
JioPrivateNet
Himadiri
DLINK45
JioPrivateNet
JioPrivateNet
JioPrivateNet
Zoro


JioPrivateNet
JioPrivateNet
JioNet@NationalInstitute
JioNet@NationalInstitute
JioNet@NationalInstitute
JioNet@NationalInstitute
OnePlus
found wifi
Himadril
JioPrivateNet
Himadiri
DLINK45
```

Figure 3.1: Detection of multiple APs by Raspberry Pi.

3.4 Machine Learning Model Training

To classify user proximity, various ML algorithms were trained on the collected dataset. The training phase was performed offline, and the final models were deployed to the authentication server.

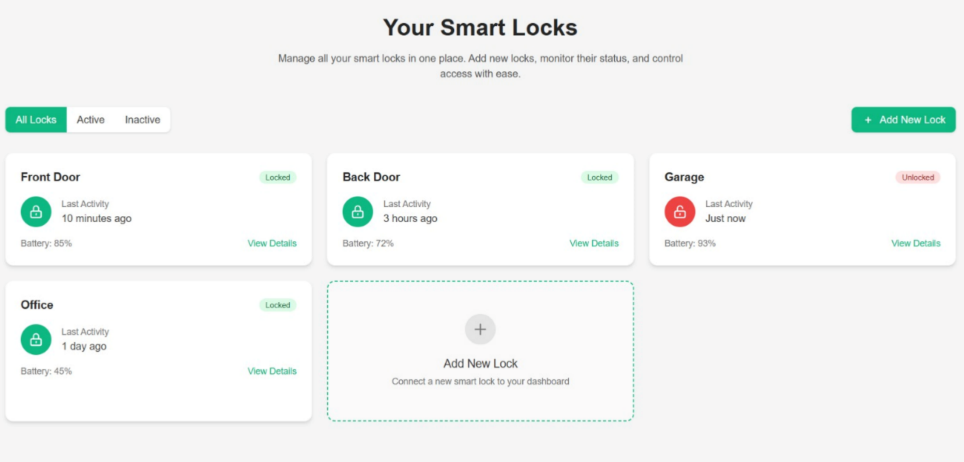
- **Algorithms Used:** Decision Tree, Random Forest, and K-Nearest Neighbors (K-NN).
- **Dataset Split:** 80% for training and 20% for testing.
- **Evaluation Metrics:** Accuracy and F1 Score.
- **Best Performer:** Random Forest achieved the highest accuracy of 93%, showing strong robustness to variations in signal patterns.

Each model was tested for its ability to distinguish between valid and invalid proximity conditions under varying signal noise and device positions.

3.5 System Testing and Real-World Validation

The complete system, which encompasses signal capture, server-based classification, and real-time lock actuation, was deployed and tested in a working environment. Key observations included:

- The system responded in real-time with a delay of less than 2 seconds from signal scan to lock decision.



Your Smart Locks

Manage all your smart locks in one place. Add new locks, monitor their status, and control access with ease.

Front Door — Locked
Last Activity
10 minutes ago
Battery: 85%
View Details

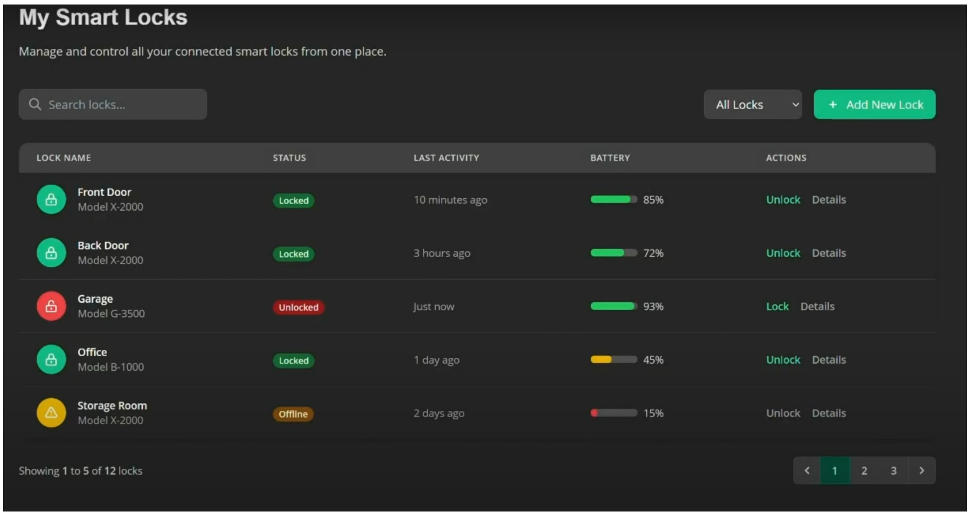
Back Door — Locked
Last Activity
3 hours ago
Battery: 72%
View Details

Garage — Unlocked
Last Activity
Just now
Battery: 93%
View Details

Office — Locked
Last Activity
1 day ago
Battery: 45%
View Details

Add New Lock
Connect a new smart lock to your dashboard

Figure 3.2: interface displaying all configures locks of an user.



My Smart Locks

Manage and control all your connected smart locks from one place.

LOCK NAME	STATUS	LAST ACTIVITY	BATTERY	ACTIONS
Front Door Model X-2000	Locked	10 minutes ago	85%	Unlock Details
Back Door Model X-2000	Locked	3 hours ago	72%	Unlock Details
Garage Model G-3500	Unlocked	Just now	93%	Lock Details
Office Model B-1000	Locked	1 day ago	45%	Unlock Details
Storage Room Model X-2000	Offline	2 days ago	15%	Unlock Details

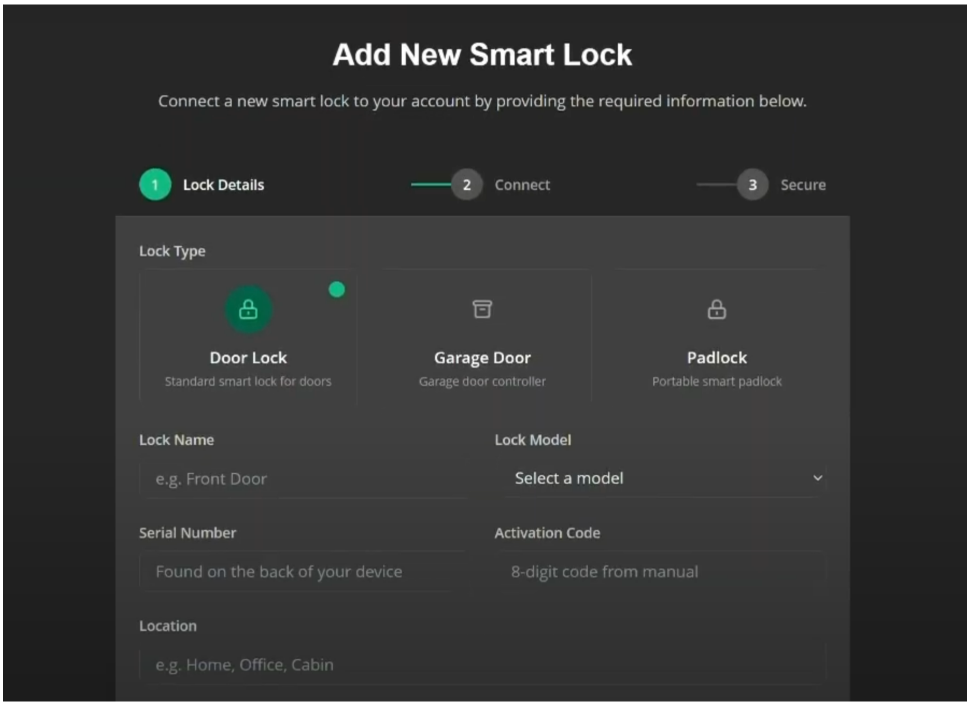
Showing 1 to 5 of 12 locks

Figure 3.3: interface to manage and monitor all locks

- Signal variability due to walls and interference was effectively handled by the trained ML models.

- Web interface integration ensured that lock status and event history were immediately visible to users after each attempt.

This testing validated the practical feasibility of the proposed solution and confirmed its ability to operate effectively in dynamic and unstructured environments.



Add New Smart Lock

Connect a new smart lock to your account by providing the required information below.

Lock Type

- Door Lock: Standard smart lock for doors ●
- Garage Door: Garage door controller
- Padlock: Portable smart padlock

Lock Name: e.g. Front Door

Lock Model: Select a model

Serial Number: Found on the back of your device

Activation Code: 8-digit code from manual

Location: e.g. Home, Office, Cabin

Figure 3.4: interface to add or configure new lock

Chapter 4

Methodology

The method used in the Wi-Fi Radio-Based Authentication Scheme for Smart Lock IoT Devices relies on the direct evaluation of Wi-Fi signal features using a server-based system in addition to machine learning (ML) to facilitate secure and easy access management. The system offers users remote control over smart locks using a web portal, where authentication is done in real-time depending on proximity.

4.1 System Architecture

The overall system is designed as a modular and scalable client-server architecture. It comprises the following core components:

- **Smart Lock Node:** There is a Raspberry Pi 3 Model B installed at the door as the smart lock. It is listening to surrounding Wi-Fi signals, gathering beacon frame information and RSSI samples, and is talking to the central server.

- **User Device Simulator:** The second Raspberry Pi 3 Model B acts as the user's device, transmitting familiar Wi-Fi signals in a typical use scenario.

- **Authentication Server:** There exists a cloud-hosted server that gets signal data from the smart lock node. It executes pre-trained machine learning models to decide proximity and make the authorization decision.

- **Web-Based Control Panel:** A web-based secure interface allows users to log in, register, or control smart locks, and provide real-time access to authentication logs. The interface allows user-specific management of locks and serves as a buffer between the server and the end user.

This architecture supports asynchronous, scalable, and low-latency communication between devices, server, and the user interface.

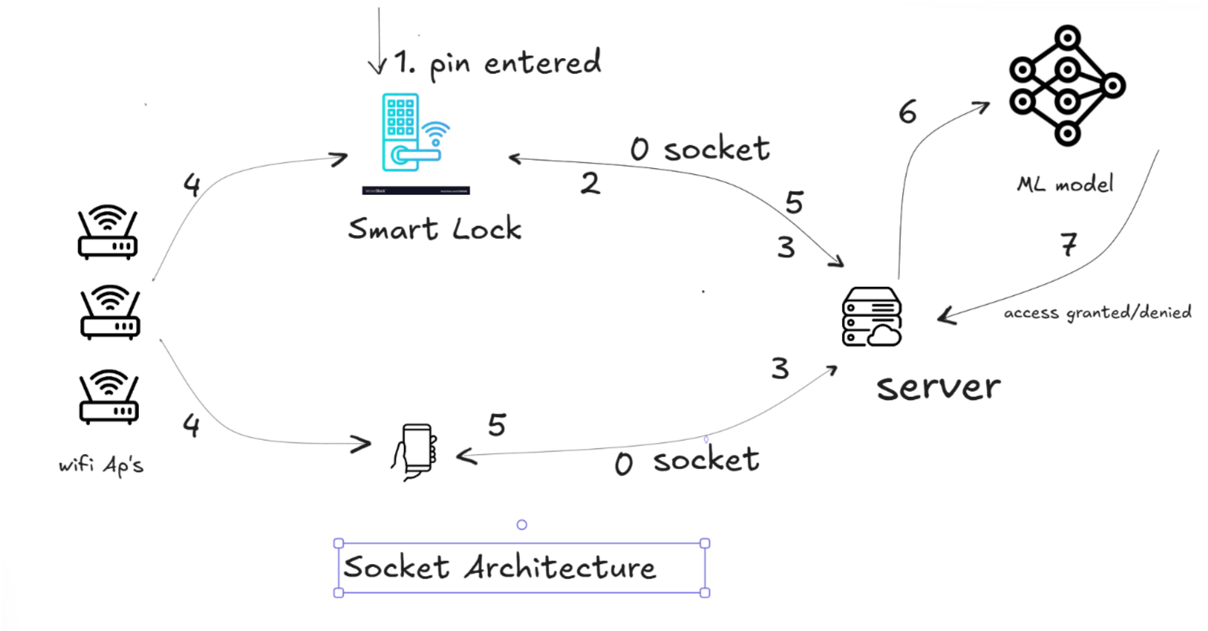


Figure 4.1: data and work flow of the system

4.2 Real-Time Data Flow

The system operates through a continuous, event-driven pipeline:

1. **Signal Scanning:** The smart lock node regularly scans nearby Wi-Fi access points and extracts RSSI and beacon frame metadata.

2. **Data Transmission:** The signal data is sent over HTTP to the central authentication server using a lightweight API.

3. **ML-Based Decision Engine:** The server uses pre-trained ML models to classify the signal into either "authorized" or "unauthorized" based on proximity thresholds.

4. **Access Control Feedback:** The server communicates the access decision (grant/deny) back to the smart lock. Simultaneously, authentication logs are updated in the user's dashboard.

5. **User Interaction via Web Portal:** Users can view their registered locks, manage access permissions, and monitor activity from the front-end web application.

4.3 Data Collection and Labeling

Data was collected from real-world Wi-Fi environments using the two Raspberry Pi devices across various physical layouts. The process included:

- Capturing RSSI values and beacon information from surrounding access points.
- Simulating authorized presence (within 7 feet) and unauthorized presence (beyond 7 feet).
- Recording over 250 labeled samples representing proximity conditions across different environments.

This labeled data was used to train the ML models deployed on the server.

4.4 Model Training and Evaluation

Multiple supervised learning algorithms were evaluated for proximity classification:

- **Decision Tree (DT):** A simple and interpretable classifier that recursively splits data based on feature thresholds.
- **Random Forest (RF):** An ensemble of decision trees that aggregates predictions to improve robustness and accuracy.
- **K-Nearest Neighbors (K-NN):** A distance-based model that classifies new instances based on the majority class of their closest neighbors.
- **Support Vector Machine (SVM):** A classifier that finds the optimal boundary between classes in a high-dimensional feature space.
- **Logistic Regression:** A probability-based classifier suitable for binary classification tasks.

- **Gaussian Naive Bayes (GNB):** A probabilistic model assuming feature independence, useful for high-dimensional but small datasets.

Dataset Split and Metrics

The dataset was divided into training and testing sets:

- **Training Set (80%):** Used to train the models.

- **Testing Set (20%):** Used to evaluate model performance.

Performance was measured using:

- **Accuracy:** Proportion of correctly classified instances.

- **F1 Score:** Harmonic mean of precision and recall to assess classification balance.

4.5 Web Application and Server Integration

The web interface was developed using modern web technologies and connected to the backend server through a secure API. Features include:

- User registration and login with role-based access.

- Dashboard displaying registered smart locks and their status.

- Real-time display of authentication attempts and results.

The backend, built using a Python-based framework, handles:

- Data ingestion from Raspberry Pi devices.

- ML inference for proximity-based classification.

- Secure communication with the web portal.

4.6 Advantages of the Proposed System

- **Real-Time Operation:** Enables instant authentication decisions via server inference.
- **Remote Access Control:** Users can manage locks and view logs from anywhere via the web.
- **Zero-Effort Authentication:** Eliminates the need for user interaction or dedicated authentication hardware.
- **Modular and Scalable:** The architecture allows for integration with multiple locks and user accounts across various settings.

The methodology marks a significant step toward practical deployment of zero-effort smart lock authentication systems by combining signal-based proximity detection, machine learning, and full-stack integration in a unified, scalable platform.

Chapter 5

Results and Analysis

The proposed **Wi-Fi Radio-Based Authentication System** was evaluated in a real-world setting using a fully functional setup that included IoT hardware, a central server, and a web-based user interface. The primary objective was to assess the system's ability to authenticate users based on proximity, using machine learning (ML) predictions delivered in real time via a backend API. This evaluation aimed to understand the practical reliability, speed, and usability of the complete system.

5.1 System-Level Evaluation

Unlike earlier offline simulations, this implementation operates in a live, end-to-end feedback loop:

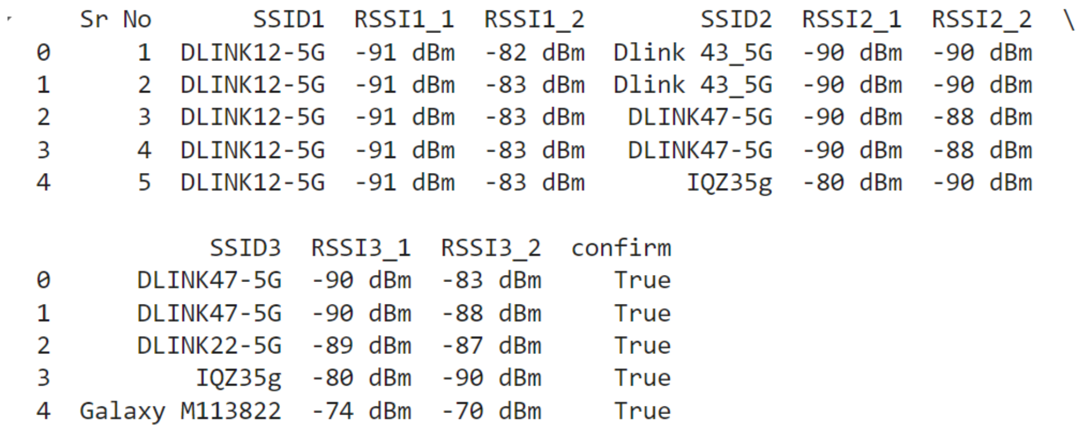
- The smart lock device (Raspberry Pi) scans nearby Wi-Fi signals, capturing characteristics such as RSSI values and beacon metadata.
- These features are transmitted instantly to a central authentication server through a RESTful API.
- The server uses a pre-trained machine learning model to determine whether the detected device is within the trusted proximity threshold.
- Based on the prediction, the server responds to the smart lock, instructing it to either grant or deny access.
- Simultaneously, each action is securely logged and visualized on the user's web dashboard, showing lock status and access history.

This live mechanism enables seamless and secure authentication, eliminating the need for manual intervention or physical credentials.

5.2 Dataset Description

Wi-Fi signal data was collected under realistic and varying conditions using two Raspberry Pi 3 Model B boards one simulating the smart lock, and the other emulating the user's mobile device.

- The dataset includes features such as SSID, BSSID, frequency, and RSSI.
- A total of **250 labeled samples** were collected. Each sample was annotated as either *authorized* (within 7 feet) or *unauthorized* (beyond 7 feet).
- The dataset was divided into 80% training and 20% testing subsets to evaluate generalization performance.



```
   Sr No       SSID1  RSSI1_1  RSSI1_2        SSID2  RSSI2_1  RSSI2_2  \
0      1  DLINK12-5G  -91 dBm  -82 dBm  Dlink 43_5G  -90 dBm  -90 dBm
1      2  DLINK12-5G  -91 dBm  -83 dBm  Dlink 43_5G  -90 dBm  -90 dBm
2      3  DLINK12-5G  -91 dBm  -83 dBm   DLINK47-5G  -90 dBm  -88 dBm
3      4  DLINK12-5G  -91 dBm  -83 dBm   DLINK47-5G  -90 dBm  -88 dBm
4      5  DLINK12-5G  -91 dBm  -83 dBm       IQZ35g  -80 dBm  -90 dBm

            SSID3  RSSI3_1  RSSI3_2  confirm
0      DLINK47-5G  -90 dBm  -83 dBm     True
1      DLINK47-5G  -90 dBm  -88 dBm     True
2      DLINK22-5G  -89 dBm  -87 dBm     True
3          IQZ35g  -80 dBm  -90 dBm     True
4  Galaxy M113822  -74 dBm  -70 dBm     True
```

Figure 5.1: Sample RSSI dataset extracted from top three strongest access points.

5.3 Web Portal Integration and Monitoring

To enhance usability and transparency, a dedicated web portal was developed. This dashboard allows users to:

- Securely log in and manage account credentials.

- Register and configure smart lock devices remotely.
- View detailed, real-time access logs showing authorized and unauthorized attempts.
- Monitor the current lock status (locked or unlocked) in sync with server-side inference.

The portal ensures that users have complete visibility into authentication events and system actions, improving overall trust and control.

5.4 Model Accuracy and Performance

Several machine learning models were evaluated to identify the most effective algorithm for proximity classification. Each model was tested in the live system:

- **Random Forest:** Achieved the highest real-time accuracy of **93%**.
- **Support Vector Machine (SVM)** and **K-Nearest Neighbors (KNN):** Both reached an accuracy of **90%**.
- **Decision Tree (DT):** Scored **89%**.
- **Logistic Regression:** Delivered **86%** accuracy.
- **Gaussian Naive Bayes (GNB):** Performed lowest, with **83%** accuracy.

5.5 Real-Time System Validation

The system was rigorously tested in real deployment conditions to verify its responsiveness and robustness:

- The entire authentication cycle from signal scanning to decision was completed in under **2 seconds**.
- The model maintained consistent accuracy even in varying indoor environments and under signal interference.

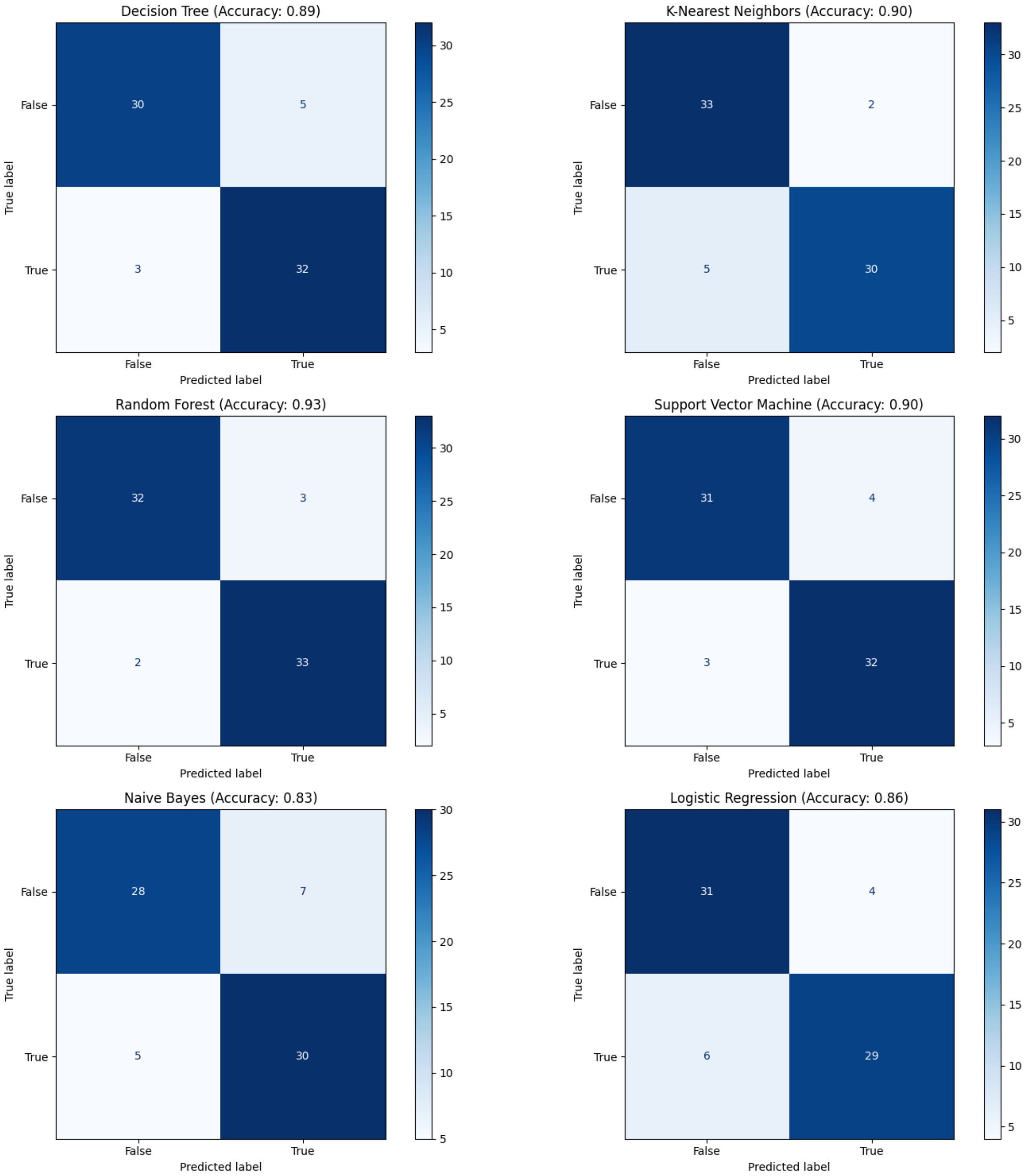


Figure 5.2: Confusion matrices for the evaluated machine learning models.

- All valid access attempts were logged and reflected in real time on the web dashboard, validating system integrity.

These results demonstrate that the system is not only technically sound but also suitable for practical, user-facing deployment.

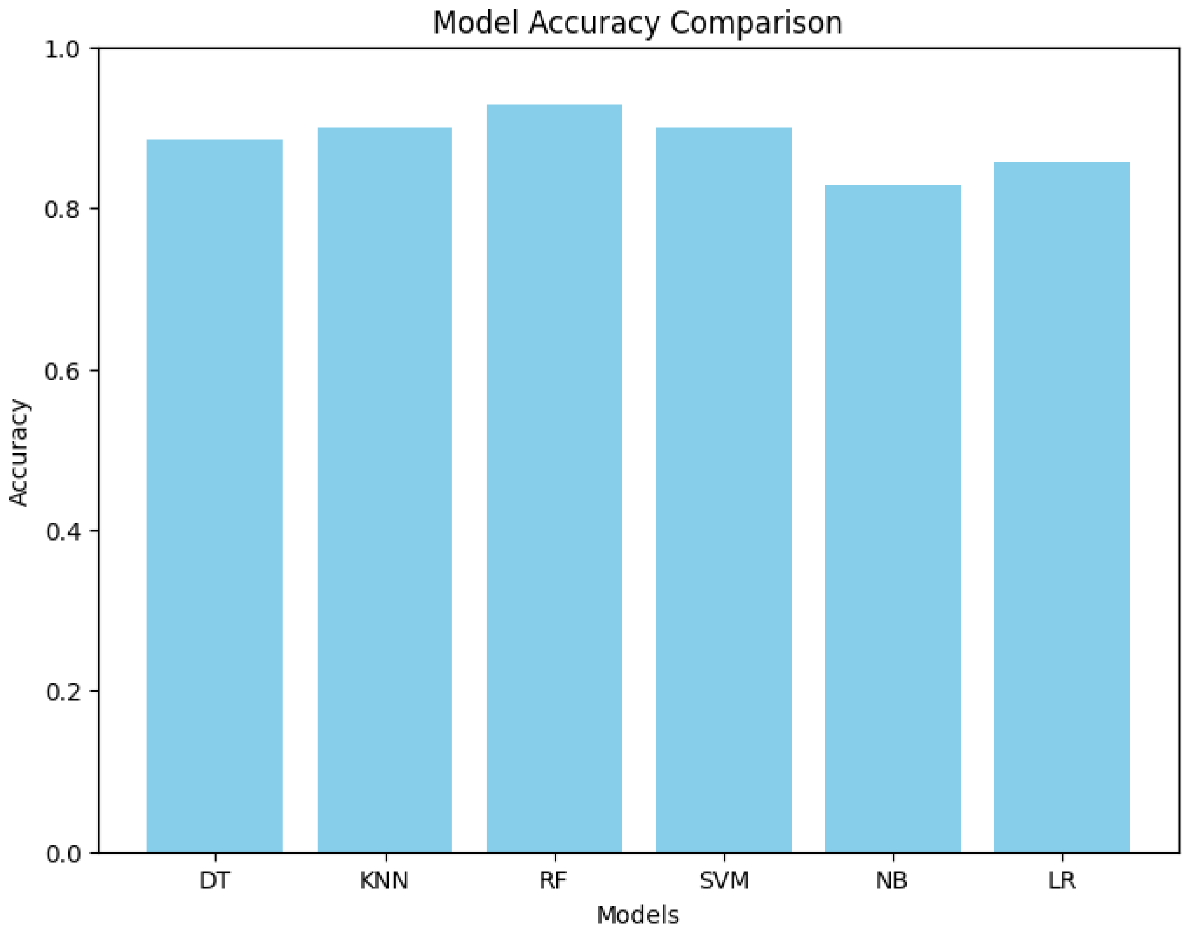


Figure 5.3: Comparison of accuracy across machine learning models.

5.6 Summary

The real-time results confirm that the proposed smart lock system, integrated with server-side machine learning and a user-friendly dashboard, is both reliable and scalable. With a maximum observed accuracy of **93%** using the Random Forest classifier, the system can effectively distinguish proximity using Wi-Fi signal data. Its responsiveness and ease of use mark it as a strong candidate for deployment in secure, IoT-based access control environments.

Chapter 6

Conclusion and Future Work

As IoT devices become more embedded in our everyday lives, the need for secure, seamless, and user-friendly authentication mechanisms has never been more critical. This project introduced a real-time, end-to-end **Wi-Fi Radio-Based Authentication Scheme for Smart Lock IoT Devices**, which uses ambient Wi-Fi signal characteristics, such as RSSI values, combined with machine learning to perform zero effort proximity-based authentication.

6.1 Conclusion

The system architecture brought together Raspberry Pi-based hardware, a centralized server hosting trained ML models, and a secure web dashboard for users. Wi-Fi signal data captured by the smart lock device was transmitted to the server in real time, where it was processed to determine access eligibility. The accompanying dashboard allowed users to remotely view access logs and monitor lock status, ensuring transparency and control over system behavior.

Evaluation results confirmed the effectiveness of this approach, with the **Random Forest model achieving an accuracy of 93%** in real-time deployment scenarios. The system was able to consistently classify users as authorized or unauthorized based on their proximity, with fast response times and minimal hardware requirements. This validates the system's suitability for real-world applications, especially in home and small-scale commercial environments.

By removing the need for manual user input or specialized biometric hardware, the proposed solution offers a low-cost, scalable, and convenient alternative to traditional authentication systems.

6.2 Future Work

While the prototype meets its current objectives, there are several promising directions for future development that can enhance functionality, scalability, and resilience:

- **Wider Real-World Deployment:** Testing the system in varied real-life environments—such as homes, offices, and shared spaces—would help assess robustness under dynamic conditions, including interference, multi-user scenarios, and environmental noise.
- **Expanded Dataset and Improved Generalization:** Collecting a larger and more diverse dataset across different times, devices, and locations would help improve model performance and reduce false positives or negatives, especially in edge cases.
- **Support for Multi-User and Multi-Lock Environments:** Extending the architecture to handle multiple users and smart locks simultaneously would allow shared access management, group permissions, and more flexible control schemes.
- **Edge-Based Inference:** Deploying lightweight versions of the ML models (e.g., via TensorFlow Lite) directly on the Raspberry Pi or similar edge devices would reduce latency, improve responsiveness, and minimize reliance on constant server communication.
- **Energy and Network Efficiency:** Investigating low-power communication protocols (e.g., MQTT), optimizing scan intervals, and enabling sleep cycles could significantly enhance battery life and sustainability of the system.
- **Enhanced Security Layers:** To strengthen the system against spoofing or unauthorized physical access, additional authentication factors—such as motion detection, facial recognition, or voice commands—could be integrated.
- **Polished Hardware Prototype and Mobile App:** Developing a user-ready prototype with a companion mobile application for setup, pairing, unlocking, and real-time notifications would help bridge the gap between research and consumer adoption.

By addressing these areas, the system can be transformed from a research prototype into a mature, market-ready product. Continued innovation in edge AI, signal processing, and IoT security will play a pivotal role in making proximity-based authentication not only viable but mainstream in the era of smart environments.

Bibliography

[1] A. A. S. AlQahtani et al. Zero-effort two-factor authentication using wi-fi radio wave transmission and machine learning. 2023.

[2] A. A. S. AlQahtani et al. Leveraging machine learning for wi-fi-based environmental continuous two-factor authentication. 2024.

[3] Reem Alrawili, Ali Abdullah S. AlQahtani, and Muhammad Khurram Khan. Comprehensive survey: Biometric user authentication application, evaluation, and discussion. *Computers and Electrical Engineering*, 2024.

[4] H. F. Atlam, Ezz El-Din Hemdan, et al. Framework for iot data collection and forensics with analysis of smart home devices.

[5] D. R. Bhuva and S. Kumar. Continuous authentication using ecg and emg biometric signals in iot devices.

[6] Aswini D, Rohindh R, Mridula C S, and Manoj Ragavendhara K S. Smart door locking system.

[7] Kaustubh Dhondge, Kaushik Ayinala, Baek-Young Choi, and Sejun Song. Infrared optical wireless communication for smart door locks using smartphones.

[8] Sandeep Gupta, Attaullah Buriro, and Bruno Crispo. Smarthandle: A novel behavioral biometric-based authentication scheme for smart lock systems.

[9] Naresh Kumar Masurkar and Priyanshu Pandey. Nfc-based biometric dual authentication access control system.

[10] Sarah Delgado Rodriguez, Sarah Prange, et al. Act2auth – a novel authentication concept based on embedded tangible interaction at desks.

[11] Shubham Saloni and Achyut Hegde. Wifi-aware as a connectivity solution for iot: Pairing iot with wifi aware technology to enable new proximity-based services. In *Proceedings of the International Conference on IoT Technologies*.

[12] Aditya Saroha, Anant Gupta, et al. Biometric authentication based automated, secure, and smart iot door lock system.

[13] Muhammad Shahzad and Munindar P. Singh. Continuous authentication and authorization for the internet of things. *IEEE Internet Computing*, 2017.