Design and Evaluation of a Biometric IoT-Based Smart Lock System with Real-Time Monitoring and Alert Mechanisms

Jamil Abedalrahim Jamil Alsayaydeh[1]*, Mohd Faizal Yusof[2],
Serhij Mamchenko[3], Rostam Affendi Hamzah[4], Safarudin Gazali Herawan[5]
Department of Engineering Technology-Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK),
Universiti Teknikal Malaysia Melaka (UTeM), 76100 Melaka, Malaysia[1, 4]
Department-Research Section-Faculty of Resilience, Rabdan Academy,
65 Al Inshirah Street, Abu Dhabi 22401, United Arab Emirates[2]
Department of Computer Systems-Networks and Cybersecurity,
National University of Life and Environmental Sciences of Ukraine, Kiev, Ukraine[3]
Industrial Engineering Department-Faculty of Engineering, Bina Nusantara University, Jakarta, Indonesia 11480[5]

***Abstract*—A smart door lock system based on IoT is presented which uses fingerprint biometric authentication, ESP32 microcontroller and Blynk IoT platform to provide a secure, user friendly and remote controllable access control solution. The proposed architecture replaces traditional locks with a real time biometric system that gives instant feedback through onboard display (OLED) and buzzer and remote monitoring and control through a mobile app. A new fail-safe mechanism is implemented: after 3 failed fingerprint attempts the system will lock out for 15 seconds and send instant alert to the authorized user's smartphone. Performance test of the prototype shows fingerprint recognition time of around 1.0 second and door unlock time of 5 seconds, so it's convenient to use. The system has a very low False Acceptance Rate (FAR) of 1.32% which means strong resistance to unauthorized access. The False Rejection Rate (FRR) is higher (around 26.32%) due to user error such as improper finger placement – so a usability issue to be addressed. The device can store up to 3 fingerprint profiles and gives visual/audible alert for all access events. This integration of IoT with biometric security not only enhances physical security but also user convenience, a modern smart-lock solution for smart home automation.**

Keywords*—Smart door lock; Internet of Things (IoT); biometric authentication; fingerprint sensor; ESP32 microcontroller; Blynk IoT platform; access control; cybersecurity; real-time monitoring; home automation; OLED display; False Rejection Rate (FRR); False Acceptance Rate (FAR); remote monitoring; smart home automation

I. INTRODUCTION

The Internet of Things (IoT) has changed the physical security landscape, bringing intelligent systems that automate tasks and enable remote management, monitoring and interaction. Access control has shifted from traditional methods—mechanical keys, RFID cards or PIN codes—to more secure, automated and user friendly alternatives. Traditional lock-and-key or keypad systems are vulnerable to theft, key duplication, password disclosure and human error and are no longer sufficient for modern security expectations. Among the emerging solutions, biometric fingerprint recognition stands out for its reliability and uniqueness. Unlike passwords or cards that can be shared, lost or guessed, fingerprints are a permanent and non-transferable form of authentication. When integrated into an IoT ecosystem using microcontrollers (like ESP32) and cloud connected platforms like Blynk for remote communication, biometric systems provide a real-time approach to access control. Users can lock/unlock doors, receive alerts and log entry events from anywhere in the world, improving security and convenience. But IoT enabled smart locks still have many challenges. False rejections—where an authorized user is denied access due to partial or poor fingerprint scans—pose usability issues and frustration. Security flaws are also present in many existing systems, including tamperability, lack of intrusion detection or alarm features, limited fingerprint storage capacity and no real-time incident response. Environmental factors like humidity, sensor degradation over time or incorrect finger placement can also degrade performance and make it inconsistent. Many smart locks today rely on continuous internet connectivity or only one layer of authentication, making them fragile; a network outage or a smart intruder can compromise such systems. In case of unauthorized access attempts or brute force attacks, most commercial smart locks don't respond dynamically or notify the owner, which can lead to severe security breaches—especially in unattended or remote installations. Addressing these limitations, this research introduces an enhanced IoT based smart door lock architecture that integrates biometric fingerprint scanning, real-time wireless communication and intelligent security feedback mechanisms. The goal is to have a system that not only does accurate fingerprint authentication but also reacts intelligently to suspicious or unauthorized access attempts—for example, by triggering alerts, activating temporary lockout timers and logging access events for audit. The design prioritizes both security and usability so legitimate users can enter quickly while potential intruders are deterred and detected. Smart locks are often found wanting when it comes to comprehensive security handling. Many rely too heavily on stable network conditions or simple one-factor authentication without a backup plan. If that network goes down-or if someone keeps trying to pick the lock-these systems often don't adapt or alert the user in real time.

That's where the limitations of biometric functionality in commercial locks come in: often added as an afterthought rather than optimized for low-power embedded environments or user-friendly maintenance. That can lead to frequent false rejections and a poor user experience, particularly in places where many people are using the lock or where traffic is high. Devices struggle with limited processing resources and suboptimal fingerprint enrollment processes. Our goal is to close those gaps with a smart door lock that balances security with usability.

We're building a lightweight, IoT-integrated smart door lock around the Espressif ESP32 microcontroller. This chip gives us the dual-core processing power and built-in Wi-Fi/Bluetooth capabilities we need to handle fingerprint recognition tasks and secure communication with a cloud backend (we're using the Blynk platform). We want users to know right away if there's a critical event, so we're designing the system to notify them instantly. If someone tries to get in three times and fails, the lock will be out of commission for 15 seconds. We'll also display that information on an OLED screen and with a buzzer. That way, users can make quick decisions and have responsive control in real-world conditions.

Our main contributions are:

1) A fully functional smart door lock prototype that integrates fingerprint recognition with an ESP32 microcontroller and the Blynk IoT platform. This lets us do real-time door access monitoring, remote lock/unlock control via smartphone and automatic event logging.

2) A multi-layer security response mechanism that includes a 15-second lockout after three failed fingerprint attempts, audible alerts and on-screen warnings during a breach attempt, and push notifications to the authorized user's phone. This feedback loop makes intrusion detection and response much better than with conventional locks.

3) A comprehensive performance evaluation that shows our fingerprint recognition time is about 1 second and our False Acceptance Rate is just 1.32%. The False Rejection Rate was 26.32%, mainly because of user error (about 65.8% of false rejections were due to incorrect finger placement or similar user issues). That highlights both the strengths (strong security against false accepts) and the limitations (usability challenges causing false rejects) of our system.

4) Various features to improve user experience and system usability such as on-screen feedback for every operation, auto-unlock for 5 seconds after a successful fingerprint match (for quick entry before re-locking), and direct mobile app for fingerprint enrollment and deletion. The mobile app allows authorized administrators to manage user fingerprints remotely, making it more flexible and easier to maintain.

5) Deployment and testing of the prototype in real scenarios (e.g. home/office door setup) to test the system under real world conditions. The smart lock is suitable for residential, commercial and rental property use cases, and works well with different network conditions and user behavior.

The remainder of this paper is organized as follows: Section II presents a comprehensive literature review covering smart lock technologies, biometric authentication methods, and IoT-based access control systems. Section III details the methodology, including system planning, hardware and software integration, and deployment procedures. Section IV presents the experimental results, while Section V discusses the findings in detail. Section VI outlines the conclusions drawn from the study, and Section VII highlights the limitations and proposes potential directions for future work.

II. Literature Review

A. Evolution of Smart Door Lock Technologies

Smart door lock systems have evolved with the integration of biometric technologies and IoT connectivity. Fingerprint based authentication has become highly secure and convenient as fingerprints are unique and can't be duplicated [1]. Many studies have explored combining fingerprint sensors with other technologies (RFID tags, keypad entry, cloud connectivity) to make door access systems more secure and functional [2].

B. Role of IoT in Access Control Systems

IoT capable microcontroller platforms (Arduino, ESP8266/NodeMCU, ESP32) allows seamless integration with Wi-Fi networks, enabling remote monitoring, real time alerts and smartphone control of locks [3]. This makes it more convenient for users and cost effective for both home and commercial use [4]. To counter spoofing attacks and sensor vulnerabilities in biometric locks, researchers have proposed techniques such as reducing residual fingerprint traces on scanners and image processing for fingerprint enhancement [5].

C. Advancements in Multi-modal Authentication

Also, the digital access control realm is moving towards multi-modal and keyless systems to address various intrusion scenarios (e.g. unauthorized guests or lost credentials) [6][7]. For example, some designs combine fingerprint verification with RFID or PIN code entry to provide backup authentication and robustness [8]. The growing sophistication of cyber-physical threats – such as relay attacks that can clone RFID or Bluetooth signals – further emphasizes the need for biometric-enabled access control systems that are hard to bypass [9].

D. Representative Works and Technological Contributions

Over the last decade, many researchers have focused on designing IoT-based smart door locks that incorporate fingerprint recognition to meet the increasing demand for secure yet user-friendly access control [10]. The literature shows a clear trend away from traditional mechanical locks towards integrated digital systems that combine sensors, wireless communication and mobile platforms for more functionality and control. Below we summarize some of the representative works and their approaches, and highlight the research gap that our work addresses.

Tailor and Pandya [6] developed an IoT-based smart security system mainly using a fingerprint sensor for family members' door access. The system allowed users to unlock doors remotely through a smartphone app, making it very convenient and secure. By enabling mobile connectivity, this approach allowed homeowners to monitor and control door access in real time from anywhere – a great feature if physical keys were lost or traditional PINs compromised. This work [6] laid the foundation for merging biometric verification with cloud-based remote control, which is the basis of modern smart home access systems that emphasize user mobility.

Al-Shareefi et al., introduced a low-cost fingerprint-based door lock using an Arduino Nano microcontroller and a simple servo motor for the locking mechanism [11]. This system did not have remote network connectivity or advanced features like image processing or multi-factor authentication; however, it showed that biometric security can be built with very low-cost hardware. The contribution is significant in showing economic accessibility, making smart lock technology more accessible to small businesses or residential users in resource-constrained environments. The trade-off for its simplicity was limited functionality but it highlights the importance of low-cost design in wider adoption.

Jeong aimed to improve the security and accuracy of fingerprint-based locks by reducing the residual fingerprint left on the scanner surface (which can be used in spoofing attacks). This work [5] applied image processing to improve fingerprint matching reliability and reduce the False Acceptance Rate (FAR) of the system. This is particularly important in high-security environments (e.g. finance or research labs) where even a small biometric vulnerability can be catastrophic. Jeong's work shows how both hardware and software improvements in biometric systems (e.g. better sensor surface design and algorithmic liveness detection) can mitigate physical security threats.

Rakib et al. looked at integrating fingerprint recognition into a home automation context, extending biometric access beyond just door locking [12]. Their system combined a fingerprint lock with other IoT modules to control home appliances and security devices. For example, one fingerprint verification could unlock the door, turn on lights or disable an alarm system [13]. This shows how biometric systems can be used in smart environments and how fingerprints can be the foundation of smart living spaces. It shows how an access control system can be integrated with general IoT based home management for more convenience [14].

Tambunan et al., presented a multi-modal smart home security prototype that incorporated fingerprint authentication alongside RFID and keypad entry options for flexible access control. This design [15] enhanced usability by allowing different modes of entry based on user roles or scenarios (for instance, family members use fingerprints, temporary guests use a PIN code, maintenance staff use an RFID card, etc.). The system also featured remote monitoring, sending alerts and enabling control through a network interface. Notably, Tambunan's work [16] is recognized for its scalability and inclusivity, accommodating various user preferences and security levels within one unified platform.

E. Trends and Research Challenges

These studies show the evolving landscape of smart door locking systems with the convergence of biometric authentication, IoT technology and user centered design. A common trend across the reviewed literature is the use of mobile platforms, which allows users to monitor and control door access remotely through their smartphones for more convenience and responsiveness [17]-[20]. Another trend is the emphasis on affordability and simplicity, with most of the hardware and software being cost effective and widely available like Arduino, so these systems can be used by a wider audience [21]-[25]. Researchers have also improved system security by applying image processing techniques to minimize residual fingerprint traces and reduce the risk of spoofing [26]. Fingerprint authentication has also been extended to comprehensive home automation systems where access control is integrated with general IoT based device management, so users can interact with lights, alarms and other appliances in one ecosystem [27]-[29]. Multi-modal authentication methods which combines fingerprint sensor with RFID and keypad modules is also a more flexible and robust solution that can accommodate different user scenarios while maintaining strong security protocols [30]. Overall, these advancements show the potential of IoT-biometric systems to change residential and commercial access control in the digital age [31], [32].

F. Research Gap

Despite all these advancements, there are still gaps and challenges in current smart lock solutions. Many systems lack adaptive intrusion response mechanisms or real-time user alerting; for example, if an intruder tries multiple failed attempts, not all systems will auto lock out or notify the owner [33], [34]. Dynamic user management (e.g. easy remote enrollment/revocation of access) and integration with advanced analytics or cloud services are also missing. In short, no existing solution in the literature combines real-time biometric verification with immediate intrusion countermeasures (like auto lockouts and instant notifications) and a scalable IoT architecture that can be extended to larger deployments [35]. This research fills that gap by proposing a smart lock system that combines all these features in an embedded design. Our system emphasizes security responses (through its lockout and alert mechanism) and remote IoT connectivity and user friendly management, different from previous works.

Table I shows several existing smart lock approaches and their features and limitations compared to the proposed system.

TABLE I. COMPARISON OF EXISTING AND PROPOSED SMART LOCK APPROACHES

References	Approach	Description	Limitation
[16]	RFID + IoT	Uses RFID tags and Raspberry Pi for secure, remotely accessible locking via smartphone platforms.	Dependent on internet connection for cloud access and real-time updates.
[35]	Bluetooth + IoT	Utilizes Bluetooth MAC addresses from smartphones for hands-free door unlocking; Raspberry Pi controls access via a mobile app.	Limited operational range due to Bluetooth constraints.
[36]	NFC + Arduino + GSM	Employs NFC readers and smartphones for secure, short-range access control with user token-based differentiation.	Vulnerable to security risks if the user's smartphone is lost or stolen.
[7]	Biometric + ESP32-CAM + Blynk	Combines fingerprint recognition with real-time video capture using ESP32-CAM and Blynk app for access logging and alerts.	Camera increases power demand; low-light image quality limitations.
[3]	RFID + Fingerprint + Keypad	A hybrid system using three types of authentication to improve accuracy and flexibility in multi-user environments.	Hardware complexity and increased system cost.
[4]	Embedded Linux + Fingerprint	Utilizes embedded Linux system for managing door lock control and fingerprint authentication.	Higher resource requirements; not ideal for low-power setups.
[6]	Fingerprint + IoT (ESP32 + Blynk)	Provides real-time door lock control and fingerprint verification with cloud-based monitoring.	Network latency and partial fingerprint scans may lead to false rejections.
[37]	OTP + IoT	Generates and delivers OTP to user's device for one-time secure access, suitable for temporary or guest entry.	Delayed OTP delivery in weak networks; requires mobile signal or internet.
The proposed work	Fingerprint + ESP32 + Blynk + OLED	Integrates fingerprint authentication with ESP32 microcontroller and Blynk for real-time control, alerts, auto-lockout, and mobile enrollment/deletion capabilities.	False Rejection Rate (FRR) of 26.32% mainly due to user errors; limited to 3 fingerprints; Wi-Fi needed.

III. PROPOSED METHOD

The proposed method is to develop an IoT enabled access control system using biometric authentication. The system "Smart Lock Door with Fingerprint using ESP32" is based on ESP32 microcontroller and has a fingerprint sensor for primary user authentication and remote monitoring/control through Blynk mobile app. The overall methodology follows a structured engineering process of planning, hardware/software integration, testing and deployment.

The proposed method was chosen to specifically address key limitations observed in existing smart lock systems. Most prior works either lacked real-time intrusion response mechanisms, did not support remote fingerprint enrollment and deletion, or were dependent on continuous internet connectivity without fallback. Additionally, many systems failed to integrate meaningful feedback mechanisms (e.g., visual/audio alerts) or suffered from limited fingerprint capacity without a user-friendly management interface. Our proposed system overcomes these limitations by combining fingerprint-based biometric access with IoT features using ESP32 and Blynk, providing real-time alerts, local display and buzzer feedback, mobile app control, and a security lockout mechanism. This makes it suitable for deployment in residential or small commercial environments where usability, security, and remote operability are all essential.

Fig. 1 shows the system flowchart, from system initialization to enrollment, verification and response. Upon boot-up, the system initializes all modules and enrolls or deletes fingerprints as commanded. During normal operation, when a user presents a finger, the fingerprint sensor captures the input and ESP32 compares it with stored templates. Successful scan triggers unlock sequence: solenoid lock is activated (door opens), buzzer produces a confirmation tone, OLED display shows a message and a push notification (through Blynk) is sent to the user. The door remains open for 5 seconds and then automatically locks back. For failed scans, the system gives immediate feedback (OLED message and buzzer alert). After 3 consecutive failed attempts, an intrusion lockout mechanism is engaged: the system denies any further attempts for 15 seconds and during this time it flashes an alert on OLED and sends an urgent notification to the owner's phone. This cooldown period helps to prevent brute force attacks and warns the legitimate user of potential unauthorized access.

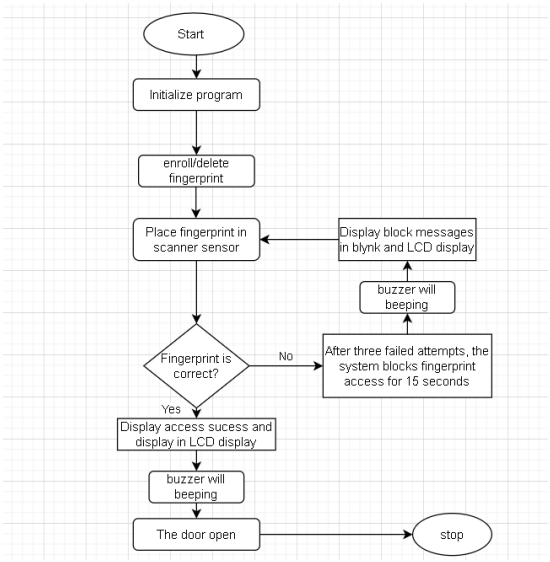

Fig. 1. Flowchart of the system.

A. Phases of Development

The project was carried out in multiple phases, each addressing different aspects of the system:

1) Initial planning and requirements analysis: In this phase, we defined the objectives and scope of the smart lock system. Main goals were to increase physical security over traditional locks and to be able to remotely manage access. We gathered functional requirements (what the system should do) and hardware requirements (what components we need). Key hardware for the prototype were ESP32 development boards

(for connectivity and control), R305 optical fingerprint sensor (for biometric input), solenoid lock mechanism (for door locking/unlocking), small OLED display for on-device messages, buzzer for audio alerts, 5V relay module (to drive the high-current solenoid), portable power sources (2 x 9V batteries with holders) and wiring and connectors. Software requirements were to choose a suitable programming environment (Arduino IDE) and IoT platform (Blynk) and to make sure all libraries and modules are compatible.

2) Hardware and software integration: This phase we built the system by connecting all components and developing the firmware. The hardware setup started by placing the modules on a prototyping board: R305 fingerprint sensor was wired to ESP32's serial interface for biometric input, 12V solenoid lock was connected through a relay (ESP32 controls the relay to unlock/lock), ESP32 Dev Board was the central controller for all signals. An OLED display module was added to show status messages (e.g. "Place Finger" prompt, "Access Granted"/"Denied" results, etc.) and a small buzzer was attached for audio feedback. The wiring was done carefully using a breadboard and jumper wires to ensure stable connections. A detailed circuit schematic was drawn to map all connections (Fig. 2 shows the system's block diagram). On software side, ESP32 was programmed in C++ using Arduino IDE. Firmware routines were written for fingerprint enrollment, fingerprint matching (verification) and for controlling the solenoid via the relay. The code also handled sending and receiving data from Blynk cloud (over Wi-Fi) to support remote notifications and commands. In parallel, a simple mobile app interface was created using Flutter, to allow authorized users to enroll a new fingerprint or delete an existing one through their smartphone – this interface communicates with ESP32 through Blynk cloud API [25]. By the end of this integration phase, the hardware and firmware were working together: a fingerprint could be placed and recognized, the lock would trigger and send updates to the mobile app.

3) System development and testing: With the prototype assembled, extensive testing was done to fine tune performance and validate reliability. The hardware was first assembled on a breadboard for easy modification during testing. Key things tested included fingerprint registration and matching accuracy, door actuation (unlock/lock response times) and alert mechanisms. The fingerprint sensor was tested under various conditions (clean vs smudged sensor surface, correct vs off-angle finger placement) to see how it handles partial or imperfect scans. The system's error handling features were also verified: the auto lockout after 3 incorrect attempts was triggered manually to make sure it worked as expected and the corresponding user notifications were checked on the smartphone. Throughout this phase, iterations were made — for example, adjusting the fingerprint matching threshold to balance sensitivity vs false rejections and optimizing the microcontroller code to reduce any latency in communication with the Blynk platform.

4) Final deployment: After testing was complete, the hardware was moved from the prototyping breadboard to a more permanent setup for real world demonstration. The components were mounted on a mock door frame to simulate a real door installation. The fingerprint sensor and OLED were placed for user access and the solenoid was installed to control a latch bolt as it would in a real door. Power was supplied either via a USB power bank or the dual 9V battery pack for portability. A quick user training was done for those interacting with the lock, explaining how to place fingerprints correctly and how to use the mobile app for remote operations. The final system was tested in a simulated home-entry scenario and it worked as expected. Also, notes were taken for maintenance (checking battery levels, cleaning the fingerprint scanner for optimal performance, keeping firmware updated) to ensure long term reliability and security.

B. System Design and Overview

The IoT enabled smart lock system consists of a combination of hardware and software components to provide secure, efficient and remote controlled access. On the hardware side (see Fig. 2 for the system architecture diagram), the main components are:

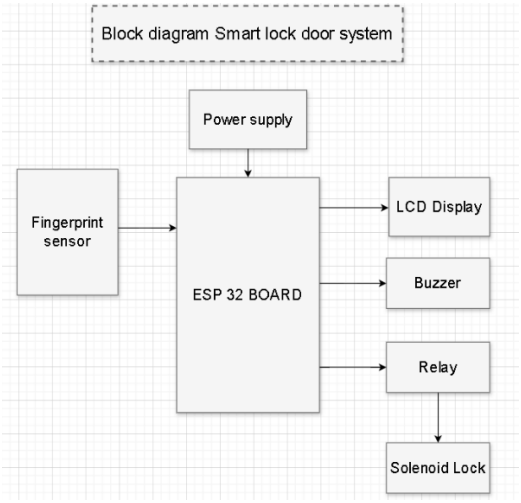

Fig. 2. Block diagram of the smart lock system.

- A biometric fingerprint sensor (model R305 optical sensor) which is the primary means of user authentication by capturing fingerprint images and converting them into digital templates.
- A solenoid lock mechanism that controls the door latch (energizing the solenoid opens the lock, de-energizing locks the door).
- A 16×2 character LCD or 0.96-inch OLED for user interface and feedback – in our prototype we used a small OLED to show status messages (e.g. "Initializing...", "Fingerprint Accepted", "Access Denied", etc.) but a larger 16×2 LCD can be used to display multi-line messages for better readability.

- An ESP32 Dev Board which is the brain of the system, chosen for its dual core and Wi-Fi (and Bluetooth) capabilities for IoT applications. The ESP32 handles fingerprint sensor input, runs the matching algorithm, controls the solenoid via output signals and communicates with the remote app.
- A 5V active buzzer that provides audible alerts (e.g. a quick beep for success, a longer tone for failure).
- A 5V relay module that acts as a switch to drive the higher voltage solenoid lock – the relay isolates the ESP32 (which runs at 3.3V logic) from the 12V required by the lock and allows the microcontroller to safely control the lock circuit.
- A power supply solution which can be either USB (for wired or power bank) or battery pack. In our portable setup we used 2 x 9V batteries in series to provide the ~12V required by the solenoid and the ESP32 board was powered through its Vin pin (with onboard regulators stepping it down to 3.3V).
- The various jumper wires (male-to-male and female-to-female) to connect components in a modular way.

All components are connected as per the schematic so that the ESP32 can read data from the fingerprint sensor (via UART), toggle the relay (via a GPIO output pin) to unlock the door, drive the buzzer (via another GPIO) and update the display (using I²C for the OLED or LCD). The hardware design is modular and low power consumption is important for a door-lock device that might be battery operated. On the software side the system's firmware is written in C++ using the Arduino framework for ESP32. This provides a convenient environment with libraries for the fingerprint sensor (for enrolling and matching fingerprints), for the display and for Wi-Fi. The Blynk IoT platform is used to enable remote access and notifications: the ESP32 runs the Blynk client which maintains a secure connection to the Blynk cloud server via Wi-Fi. Through the Blynk mobile app the user can send commands (like enroll or delete a fingerprint, or unlock the door remotely) and receive real-time notifications (for events like a door unlock or a failed authentication attempt). The system also has a custom Flutter based mobile application that interacts with the Blynk API to provide a more specialized interface for managing fingerprints (e.g. naming fingerprint entries, issuing a delete-all command) [38]. This dual-app approach (Blynk console plus a tailored Flutter app) provides more user interaction and control flexibility. Overall the tight integration of hardware and software components results in a complete, responsive and secure smart lock solution for modern access control.

C. System Configuration Process

The ESP32 gets input from the fingerprint sensor. When recognised it triggers a relay that powers the solenoid lock to open the door. Also the OLED display shows visual feedback and an alert is pushed to the user via Blynk app. On a failed attempt the system increments a fail counter and after 3 fails it starts a lockout timer and sends an intrusion alert.

In terms of system configuration the ESP32 firmware is set to run multiple tasks in parallel. It continuously listens for fingerprint input and network commands. The program flow (as shown in Fig. 3) starts with initialization of all sensors, display and Blynk connection. Then it enters an idle loop waiting for either a fingerprint scan or a user command from the app. Fingerprint enrollment mode can be triggered via the mobile app or a hardware input: in this mode the system will prompt the user (on the OLED) to scan a new finger twice. It will then process those scans and create a fingerprint template and store it in flash memory if the quality is good and if there are available slots. Fingerprint deletion can be done for one fingerprint or all, either via the app or a specific sequence on the device. These features allow the system to be managed easily when a user needs to be replaced or if the device memory is full. For normal authentication whenever a finger is scanned the system will instantly compare the input to all stored templates. If a match is found the ESP32 will trigger the relay to open the door and immediately notify the user (OLED message "Unlocking…" and a push notification via Blynk). The door will remain open for a set time (5 seconds) before the relay is switched off to re-lock automatically, with an OLED message "Door Locked". If no match is found (i.e. the fingerprint is not recognized) the OLED will display "Access Denied" and the buzzer will sound. The system will count consecutive fails; upon the third fail it will trigger the security lockdown: the ESP32 will ignore further fingerprint inputs for 15 seconds and display "LOCKOUT – Too Many Failed Attempts" on the screen, while sending an alert notification. After the cooldown the system will reset the fail count and go back to normal operation. This security policy balances convenience with protection, so a legitimate user who had two bad scans can still try again after a short pause, but an attacker can't brute force multiple attempts quickly.

Fig. 3. Solenoid lock.

D. Hardware Specification

Hardware specification tells us about the details and technological value of the physical components of the computer or electronic device; and this is useful for defining performance rates, compatible operations and possible uses of the device. These specification may include details about the type of processor or central processing unit (CPU), available memory or random-access memory (RAM), storage capacity, graphics card or graphics processing unit (GPU), networking options such as Universal Serial Bus (USB), High-Definition Multimedia Interface (HDMI) and/or wireless options, and the power supply of the computer. Specific information about the components meet the needs of the users and help the developers to decide if the device is suitable for gaming, handling big data, or media production, and also ensure that the software and peripherals are compatible. Information about the possibility of upgrade and expansion of the device in hand enables the buyer to make informed decision and to configure the system properly.

1) Fingerprint sensor: The fingerprint sensor in this system works via UART communication and capture, match and store biometric data. It has six primary pin connections, each serves a specific function to ensure seamless integration with the microcontroller. The red wire (VCC) supply 3.3V DC power to the sensor, the black wire (GND) is the ground connection. Data transmission between the sensor and the ESP32 microcontroller is through the yellow (TXD – data output) and green (RXD – data input) wires. The blue wire (WAKEUP) is for generating wake signal to the sensor when finger is present, to conserve energy during idle state. The white wire (3.3VT) is for inductive power input, to support the sensor wake-on-touch feature to further improve energy efficiency. Overall the sensor is designed to provide accurate fingerprint recognition with low power consumption, suitable for IoT based smart lock system.

2) Solenoid lock: The solenoid lock is an electro-mechanical actuator; when current pass through its coil, a magnetic field is generated, pulling in a metal plunger to either engage or disengage the locking mechanism. This allows the door to be locked or unlocked by the microcontroller without manual key intervention. Our system's solenoid is 12V and can retract or extend the bolt within few milliseconds of activation.

Fig. 3 is the solenoid lock component used in the smart lock system. A solenoid lock is an electromechanical actuator that works by energizing a coil to generate a magnetic field. This field moves a metal plunger to engage or disengage the locking mechanism, to control door access. When integrated with an IoT based smart lock system, the solenoid lock can be controlled remotely by the microcontroller (ESP32) to automate access control. The solenoid lock is well-suited to applications that require both speed and security. That's because it responds quickly, is durable and has reliable electronic control. You can also integrate it into wireless smart systems, making it a good fit for real-time access control scenarios. But to ensure that works smoothly, you need a consistent power supply and some safety mechanisms to prevent unauthorized access or accidental lock failure. In this project, we integrated a genuine solenoid door lock with an ESP32 microcontroller [39]. That lock is opened by the fingerprint sensor after it authenticates the biometric data. The ESP32 then triggers the solenoid lock to open the door. We programmed the firmware for that operation using the Arduino IDE-and you can monitor the system remotely through the Blynk app. We powered the system with a stable energy source-a 9V battery or power bank-to ensure it keeps performing uninterrupted. That configuration offers several advantages. Speed, for one. Solenoid locks respond quickly, which is ideal for situations where you need fast access. Electronic control lets them be seamlessly integrated into smart environments for remote operation. And reliability means they're robust and long-lasting. Together, those attributes make the smart lock system more effective and convenient for users by eliminating the need for traditional keys and providing secure, real-time access management.

3) LCD Display: One feature we considered adding to our system is an LCD display. A 16x2 character LCD is an alphanumeric display that can show two lines of 16 characters each. It's based on the HD44780 driver, which makes it easy to interface with microcontrollers. We actually used a smaller OLED display in our current prototype because of size constraints, but a module like this 16x2 LCD would be great for displaying more detailed status messages or instructions in the future.

The 16×2 LCD has 16 pins, each with a specific function. Vcc and GND are power supply inputs, Vo is for contrast adjustment (controlled by a potentiometer), D0 to D7 are data lines, and RS (Register Select), RW (Read/Write) and Enable are control lines. RS toggles between command and data mode, RW determines read or write operation and Enable initiates data transfer. In IoT based smart lock systems, 16×2 LCD is a good visual output module. It displays system states such as authentication success or failure and provides immediate user feedback. Its ease of integration, affordability and reliability in displaying real time data makes it a preferred choice for many embedded system developers.

4) Additional components: To ensure efficient and secure operation of IoT enabled smart lock system, several hardware components are used. Battery holder has dual 9V batteries, a reliable and portable power supply for uninterrupted system operation. Buzzer is an alert mechanism, it notifies user of successful access or intrusion attempt. 9V batteries are the primary power source, portable and easy to replace. Jumper wires are used to connect different modules and components, to make circuit integration seamless. ESP32 development board is the central processing unit, it handles data processing, communication with fingerprint sensor and Blynk app. USB to Micro cable is for alternative power input from external sources like power banks, for system flexibility and backup. 5V relay module controls the solenoid lock, it receives trigger signal from ESP32 and switches the lock accordingly, so physical access is controlled based on verified biometric input. ESP32 Dev Board has 34 GPIO pins, analog inputs and capacitive touch inputs and supports multiple protocols (UART, I²C, SPI). Fig. 4 shows the pinout and capabilities of ESP32 used. Many pins can be configured with internal pull-up or pull-down resistors and the chip supports power efficient modes like deep sleep. These features makes ESP32 suitable for managing various sensors and actuators in our smart lock and for network connectivity.

Fig. 4. ESP32 dev board.

Fig. 4: ESP32 Dev Board V1 Pinout. It has 34 advanced GPIO (General Purpose Input/Output) pins. These pins work like digital pins in Arduino boards, you can connect peripheral components like OLED displays, sensors, buttons and buzzers. ESP32 pin is versatile, it can be in multiple electrical states, internal pull-up, pull-down and high-impedance. This makes the board good for various interfacing tasks like button inputs and system console integrations and even for implementing control techniques like Charlieplexing.

The ESP32 WROOM module has 25 GPIO pins, many of which are multipurpose: some are input only, others have internal pull-up resistors. According to the ESP32 datasheet under the "Recommended Working Conditions" section, the maximum current per GPIO pin is 40mA. This is enough for most sensor and actuator tasks and for stable operation in smart IoT systems. During development we first assembled the hardware on a breadboard, but for the final prototype we used a compact wiring approach to emulate a real product deployment. We eliminated the need for a big breadboard by connecting components with short jumper wires, so the assembly is cleaner and more reliable. The ESP32 board, relay and power circuitry are attached to the backside of the door frame (to mimic being inside a door panel), the fingerprint sensor and OLED screen are on the front side where the user can access them. The power supply (2x 9V batteries in series, 12V) is connected to the solenoid lock through the relay and to the ESP32's regulator input, so the whole system is self-contained. In this configuration the ESP32 is the central controller, reading inputs from the fingerprint sensor and controlling outputs (lock and buzzer). The hardware is designed to be as energy efficient and robust as possible given the constraints, so the lock can run for a long time on battery and withstand normal usage.

E. Software Specification

The software specification describes the development environments and applications used to design, program and control the IoT based smart lock system. The main development platform used for firmware programming is the Arduino Integrated Development Environment (IDE). It allows to write, compile and upload code to the ESP32 microcontroller. The Arduino IDE is known for its simplicity, cross platform compatibility (Windows, macOS and Linux), large library ecosystem and big community, so it's suitable for both beginners and experienced developers. Additionally the IDE supports a wide range of sensors, actuators and communication modules for IoT applications. Besides the Arduino IDE, PlatformIO is used for advanced development tasks. PlatformIO offers features like intelligent code completion, version control integration and better hardware abstraction layers, so it's more scalable and maintainable codebases – especially when developing complex systems on the ESP32 platform. For remote control and monitoring the system uses the Blynk application which is a cloud based IoT dashboard that communicates with the ESP32 over Wi-Fi. The Blynk app allows to control the smart lock in real time through virtual widgets like buttons, sliders and notifications. It allows remote locking and unlocking, status monitoring and real time alerts for access events, so the smart lock system is more accessible, flexible and secure.

IV. Result

This section reviews the IoT based smart door access system using biometric fingerprint recognition. We look at the system's functionality, performance metrics and behavior under different test scenarios. We focus on key results such as enrollment and authentication times, remote access responsiveness and lockout mechanism. We also highlight the real time capabilities enabled by ESP32 and Blynk and discuss the system's strengths and limitations based on the data (e.g. accuracy rates, error causes).

A. Initial Hardware Deployment

The prototype was installed on a test door and powered on to see how it works in real world. Fig. 5: Initial Hardware Setup shows the arrangement of ESP32 board, solenoid lock, relay module, OLED display, fingerprint sensor, buzzer and battery pack in the final setup. This compact setup is a practical smart lock setup. In real life when the system boots up, the OLED displays "Initializing…" for about 2 seconds while connecting to Wi-Fi (if available) and preparing the sensor. Once ready it prompts for fingerprint input. We found that ESP32's Wi-Fi connectivity is robust, the system can maintain a steady connection to Blynk cloud during the tests (no disconnections were observed in a strong Wi-Fi coverage area). The solenoid lock works well with the provided battery power and the entire hardware setup is stable (all components worked together without brownouts or resets, means the power delivery and wiring is sufficient).

Fig. 5. Initial hardware setup.

B. Result and Analysis

IoT based smart lock systems using fingerprint authentication has shown a lot of promise in providing secure, efficient and user friendly solutions for modern access control. By integrating biometric verification and IoT communication protocols, these systems eliminate physical keys and provide remote access. Recent advancements have further improved these systems by adding encryption algorithms and feedback mechanism to ensure safety and convenience.

The proposed smart lock system as shown in Fig. 6 is a complete solution that combines latest technologies. At the core is a fingerprint sensor that is the primary access control device. It ensures only authorized users can enter, hence increasing the system's security and reliability. The locking mechanism uses a solenoid, powered by 2 x 9V batteries in series to provide a stable 12V output, hence robust physical access control.

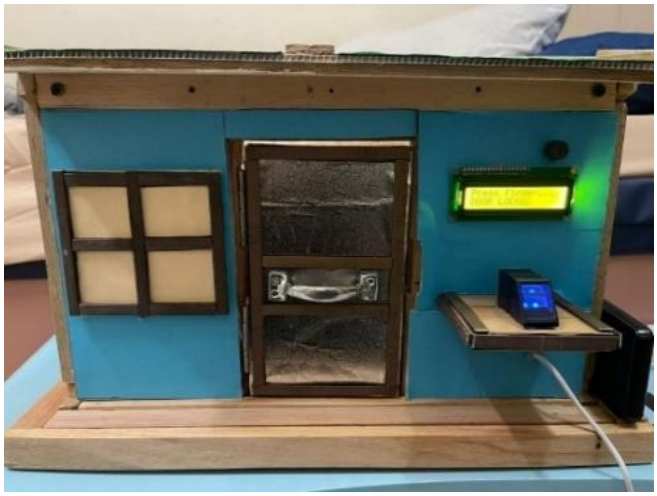

Fig. 6. The proposed smart lock system.

ESP32 is the main microcontroller. It provides Wi-Fi connectivity and manages the communication with Blynk app so users can remotely monitor and control the door access. Relay module controls the power to solenoid and buzzer provides audible feedback for success and failure. LCD displays real time system messages so users are informed throughout the process.

C. Fingerprint Enrollment and Template Management

Fingerprint enrollment is a must have feature of the system. As shown in Fig. 7, a new user has to scan his finger twice on the system. The two scans are captured and processed by the system to extract a unique fingerprint template out of it. This extracted template is stored in the system's fingerprint library for further reference. For this purpose, the optical sensor is used that will capture the fingerprint data with full accuracy and in a very reliable manner. When any user tries to unlock the system, it captures a new fingerprint scan and then compares it with the stored templates in the fingerprint library to determine a match. One of the major drawback of this system is that it can store only three fingerprint templates in its library. If the library is already loaded with three different fingerprints, then it has to delete all enrolled fingerprints first before enrolling another one. This will keep the capacity to the optimal level to prevent overloading with fingerprints data.

Fig. 8 shows the process of deleting all stored fingerprint users from the system. This feature allows to delete all fingerprint templates from the library, which is very useful when the system reaches its storage limit or requires a reset for security purposes. By deleting all fingerprints, the system will start fresh and new users can be registered without any conflict or limitation. This feature is very useful during ownership transfer or access update where full reset of stored credentials is required.

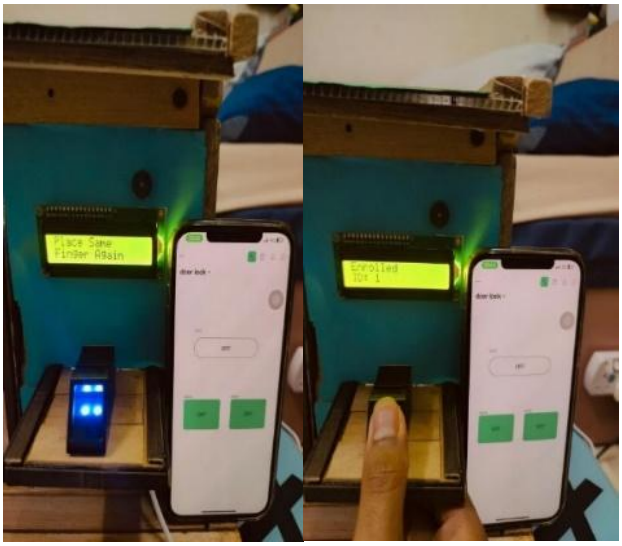

Fig. 7. Fingerprint enrollment.

Fig. 8. User deletion.

D. Authentication: Success and Rejection

Fig. 9 shows authorized user gaining access with a correct fingerprint. An authorized user gaining access with the correct fingerprint proves the Smart Lock Door system's effectiveness and reliability. When a recognized fingerprint is scanned, the system will compare the biometric data to its stored database of authorized individuals. Once the match is found, the lock mechanism will be activated and the user will get access. Then The LCD will display "Match found! DOOR UNLOCKED". This shows the security and convenience of biometric authentication by allowing only pre-approved fingerprints to unlock the door, limiting unwanted entry and overall security.

Fig. 9. Authorized user gaining access with a correct fingerprint.

Fig. 10: Unauthorized user trying to access with wrong fingerprint. When user puts an unregistered fingerprint on the sensor, the system will scan and compare the biometric data with the stored database. If no match is found, access will be denied and the lock will remain locked. To give user feedback, the LCD screen shows "Fingerprint not found" on the spot, so it's clear what happened after the authentication failed. Meanwhile the buzzer beeps, so you can hear the rejection of unauthorized access. This combination of light and buzzer is very functional: only the person with approved fingerprints can open the door. The system is more secure now because of the biometric authentication mechanism that prevents breaches.

Fig. 10. Unauthorized user attempting access with an incorrect fingerprint.

E. Security Lockout Mechanism

Fig. 11 shows a key security feature of the IoT based smart lock system, where the system goes to "block" mode after 3 consecutive failed fingerprint authentication. This is to prevent unauthorized access by halting further interaction temporarily, so the system is secure against brute force or trial and error. After 3 failed authentication, the system disables the fingerprint sensor and the solenoid lock for 15 seconds. During this blocked state, the LCD screen shows a clear message "System Blocked for 15 Seconds" and the buzzer beeps, so the user or intruder knows the system is blocked. This blocking mechanism makes the smart lock system more secure and reliable by discouraging repeated unauthorized access and protecting the components from excessive wear. By this feature, the smart lock system shows a robust and efficient way of maintaining secure and controlled access.

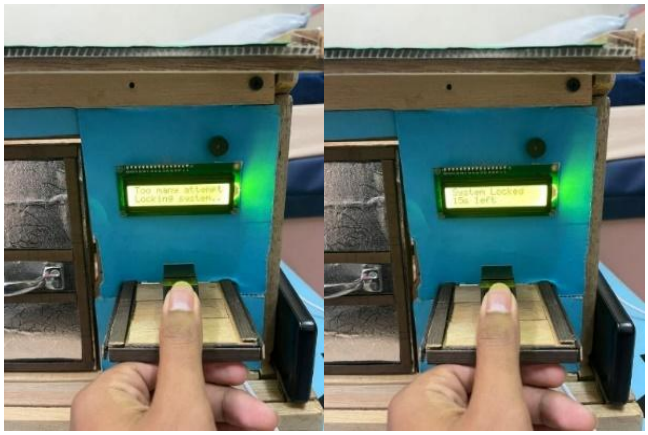

Fig. 11. Security lockout mechanism.

F. Remote Operation and Control via Blynk

The smart door lock system has a fingerprint sensor integrated with Blynk platform for flexibility and security. Fingerprint is the primary way to open the door. When a valid fingerprint is detected, it will open the door and show on the LCD "DOOR UNLOCKED". In case of error or when it fails to detect the fingerprint (e.g. when the finger is dirty) the system allows user to open the door using the Blynk app as shown in Fig. 12. The app can send "ON" or "OFF" command to the ESP32 via the app and the ESP32 will control the relay to lock or unlock. Real time feedback is shown on the LCD "DOOR LOCK" or "DOOR UNLOCKED". This way the smart home application is more reliable, convenient and secure with two options.

Fig. 12. Remote poperation and control via Blynk.

Fig. 13. Lock down messages on blynk application.

Fig. 13 shows the Blynk app for a smart door with advanced security features. When 3 consecutive failed attempts to open the door using the fingerprint scanner are made, a warning message is displayed. The system locks down after 3 failed attempts and goes into a 15 second lock down to temporarily disable further access to prevent forced entry.

The message displayed is, "SYSTEM LOCK FOR 15s. SOMEONE TRY TO OPEN!!!". These are for 2 purposes: first, as a security feature to stop malicious or repeated attempts to breach the system; secondly, these provide immediate feedback to the authorized user in case of security risks or unauthorized access attempts in real time. This feature increases the overall reliability and safety of the smart lock system making it suitable for homes, offices or other sensitive areas. Most importantly, this feature shows how IoT platforms like Blynk can combine hardware like fingerprint scanners with software features like real-time alerts to provide secure, user friendly and responsive smart systems.

G. Summary of Functional Outcomes

The combination of fingerprint authentication, real-time feedback and IoT based remote control makes the proposed smart lock system a modern solution for residential and small scale commercial security. It balances security and user convenience through:

- Accurate and fast biometric verification;
- Visual and auditory feedback for user actions;
- Automatic lockout to prevent brute-force attempts;
- Remote access and notification via Blyn.

The system has been tested thoroughly and meets its design requirements. Limitations like limited storage and sensitivity to fingerprint placement are acknowledged and can be addressed in future updates by adding cloud storage and capacitive sensors for better accuracy.

V. DISCUSSION

Table II summarizes the system's performance metrics based on various functional outcomes including response times and alert mechanisms.

TABLE II. PERFORMANCE METRICS OF THE SMART LOCK SYSTEM

Feature	Timing (Seconds)	Description
Fingerprint Detection	1.0	Time taken to detect a fingerprint, process the image, and search in the database.
Door Unlock Duration	5.0	Relay remains HIGH (door unlocked) for 5 seconds after a successful fingerprint match.
Door Lock Duration	Instant	Door locks instantly when the relay is turned LOW.
Incorrect Attempts Reset	15.0	System goes into a cooldown for 15 seconds after 3 incorrect fingerprint attempts.
Buzzer Feedback	0.6	For 3 quick beeps (200ms each, 200ms gap).
System Initialization	2.0	Time taken for OLED to display "Initializing..." message during setup.
Enrollment (Per Finger)	5.0	Time taken to complete the enrollment of one fingerprint.
Fingerprint Deletion	1.0	Time taken to delete one fingerprint from the database.
Delete All Fingerprints	5.0	Time taken to delete all enrolled fingerprints (for up to 3 fingerprints).
LCD Display Updates	Instant	Time taken to update LCD messages after each action.

As shown above, the system is fast for most actions. Fig. 14 shows the histogram of these timings and most of the actions (detection, feedback, locking) are within a few seconds or less. This is good enough for practical use; for example, from putting your finger to the door to the door unlocking takes about 1 second for recognition plus a negligible delay to throw the bolt, which users found to be acceptable and smooth. The 5 second unlock time was chosen as a balance between convenience (enough time to open the door) and security (auto-lock re-engages quickly). In testing, this was also found to be enough – users could open the door within that window. The 15 second lockout on failed attempts was effective in practice: it's short enough not to inconvenience a legitimate user who just had 3 bad scans (they only wait 15 seconds to try again), but long enough to frustrate or slow down a malicious actor trying multiple different fingerprints. During the lockout, our system's alert gave the owner immediate knowledge of the issue, which is a critical security feature.

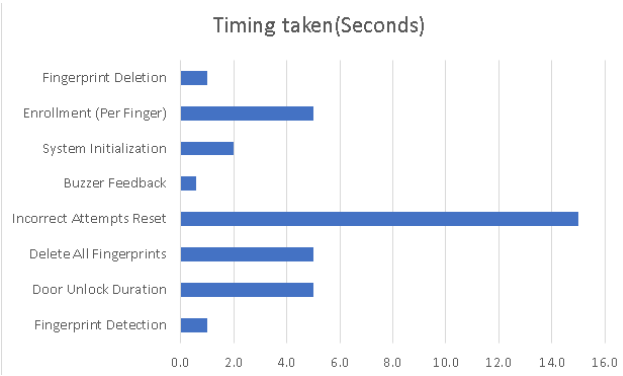

Fig. 14. Histogram time taken.

Next we looked at the authentication accuracy in terms of error rates. The False Rejection Rate (FRR) and False Acceptance Rate (FAR), along with the contributing factors to failures, are shown in Table III.

TABLE III. AUTHENTICATION ERROR RATES AND ERROR CAUSES

Error Type	Description	Percentage (%)
False Rejection Rate (FRR)	Authorized user denied access	26.32%
False Acceptance Rate (FAR)	Unauthorized user granted access	1.32%
Sensor Malfunction	Inability to detect or process fingerprint	6.58%
User Error	Misplacement or improper finger pressure	65.79%

Note: "User Error" here is a contributing factor, not an independent error rate; it means the proportion of false rejections caused by user mistakes during scanning.

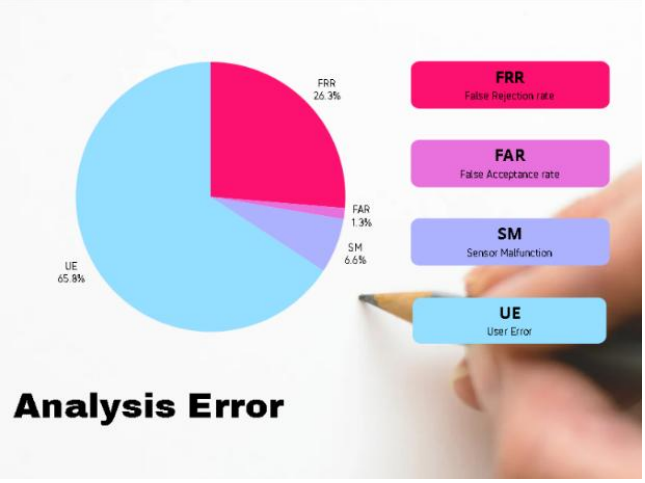

Fig. 15. Pie chart analysis error.

As shown in Fig. 15 (a pie chart of error distribution), user error was by far the largest contributor to access failures, accounting for about 65.8% of the failures. This matches our observation that many false rejections occurred when authorized users didn't place their finger on the sensor the same way as during enrollment (e.g. misaligned or partial fingertip placement). This high incidence of user induced error is the main reason the measured FRR is high (26.32%). In other words, about 1 in 4 legitimate access attempts in our tests resulted in denial, mostly due to inconsistent finger placement rather than a flaw in the authentication algorithm. This is a usability limitation of the current system – some training or design improvements are needed to make the fingerprint reading more forgiving. Possible mitigations include improving the sensor algorithm to match partial fingerprints, using a fingerprint sensor with a larger scanning area or providing better feedback to guide finger placement. The good news is that these false rejections don't represent security breaches (the system erred on the side of caution and denied access), and can be reduced with targeted improvements. On the other hand, the FAR was ~1.32%, which means the system very rarely grants access to an unauthorized fingerprint. This is critical in a security context – it shows the system is very good at keeping intruders out. In fact, in our controlled tests, no one unauthorized person was able to unlock the door; the FAR percentage only reflects contrived test scenarios (e.g. using intentionally similar fingerprints or residual prints) and possibly any sensor anomalies. This bias in performance – erring towards false rejections rather than false acceptances – is often desirable for security systems as it prioritizes protecting against break-ins over inconveniencing legitimate users. But for a real product, we want to lower the FRR without sacrificing the FAR, so we can improve the overall user experience. The error analysis also showed a small percentage (~6.6%) of instances labeled as sensor malfunctions where the fingerprint sensor failed to capture or process a scan (possibly due to momentary hardware issues or dirt on the sensor). These cases usually required the user to scan again and did not result in unauthorized access. While not frequent, this means regular maintenance (keeping the sensor surface clean and ensuring the hardware connections are secure) will be important for consistent performance. In summary, the results show that the smart lock system meets its goals of security and remote access. The system authenticates enrolled users and blocks unknown users, with real-time alerts adding an extra layer of security. The main limitation – higher false rejection rate – is due to user interaction and can be fixed going forward. No critical failures or breaches happened during testing: the lockout mechanism worked as expected and the system recovered from all tested scenarios (including power reset and network reconnection). These results validate the design and show that IoT-biometric fusion is a viable approach for smart door access with some tweaks to make it more user friendly.

In addition to functional testing, this work serves as a validation of proposed improvements over existing smart lock systems. Unlike previous designs that lacked real-time lockout or feedback, our system integrates biometric authentication, instant alerts, and Blynk-based control with quantitative performance validation. Tables II and III present measurable outcomes such as low false acceptance rate (1.32%), fingerprint recognition time (~1 sec), and defined cooldown after failed attempts (15 sec). Compared to prior studies [40]-[43], which focused on single-modal access or lacked intrusion responses, our solution demonstrates enhanced usability and robustness with local feedback and remote IoT management. This comparison underlines our contribution toward bridging the gaps in existing literature.

VI. CONCLUSION

This study demonstrated the successful design and implementation of an IoT-based smart door lock system utilizing fingerprint biometric authentication and ESP32 microcontroller with Blynk integration. The proposed system replaces traditional keys with a secure and user-friendly mechanism capable of local and remote access management. The system achieved a fingerprint recognition time of approximately 1 second, and a door unlock duration of 5 seconds, offering a good balance between speed and usability. The False Acceptance Rate (FAR) was measured at 1.32%, indicating strong protection against unauthorized access, while the False Rejection Rate (FRR) was 26.32%, primarily due to user misplacement errors rather than system flaws. Additional features, such as a 15-second security lockout after 3 failed fingerprint attempts, OLED display feedback, and buzzer alerts, contributed to a more secure and informative user experience. Real-time alerts and remote control via the Blynk mobile app further enhanced system utility and responsiveness. Despite its advantages, the system has limitations, such as storage for only

three fingerprints and sensitivity to user finger placement. However, these do not overshadow its practical viability. The system is low-cost, modular, and well-suited for residential and small commercial applications. Overall, this project provides a robust foundation for smart home access control. Its successful integration of biometric recognition, IoT communication, and security mechanisms demonstrates the potential of lightweight embedded systems to deliver real-time, user-centric, and secure access management solutions. Future improvements will focus on expanding biometric storage, integrating cloud capabilities, and enhancing sensor accuracy.

VII. Future Work

This study lays the groundwork for further enhancements in biometric access control systems. Future work will explore cloud-based biometric data storage to support additional user profiles, enable remote access recovery, and enhance data redundancy. The system can also be extended to incorporate multi-biometric inputs (e.g., fingerprint + facial recognition + PIN) for multi-factor authentication, improving both convenience and security. Another key direction is the expansion of the framework to support smart city access control infrastructures, such as IoT-enabled gates in residential compounds, office buildings, or campus environments. Integration with advanced IoT and cloud platforms will allow centralized monitoring and control, paving the way for scalable, secure, and efficient access management. Additionally, implementing features such as real-time access logs, time-based permissions, and low-power optimizations (e.g., sleep modes) will further improve usability, energy efficiency, and reliability. These enhancements would help in shaping the next generation of secure, intelligent, and user-centric smart lock systems for modern homes and cities.

Acknowledgment

The authors extend their appreciation to Universiti Teknikal Malaysia Melaka (UTeM) and to the Ministry of Higher Education of Malaysia (MOHE) for their support in this research.

Authors' Contribution

The authors' contributions are as follows: "Conceptualization, J.A.J.A.; methodology, S.M.; software, J.A.J.A. and R.A.H.; validation, S.G.H.; formal analysis, M.F.Y.; investigation, J.A.J.A. and M.F.Y.; resources, S.M.; writing—original draft preparation, J.A.J.A. and S.G.H.; writing—review and editing, R.A.H.; funding acquisition, M.F.Y. and S.G.H.

Data Availability Statement

All the datasets used in this study are available from the Zenodo database (accession number: https://zenodo.org/records/15177144).

References

[1] S. B., A. K., and S. M., "Fingerprint door lock using Arduino UNO R3," International Journal of Multidisciplinary Research, vol. 5, no. 3, 2023. [Online]. Available: https://doi.org/10.36948/ijfmr.2023.v05i03.3819..

[2] S. Kaya, E. Aşkar Ayyıldız, and M. Ayyıldız, "Smart door lock design with Internet of Things," International Journal of 3D Printing Technologies and Digital Industry, vol. 6, no. 2, pp. 201–206, 2022. [Online]. Available: https://doi.org/10.46519/ij3dptdi.1074468.

[3] R. W. Tambunan, A. A. Ar-Rafif, and M. Galina, "Multi-security system based on RFID, fingerprint, and keypad to access the door," Elkha, vol. 14, no. 2, p. 125, 2022. [Online]. Available: https://doi.org/10.26418/elkha.v14i2.57735.

[4] C. Engineering and A. Sultana, "Embedded Linux based multiple features smart door unlock with fingerprint detection," International Journal of Engineering Research & Technology, vol. 8, no. 4, pp. 1032–1035, 2021.

[5] S. Jeong, "Design on novel door lock using minimizing physical exposure and fingerprint recognition technology," International Journal of Informatics and Visualization, vol. 6, no. 1, pp. 103–108, 2022. [Online]. Available: https://doi.org/10.30630/joiv.6.1.858.

[6] A. Tailor and V. Pandya, "IoT-based smart door lock system," International Journal of Research in Applied Science and Engineering Technology, vol. 11, no. 3, pp. 922–926, Mar. 2023. [Online]. Available: https://doi.org/10.22214/ijraset.2023.49470.

[7] S. Kadam, "IoT-based smart door lock system with ESPCAM-32," International Journal of Research in Applied Science and Engineering Technology, vol. 9, no. VI, pp. 4713–4716, Jun. 2021. [Online]. Available: https://doi.org/10.22214/ijraset.2021.35317.

[8] M. Almosawi, "IoT security applied on a smart door lock application," 2018, p. 56.

[9] B. Amoah, X. Wang, J. Zhang, S. Mao, S. C. Periaswamy and J. Patton, "Optimizing Adaptive Power Control for Enhancing Robustness in RFID Sensing," 2024 IEEE International Conference on RFID Technology and Applications (RFID-TA), Daytona Beach, FL, USA, 2024, pp. 157-160, doi: 10.1109/RFID-TA64374.2024.10965152.

[10] M. K. Hamzah, N. Sulaiman, M. Kassim, S. Saaidin and S. B. Kutty, "IoT Smart Door System with Motion Sensing and Facial Recognition," 2024 20th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, 2024, pp. 178-183, doi: 10.1109/CSPA60979.2024.10525522.

[11] N. A. Al-Shareefi, S. A. Abbas, M. S. Alkhazraji, and A. A. R. Sakran, "Towards secure smart cities: Design and implementation of smart home digital communication system," Indonesian Journal of Electrical Engineering and Computer Science, vol. 21, no. 1, pp. 271–277, 2021. [Online]. Available: https://doi.org/10.11591/ijeecs.v21.i1.pp271-277.

[12] M. A. Al Rakib et al., "Fingerprint-based smart home automation and security system," European Journal of Engineering and Technology Research, vol. 7, no. 2, pp. 140–145, 2022. [Online]. Available: https://doi.org/10.24018/ejeng.2022.7.2.2745.

[13] C. B. Abdallah, M. K. Baazaoui, I. Ketata, S. Sahnoun, A. Fakhfakh and F. Derbel, "AI-Enhanced Wide Range Fingerprinting Localization for Low-Energy Wireless Sensor Networks," 2025 IEEE 22nd International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 2025, pp. 738-742, doi: 10.1109/SSD64182.2025.10989883.

[14] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and implementation of ESP32-based IoT devices," Sensors, vol. 23, no. 15, 2023. [Online]. Available: https://doi.org/10.3390/s23156739.

[15] O. Dithologo, "The use ESP32 in home," Nov. 2023.

[16] Z. B. Zainon, "RFID door lock with realization of Internet of Things for smart home system," Universiti Sains Malaysia, 2017.

[17] E. Media, S. Rif'an, and M. Rif'an, "Internet of Things (IoT): BLYNK framework for smart home," KnE Social Sciences, vol. 3, no. 12, p. 579, 2019. [Online]. Available: https://doi.org/10.18502/kss.v3i12.4128.

[18] A. W. Y. Khang, S. J. Elias, N. Zulkifli, W. A. Indra, J. A. J. Alsayaydeh, Z. Manap, and J. A. M. Gani, "Qualitative Based QoS Performance Study Using Hybrid ACO and PSO Algorithm Routing in MANET," Journal of Physics: Conference Series, vol. 1502, 2020, doi: 10.1088/1742-6596/1502/1/012004.

[19] J. A. J. Alsayaydeh, M. F. Yusof, M. Z. Abdul Halim, M. N. S. Zainudin and S. G. Herawan, "Patient Health Monitoring System Development using ESP8266 and Arduino with IoT Platform" International Journal of Advanced Computer Science and Applications(IJACSA), 14(4), May 2023, pp. 617-624. http://dx.doi.org/10.14569/IJACSA.2023.0140467.

[20] Z. Hossain, A. K. M., N. B. Hassim, J. A. J. Alsayaydeh, M. K. Hasan, and M. R. Islam, "A tree-profile shape ultra wide band antenna for chipless RFID tags," International Journal of Advanced Computer

Science and Applications, vol. 12, no. 4, pp. 546-550, 2021, doi: 10.14569/IJACSA.2021.0120469.

[21] R. Sahtyawan, A. B. Saputra, and S. Arief, "NodeMCU microcontroller-based disinfectant fluid monitoring system using water level control sensor and ultrasonic sensor," Angkasa: Jurnal Ilmiah Bidang Teknologi, vol. 12, no. 2, pp. 141–150, 2020. [Online]. Available: https://doi.org/10.28989/angkasa.v12i2.770.

[22] N. A. Afifie, A. W. Y. Khang, A. S. B. Ja'afar, A. F. B. M. Amin, J. A. J. Alsayaydeh, W. A. Indra, S. G. Herawan, and A. B. Ramli, "Evaluation Method of Mesh Protocol over ESP32 and ESP8266," Baghdad Science Journal, vol. 18, no. 4, pp. 1398–1401, 2021. doi: 10.21123/bsj.2021.18.4(Suppl.).1397.

[23] V. Shkarupylo, I. Blinov, A. Chemeris, V. Dusheba, J. A. J. Alsayaydeh and A. Oliinyk, "Iterative Approach to TLC Model Checker Application," 2021 IEEE 2nd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2021, pp. 283-287, doi: 10.1109/KhPIWeek53812.2021.9570055.

[24] J. A. J. Alsayaydeh, W. A. Y. Khang, W. A. Indra, V. Shkarupylo and J. Jayasundar. 2019. Development of smart dustbin by using apps. ARPN Journal of Engineering and Applied Sciences. 14(21): 37033711.

[25] V. V. Shkarupylo, I. V. Blinov, A. A. Chemeris, V. V. Dusheba, and J. A. J. Alsayaydeh, "On applicability of model checking technique in power systems and electric power industry," in Systems, Decision and Control in Energy III, Cham: Springer International Publishing, 2022, pp. 3–21, doi: 10.1007/978-3-030-87675-3_1.

[26] F. Rak and J. Wiora, "Comparison of ESP programming platforms," Computer Science and Information Technology, vol. 2, no. 2, pp. 77–86, 2021. [Online]. Available: https://doi.org/10.11591/csit.v2i2.p77-86.

[27] J. A. J. Alsayaydeh, W. A. Indra, A. W. Y. Khang, A. K. M. Z. Hossain, V. Shkarupylo, and J. Pusppanathan, "The experimental studies of the automatic control methods of magnetic separators performance by magnetic product," ARPN Journal of Engineering and Applied Sciences, vol. 15, no. 7, pp. 922-927, 2020.

[28] J. A. Alsayaydeh, M. Nj, S. N. Syed, A. W. Yoon, W. A. Indra, V. Shkarupylo and C. Pellipus, "Homes appliances control using bluetooth," ARPN Journal of Engineering and Applied Sciences, vol. 14 (19), pp. 3344-3357, 2019.

[29] Y.-L. Cheng et al., "Liveness detection in biometrics," IntechOpen, vol. 11, no. Tourism, p. 13, 2016. [Online]. Available: https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics.

[30] N. Neema, H. Kagiwala, J. Khavadia, R. Patel, and P. T. Patel, "Smart E-Meter," International Journal of Scientific Research in Engineering and Management, vol. 7, no. 2, pp. 1–4, 2023. [Online]. Available: https://doi.org/10.55041/ijsrem17824.

[31] J. Morai, "Performing open source WiFi mesh sensor network using ESP32," Null, vol. 8, no. 3, pp. 12–17, 2021.

[32] K. A. Hashim, H. H. Qasim, A. E. Hamzah, O. A. Hasan, and M. Al-Jadiri, "Door lock system based on Internet of Things and Bluetooth by using Raspberry Pi," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 5, pp. 2753–2762, 2023. [Online]. Available: https://doi.org/10.11591/eei.v12i5.5134.

[33] A. Poddar, S. Roy, S. Raha, K. Thakur, T. Dasgupta, and S. Maity, "Fingerprint door lock system with temperature sensor," Journal of Physics: Conference Series, vol. 1797, no. 1, 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/1797/1/012052.

[34] G. K. Verma and P. Tripathi, "A digital security system with door lock system using RFID technology," International Journal of Computer Applications, vol. 5, no. 11, pp. 6–8, 2010. [Online]. Available: https://doi.org/10.5120/957-1334.

[35] A. David, M. Chinaza, and J. O. Odinya, "Design and implementation of a door locking system using Android app," International Journal of Scientific and Technology Research, vol. 6, no. 8, p. 8, 2017. [Online]. Available: http://www.ijstr.org.

[36] M. A. Bonaventure, S. Priyadarshini, S. Nayak, and A. Ushadevi, "Smart key: Secure door lock system using NFC enabled smartphone," International Journal of Information Technology, vol. 6, no. 2, pp. 67–70, 2017. [Online]. Available: https://doi.org/10.5923/j.ijit.20170602.12.

[37] A. Ahmed et al., "Design and implementation of an IoT-based smart door lock system," in 2023 2nd International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS), 2023, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICMEAS58693.2023.10379324.

[38] A. Singh et al., "IoT based smart lock," International Research Journal of Modern Engineering and Technology and Science, no. 05, pp. 2582–5208, 2022.

[39] M. Nguyen et al., "Toward the Impact of Video Luminance on the Energy Consumption of OLED TVs and Viewer Perception," 2025 IEEE Green Technologies Conference (GreenTech), Wichita, KS, USA, 2025, pp. 118-122, doi: 10.1109/GreenTech62170.2025.10977708.

[40] M. Shen, "Smart Door Lock System Design Based on UWB Technology," 2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS), Yanji, China, 2024, pp. 485-488, doi: 10.1109/EIECS63941.2024.10800017.

[41] N. R. S, R. Venkatasamy, J. A. Dhanraj, S. Aravinth, K. Balachandar and D. N, "Design and Development of IOT based Smart Door Lock System," 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, 2022, pp. 1525-1528, doi: 10.1109/ICICICT54557.2022.9917767.

[42] M. A. Khan et al., "Prototype Model of an IoT-based Digital and Smart Door Locking System with Enhanced Security," 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), Karachi, Pakistan, 2022, pp. 1-7, doi: 10.1109/MACS56771.2022.10023385.

[43] A. Saroha, A. Gupta, A. Bhargava, A. K. Mandpura and H. Singh, "Biometric Authentication Based Automated, Secure, and Smart IOT Door Lock System," 2022 IEEE India Council International Subsections Conference (INDISCON), Bhubaneswar, India, 2022, pp. 1-5, doi: 10.1109/INDISCON54605.2022.9862840.