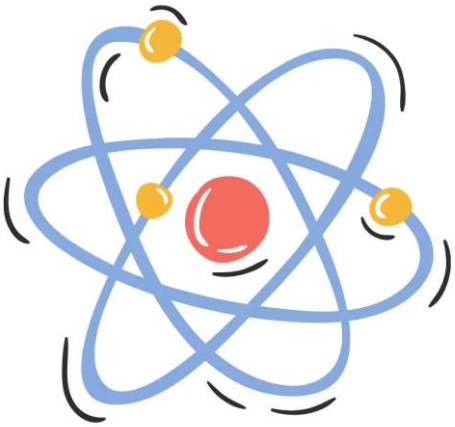


JOURNAL OF DYNAMICS AND CONTROL

VOLUME 8 ISSUE 12

DUAL AUTHENTICATED SMART DOOR LOCKS USING MACHINE LEARNING TECHNIQUE

Satvik V, Rohit M and U. Vignesh
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai,
Tamil Nadu - 600127, India

DUAL AUTHENTICATED SMART DOOR LOCKS USING MACHINE LEARNING TECHNIQUE

Satvik V, Rohit M and U. Vignesh*
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai, Tamil Nadu - 600127, India
**vignesh.u@vit.ac.in*

Abstract: *This research is about a door lock system that combines RFID and keypad authentication, along with a simple user interface for easy use. The system records each successful entry in a database, noting the time and method of access. The keypad password is stored encrypted in the database for security reasons. The main goal is to improve security by using machine learning to spot unusual access patterns that could mean a potential security issue. By using past access data, a model is created to identify unusual times when the door is opened, which could indicate unauthorized access. The machine learning model learns from regular access records to recognize normal patterns and alert users if anything out of the ordinary happens. This system not only makes traditional door locks more secure but also adds smart features that adapt and learn over time, making it a proactive solution for detecting intruders. This combination of hardware, data analysis, and machine learning helps create a strong security solution for homes and businesses. Additionally, the system is designed to be energy efficient by using a PIR sensor to only power the system when movement is detected, ensuring minimal energy consumption.*

Keywords: RFID, Keypad Authentication, Door Lock System, Machine Learning, Anomaly Detection, Security Systems, Access Control, Data Encryption, Smart Locks, Internet of Things (IoT).

1. Introduction

Ensuring the safety of homes and workplaces has become more critical than ever. Over the time, the vulnerabilities of traditional security methods have also become more apparent as burglars and unauthorized individuals develop more sophisticated techniques to bypass them. Locks that rely solely on mechanical mechanisms can be easily picked or forced open, leaving homes and work- places exposed. Additionally, the widespread use of duplicated keys, either intentionally or through unauthorized means, further exacerbates the problem. These duplicated keys can fall into the wrong hands, compromising the security of properties without the owner's knowledge. Furthermore, outdated locks do not offer any form of notification or alert system, which means that any unauthorized entry often goes unnoticed until it is too late. The lack of advanced security features such as tamper detection or activity tracking leaves properties vulnerable and increases the risk of successful break-ins and thefts, undermining the sense of safety that traditional locks are meant to provide.

As illustrated in Fig.1.[8] , theft accounts for the highest number of police-recorded offences per 100,000 inhabitants across various European countries for 2022, followed by burglary and robbery. This data highlights significant disparities in crime rates among different nations, with some countries experiencing a markedly higher incidence of these offences. The prevalence of theft and burglary underscores the importance of strengthening security measures and implementing comprehensive strategies to prevent these crimes. Traditional security approaches may not be sufficient, making it crucial to adopt modern smart security solutions to protect homes, workplaces, and public spaces more effectively.

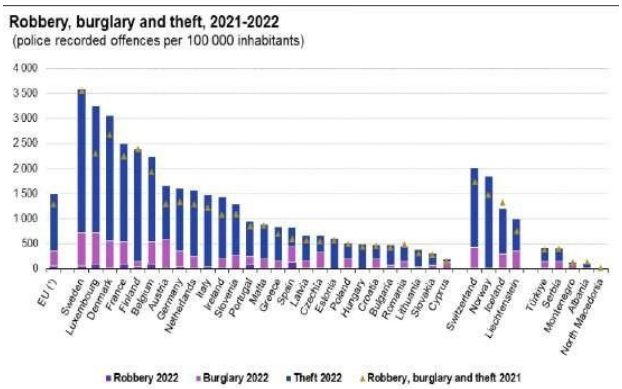


Figure 1. Rates of police-recorded offences for robbery, burglary, and theft per 100,000 inhabitants across various European countries for the years 2021 and 2022.

To address these issues, a smart door lock system has been developed that integrates RFID technology with keypad [10] access. This system connects to an Oracle SQL database to store essential information, such as timestamps, access methods, and the identity of the individual accessing the door. This integration simplifies data management and provides a robust foundation for analyzing access patterns. A key component of the project is a machine learning model designed to scrutinize access data, detect irregularities, and enhance security. If an intrusion is detected, the system promptly sends an alert to the owner.

The user-friendly interface includes a calendar view that dis- plays door access activity, detailing the individuals and times in- volved. This feature enhances security management and is particularly useful for parents monitoring when family members enter or exit the home. By combining these features with intelligent data analysis, this solution provides a more secure and insightful way to control access. Unlike traditional locks, this system leverages machine learning to monitor events and identify unusual patterns indicative of potential break-ins. It ensures password encryption and continuous historical data analysis for high-level security and proactive protection. This approach not only enhances the security of traditional door locks but transforms them into adaptive smart systems. With real-time alerts and comprehensive access history, users gain peace of mind by knowing who enters their space and when. This project bridges the gap between conventional security methods and modern smart technology, offering a comprehensive solution to safeguard homes and businesses.

2. Related Works

There have been several studies that aim to improve door lock security systems by combining traditional mechanisms with modern technologies like RFID, machine learning, and data encryption. A variety of door lock security systems, including both mechanical and electronic solutions, have been reviewed in existing research [7]. Studies have shown that combining traditional locks with smart features like RFID [9][11]can im- prove security. However, one limitation of this approach is that it often does not address how to analyze access data for suspicious patterns. This project addresses that gap by incorporating machine learning to detect unusual activity.

Previous work on door lock systems has included RFID tech- nology with cloud monitoring [1][12]. This approach inspired the inclusion of RFID in the current system, but potential drawbacks such as reliance on the cloud—which can lead to vulnerabilities during network outages—were mitigated by using a local Oracle SQL database to ensure continuous access data availability. Smart locks with keypad authentication and centralized data tracking have also been developed [2], aligning with the use of an Oracle SQL database

in this project. However, those systems often lack integration with machine learning for detecting unusual activity, a feature included here for enhanced security.

The concept of using machine learning to detect unusual activities in smart homes by analyzing access data provided a foundation for the anomaly detection model used in this system [3][18]. While previous research focused on individual door locks, this project integrates multiple authentication methods to enhance security. Reviews of various machine learning techniques for anomaly detection in IoT devices were instrumental in selecting a suitable model for the current door lock system [4][13]. One noted challenge was the need for substantial data to make these models effective, which may not always be feasible. This project over- comes that by using a hybrid approach that combines different models to function effectively with smaller datasets.

Surveys on machine and deep learning techniques for intrusion detection in IoT systems have highlighted the strengths of deep learning for detecting intrusions, but also pointed out the computational demands of such models [5][14]. To balance efficiency and complexity, simpler yet effective machine learning models were chosen. Proposals for highly secure door lock systems using multiple sensors for enhanced security have influenced the design of this project [6][16]. While those systems are generally tailored for high-security areas, this project adapts those concepts to make them more accessible and affordable for typical households.

A study on securing smart doors using RFID and IoT tech- nologies proposed a system that integrates RFID with Internet of Things (IoT) devices to provide enhanced access control [12]. The system leverages RFID tags for user identification, allowing seamless access while utilizing a central IoT platform for real- time monitoring and control. This approach improves traditional security measures by offering connectivity and remote control through IoT devices. However, the system primarily focuses on RFID authentication and does not incorporate machine learning for anomaly detection. In contrast, the current work enhances access control by combining RFID-based authentication with machine learning algorithms to detect anomalies, providing an additional layer of security by identifying unusual access patterns that go beyond simple credential verification.

A review on RFID applications for secure door access in IoT environments explored the use of RFID technology for enhancing security in smart door systems[15]. The paper discussed how RFID can be integrated into IoT-based environments to provide efficient, contactless access control, offering benefits like ease of use and fast access time. Additionally, the paper highlighted the potential of RFID for secure user identification and the integration of IoT sensors to improve system functionality. While the focus of the work was on the practical deployment of RFID in IoT environments, it did not explore advanced anomaly detection techniques. In contrast, the current research goes further by combining RFID-based authentication with machine learning for real-time anomaly detection, adding an extra layer of intelligence to identify unusual access attempts beyond typical credential verification.

2. Proposed methodology

The proposed door lock system combines hardware compo- nents like RFID and keypad with software features like machine learning-based anomaly detection and a graphical user inter- face (GUI) for monitoring and management. The methodology involves three key components: the hardware system, machine learning model, and user interface. Below are the detailed steps:

2.1. Hardware Implementation

The hardware implementation of the smart door lock system is designed to optimize both security and energy efficiency. The primary components of the system include an RFID reader, a 4x3 keypad, a PIR sensor, a relay door lock, a solenoid lock, a buzzer, an Arduino Nano microcontroller, a breadboard, jumpers, a buck converter, and a switch that allows the selection between RFID and keypad modes.

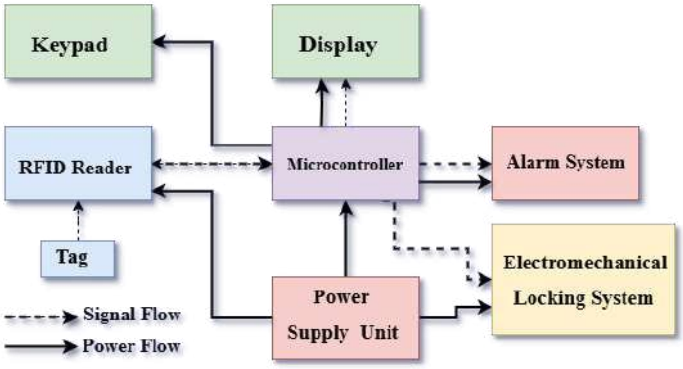


Figure 2. System Architecture

The PIR sensor is the first component in the sequence and plays a crucial role in energy management[18]. It is positioned to detect any movement near the door. When motion is detected, the PIR sensor activates the rest of the system by powering on the components, including the Arduino Nano, RFID reader, and keypad. This design ensures that the system remains inactive when there is no activity, thereby saving energy and making the system highly efficient.

Once the system is powered up by the PIR sensor, it waits for user authentication. Users have two options for unlocking the door—either using an RFID card or entering a passcode through the keypad. The system is equipped with a switch that determines the active mode. When the switch is set to position 1, the RFID reader is active, allowing users to unlock the door using an RFID card. When the switch is set to position 2, the keypad becomes active, enabling users to unlock the door using a passcode. The RFID reader is used to scan RFID cards, which contain a unique identifier stored in the system's memory. If a valid card is detected, the Arduino Nano sends a signal to the relay, which activates the solenoid lock, allowing the door to be opened. The door remains unlocked for a few seconds before the relay returns it to the locked position.

If the RFID card is not available, users can also unlock the door by entering a password on the 4x3 keypad. The keypad is connected to the Arduino Nano, and the entered code is processed by the microcontroller.Once the user provides their credentials (password) via the keypad or interface , the system checks the provided password against the stored encrypted password information in the database. The provided password is hashed securely and compared to the stored encrypted hash (decrypting the AES-256 part to reveal the original SHA-3 hash). If the hashes match, the authentication is successful. Once authenticated, the current time and user details are recorded in the database as a log entry indicating successful access. The lock is triggered to open, allowing the user to proceed. The user receives feedback, such as a message on the display ("Access Granted"). In the case of incorrect authentication—either an invalid RFID card or an incorrect password the buzzer is activated to alert the owner of unauthorized access attempts. This adds an additional layer of security, making it harder for intruders to bypass the system without being noticed.

The components are connected to the Arduino Nano as follows: the RFID reader communicates via the serial port, the keypad is connected using multiple digital input pins, and the PIR sensor is connected to detect motion using a digital input pin. Specifically, the PIR sensor is connected to pin 9, the relay that controls the solenoid lock is connected to pin 11, the condition switch (which controls RFID or keypad mode) is connected to pin 12, and the buzzer is connected to pin 10. The use of a buck converter ensures that the system components receive a stable voltage, which is crucial for reliable operation.

The entire setup is mounted on a breadboard, with jumpers used to establish connections between the various components. The use of a breadboard makes the system modular and easy to adjust or expand if additional sensors or features are added in the future. The relay, which controls the door lock mechanism, is connected to a digital output pin to receive signals from the microcontroller. The solenoid lock is powered via the relay, which enables or disables the door lock based on the authentication result. Finally, the buzzer is also connected to a digital output pin, which allows it to sound an alarm when unauthorized access is attempted.

This interconnected system works in tandem to ensure secure access to the premises while also optimizing energy consumption. By only powering the components when needed, the design provides a practical solution to balance security requirements with energy efficiency.

Figure 3. Hardware Setup of the Door Lock System

2.2. Graphical User Interface (GUI) Implementation

The GUI is implemented using several Python libraries to create a user-friendly interface for monitoring and managing the door lock system:

PySimpleGUI is employed to design the graphical user interface, allowing for the simple creation of buttons, text fields, and other interface elements. This tool simplifies the process of building an interactive interface, making the system user- friendly and accessible for a

range of users. With PySimpleGUI, developers can design intuitive layouts that facilitate efficient interaction with the smart door lock system [17].

Matplotlib is utilized to create visualizations, such as event frequency graphs, which are displayed in separate windows. This library provides powerful plotting capabilities that allow the system to present data in a clear and informative manner. The visual representations of access patterns help users easily identify trends or unusual activities, enhancing the overall monitoring experience. Through Matplotlib, users can gain insights into access events and their frequency, aiding in proactive security management.

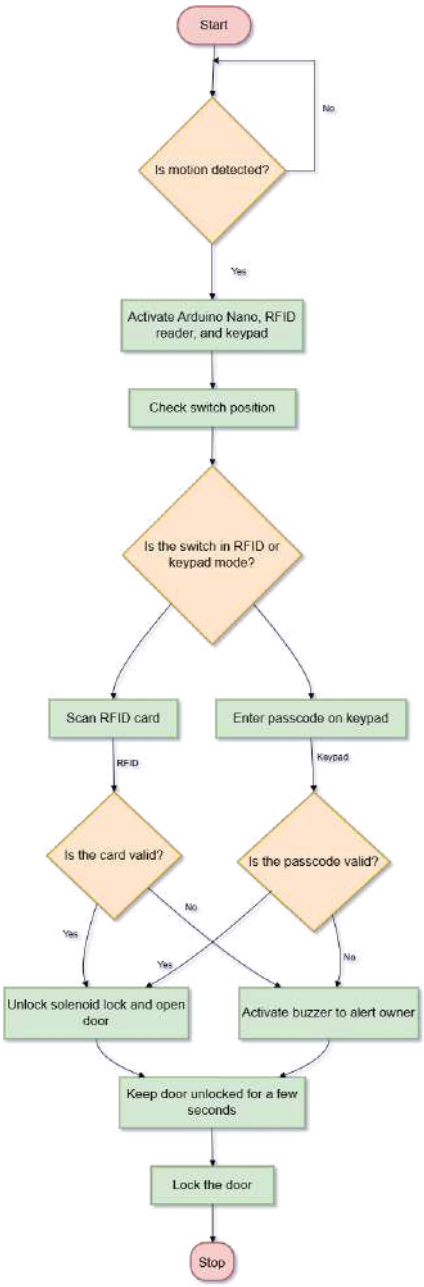


Figure 4. Operational Flow Chart of the Smart Door Lock System

Pandas is instrumental in data manipulation and analysis, making it easier to process access data efficiently. By leveraging Pandas, the system can handle large datasets, filter records, and perform complex data transformations. This enables quick and seamless data

handling, allowing the system to analyze access logs in real-time. The use of Pandas ensures that the information is processed effectively, contributing to the timely detection of any irregularities or potential security threats.

Cx Oracle provides connectivity to the Oracle SQL database, enabling the storage and retrieval of access events. This connectivity ensures that all access data is securely stored and easily accessible for analysis. By using Cx Oracle, the system can communicate with the database to log new entries, update records, and retrieve historical data as needed. This integration ensures that the database remains an integral part of the overall system, supporting robust data management and security analysis.

Additionally, the Serial library is used for serial communication with the Arduino Nano to receive sensor data and control the door lock system. This communication channel is essential for the system to interact with hardware components, allowing real-time responses to sensor inputs and user actions. The Serial library facilitates data transmission between the software and the Arduino, ensuring smooth operation of the door lock mechanism. This integration allows for effective coordination between the software logic and hardware actions, enhancing the overall re- liability and responsiveness of the system.

The interface displays various indicators to monitor the real- time status of the door lock system:

PIR Sensor Indicator: Indicates whether movement is detected near the door. When the PIR sensor picks up movement, the system powers up, and the indicator turns green to signal that the system is active and ready for user authentication.

RFID Status Indicator: Displays the status of RFID card authentication. If an RFID card is successfully scanned and verified, this indicator turns green to show that access is granted. If the scan fails, the indicator remains unchanged or displays a different color to alert the user.

Keypad Status Indicator: Shows the current state of the keypad-based authentication process. When a valid passcode is entered, the indicator changes to green to confirm successful authentication. If an incorrect passcode is entered, the indicator switches to a different color, such as red, to indicate a failed attempt and potentially alert the owner of unauthorized access.

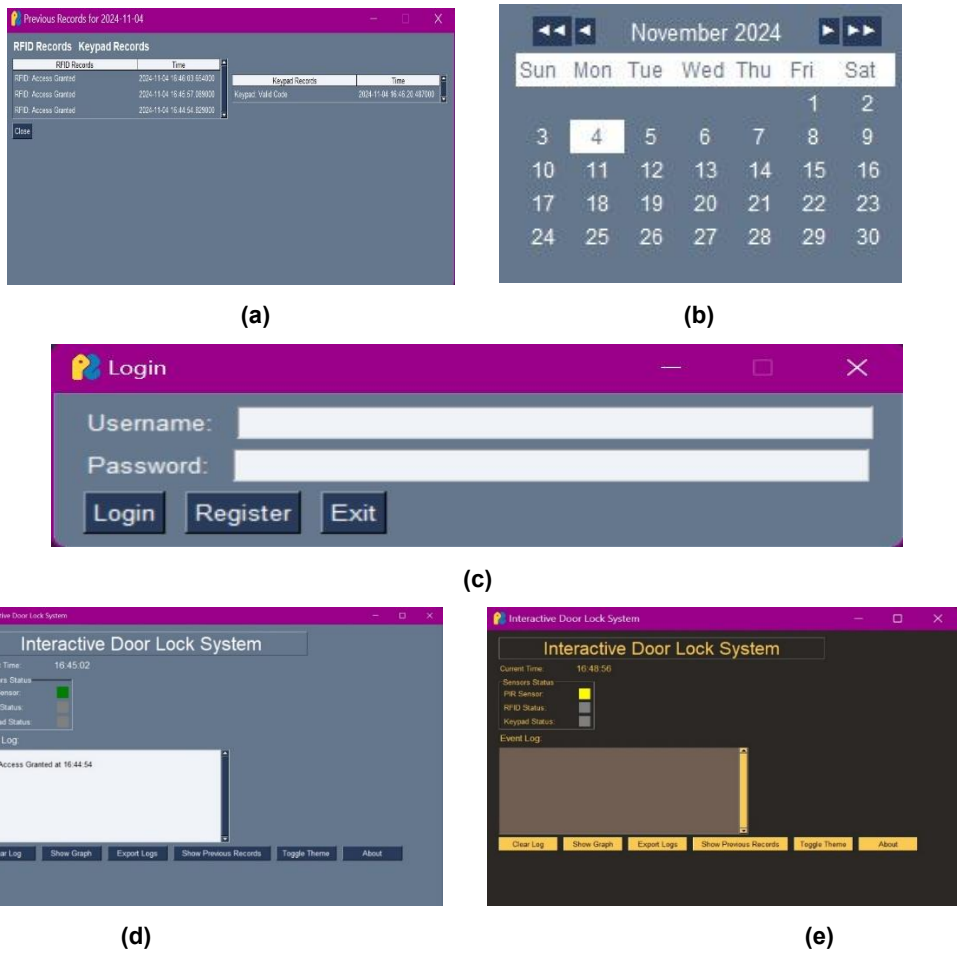
The GUI components are directly connected to the physical door lock system through serial communication, where an Arduino Nano sends signals to the GUI to update the indicators. This interaction allows the user to view the current status of the system in real-time. The system logs all events, including PIR detection, RFID authentication, and keypad access attempts, in the GUI's event log panel. The events are also stored in an Oracle SQL database, which acts as the central repository for all access data. The Oracle SQL database keeps a detailed record of:

Table 1. Example Records of Event Logs

Type Of Event	Timestamp Of The Event	Event Status
PIR Detection	2024-11-13 10:15:45	Successful
RFID Access Granted	2024-11-13 10:16:10	Successful
RFID Access Denied	2024-11-13 11:20:05	Denied
Keypad Access Granted	2024-11-13 12:05:30	Successful
Keypad Access Denied	2024-11-13 13:45:22	Denied

The Oracle database is used to ensure that access records are stored safely and are easily retrievable for further analysis. The database integration helps maintain a historical record of access activities, which is crucial for identifying anomalies and improving security. Whenever a new event occurs, the GUI captures it and logs it into the Oracle SQL database through a direct connection. This data is also used by the anomaly detection model.The GUI allows users to visualize data using a graphical representation of event frequencies. A separate window is created using Matplotlib to show the occurrence of different events, such as how many times the door was accessed through RFID or keypad authentication. Users can easily view the frequency of various types of access to identify any unusual patterns.

Moreover, users have the option to export logs to a CSV file for offline analysis or for sharing purposes. This is implemented through Python's csv module, allowing users to store access logs securely. The exported CSV can be opened directly in the default spreadsheet viewer for convenience. The GUI offers buttons for exporting logs, clearing logs, and showing graphs, providing ease of use for the end user. The GUI also includes a login system, Fig.5(c) which ensures that only authorized individuals can access and manage the door lock data. This is crucial for maintaining the security of the door lock system. The login credentials are stored in the Oracle SQL database, and users can either register a new account or log in to access the main system. Once logged in, users can access detailed information about previous access records through a calendar-based view. By selecting a specific date, users can retrieve the history of door access events on that day, providing detailed insights into who accessed the door and when. This feature is particularly useful for parents or homeowners who want to monitor daily activities in and out of their premises.



(a) (b)

(c)

(d) (e)

Figure 5. Different aspects of GUI

The calendar feature in the GUI allows users to easily navigate and review historical access records for the door lock system based on specific dates. This feature is particularly helpful for tracking and auditing access patterns and for monitoring security over time:

1) Calendar-Based Selection: The calendar view is implemented in the GUI using PySimpleGUI's Calendar Button Fig.5(a). Users can click on the button to bring up a calendar interface, from which they can select a date of interest. Once a date is selected, the system retrieves all access events that took place on that date. This includes entries made using either RFID cards or keypad authentication.

2) Fetching Data from Oracle Database: Upon selecting a date, the GUI connects to the Oracle SQL database to fetch the relevant records. The show_records_for_date function queries the database to get all entries for the chosen date. The database contains details about the type of event (e.g., RFID access granted/denied, keypad valid/invalid), the timestamp of the event, and the specific method of access. These details are presented to the user in a clear tabular format.

3) Display of Records: After the records are fetched, they are presented in a new window in the GUI. The records are divided into two categories: RFID records and Keypad records. Each category is displayed in a separate table for better organization. This separation allows users to quickly see how the door was accessed throughout the day, providing more granular control and monitoring capabilities Fig.5(b).

EVENT_TYPE	EVENT_TIME
RFID: Access Granted	10-OCT-24 09.28.49.135000 AM
Keypad: Valid Code	10-OCT-24 09.28.55.005000 AM
RFID: Access Granted	04-NOV-24 04.44.54.829000 PM
RFID: Access Granted	04-NOV-24 04.45.57.089000 PM
RFID: Access Granted	04-NOV-24 04.46.03.654000 PM
Keypad: Valid Code	04-NOV-24 04.46.20.487000 PM

Figure 6. Table data in Oracle sql

4) Use Cases for Calendar Feature: The calendar feature serves multiple use cases, including security monitoring, anomaly detection, and auditing purposes. Users can easily identify who accessed the door and when, allowing them to cross-check the data against expected behavior and maintain an audit trail of all access events.

5) User-Friendly Interaction: The intuitive calendar interface makes it easy for even non-technical users to select a date and view historical data. The GUI provides a smooth interaction where users can click the "Show Records" button, and the system immediately displays all relevant logs Fig.5(d),Fig.5(e).

6) Timely Analysis of Events: Combined analysis of PIR, RFID, and keypad trends can identify suspicious behavior, such as loitering or unauthorized attempts at entry, especially during non-working hours. Monitoring trends over time aids in recognizing patterns and peak times of activity, allowing for proactive enhancements to security measures and better protection against vulnerabilities.

The proposed door lock system seamlessly integrates hardware components, machine learning algorithms, and an easy-to-use GUI to provide a comprehensive security solution. By combin- ing multiple layers of authentication (RFID and keypad) with intelligent analysis of access patterns, and optimizing energy consumption using a PIR sensor, this solution aims to offer enhanced security that is both effective and user-friendly.

Fig.7 provides insights into the frequency of security-related events in a monitored environment. High PIR (Passive Infrared) counts indicate frequent movement, which could be normal in high-traffic areas or signal unauthorized activity in restricted zones,

particularly after hours. Investigating the timing of these events helps determine if they align with expected activity. RFID Granted versus RFID Denied trends reveal access patterns; higher RFID Granted counts suggest authorized use, while ele- vated RFID Denied counts could indicate unauthorized access attempts and potential intruders, prompting a review of security protocols. Keypad Valid versus Keypad Invalid trends highlight user behavior, where numerous invalid entries may indicate user error or attempts to guess the code, suggesting security risks. This data can be leveraged to enhance security protocols proactively, ensuring the environment remains secure and any vulnerabilities are promptly addressed via extensive analysis, with the help of an AutoEncoder machine learning model.

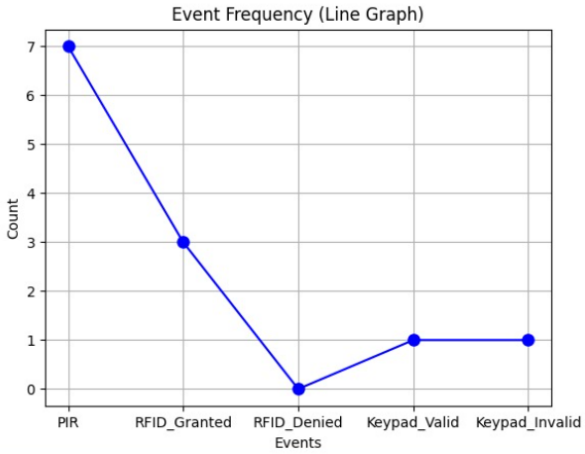


Figure 7. Event Frequency graph

2.3. Machine learning Implementation

The machine learning component of the door lock system is centered around anomaly detection. This part of the project uses historical access data to train a machine learning model that can identify unusual or suspicious access attempts, thus enhancing the overall security of the door lock system. The model implemented is based on an autoencoder, which is particularly effective for detecting anomalies. Below is a detailed breakdown of the steps involved in the implementation:

The first step in creating the anomaly detection model is to collect and prepare the data. The system continuously logs all door access events into an Oracle SQL database, which includes details such as the type of access (e.g., RFID or keypad), the timestamp of each access, and the outcome of the access (e.g., access granted or denied). Once the data is collected, the features are extracted. These features include the year, month, day, hour, minute, and second of the access event. Additionally, a flag is created to indicate whether the access occurred during working hours. This feature extraction helps capture seasonal or timebased trends in the access data. After the feature extraction, the data is standardized using StandardScaler from Scikit-Learn. Normalization ensures that all features are on the same scale, making it easier for the machine learning model to learn effectively. The normalization process is defined as:

$$X_{scaled} = \frac{X - \mu}{\sigma} \quad (1)$$

where:

- X : The raw feature,

- μ : The mean of the feature,
- σ : The standard deviation of the feature.

The anomaly detection model used here is an autoencoder. An autoencoder is a type of neural network designed to learn a compressed representation of input data and reconstruct it. The model is trained exclusively on normal access events, meaning it learns what normal access patterns look like. When an unusual activity occurs, it cannot be reconstructed by the autoencoder, resulting in a high reconstruction error.

The architecture of the autoencoder includes the following components. The input layer takes in the normalized features of the access data. Multiple hidden layers reduce the dimensionality of the data and then expand it back to the original size. This helps the model learn an efficient representation of normal access events. The output layer reconstructs the original input from the compressed representation.

The autoencoder is trained to minimize the reconstruction error, which is the difference between the original input and the reconstructed output. The reconstruction error, also known as the Mean Squared Error (MSE), is defined as:

$$\text{Reconstruction Error (MSE)} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \quad (2)$$

where:

- x_i : The original input
- $\hat{x}_i$: The reconstructed input.
- N : The number of data points

The model is trained using only normal access data, which includes events that took place during expected working hours or with authorized credentials. By training the model on this data, it learns to recognize typical patterns of door access activities. The training is performed using Mean Squared Error (MSE) as the lossfunction. The MSE measures how well the model can recreate theinput data after encoding and decoding it. The formula for MSE during training is:

$$\text{Loss (MSE)} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \quad (3)$$

where:

- x_i : The original input
- $\hat{x}_i$: The reconstructed input.
- N : The total number of samples.

Once the model is trained, a threshold is set to classify an access event as normal or anomalous. During inference, the model reconstructs the features of each access event, and the reconstruction error is computed by calculating the MSE between the original event and its reconstruction. A threshold is set based on the 85th percentile of the reconstruction errors from the training data. This ensures that most normal events fall below this threshold, and anything above this threshold is flagged as an anomaly. Any event with a reconstruction error greater than this threshold is considered anomalous. The threshold is computed as:

$$\text{Threshold} = \text{Percentile}_{85}(\text{Reconstruction Errors from Normal Data}) \quad (4)$$

In addition to the model-based anomaly detection, manual rules are incorporated to enhance the system's security. One such rule is that any access event occurring after 8:30 PM is flagged as anomalous. This hybrid approach combines both data-driven anomaly detection from the machine learning model and rule based detection from manually set

policies. The hybrid system ensures a more reliable detection mechanism by combining machine learning with manual policies. This is especially useful in situations where the model might not have enough data to learn certain patterns, such as infrequent night-time access.

During real-time operation, each access event is processed by the anomaly detection system. The following steps are performed:

1. The features of the new event are extracted and normalized.
2. The event is passed through the trained autoencoder, and the reconstruction error is calculated.
3. If the reconstruction error exceeds the predefined threshold, the event is flagged as anomalous.
4. If an anomaly is detected, the system immediately sends an alert to the owner.

This alert can be in the form of a pop-up message in the GUI as well as an entry in the event log. The purpose is to ensure that any suspicious activity is immediately brought to the attention of the owner.

Autoencoders provide several advantages for anomaly detection. First, they do not require labeled anomalous data, as they are trained only on normal events. This is especially useful because anomalous events are often rare and difficult to obtain. The model can be retrained periodically as more data becomes available, ensuring it adapts to any changes in access patterns over time. Furthermore, the combination of manual rules and the autoencoder ensures that the system remains effective even with limited data, as the rules act as a fallback mechanism when the model is unable to detect anomalies.

Figure 8 shows the relationship between the training loss (Mean Squared Error, MSE) and the number of epochs during model training. As observed in the graph, the training loss decreases steadily as the model learns to reconstruct normal access data. This decrease indicates that the model is effectively capturing the typical patterns of normal access events. Additionally, the validation loss, plotted alongside the training loss, helps in monitoring the model's generalization ability. Ideally, both the training and validation losses should decrease towards a low value without significant divergence, which could indicate overfitting. This graph is essential for understanding how well the model is learning and whether adjustments are needed to improve performance.

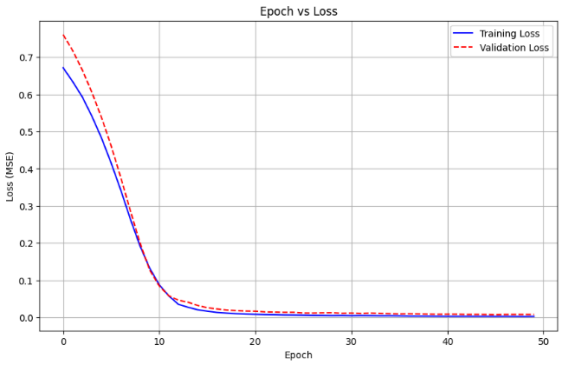


Figure 8. Loss vs Epoch during Model Training

Figure 9 provides a visual representation of the loss function's behavior with respect to two hyperparameters during the optimization process. The x and y axes represent two hyperparameters (e.g., weights or neuron counts), and the z axis represents the loss value. This 3D surface plot helps to visualize how the model's optimization algorithm navigates the parameter space in search of the minimum loss. The objective of the training process is

to find the global minimum, represented by the lowest point on the surface. However, the landscape often contains several local minima where the optimization process might get stuck. Understanding the loss landscape helps explain the optimization challenges faced during training and the model's journey toward an optimal solution.

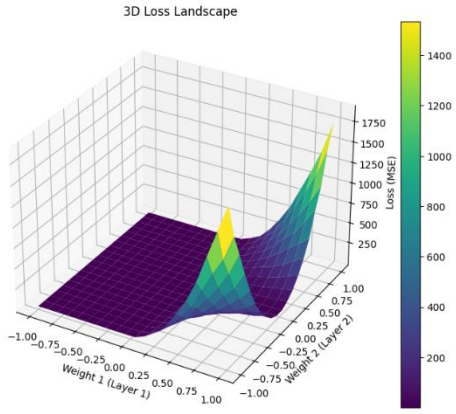


Figure 9. 3D Loss Landscape for the Autoencoder Model

These two graphs offer valuable insights into the model's training and optimization process. Figure 8 helps track the progress of the model's learning over time, while Figure 9 illustrates the challenges involved in the optimization process. Together, they provide a clear understanding of the model's behavior, performance, and the optimization techniques used during training.

4. Results

4.1. Hardware and system functionality results

The PIR sensor played a crucial role in efficiently managing the system's power consumption. It was able to detect any movement near the door, ensuring that the system was only powered on when necessary. This behavior not only improved the overall energy efficiency of the system but also contributed to extend- ing its lifespan by minimizing unnecessary power usage. When movement was detected, the PIR sensor triggered the activation of the system, enabling the access mechanisms to function.

Once the RFID reader was activated, it successfully granted access to authorized users, ensuring that only those with valid credentials could enter. For any unauthorized attempts, the RFID reader promptly denied access, providing a secure and seamless method of entry. This process helped maintain a secure environ- ment, reducing the risk of unauthorized access. Additionally, the keypad authentication provided an alternative access method for users, ensuring flexibility in the system. If an invalid password was entered, the buzzer was activated to alert the user about the failed access attempt, thereby notifying the owner of potential security breaches.

Furthermore, in cases where the system detected an intruder, an alert was immediately triggered. This alert was vital for ensuring the security of the space, as it notified the owner of suspicious activity, allowing them to take prompt action. The combination of these sensors and components worked together to form a highly secure and energy-efficient access control system.

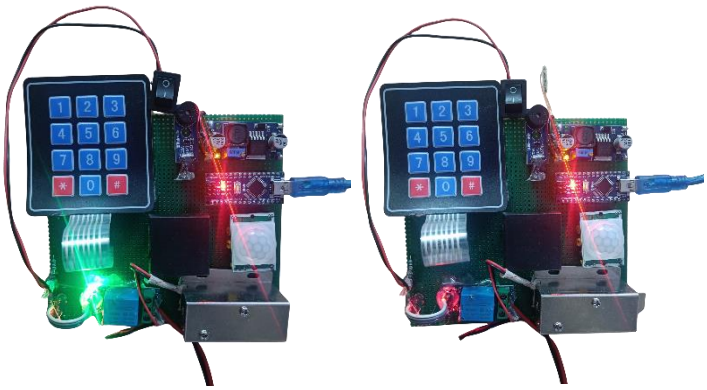


Figure 10. Access Accepted **Figure 11. Access Denied**

4.2. GUI results

The Graphical User Interface (GUI) played a pivotal role in ensuring the smooth interaction between the user and the door lock system. It provided real-time updates on the system's status, offering users immediate feedback about ongoing events. The GUI displayed important information such as whether movement was detected by the PIR sensor, whether the RFID authentication was successful or denied, and whether there were any keypad access attempts. This real-time visibility helped the user monitor the system's activity as it happened, ensuring that any unusual or unauthorized access could be quickly identified.

Figure 12. Alert Based On Anomaly Detection

Additionally, the event log in the GUI allowed users to track access events by displaying detailed information about who accessed the door and at what time. Each log entry also indicated whether the access attempt was successful or denied, providing a comprehensive record of all interactions with the door lock system. This feature ensured that users could keep an organized and accessible history of access events, which was particularly useful for security monitoring. The interactive features in the GUI further enhanced the user experience. For instance, the calendar interface allowed users to view past records easily, enabling them to track access events on specific dates. This feature made it simple to retrieve historical data, whether for security audits or to monitor regular access patterns. Furthermore, the option to export logs to a CSV file was another valuable tool, enabling users to download the event logs and store or analyse them outside the system. This functionality provided a high level of flexibility, allowing users to back up or further

examine the recorded data. Together, these features made the GUI a highly effective, user-friendly tool for managing and monitoring the door lock system.

4.3. Machine learning model results

This section presents the results of the machine learning model used for anomaly detection in the door lock system. The autoencoder model was trained to detect abnormal access attempts by measuring the reconstruction error. Various graphs are used to evaluate the model's performance and provide insights into its effectiveness.

Figure 13 shows the 3D Visualization of Reconstruction Errors. This graph provides a three-dimensional view of how reconstruc- tion errors vary across three features: hour, minute, and second. Each point represents a reconstruction error, and the color gradient indicates the magnitude of the error. Anomalous access events tend to have higher reconstruction errors, and these events are shown as points that deviate from the normal pattern, marked by higher values in the color scale. The model successfully detects these deviations in the three-dimensional feature space, helping identify suspicious access attempts.

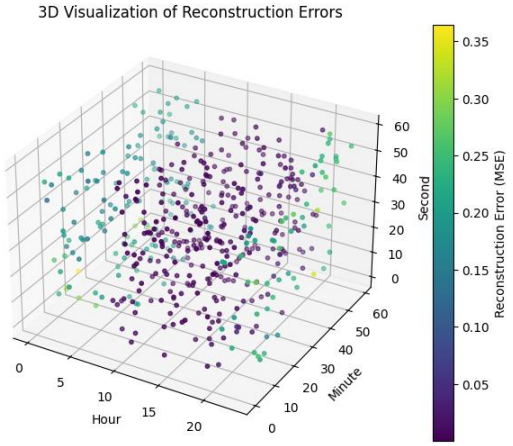


Figure 13. 3D Visualization of Reconstruction Errors for Anomaly Detection

Figure 14 presents the Correlation Heatmap of Features. This heatmap visualizes the correlations between various features, including year, month, hour, minute, second, working hours, and reconstruction error. It can be observed that features such as the hour and minute have a notable correlation with the reconstruction error. The working hours feature, which indicates whether an event occurred during regular working hours, shows a strong negative correlation with reconstruction errors, suggesting that the model performs better during these hours. This analysis helps in understanding which features contribute the most to anomaly detection.

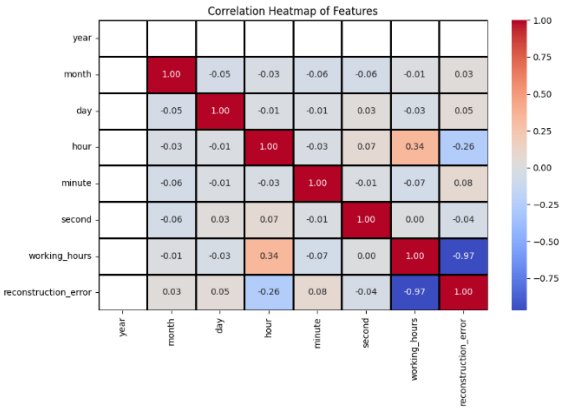


Figure 14. Correlation Heatmap of Features

The Anomaly Detection: Reconstruction Error vs Time graph, shown in Figure 15, tracks reconstruction error over time. The blue dots represent normal events, while the red dots represent anomalies. The horizontal dashed line indicates the threshold above which reconstruction errors are considered anomalous. As seen in the graph, the reconstruction error remains low for most of the data points, but for events above the threshold, the points shift to the red region, indicating anomalies. This helps visualize how the model can automatically flag suspicious activities based on the reconstruction error.

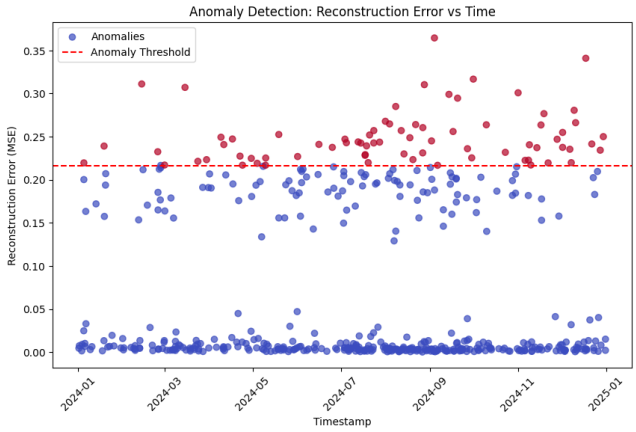


Figure 15. Anomaly Detection: Reconstruction Error vs Time

The Histogram of Reconstruction Errors, shown in Figure 16, displays the frequency of reconstruction errors for both normal and anomalous events. The histogram clearly shows that the majority of normal events have a low reconstruction error, while the anomalous events have a higher reconstruction error. This distinction helps the system detect anomalies based on how poorly the autoencoder is able to reconstruct the event. The higher the reconstruction error, the more likely it is that the event is anomalous.

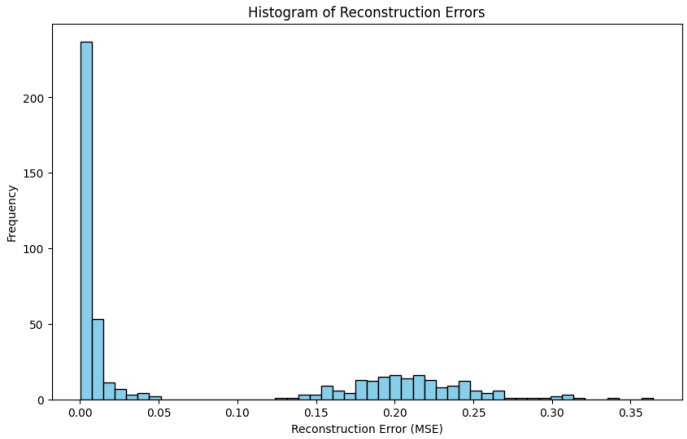


Figure 16. Histogram of Reconstruction Errors from Normal Data

Figure 17 shows the Box Plot of Reconstruction Errors for normal and anomalous events. This plot helps visualize the spread of reconstruction errors. The lower reconstruction errors for normal events are clearly separated from the higher errors of anomalous events. The plot also highlights any outliers in the reconstruction error, which may correspond to particularly unusual access attempts. This box plot serves as a further confirmation that the model effectively distinguishes between normal and anomalous access events.

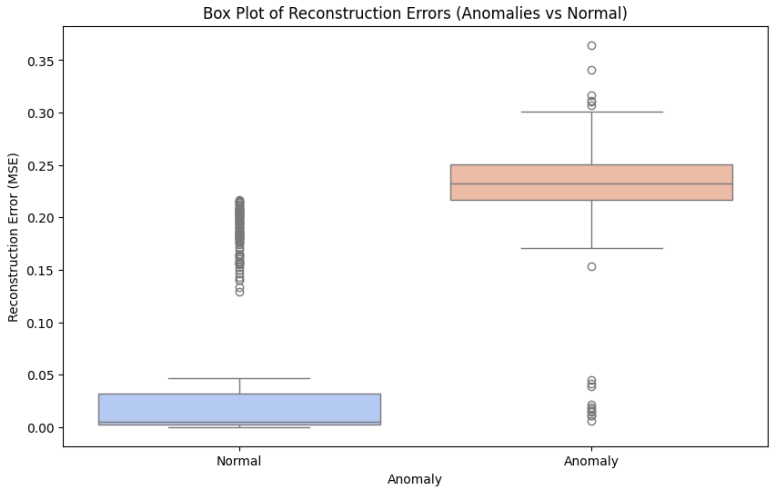


Figure 17. Box Plot of Reconstruction Errors (Anomalies vs Normal)

Together, these visualizations confirm the model's ability to detect anomalies based on reconstruction errors. The 3D vi- sualization, correlation heatmap, anomaly detection over time, histogram, and box plot all highlight different aspects of the model's performance. These results demonstrate the effectiveness of the autoencoder model in detecting unusual access attempts, thereby enhancing the security of the door lock system by quickly identifying potential threats.

4.4. Oracle SQL database integration

All access events were successfully recorded in the Oracle SQL database, ensuring that a complete and detailed history of door access was maintained. This allowed the system to store important data on every access attempt, making it easy for users to retrieve and review information about door activity at any given time. By having this historical data stored

securely, users could track patterns and identify unusual events, contributing to the overall security of the system.

The calendar feature in the GUI was especially useful for accessing past data. It allowed users to easily search for and view historical records by selecting specific dates. This functionality made it convenient to quickly retrieve information on access events from any particular day, helping users stay informed about system activity over time.

Additionally, the data was organized into two main categories: RFID records and keypad records. This clear organization of data ensured that users could easily distinguish between access attempts made with RFID cards and those made with keypads. By separating the two types of access, the system made it simpler for users to monitor and review different types of door interactions, enhancing the overall clarity and usability of the monitoring process.

4.5. Security enhancement

The buzzer and alert mechanisms worked effectively during unauthorized access attempts, providing an immediate response to potential security threats. Whenever an invalid access attempt was detected, either through RFID or keypad authentication, the buzzer was triggered to alert the user about the unauthorized attempt. In addition, if the system detected a potential intruder, an alert was immediately activated, notifying the user in real-time. This ensured that any unusual activity was quickly brought to the user's attention, enhancing the overall security of the system.

By incorporating both RFID and keypad authentication methods, the system provided a double layer of security. The use of two separate methods for access control significantly reduced the chances of unauthorized entry, as it would require bypassing both authentication methods to gain access. This two-factor approach ensured that even if one method was compromised, the other would still provide protection, making the system more reliable and secure.

5. Conclusion And Future Work

5.1. Conclusion

This paper discusses the development of a machine learning- based anomaly detection system designed to enhance the security of a door lock system. Traditional door lock systems rely on simple mechanisms such as RFID or keypads, which can be bypassed or compromised. The goal of this research was to introduce a more advanced and reliable security measure using machine learning, specifically by implementing an autoencoder model for anomaly detection.

The autoencoder model was trained on normal access data, meaning it learned the usual patterns of door access. The model operates by comparing the original access data with the recon- structed data generated by the autoencoder. When the reconstruc- tion error is low, it indicates that the access attempt is normal. However, if the reconstruction error is high, this suggests that the access event is unusual or potentially malicious, flagging it as an anomaly.

To evaluate the performance of the model, several important visualizations were used. The Loss vs Epoch graph shows how the model improved its performance over time by minimizing the loss function. The 3D Loss Landscape graph provides a deeper insight into how the model's parameters are adjusted during training. Additionally, the Histogram of Reconstruction Errors and Box Plot clearly illustrate how the model differentiates between normal access events and anomalies based on the reconstruction errors. These graphs help validate the model's effectiveness in detecting anomalous access attempts.

In real-world applications, the system has proven successful in identifying unauthorized access attempts by learning from patterns of normal access data. The model uses a hybrid approach, which combines machine learning techniques with manually set rules to enhance security. This combination ensures that the system remains reliable, even when faced with unpredictable or rare access patterns. The hybrid model can be adapted to various scenarios, making it a highly flexible and effective security solution for homes and businesses alike.

5.2. Future Work

While the current model works well, there are several ways it could be improved in the future:

1) Better Model Performance: The current model works well for normal access data, but its ability to detect more complex or rare anomalies could be improved. Exploring more advanced models, such as LSTM-based autoencoders, could help the system better capture patterns in sequential data and improve anomaly detection.

2) Handling Rare Events: The system might not perform as well for rare or unusual events that the model has not seen before. Using synthetic data or data augmentation techniques could help train the model on a wider range of potential anomalies.

3) Real-Time Data Processing: Currently, the system processes data stored in a database. In the future, it could be enhanced to process data in real-time using technologies like Apache Kafka for streaming data. This would allow the system to detect anomalies as they happen, without waiting for batch processing.

4) Edge Computing: To improve response time and reduce reliance on cloud services, the system could be moved to edge devices. This would allow the model to run directly on local hardware, making it faster and more efficient.

5) Integration with Other Security Systems: The door lock system could be combined with other security measures like cameras or motion detectors. This would allow the system to use multiple sources of data for better anomaly detection.

6) User Authentication: Future versions of the system could include more secure authentication methods, such as bio- metrics (fingerprints, facial recognition), in addition to RFID and keypad access.

7) Data Privacy and Security: As the system handles sensitive data, it is important to ensure that this data is secure. Future improvements could focus on implementing encryption and ensuring the system follows privacy regulations, such as the General Data Protection Regulation (GDPR).

8) Long-Term Adaptation: The system could be updated over time by retraining the model with new data. This would allow the system to adapt to changing access patterns and remain effective as new types of anomalies emerge.

In conclusion, the machine learning-based anomaly detection system is an effective way to enhance the security of door lock systems. By combining machine learning with rule-based anomaly detection, the system is both reliable and adaptable. Future work will focus on improving the model, adding new features, and ensuring the system remains secure and effective over time.

Acknowledgment

We would like to express our heartfelt thanks to Vellore Institute of Technology, Chennai, for offering the resources and platform that enabled the completion of this research project. We are grateful for the continuous support and encouragement, which played a key role in

our ability to explore and integrate various hardware and technologies. This opportunity has been essential in helping us successfully execute and finalize this work.

References

[1] *Nandhini, P., et al. (2021). RFID-Based Access Control System with Cloud Monitoring. IEEE Access.*

[2] *Singh, R., et al. (2020). IoT-Enabled Smart Lock System Using Keypad Authentication. International Journal of Smart Home Technologies.*

[3] *Patel, A., Sharma, R. (2019). Machine Learning for Anomaly Detection in Smart Home Security Systems. Journal of Intelligent Systems.*

[4] *Choudhury, B., et al. (2018). Survey on Anomaly Detection Techniques for IoT Devices. IEEE Communications Surveys & Tutorials.*

[5] *Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., Kim, T.-H. (2020). Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. IEEE Access.*

[6] *Mathew, M., Divya, R. S. (2022). Super Secure Door Lock System for Critical Zones. Mar Baselios College of Engineering.*

[7] *Shetty, S., Shetty, S., Patil, S., Vishwakarma, V. (2022). Review Paper on Door Lock Security Systems. Atharva College of Engineering.*

[8] *Eurostat. (2022). Robbery, Burglary and Theft, 2021-2022: Police Recorded Offences Per 100,000 Inhabitants. Retrieved from Eurostat Crime Statistics.*

[9] *Fernandez, L., et al. (2021). RFID Technology in Modern Smart Door Lock Systems: Security and Usability Perspectives. Journal of Advanced RFID Applications.*

[10] *Gupta, H., Patel, R. (2022). RFID-Based Dual Authentication System for Residential Access Control. International Journal of Smart Home Security.*

[11] *Banerjee, T., Singh, A. (2020). The Integration of RFID for Enhanced Door Lock Mechanisms in Smart Homes. IEEE Transactions on Consumer Electronics.*

[12] *Kumar, V., Sharma, N. (2019). Securing Smart Doors Using RFID and IoT Technologies. Journal of Secure Automation Systems.*

[13] *Puri, M., Gupta, S. (2021). RFID and IoT-Based Access Control Solutions for Smart Living. International Journal of Smart Technology and Systems.*

[14] *Ahmed, Z., Khan, F. (2023). Advancements in RFID-Based Locking Mechanisms for Smart Homes. Journal of IoT Security and Implementation.*

[15] *Lee, J., Choi, K. (2020). Review on RFID Applications for Secure Door Access in IoT Environments. IEEE Access.*

[16] *Ramachandran, P., Verma, S. (2018). Enhancing Home Security with RFID-Based Smart Door Locks. Journal of Embedded Security Solutions.*

[17] *Smith, C., Green, R. (2022). Next-Generation RFID Locks for Smart Homes: Trends and Challenges. Journal of Digital Security.*

[18] *Yadav, A., Iyer, M. (2019). Implementing RFID in Smart Lock Systems: A Case Study. International Journal of RFID and Smart Applications.*