Reimagining Low Level Security

Pranit Sehgal
School of Computing and Augmented Intelligence
Arizona State University
pvsehgal@asu.edu (1225456193)

Abstract

In the contemporary realm of security systems, there is a pressing need for innovative approaches that offer both robustness and user engagement. This project presents a novel security system utilizing the Arduino Nano 33 BLE Sense, which introduces a dual-layered approach to access control: a gesture-based lock and a rhythm-based lock. The gesture lock leverages the Arduino's gesture sensor, allowing users to employ hand gestures as a unique access code. In contrast, the rhythm lock advances the concept by incorporating keystroke dynamics analyzed through a machine learning (ML) model. Initially, the ML model, trained with a limited dataset of 100 data points, exhibited modest performance. However, upon expanding the dataset to 200 points and optimizing hyperparameters, the model's accuracy significantly improved. This project not only contributes to the field of secure access systems by integrating gesture and rhythm recognition but also enhances the user experience by making technology interaction more engaging and personalized. The successful implementation of this system underscores the potential of combining embedded hardware with ML algorithms, paving the way for more intuitive and secure access control solutions.

Introduction

Innovating Security: Beyond Traditional Locks

In the realm of security systems, especially in access control, the predominant reliance on static passwords or PINs has been a longstanding vulnerability. These methods, while simple, offer limited protection against determined intruders, as they can be easily compromised or guessed. This presents a significant problem in the security domain, where the integrity of access control systems is paramount. Moreover, the lack of user engagement and interactivity in traditional security methods often leads to a mundane user experience.

Research Gap and Novel Contributions

This project addresses the critical gap in the security field by integrating the concept of keystroke dynamics with a gesture-based system. Keystroke dynamics, the study of the time intervals between key presses, add an additional layer of security beyond the conventional PIN or password. Even if an unauthorized person knows the correct PIN, they would be unable to replicate the unique timing pattern of the legitimate user's keystrokes, making unauthorized access significantly more challenging. This approach is not widely adopted in standard security systems, especially those utilizing embedded systems like the Arduino Nano 33 BLE Sense.

The unique contribution of this project lies in its novel application of machine learning to the realm of keystroke dynamics. By applying ML algorithms to the timestamps of key presses, the system can accurately identify and authenticate the unique rhythm patterns of different users. This not only enhances security by adding an intricate layer that is difficult to replicate but also makes the system more dynamic and interactive.

Furthermore, this project stands out in its deployment and optimization of the ML models in a resource-constrained environment, like that of the Arduino Nano 33 BLE Sense. Adapting complex ML models to run efficiently on such devices, without compromising their accuracy, is a significant achievement in the field of embedded systems and machine learning.

In essence, this project traverses multiple dimensions of innovation - from system design and application to algorithm deployment and optimization. It presents a groundbreaking approach to secure access systems, marrying the robustness of ML with the versatility and user-friendliness of gesture and rhythm-based interactions.

Related Work

Building on Foundations: A Review of Existing Security Systems

The current state of the art in security systems primarily revolves around biometric recognition, RFID technology, traditional alphanumeric passwords, and pattern locks. Biometric systems, such as fingerprint and facial recognition, are widely praised for their uniqueness to each individual but face challenges in terms of privacy concerns and vulnerability to sophisticated spoofing attacks. RFID-based systems offer convenience but can be compromised through cloning or signal interception. Traditional passwords and pattern locks, while common, suffer from predictability and are vulnerable to brute-force attacks.

Gap in Existing Literature

Despite these advancements, a significant gap remains in the literature concerning the integration of temporal dynamics in access control. Most systems focus on ’what’ the authentication factor is (like a password or biometric trait) rather than ’how’ it is entered. The concept of using keystroke dynamics, particularly in the context of embedded systems like Arduino, is relatively underexplored. While keystroke dynamics have been studied in the domain of computer security, their application in physical access control systems, especially with a layer of machine learning analysis, is novel.

Unique Contributions in Perspective

This project expands on the existing body of work by incorporating the rhythm of user input as a key security factor, a concept not widely implemented in current security systems. The application of machine learning to analyze these timing patterns in the context of an Arduino-based system is an innovative approach that stands out from conventional methods. This approach adds a new dimension to security - the rhythm with which an input is entered becomes as important as the input itself.

Researchers have explored machine learning in various security applications, but its use in analyzing keystroke dynamics for physical access control is relatively unique. The project’s success in deploying and optimizing these ML models for a resource-constrained environment like Arduino adds a valuable contribution to the field. It demonstrates the feasibility and effectiveness of implementing sophisticated ML algorithms in smaller, embedded systems, broadening the scope of where and how machine learning can be applied in security technologies.

In summary, while building upon the principles established in existing security systems, this project diverges significantly in its application of rhythm recognition and machine learning in an embedded system context. It provides a fresh perspective on enhancing security measures, bridging a gap in the literature, and pushing the boundaries of current technology in the field.

System Design

Engineering a Dual-Layer Security System

The system designed in this project is a dual-layer security system using an Arduino Nano 33 BLE Sense. It comprises two main components: a gesture lock and a rhythm lock, both enhanced with machine learning for improved accuracy and security.

Gesture Lock

1. **Component:** The gesture lock utilizes the Arduino’s onboard APDS-9960 sensor, which is capable of detecting directional gestures.
2. **Functionality:** Users perform a sequence of hand gestures (like up, down, left, right) as a password. The sensor detects these gestures and sends the data to the Arduino for processing.

Rhythm Lock

1. **Component:** This lock uses a 4x4 keypad connected to the Arduino.
2. **Functionality:** When a user enters a PIN, the system records the time intervals between each keypress. These intervals are crucial as they form a unique rhythm pattern for each user.

Machine Learning Architecture

1. **Model Choice:** A TensorFlow/Keras Sequential model was chosen for its flexibility and ease of implementation on embedded systems.
2. **Architecture:**
   - Input Layer: Matches the shape of the training data.
   - Scalar Standardization Layer: Normalizes the data, crucial for effective learning.
   - Dense Layers: Multiple layers with ReLU activation to capture complex patterns in the data.
   - Dropout Layers: Included for regularization to prevent overfitting.
   - Output Layer: Softmax activation for multi-class classification.
3. **Reason for Choice:** This architecture was selected for its ability to efficiently learn from time-series data (key press intervals) and its compatibility with the computational constraints of the Arduino platform.

Data Preprocessing

1. **Standardization:** The raw time intervals between keypresses are standardized to have a mean of zero and a standard deviation of one. This process is vital for the model to perform effectively.
2. **Data Splitting:** The dataset is split into training and validation sets, ensuring the model is not biased and generalizes well.

Model Optimization for Embedded Systems

1. **Size Reduction:** The model’s size is crucial for deployment on the Arduino. Techniques like pruning and quantization were considered to reduce the model’s footprint without significantly impacting accuracy.
2. **Efficiency Improvement:** Optimization techniques were applied to ensure the model runs efficiently in real-time on the Arduino, a device with limited computational power and memory.

The system’s design intricately blends hardware components with advanced software algorithms, particularly in machine learning, to create a highly secure yet user-friendly access control system. The choice of components and the model architecture were dictated by the need for efficiency, accuracy, and real-time processing capabilities in an embedded system environment.

Evaluation Approach

Data Collection and Dataset Utilization

1. **Initial Dataset:** The project began with a small dataset of 100 data points, collected using the Arduino system. This dataset comprised timing intervals for different PIN types entered on the keypad.
2. **Expanded Dataset:** To enhance the model's performance, the dataset was expanded to 200 data points. This expansion provided a more diverse range of timing patterns, crucial for the model to learn and generalize effectively.

Evaluation Metrics

1. **Accuracy:** This metric measured the percentage of correctly identified PIN entries and was the primary metric for evaluating the model's performance.
2. **F-Score:** The F-score, combining precision and recall, was used to understand the model's balance between correctly identifying true positives and not misclassifying negatives.
3. **Model Size:** Given the limited memory and storage on the Arduino Nano 33 BLE Sense, the size of the trained model was a critical metric. It was important to ensure the model was compact enough for deployment without sacrificing accuracy.
4. **Power Consumption:** As the system is intended for real-world applications, power efficiency was crucial. The power consumption of the Arduino while running the model was monitored to evaluate the system's viability in energy-constrained scenarios.

Methodology

1. **Training and Validation:** The model was trained on the collected dataset, with a split of 80
2. **Testing:** After training, the model was tested on a separate set of data to evaluate its real-world performance.
3. **Performance Analysis:** Post-testing, the model's accuracy and F-score were calculated. Additionally, the model's size was measured, and power consumption metrics were gathered during its operation on the Arduino.

Additional Evaluations

- **Threshold Analysis:** The effectiveness of the threshold factor in enhancing security was assessed. This involved testing the model's ability to differentiate between correct and incorrect rhythms for the same PIN.
- **Real-Time Performance:** Since the system is designed for real-world use, its performance in real-time was critically evaluated. This included assessing the latency between input and authentication decision.

Results and Discussion

Performance Results

1. **Training and Validation:** The final machine learning model achieved a training accuracy of 83.56
2. **Testing:** On the testing dataset, the model maintained a high accuracy level of 82.61
3. **Real-Time Efficiency:** The system showed excellent real-time performance, with minimal latency observed between input (gesture or PIN entry) and system response. This efficiency is critical for user acceptance and practical application.

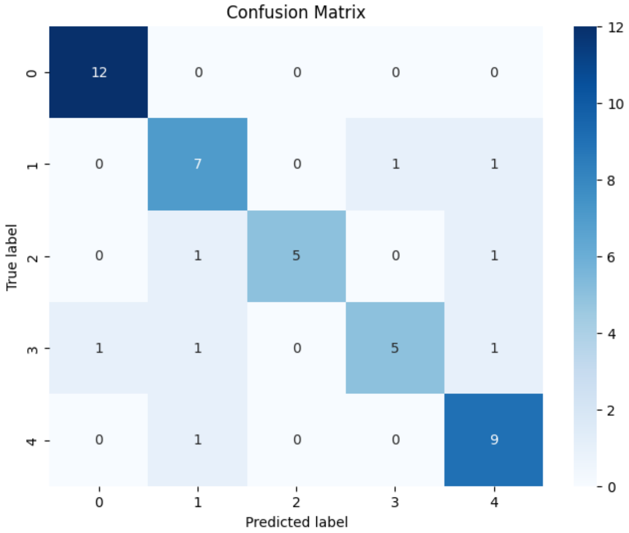

Figure 1: Confusion Matrix

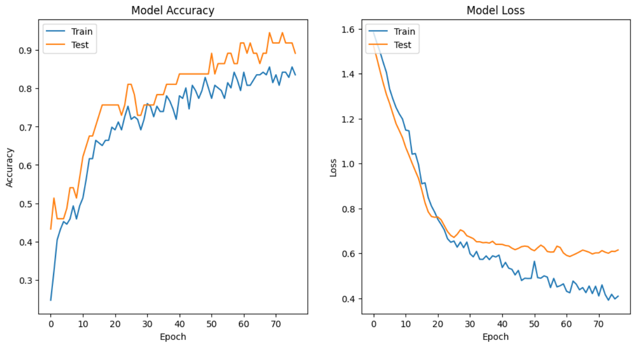

Figure 2: Model Accuracy and Loss

Future Work and Applications

Adapting to User Preferences: Evolution of the Rhythm Lock

Based on feedback from a survey conducted among classmates and friends, it was clear that entering a PIN 20 times for model training was viewed as cumbersome. This insight led to the development of a more user-friendly approach to setting up the rhythm lock.

Initial Solution: Simplified Rhythm Setup

1. **Simplified Entry Process:** Instead of 20 entries, users will now set their PIN by entering it just three times.
2. **Temporary Threshold Code:** The system calculates the average of the time intervals from these three entries and adds a margin of 0.100 seconds to account for human error. This average serves as the initial threshold for rhythm detection.
3. **Database Integration:** Each successful login attempt records the PIN's timing data. After 40 successful entries, this accumulated data is sufficient to run the ML model specifically tailored to the user's rhythm.
4. **Benefit:** This method simplifies the setup process, making it more practical and user-friendly while retaining the security benefits of rhythm analysis.

Alternative AI-Driven Approach Another proposed solution involves the development of an AI model capable of generating a comprehensive range of 5-character beats within a 3-second window.

1. **Automatic Beat Assignment:** When a user sets their PIN, the system automatically assigns a rhythm or beat to it, based on the AI model's database.
2. **Advantage:** This method further simplifies the process, eliminating the need for the user to input their PIN multiple times during the setup phase.

Project's Motive and Real-World Testing

The primary objective of this project was to test the feasibility of applying keystroke dynamics in a hardware-based security system. The successful implementation and positive results validate the concept of integrating rhythm analysis into physical access control systems.

Potential Applications and Broader Impact

This system has potential applications in various domains where secure yet user-friendly access control is required. Examples include residential security systems, secure access to personal devices, and even secure access in corporate settings.

The broader impact lies in demonstrating the practicality of ML and AI in enhancing traditional security systems, paving the way for more innovative, efficient, and user-centered security solutions in the future.

Conclusion

Looking ahead, the project is poised for further refinement and expansion. The focus will be on enhancing user experience and simplifying the setup process while maintaining the high-security standards achieved. These advancements will solidify the project's standing as a pioneering effort in the practical application of keystroke dynamics in hardware security.