IoT Engineer intern at Dataviv Technologies

Internship Report

Electrical Engineering Department
Veermata Jijabai Technological Institute

A thesis submitted for the degree of

Bachelor's of Technology in Electronics Engineering

(Amir Zakaria)

July 2024

IoT Intern at Dataviv Technologies

In Partial Fulfilment of the Requirements
for the Degree of
BTech in Electronics & Telecommunication Engineering

Submitted by
Amir Zakaria
(211090078)

Under the guidance of
Dr. Niteshkumar Agrawal and Dr. Gajanan Galshetwar

Electrical Engineering Department
Veermata Jijabai Technological Institute, Mumbai - 400 019
(Autonomous Institute Affiliated to University of Mumbai)

June 2024

Statement by the Candidate

I wish to state that the work embodied in this dissertation titled “IoT Intern at Dataviv Technologies”, forms my own contribution to the work carried out under the guidance of ‘Dr. Niteshkumar Agrawal and Dr. Gajanan Galshetwar‘” at the Veermata Jijabai Technological Institute. This work has not been submitted for any other Degree or Diploma of any University/Institute. Wherever references have been made to previous works of others, it has been clearly indicated.

Amir Zakaria
VJTI, Mumbai
July 2024

Certificate

This is to certify that Amir Zakaria, a student of Final Year BTech in Electronics and Telecommunication Engineering, has completed the dissertation "IoT Engineer Intern at Dataviv Technologies" to our satisfaction.

Dr. Niteshkumar Agrawal and Dr. Gajanan Galshetwar
Supervisor

Dr. Sushama R Wagh
Head, Electrical Engineering Department

Director
VJTI, Mumbai

Approval

The dissertation, **"IoT Engineer intern at Dataviv Technologies"** submitted by **Amir Zakaria**, is found to be satisfactory and is approved for the degree of B.Tech. in Electronics and Telecommunication Engineering from Veermata Jijabai Technological Institute affiliated to University of Mumbai.

Supervisor:

Dr. Niteshkumar Agrawal

Dr. Gajanan Galshetwar

Examiner

Acknowledgements

“The only limit to our realization of tomorrow is our doubts of today.”
- Franklin D. Roosevelt

First and foremost, I would like to express my deepest gratitude to my mentor, **Dr. Niteshkumar Agrawal and Dr. Gajanan Galshetwar**, for his invaluable guidance, unwavering support, and insightful feedback throughout my internship.

I am immensely grateful to the Head of Department, **Dr. Sushama R Wagh**, for providing me with this incredible opportunity and for fostering a nurturing environment for learning and growth.

I would like to extend my heartfelt gratitude to **Dr. Sachin D. Kore**, the Director of VJTI, for the continuous academic support and resources that enabled me to pursue this internship and enhance my practical knowledge.

Lastly, I am deeply grateful to all the individuals at Dataviv Technologies who shared their knowledge and experiences with me, making this internship a memorable and enriching experience. The skills and insights I have gained will undoubtedly play a vital role in my future endeavors.

June, 2024
VJTI, Mumbai

Amir Zakaria

Abstract

During my internship at Dataviv Technologies in IoT, I gained a strong foundation in hardware based problem-solving and analytical thinking. One of the most significant accomplishments of this experience was the development of a fully automated Linear Actuator aimed at empowering users to effectively control systems remotely via a web interface. This project involved end-to-end development, from conceptualizing the application's features to building the solution on the hardware aspect, software aspect as well as the mechanical part. Through this project, I enhanced my technical proficiency, improved my understanding of IoT systems, and contributed to creating a user-friendly solution.

Throughout the internship, I gained hands-on experience with a range of hardware-focused tools and technologies such as Embedded C, Python for hardware interfacing, SQL (PostgreSQL) for data storage, Arduino C for microcontroller programming, and PCB design for creating custom circuits. I was involved in the design, manipulation, and analysis of hardware components, including sensors and actuators, ensuring smooth integration into systems. Collaborating with cross-functional teams and working under the guidance of experienced mentors enhanced my understanding of real-world hardware challenges and the practical applications of embedded systems and electronic design.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

The advent of the Internet of Things (IoT) has revolutionized industries by enabling intelligent communication between devices, sensors, and systems. This paradigm shift has opened new avenues for innovation in automation, connectivity, and data-driven decision-making. Dataviv Technologies, a pioneer in IoT solutions, provided an excellent platform for applying theoretical knowledge to practical challenges during my internship.

The primary focus of my internship was to design and implement hardware solutions for real-world IoT applications, with projects encompassing precision actuators, wireless control systems, and multi-device communication. The work demanded an interdisciplinary approach, combining mechanical design, embedded systems programming, and IoT protocol integration.

My motivation stemmed from the desire to bridge the gap between theoretical learning and industrial implementation. This internship presented a unique opportunity to contribute to innovative projects while developing a deep understanding of hardware design and IoT integration. Solving challenges such as designing custom PCBs, ensuring seamless communication between microcontrollers, and optimizing system performance fueled my passion for engineering and technology.

Furthermore, the internship aligned with my long-term goal of developing cutting-edge IoT systems that enhance efficiency and automation in industrial and consumer applications. The dynamic environment at Dataviv Technologies provided invaluable exposure to the complexities of real-world engineering, instilling confidence and equipping me with the skills to excel in future endeavours.

1.2 Objectives of the Internship

Objectives

The primary objectives of my internship at Dataviv Technologies were structured to ensure a comprehensive learning experience and significant contributions to ongoing projects.

These objectives included:

1. Hardware Design and Development:
    - Design and develop a precision linear actuator with custom mechanical components, ensuring high accuracy and reliability for IoT applications.
    - Create and optimize a wireless AC motor control system integrating relays, microcontrollers, and PCBs for efficient and safe operations.

2. IoT System Integration:
    - Develop a LAN-based multi-microcontroller communication system with host-slave logic for seamless device coordination.
    - Implement IoT protocols such as MQTT and HTTP to enable remote monitoring and control of hardware systems.

3. Custom Electronics Development:
    - Build an IR-based remote control system tailored to industrial requirements.
    - Enhance actuator functionality by integrating stepper motors, stepper drivers, and ESP8266 microcontrollers.

4. Testing and Optimization:
    - Conduct rigorous testing of hardware and software systems to ensure durability, reliability, and efficiency.
    - Identify and resolve challenges related to performance bottlenecks and hardware-software integration issues.

5. Documentation and Reporting:
    - Document all phases of design, development, and testing to create comprehensive reports for future reference and scalability.
    - Prepare project summaries and performance analysis for stakeholders.

6. Skill Development:
    - Gain hands-on experience in hardware architecture, embedded programming, and IoT integration.
    - Enhance professional skills in teamwork, problem-solving, and project management.

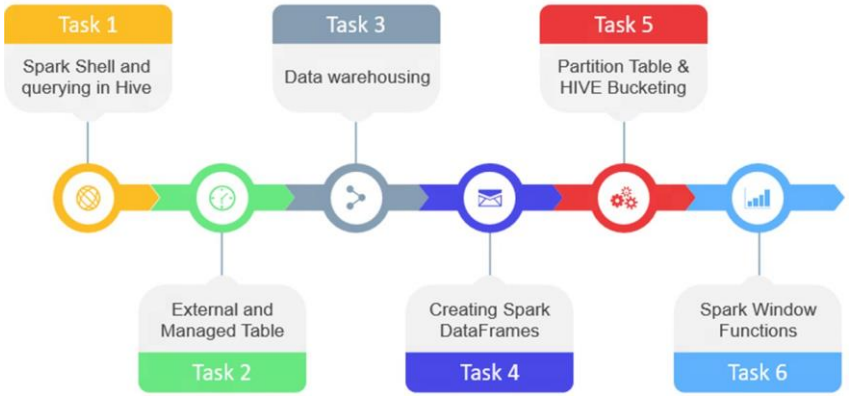


Figure 1.1: Learning Curve

1.3 Scope of the Work

The scope of work during my internship at Dataviv Technologies encompassed a broad range of tasks and responsibilities, aimed at addressing real-world challenges and contributing to the company's IoT initiatives. The work involved designing, developing, and testing IoT-based hardware systems while ensuring scalability, reliability, and efficiency.

Key areas of focus included:

1. Precision Linear Actuator Development

- Designed and implemented a custom mechanical system with a GT belt mechanism, V-plate support, and a 4-axis channel.
- Conducted detailed mechanical calculations and optimized the design for industrial applications requiring precise movement.

2. Wireless Control System Design

- Developed a wireless AC motor control system integrating relays, microcontrollers, and PCBs.
- Ensured safe and efficient remote operation of high-power devices.

3. IoT Communication Systems

- Built a LAN-based multi-microcontroller communication system using host-slave logic, enabling seamless coordination between multiple devices.
- Implemented IoT protocols such as MQTT and HTTP for remote monitoring and control.

4. Custom Electronics Development

- Created an IR-based remote control system tailored to specific industrial requirements.
- Enhanced actuator functionality by integrating stepper motors, drivers, and ESP8266 microcontrollers for efficient operation.

5. Testing and Debugging

- Performed comprehensive testing of hardware systems to ensure reliability, durability, and accuracy under various conditions.
- Identified and resolved challenges related to hardware-software integration and system performance bottlenecks.

6. Documentation and Knowledge Sharing

- Documented the entire development process, including design schematics, programming workflows, and testing protocols, to ensure reproducibility and future scalability.
- Prepared detailed reports to communicate project outcomes and performance metrics to stakeholders.

7. Future Expansion and Scalability

- Proposed strategies for scaling the developed systems for industrial IoT use cases, including predictive maintenance and real-time analytics using AI/ML integration.

By encompassing these aspects, the scope of work ensured a comprehensive approach to IoT engineering, allowing for technical contributions while fostering skill development and understanding of industrial practices.

1.4 Organization of the Report

Dataviv Technologies is a dynamic IoT solutions provider specializing in designing and developing hardware and software systems for industrial and consumer applications. The company focuses on delivering innovative IoT-based products that enhance automation, connectivity, and operational efficiency.

Dataviv's expertise spans multiple domains, including precision hardware design, embedded systems programming, and IoT protocol integration. Its mission is to provide cutting-edge solutions that solve real-world challenges while maintaining high standards of quality and innovation.

During my internship, I contributed to several projects aligned with the company's goals, involving the development of precision actuators, custom electronics, and LAN-based communication systems. These projects emphasized both technical rigor and practical applications, offering an invaluable opportunity for learning and growth.

Key Responsibilities:

- Designed and developed a precision linear actuator with custom mechanical components and IoT integration.
- Built a wireless AC motor control system with microcontroller logic and PCB design for remote operation.
- Implemented a multi-microcontroller communication system using host-slave logic for seamless device coordination.
- Developed an IR-based remote control system tailored for specific industrial requirements.
- Documented workflows, prepared technical reports, and collaborated with the team to deliver high-quality outcomes

Chapter 2

Technical Concepts

1. **Programming Languages:**
    - **Python:** Used for control algorithms, hardware interfacing, and data processing.
    - **Arduino C:** A simplified version of C/C++ used to program Arduino boards, enabling quick development for embedded systems and hardware interfacing.
    - **Embedded C:** A high-level programming language designed for programming embedded systems, providing low-level control over hardware with efficient resource usage.
2. **Software Tools:**
    - **MATLAB:** Utilized for modelling and algorithm testing.
    - **CAD Tools (e.g., FreeCAD, Eagle):** For designing custom PCBs and mechanical components.
3. **Hardware Platforms:**
    - **Microcontrollers (e.g., ESP8266, Arduino):** For controlling sensors, actuators, and communication.
    - **Stepper Motor and Driver (A4988):** For motion control.
    - **Relay Modules & Power Regulation:** For managing high-power devices and voltage stability.
4. **Development Environments:**
    - **Arduino IDE & Embedded C:** For firmware development and debugging.
    - **Version Control (Git):** For code management and collaboration.

These technologies enabled the successful development and integration of the project, balancing hardware and software components.

A) Software Related Concepts

2.0.1 Python

Python is a high-level programming language that emphasizes code readability and simplicity. It is widely used for both scripting and building complex applications. Python's versatility makes it suitable for a wide range of tasks, from web development to machine learning. In this project, Python is used to handle data processing, control systems, and integrate with other tools.

Example Code:

```
import os
def process_data(file_path):
   with open(file_path, 'r') as file:
      data = file.read()
   return data
```

Key Pointers:

- Supports a vast ecosystem of libraries, including NumPy and pandas for data processing.
- Easy integration with other technologies like databases and web frameworks.

2.0.2 Embedded C

Embedded C is a programming language tailored for developing software for embedded systems, particularly in hardware-intensive applications. It is used to program microcontrollers and other hardware devices. In this project, Embedded C is used to write firmware for controlling hardware platforms such as sensors and actuators.

Example Code:

```
#include <avr/io.h>
void init_motor() {
   DDRB |= (1 << PB0);  // Set pin for motor control
}
void start_motor() {
   PORTB |= (1 << PB0);  // Activate motor
}
```

Key Pointers:

- Efficient in handling hardware-level control.
- Utilizes direct memory access for fast execution.

2.0.3 Arduino C

Arduino C is a simplified version of C used in programming Arduino microcontrollers. It is widely used in prototyping and embedded systems projects due to its simplicity and vast community support. In this project, Arduino C is used to interface with sensors and other components of the system.

Example Code:

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Set up LED pin
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // Turn on LED
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);  // Turn off LED
  delay(1000);
}
```

Key Pointers:

- Focuses on simplicity and ease of use for hardware prototyping.
- Large ecosystem of libraries and shields for easy expansion.

2.0.4 FreeCAD

FreeCAD is an open-source 3D computer-aided design (CAD) software that can be used for creating detailed designs of mechanical components and assemblies. In this project, FreeCAD is used to design and model hardware components such as enclosures and brackets for sensor mounting.

2.0.5 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a simple yet powerful tool used for writing, compiling, and uploading code to Arduino boards. The IDE supports both Arduino C and other embedded programming languages, making it a critical tool for embedded system development.

2.0.6 Git

Git is a distributed version control system commonly used for software development and tracking changes in code. In this project, Git is used for version control to manage changes in both hardware and software components of the project.

B) Hardware Related Concepts

2.0.1 ESP8266

The ESP8266 is a low-cost Wi-Fi microchip with a full TCP/IP stack and microcontroller capability. It is widely used in Internet of Things (IoT) projects for wireless communication. The ESP8266 enables devices to connect to Wi-Fi networks and exchange data with servers or other devices over the internet. It is often programmed using the Arduino IDE or similar tools for custom applications.

Key Pointers:

- Ideal for Wi-Fi-enabled IoT projects.
- Supports GPIO pins for interfacing with sensors and actuators.
- Low power consumption for battery-operated devices.

2.0.2 Stepper Motor

A stepper motor is a type of DC motor that divides a full rotation into a series of discrete steps. This precise control makes stepper motors ideal for applications requiring accurate positioning and rotation. In this project, a stepper motor is used to control mechanical movements such as actuators or moving parts in a system.

Key Pointers:

- Provides precise control of rotational position without needing encoders.
- Often used in 3D printers, robotics, and CNC machines.
- Requires a stepper driver to control rotation.

2.0.3 Stepper Driver A4988

The A4988 is a microstepping driver for controlling stepper motors. It allows for precise control of stepper motors by managing current flow to the motor coils, enabling finer steps of motion. The A4988 is capable of driving bipolar stepper motors and is often used in robotics, CNC machines, and 3D printers.

Key Pointers:

- Supports up to 2A per coil and can drive bipolar stepper motors.
- Provides microstepping capability for smoother motion.
- Includes over-temperature and over-current protection.

2.0.4 IC 7805

The 7805 is a voltage regulator IC that provides a constant 5V output from a higher voltage input. It is commonly used to power components like microcontrollers and sensors in embedded systems. The 7805 is a linear regulator, meaning it converts excess voltage into heat, which requires adequate cooling for stable operation.

Key Pointers:

- Provides a fixed 5V output from a wide range of input voltages (7-35V).
- Often used to power microcontrollers, sensors, and other low-power devices.
- Requires heat dissipation for high current loads.

2.0.5 Step Down Module (DC-DC Converter)

A step-down module, also known as a buck converter, is used to convert a higher input voltage to a lower, regulated output voltage. It is more efficient than linear regulators like the 7805, as it reduces energy loss by switching rapidly and controlling the output voltage.

Key Pointers:

- Provides higher efficiency compared to linear regulators.
- Can step down voltages from sources like 12V or 24V to 5V or other required voltages.
- Widely used in battery-powered devices for power conservation.

2.0.6 Relay Module

A relay module is an electrically operated switch used to control high-voltage devices with a low-voltage signal. The relay can be controlled by a microcontroller or GPIO pins to switch large currents, enabling the control of motors, lights, or other heavy-duty devices from low-power systems.

Key Pointers:

- Allows control of high-voltage or high-current devices using a low-voltage signal.
- Commonly used for automation tasks like controlling home appliances.
- Includes protection circuits to handle inductive loads safely.

2.0.7 GPIO PCB Board

A GPIO (General Purpose Input/Output) PCB board is a custom-designed printed circuit board that provides dedicated connections for the GPIO pins of microcontrollers like the ESP8266. It simplifies the process of connecting and interfacing sensors, actuators, and other external components to the microcontroller.

Key Pointers:

- Provides easy access to GPIO pins for sensor and actuator connections.
- Can include features like pull-up resistors, capacitors, and voltage regulation.
- Ideal for custom-built hardware projects with multiple I/O requirements.

2.0.8 Power Supply Module

A power supply module is used to provide a stable and regulated power output to various components in a project. It can convert different input voltages into specific voltages needed by various parts of a system, such as 5V, 12V, or 3.3V.

Key Pointers:

- Often includes multiple outputs for powering different components.
- Ensures stable voltage and current for reliable operation.
- Common in projects involving sensors, microcontrollers, and actuators.

Chapter 3

Precision Linear Actuator Project

3.1 Introduction

As part of my internship at Dataviv, I worked on a Precision Linear Actuator project that involved designing and building a high-precision, motorized actuator system for various applications. This project combined mechanical and electrical engineering skills with programming expertise.

3.2 Problem Statement

The project aimed to develop a highly accurate and efficient linear actuator system using a stepper motor, ESP8266 microcontroller, and a stepper motor driver. The goal was to create a solution that could provide smooth and reliable motion control, allowing for precise movement in applications like robotics or automation systems.

3.3 Project Overview

I designed the actuator system to integrate a stepper motor with the A4988 stepper driver, controlled via a wireless interface using an ESP8266 microcontroller. The system allows for fine-grained control of the motor's movement, achieving high precision in positioning. The project also involved creating a user-friendly interface to control the actuator remotely and testing the system for performance and reliability. The actuator was tested in various scenarios to ensure smooth, responsive, and consistent motion, meeting the design specifications.

Figure 3.1: Home Screen of The FinVoice-360

3.4 API Endpoints

The project included a web interface for controlling the linear actuator. Below are the key API endpoints:

3.4.1 Web Based Controller

This controller handles the movement of the actuator. The endpoints include:

- **POST /move**: Moves the actuator to a specified position.
- **GET /status**: Retrieves the current position of the actuator.

3.4.2 System Controller

This controller monitors the health and status of the actuator system. The endpoints include:

- GET /health: Checks the health status of the system.
- **POST /calibrate: Calibrates the actuator to ensure precise movement.**

3.4.3 Invoice Controller

The invoice controller handles invoice creation and retrieval. The endpoints include:

- **POST** /add-invoice: Creates a new invoice and saves it in the system.
- **GET** /get-invoices/{userId}: Fetches all invoices for a user identified by their userId.

3.4.4 HTTP Methods and Color Codes

The API uses standard HTTP methods to define the behavior of each endpoint. The methods are color-coded for clarity:

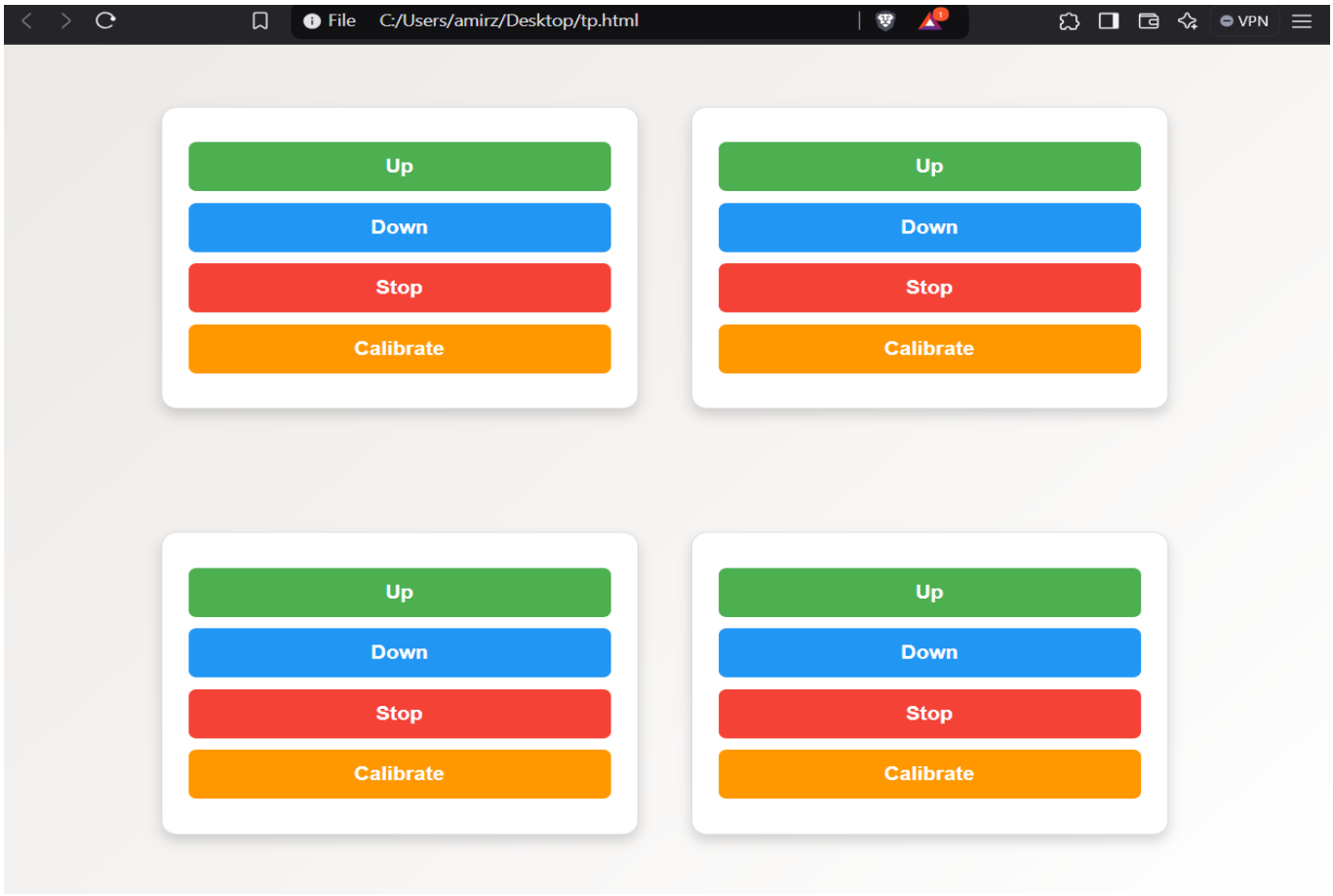
- **POST**: Used to send data for actuator movement and calibration.
- **GET**: Used to retrieve actuator status and system health.

3.4.5 API Overview Table

The following table provides a summary of all the endpoints:

Table: API Endpoint Overview

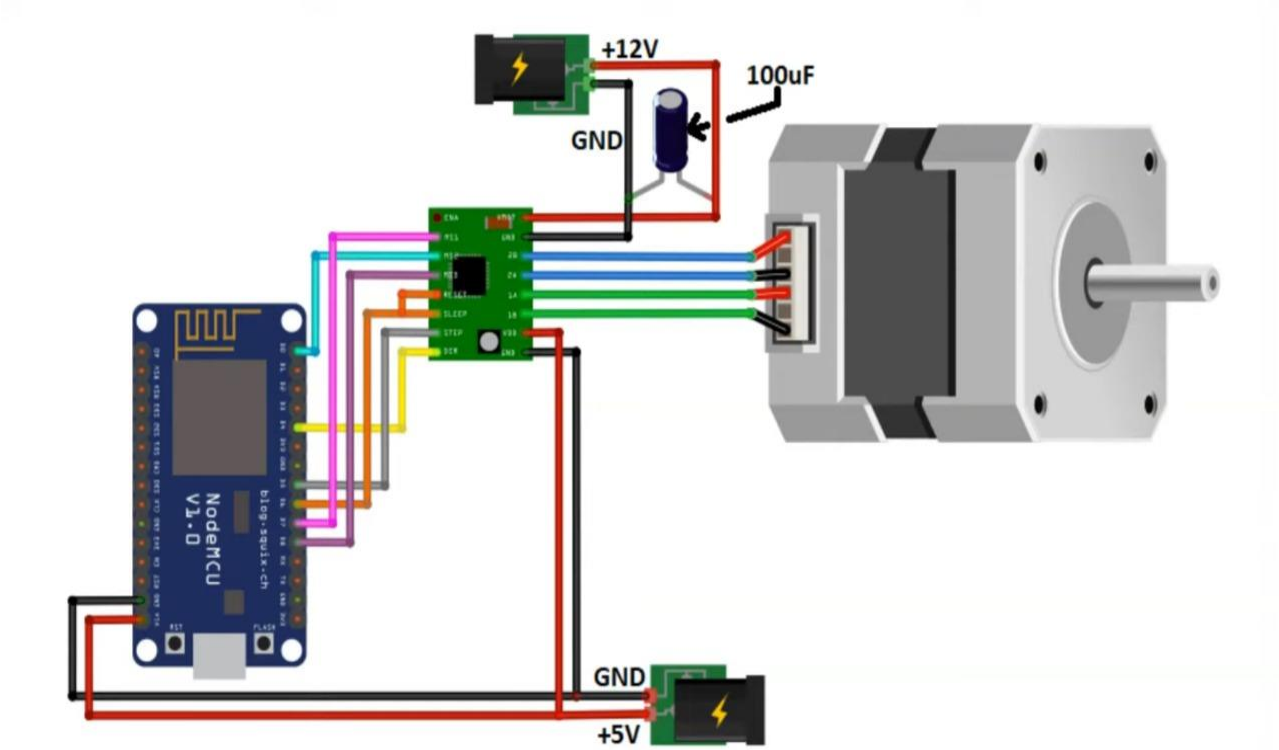
Controller	HTTP Method	Endpoint	Description
Control	POST	/mover	Move actuator to specified position
Control	GET	/status	Retrieve current actuator position
System	GET	/health	Check system health
System	POST	/calibrate	Calibrate the actuator

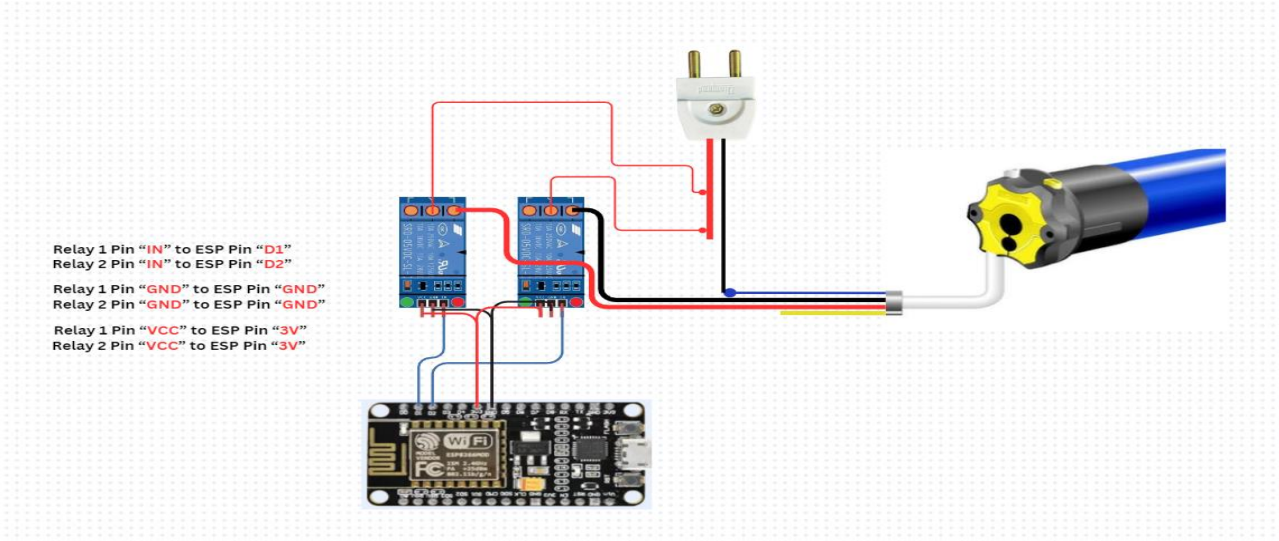


Web Based Control Interface

3.5 Architecture of The Project

The actuator system is built using an ESP8266 microcontroller for wireless control and an A4988 stepper driver to manage the motor's movement. The system architecture supports remote monitoring and control, ensuring flexibility and precision in operation. The design incorporates real-time feedback loops to ensure smooth and reliable actuator movement.





Chapter 4

ML Based Anti-Virus

4.1 Introduction

During my internship at Dataviv, I worked on an ML-based Antivirus Scanner project that aimed to leverage machine learning for the detection and classification of malware. The project involved creating an intelligent system capable of analyzing files, system behavior, and network traffic to identify potential security threats.

4.2 Problem Statement

The primary goal was to develop an antivirus scanner that could accurately identify and block malware in real-time using machine learning. The challenge was to build a solution capable of detecting both known and unknown threats, ensuring the system could adapt to new, evolving malware without constant manual updates.

4.3 Project Overview

I designed a machine learning model that classified files and processes as either benign or malicious. The model was trained using large datasets of malware and legitimate software, employing techniques like decision trees, random forests, and neural networks. The scanner analyzed file attributes, behavior patterns, and metadata, flagging files with unusual characteristics for potential threats. The system also incorporated a self-learning mechanism, allowing it to improve its detection accuracy over time.

4.4 Key Features and Components:

4.4.1 Real-time Threat Detection

The system continuously monitored file and system activity, analyzing patterns and behaviors in real-time to detect deviations indicative of malware. Files exhibiting suspicious behavior were flagged and isolated for further analysis.

4.4.2 Behavioral Analysis

Instead of solely relying on signature-based detection, the scanner utilized behavioral analysis to detect zero-day attacks. By evaluating how a file interacts with the system, it could identify previously unseen malware based on its activity rather than its signature.

4.4.3 Machine Learning Model

The heart of the system was a machine learning model trained on a comprehensive dataset of malware and benign software. It classified files and processes based on features extracted from the data, such as system calls, file modifications, and network activity.

4.4.4 Continuous Learning

As the system processed more data, it became more efficient at distinguishing between safe and malicious files. The continuous learning feature ensured that the model adapted to new types of malware, improving detection capabilities with minimal human intervention.

3.5 System Architecture

The architecture of the ML-based Antivirus Scanner was built around a modular framework. The core components included the data preprocessing module, machine learning model, and scanning engine. The scanner's real-time analysis was powered by a data pipeline that extracted features from running processes and files, which were then passed to the machine learning model for classification. The system was optimized for minimal impact on system performance, running in the background without slowing down user activities.

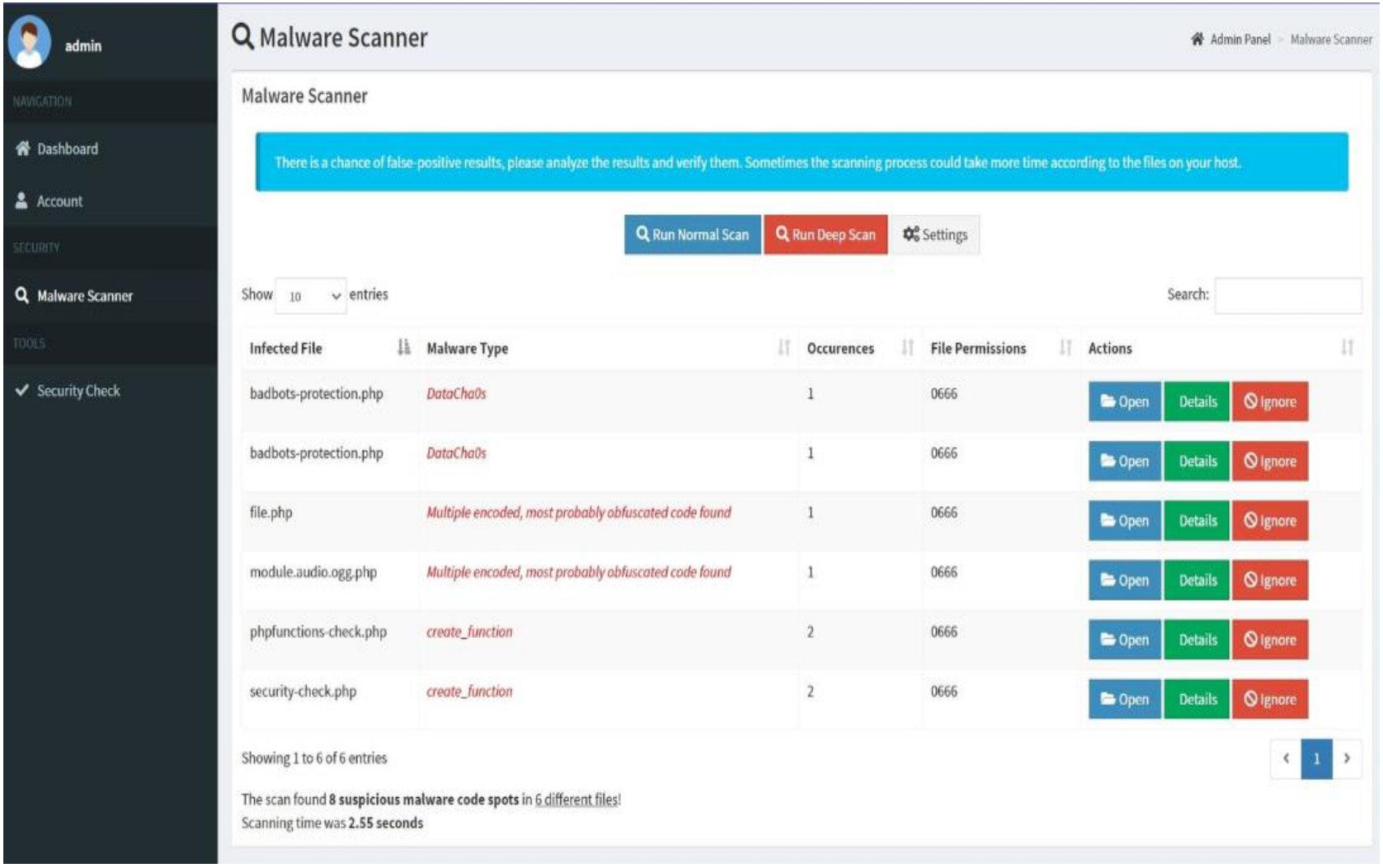
3.6 Threat Identification Process

The threat detection process followed these stages:

1. Data Collection: Information was gathered from files, system behavior, and network activity.
2. Feature Extraction: Relevant attributes such as API calls, file metadata, and interactions with the operating system were extracted for analysis.
3. Malware Classification: The machine learning model analyzed the extracted features and classified the file or activity as benign or malicious.
4. Threat Response: If a threat was detected, the file was quarantined, and the user was alerted.

3.7 Performance Evaluation

To evaluate the performance of the ML-based antivirus scanner, I conducted testing with both known and unknown malware samples. The system's detection accuracy, false positive rate, and resource usage were measured. The results showed high accuracy in identifying both known and zero-day malware, with minimal system resource usage.


admin
NAVIGATION
Dashboard
Account
SECURITY
Malware Scanner
TOOLS
Security Check
Malware Scanner
Admin Panel
Malware Scanner
Malware Scanner
There is a chance of false-positive results, please analyze the results and verify them. Sometimes the scanning process could take more time according to the files on your host.
Run Normal Scan
Run Deep Scan
Settings
Show 10 entries
Search:
Infected File
Malware Type
Occurences
File Permissions
Actions
badbots-protection.php
DataCha0s
1
0666
Open
Details
Ignore
badbots-protection.php
DataCha0s
1
0666
Open
Details
Ignore
file.php
Multiple encoded, most probably obfuscated code found
1
0666
Open
Details
Ignore
module.audio.ogg.php
Multiple encoded, most probably obfuscated code found
1
0666
Open
Details
Ignore
phpfunctions-check.php
create_function
2
0666
Open
Details
Ignore
security-check.php
create_function
2
0666
Open
Details
Ignore
Showing 1 to 6 of 6 entries
1
The scan found 8 suspicious malware code spots in 6 different files!
Scanning time was 2.55 seconds

References

[1] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, ”Applied Logistic Regression,” 3rd ed., John Wiley Sons, 2013.

[2] J. Dean and S. Ghemawat, ”MapReduce: Simplified Data Processing on Large Clusters,” in *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI)*, San Francisco, CA, USA, 2004, pp. 137–150.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ”SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[4] S. Garc´ıa, J. Luengo, and F. Herrera, ”Data Preprocessing in Data Mining,” Springer, 2015.

[5] A. G. Cohn, P. E. Johnson, and L. O. Hall, ”Big Data Analytics: A Survey,” in *Proceedings of the IEEE International Conference on Big Data*, Washington, DC, USA, 2013, pp. 323–331.

[6] K. P. Murphy, ”Machine Learning: A Probabilistic Perspective,” MIT Press, 2012.

[7] J. K. S. M. R. Chien, C. R. R. H. P. R. R. Y. Z. Liu, ”Big Data Analytics: Technologies, Tools, and Techniques,” *Springer*, 2015.

[8] S. J. Redman, ”Data Quality: The Field Guide,” *Digital Press*, 2001.

[9] H. He, Y. Bai, E. A. Garcia, and S. Li, ”ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning,” in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, Orlando, FL, USA, 2008, pp. 1322–1328.

[10] X. Chen, D. A. D. R. R. Wang, ”Exploring Big Data: A Survey of Big Data Techniques,” *IEEE Transactions on Big Data*, vol. 1, no. 2, pp. 1–16, 2015.

[11] Medium, ”ElasticNet Regression Fundamentals and Modeling in Python” Available: https://keremkargin.medium.com/elasticnet-regression-fundamentals-and-modeling-in-python-8668f3c2e39e. [Accessed: 17-06-2024].

[12] Medium, ”Borderline KNN— SVM and ADAYSN SMOTE” Available: https://bobrupakroy.medium.com/borderline-knn-svm-and-adaysn-smote-1d74756fb049 [Accessed: 22-06-2024].

[13] Medium, ”Understanding ‘ Borderline SMOTE Oversample (Py)’” Available: https://medium.com/@analyttica/understanding-borderline-smote-oversample-py-e986d07fc0c [Accessed: 12-06-2024].