1. Introduction and Problem Definition

Bicycle theft is a problem in many cities and universities worldwide. Despite the use of protective tools, such as locks or padlocks, that we have used over the years, these methods remain vulnerable to theft or vandalism. In many cases, the bicycle owner is not nearby and cannot find out what has happened to their bike until it is too late, so security is not good enough. This problem is especially serious in areas with a large number of students, open-air parking areas, or public spaces without adequate surveillance.

In this context, the Internet of Things (IoT) offers us a development opportunity with smart solutions that can detect these events in real time, generate immediate alerts, and record all relevant information about incidents involving parked bicycles. Furthermore, the incorporation of Edge AI improves system reliability by reducing false alarms and avoiding cloud dependency for decision-making, distinguishing between movements that pose no threat and those that truly warrant alert.

The main objective of this project is to design a bicycle alarm capable of detecting suspicious movements using vibration sensors, analyzing these events with Edge AI logic, and communicating the results to a cloud platform for visualization and analysis, such as graphs of the detected data. The system was created using an Arduino UNO R4 WiFi.

Besides creating a new, functional technological device, this project also seeks to explore the challenges of the IoT and Edge AI world, such as reliability, data management, latency, privacy, and the ethics of implementing smart technologies in everyday life.

2. Project Requirements

The system development began by differentiating between the functional and non-functional requirements of our project, which served as a guide throughout the design and implementation process. These requirements are divided into three main categories: hardware, software, and cloud services.

Hardware Requirements

The core of the system is an Arduino UNO R4 WiFi, as it has greater processing power compared to other versions and offers WiFi connectivity. This microcontroller is an IoT device and Edge Computing node, responsible for data collection and local analysis.

For motion detection, an SW-420 vibration sensor is used, which provides a digital signal upon detecting vibrations and an analog signal that allows for measuring the intensity of the movement. This dual output is especially useful for applying event filtering and classification techniques.

The system also includes an active buzzer that emits an audible alarm upon detection of a confirmed event, indicating the system status, the operation of the Edge AI, and WiFi connectivity.

Component Cost Breakdown (Hardware):

Material	*Price*
Switch	£1.33
Lithium Battery Charging Module	£1.49
Tupperware (Enclosure)	£2.50
18650 Batteries	£8.20
Battery Tray	£6.40
Female-to-Male Jumper Wires (10x)	£1.60
Male-to-Male Jumper Wires (10x)	£1.34
Arduino UNO R4 WiFi Development Board	£22.22
USB Cable, Type A Plug to Type C Plug	£3.95
Vibration Sensor Module	£3.99

Software Requirements

The system software was developed entirely in the Arduino environment, using C/C++ and the official libraries provided for Arduino. The code is structured into distinct functions, which facilitates its maintenance and use.

A project requirement was that the system function even without a Wi-Fi connection. Therefore, the main logic for detecting and triggering alarms runs locally, independent of external services. The internet connection is used only to send data to the cloud platform when available.

Cloud Service Requirements

For data storage and visualization, the ThingSpeak platform was used due to its easy compatibility with Arduino and its ability to generate graphs in real time. The system sends information about detected events, vibration intensity, and statistics related to the performance of Edge's AI.

ThingSpeak allows for analyzing the system's behavior over time and evaluating the effectiveness of the implemented logic in reducing false alarms, thus fulfilling the project's data analysis and visualization requirements.

3. System and Circuit Design

The system design was approached from a modular perspective, clearly separating data acquisition, local processing, physical actuation, and cloud communication.

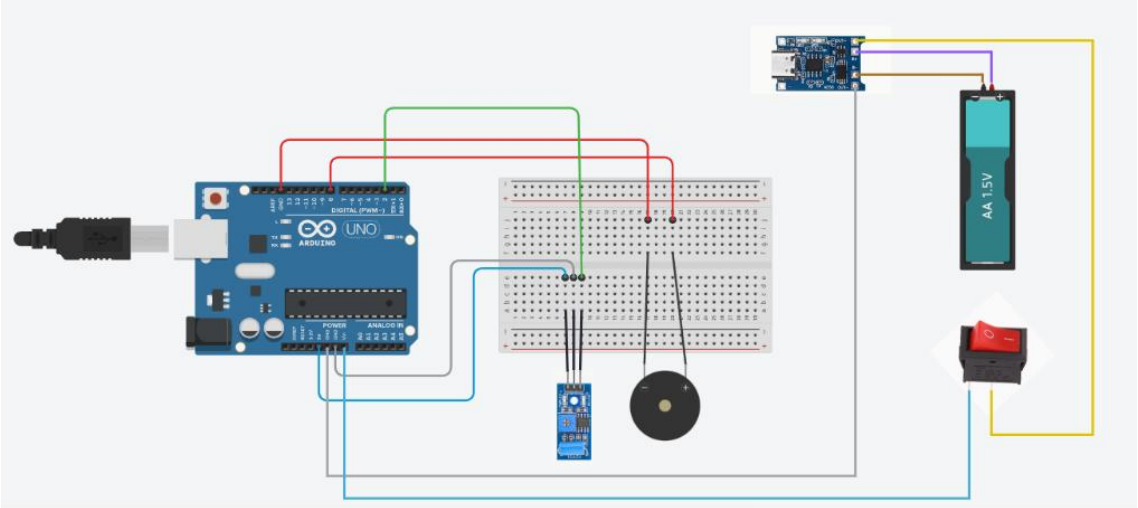
Circuit Design

The circuit is relatively simple, facilitating reproduction and reducing costs. The SW-420 vibration sensor connects to one digital and one analog input of the Arduino. The digital output detects the vibration the machine senses, while the analog output measures the intensity of that vibration.

Power Supply Subsystem: A fundamental part of the design is the portable power supply. An 18650 lithium battery (housed in a tray) provides the main power source for our alarm. Its output is connected to a TP4056 lithium battery charging module, which manages safe charging and regulates the output voltage. The OUT+ and OUT- terminals of the TP4056 module directly power the VIN and GND pins of the Arduino. A switch

is placed on the positive line between the TP4056 and the Arduino to allow control of the machine (when it is on and when it is off). The buzzer is connected as an output, allowing different sound patterns to be generated depending on the severity of the detected event.between the TP4056 and the Arduino to allow for manual power control, which is essential for a portable device.

The buzzer is connected to a digital pin configured as an output, enabling the generation of different sound patterns based on the severity of the detected event.

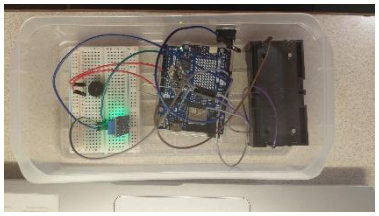


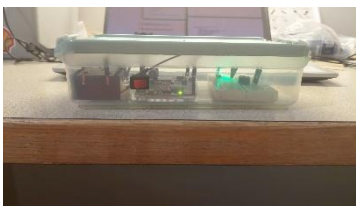
All the electronic components are housed in a Tupperware container, which acts as a robust and waterproof enclosure for the alarm that can be attached to a bicycle.

This design was accepted after passing a series of simulation and basic tests, ensuring that it would be ideal for storing the alarm's components before coding.

System Architecture

The system follows a typical IoT model with edge computing. Arduino acts as an edge device, capturing environmental data and making decisions locally. Only relevant results are sent to the cloud, improving the quality of the data displayed graphically. The edge AI logic is implemented directly on the microcontroller using a simplified, rule-based weighted model. While a machine learning library like TensorFlow Lite isn't used, the approach employed simulates the behavior of an inference model, evaluating multiple variables before confirming the veracity of an event.





4. System Testing and Validation Phase

The testing phase was crucial to ensuring the system's reliability and fine-tuning the detection parameters. Tests were conducted under both controlled conditions and in real-world scenarios, simulating various types of vibrations and movements.

Functional Tests

First, each component was verified to be functioning correctly. The vibration sensor was checked to ensure it responded correctly to different levels of movement, and the buzzer was confirmed to emit a sound when activated, indicating that the system was working properly.

Subsequently, the main software functions were tested, such as sensor reading, alarm activation, and Wi-Fi connection. These tests helped detect and correct errors in system synchronization and status.

Simulated Edge AI Tests

One of the most important aspects of the project was validating the Edge AI logic. For this, multiple tests were conducted comparing the system's behavior with and without the filtering logic activated. Real events were recorded, such as strong impacts or prolonged movements, and minor events, like small vibrations caused by wind or people passing by.

The results showed that using a history of vibrations and weighted thresholds significantly reduced the number of false alarms. The system only activated the alarm when the confidence score exceeded a determined value, demonstrating the utility of intelligent local processing.

Connectivity and Data Transmission Tests

Finally, tests were conducted to ensure everything connected correctly and that data reached ThingSpeak when a Wi-Fi connection was available. The system was also tested to ensure it functioned offline, activating the local alarm without attempting to send data to the cloud.

5. Data Analysis and Visualization

Data analysis is an important part of this project because it evaluates system performance and justifies design decisions. ThingSpeak is the ideal platform for storing and visualizing the data sent by the device.

Data Collected

The data sent to the cloud includes alarm activation, event confirmation by the Edge AI, the intensity of the detected vibration, and accumulated statistics such as the total number of events, confirmed events, and false positives.

This data allows for analyzing not only when the alarm is triggered but also the quality of the decisions made by the Edge AI logic.

Visualization and Evaluation

ThingSpeak generates timeline graphs that show the evolution of different parameters. For example, it shows how the vibration intensity varies over time and at each alarm activation. At the same time, it distinguishes between false positives and alerts triggered by excessive vibration.This analysis demonstrates that the use of Edge AI, even in a simplified form, adds real value to the system, improving its reliability and reducing unnecessary inconvenience for the user.

6. Legal, Ethical, and Social Evaluation

From a legal and ethical standpoint, the project considers minimal data related to privacy. While the system is far from collecting personal data, it does record information about physical events and behaviors, which could be sensitive in other contexts unrelated to the project's primary objective.

The fact that most of the processing takes place on the device, rather than in the cloud, contributes to improved user privacy. Only the aggregated data necessary for analysis is sent, minimizing the risk of misuse of information.

In terms of social responsibility and data accountability, the project promotes the responsible use of technology to enhance personal security without resorting to invasive systems. Furthermore, it is a low-cost and easily replicable solution, thus fostering equal access and sustainability.

7. Conclusion

This project demonstrates how IoT and edge computing technologies can be combined to effectively solve a real-world problem. This smart bicycle alarm is a functional, reliable, and adaptable system that leverages local processing to refine event recognition and distinguish suspicious movements.

Key concepts from IoT, edge computing, and data analytics were applied during the project's development, thus fulfilling the module's technical, ethical, and academic requirements. It demonstrates that this type of project can benefit from the use of edge computing, paving the way for future improvements and expansions.

8.References

Arduino Documentation.
ThingSpeak Documentation.
Course material from the IoT and Edge AI module.

ThingSpeack link https://thingspeak.mathworks.com/channels/3229333

GitHub link: https://github.com/pablorodg58/IoTProject