University of Trento

Department of Industrial Engineering

Master's Degree in Mechatronics Engineering
Curriculum Electronics and Robotics

~ · ~

Academic Year 2023–2024

IoT Sensor System for Fruit Stress Monitoring in Apple Grading Industrial Processes

Supervisors
Prof. Davide Brunelli
Dr. Matteo Nardello

Graduate Student
Marco Perini
229203

To my family, friends and cats

Contents

Introduction

In recent years, the advent of Internet of Things (IoT) technologies has significantly transformed multiple sectors, enabling enhanced connectivity between devices for efficient information sharing and minimal resource use. These innovations have found widespread application across various fields, facilitating improvements in environmental monitoring to safeguard public health, and advancing quality control in manufacturing processes. Such technological progress supports distributed, remote system access, allowing for effective real-time monitoring. Industries have experienced seamless integration of IoT, combining traditional and innovative methods to establish custom assembly lines tailored to specific monitoring and analysis goals. Following these advancements, the agri-food supply chain emerges as a sector poised to benefit substantially from IoT technology, particularly in maintaining product quality from the initial harvest to the final delivery to consumers. Despite the potential, the adoption of IoT within the agri-food sector is still in its early stages, highlighting a distinct need for innovation. Especially in fruit production, the application of IoT promises a new level of precision in quality control. Manufacturers are beginning to leverage IoT to automate their production lines, enhance consistency, reduce errors, and optimize operational efficiency. The deployment of real-time monitoring systems equipped with various sensors enables the tracking of environmental conditions and equipment performance, ensuring product quality is upheld throughout the manufacturing process. Furthermore, this digital oversight includes predictive maintenance and remote monitoring capabilities, offering protection against potential damage due to equipment failure or sub-optimal environmental conditions.

An Italian reality that is investing heavily in innovation to provide an increasingly higher quality product to the consumer is Melinda, a consortium specialized in the production and supply of apples located in Val di Non, Trentino. Melinda's efforts reflect a broader industry trend towards leveraging technology not just for efficiency and cost reduction, but also as a means to elevate the product quality and deliver it to the consumers in the best possible condition. Despite these advancements and efforts towards integrating cutting-edge technologies, challenges persist within the agri-food sector, particularly in ensuring the post-harvest quality of the product. A notable concern for Melinda, and indeed a prevalent issue in the fruit production industry, is the emergence of bruises on apples days after the packing process has been completed. This problem not only undermines the quality of the product but also impacts consumer satisfaction and potentially leads to significant financial losses due to increased waste. The delayed appearance of these bruises complicates detection and prevention, as the damage is not immediately visible at the time of packing. This phenomenon highlights a critical gap in the current quality control processes, underscoring the need for innovative solutions that can predict or quickly identify such damage before the products reach the consumer. Melinda's commitment to quality and sustainability drives the consortium to seek out advanced methodologies and technologies that can address this challenge, further enhancing their production processes and ensuring that their apples maintain the highest standards of quality from the orchard to the consumer's hands.

To address the challenge presented by Melinda, this thesis outlines the development of an innovative solution: a sensor device specifically designed to monitor apples during the critical final stages of processing within the facility. Prior to this intervention, no such device existed, leaving a gap in the capability to detect and analyze the conditions leading to bruising and damage during the post-harvest process. The journey to the creation of this sensor involved a multidisciplinary approach, blending principles from mechanical engineering, computer science, and agricultural science to craft a device that could seamlessly integrate into the existing infrastructure. The sensor, designed to mimic the core of a real apple, is equipped to record impacts in real-time as apples undergo washing, photographic analysis, and sorting. Leveraging Bluetooth Low Energy (BLE) technology, the device wirelessly transmits collected data for subsequent analysis. This innovative approach enabled the identification of specific locations within the production line that were most likely to cause damage to the apples. By presenting the analyzed data through a user-friendly dashboard, the system provides immediate insights into the direction and intensity of impacts, as well as the distribution of these critical impacts across various stages of the production line.

This breakthrough has not only furnished Melinda with an adavnced monitoring system for their production processes but has also opened avenues for broader applications across the agri-food sector. The sensor's adaptability to various contexts—from monitoring different types of fruits to potentially other agricultural products—underscores its significant development potential. Most importantly, the insights gleaned from the data have pinpointed parts of the production process with a higher likelihood of causing damage, enabling targeted interventions to enhance overall product quality. The results of this thesis represent a pivotal advancement in the field of agricultural technology, offering a tangible solution to a longstanding problem faced by producers like Melinda. The implementation of this sensor device has the potential to revolutionize quality control practices not only within the apple industry but across the broader spectrum of agri-food production, marking a significant step forward in the journey towards more sustainable and efficient agricultural processes.

- **Chapter 1** introduces the research problem, including a thorough literature review and analysis of the state-of-the-art. Additionally, the main frameworks and definitions are described, setting a foundational understanding for the study.
- **Chapter 2** offers a detailed description of the system architecture and its components, explaining their functionalities and interactions. This section is crucial for understanding how each part contributes to the overall system performance and objectives.
- **Chapter 3** is dedicated to the presentation of the sensor-equipped apple, covering both the hardware development and firmware analysis. This chapter highlights the innovative approach to integrating sensors within agricultural products to gather data.
- **Chapter 4** and **Chapter 5** delve deeper into the implementation of the IoT Gateways and the server-side device, both based on Raspberry Pi 4. These chapters discuss the technical specifications, software, and network protocols used to facilitate communication and data processing within the system.
- **Chapter 6** provides an overview of the production process at the Melinda plant, detailing the various zones of the production line and the machinery used to process the apples. This contextualizes the environment in which the sensor-equipped apples are deployed and monitored.
- **Chapter 7** presents the results obtained from data acquisition from the sensor-equipped apples, both in a controlled environment through synthetic tests and in actual production settings. This chapter evaluates the system's performance and the quality of data collected under different conditions.

Chapter 1

State-of-the-Art and Frameworks

1.1 Introduction

1.1.1 Overview of the Problem

In the agri-food sector, maintaining product quality from harvest through to the final consumer is a critical factor distinguishing companies and enabling consumers to make informed choices. Various practices assess the quality of a product not only through direct analysis but also by evaluating the production processes involved, from careful handling of the product at all stages to the use of sanitized and regulated machinery. Studies have shown that these factors can directly and indirectly influence food safety knowledge, as outlined in Figure 1.1, ensuring, among other things, a high level of healthiness [14]. The HACCP system, which stands for "Hazard Analysis and Critical Control Points," is particularly noteworthy. It is a systematic and scientifically based approach that focuses on monitoring potential contamination hazards—biological, chemical, or physical—during food processing. Its purpose is to identify and analyze hazards and to develop suitable control systems [3].

Melinda is committed to adhering to these issues and regulations, which is why it has partnered with the University of Trento to conduct a research closely related to the quality of their apples. They noticed that some apples developed bruising days after packaging, potentially compromising not just the individual fruit but the integrity of the entire batch. After harvesting, the apples are processed within Melinda's facilities, undergoing various machinery and production phases before being sorted into bins based on intrinsic characteristics such as size, color, and porosity, and then moving on to the packaging phase.

Figure 1.1: Conceptual framework of food safety knowledge integration, highlighting the critical control points from production and processing to cultural practices and system regulations that collectively inform and uphold food safety standards in the agri-food industry.

1.1.2 Purpose of the Study

Melinda hypothesizes that apples may be damaged when transitioning between machines along the production line. They have tasked us with analyzing the movements and impacts that apples receive in various areas of the facility to identify critical points and act accordingly. The project's goal is to create a system that monitors the impacts on apples and conducts an in-depth analysis to extract as much information as possible. Addressing these points of concern will serve a dual purpose: reducing the number of apples that are discarded due to quality issues and, as a result, cutting down on the associated costs. Fewer damaged apples mean lower operational losses for Melinda and a decrease in food waste, aligning with sustainable environmental practices. This project's successful implementation will not only streamline Melinda's production process but also contribute to the broader goal of sustainability in the agri-food sector. The development of an IoT-based monitoring system is central to this endeavor, providing a technological solution to a significant agricultural challenge.

1.2 State of the Art

1.2.1 Current Knowledge and Practices

Numerous studies and research have been published over the years in response to increasing demands for product quality from governments worldwide. In the agricultural sector, the focus is often on the fruit harvesting phase in fields, a process involving repetitive and strenuous tasks for workers. There's a push towards automating harvesting with robotic systems that typically move autonomously and identify fruits to be harvested using artificial vision systems. This introduces a multitude of factors that make the process more complex, necessitating the design of increasingly advanced systems to avoid damaging fruits during harvesting and transport and to reduce false positives in food classification. Upon a comprehensive review of current literature, it becomes clear that scholarly contributions can be broadly categorized into three distinct domains: (1) the innovation of robotic mechanisms for in-field fruit harvesting, (2) the advancement of artificial vision systems for accurate fruit identification, and (3) the formulation of algorithms and methodologies dedicated to assessing fruit damage. Each category represents a crucial aspect of agricultural modernization, aiming to streamline operations, enhance yield quality, and meet the stringent quality expectations of both consumers and regulatory bodies.

1. **Robotics Systems**: In agricultural robotics, advancements in apple and raspberry harvesting highlight significant progress. Variable impedance control enhances apple harvesting robots' efficiency by improving grasping force tracking [7]. Additionally, the design and control of such robots have been fine-tuned for better performance [1]. For raspberries, an artificial physical twin facilitates the development of harvesting techniques by simulating the fruit's properties, optimizing the picking process [8]. These innovations collectively reduce mechanical damage and boost harvesting efficiency.

2. **Vision-based Systems**: Recent studies have demonstrated a multi-faceted approach towards enhancing fruit detection, recognition, and harvesting using a vision-based approach. The integration of finite element method simulations with 3D scanning and high-speed camera technology offers a novel perspective on fruit deformation, essential for assessing mechanical damage during harvesting [5]. Concurrently, the exploration of non-destructive imaging techniques such as biospeckle, X-ray, and hyperspectral imaging, presents a comprehensive strategy for fruit damage detection, underscoring the synergy of combining these methods for improved quality monitoring [13]. The development of a multi-camera system for in-line apple defect detection marks a significant step towards automating fruit grading processes, showcasing the potential of machine vision in agricultural applications [21]. This technological progression is further supported by advancements in machine learning and deep learning algorithms, which are instrumental in realizing real-time fruit recognition and robotic grasping, significantly enhancing the efficiency and accuracy of robotic apple harvesting [10, 4].

3. **Fruit Damage Assessment Methods**: section integrates insights across multiple studies to explore strategies for minimizing mechanical damage to fruits, combining findings from finite element analysis, vibration and impact studies, and innovative sensing technologies. Kabas and Vladut's work [9] underscores the predictive power of FEA in assessing peach damage from drops, while Vursavuş and Özgüven [19] highlight the impact of vibration parameters on apple damage during transport. Szeptycki's exploration [16] of apple cultivars' resistance to mechanical stress complements Antunes et al.'s examination [6] of mechanical damage in Brazilian apple packing lines, suggesting the importance of packing line optimization. Xia et al.'s development [20] of flexible sensors for cold chain monitoring and Luo et al.'s hyperspectral imaging [12] for early bruise detection represent significant advancements in non-destructive damage assessment. This discussion illustrates a multi-disciplinary approach to reducing postharvest losses, emphasizing the importance of technological innovation and adaptive strategies in agricultural practices.

1.2.2 Gaps in Existing Literature

Despite the extensive research into agricultural robotics, vision systems, and fruit damage assessment, there remains a notable gap in the application of sensor technology directly to real fruits during the post-harvest phase. Current studies predominantly utilize artificially created fruits equipped mainly with sensors that measure environmental variables such as temperature and humidity, but these do not account for the physical stresses fruits undergo during movements, including within production processes at facilities. This limitation is significant, as these artificial fruits fail to replicate the exact physical characteristics, like inertia and mass, of genuine produce. Consequently, they overlook the complex dynamics actual fruits experience during handling and storage in controlled environments. Moreover, while existing literature largely focuses on pre-harvest technologies and methodologies in orchards, there's a marked deficiency in exploring IoT technologies' role in monitoring real fruits' conditions, specifically the stresses and strains they are subjected to during post-harvest processes. This research aims to bridge this gap by investigating IoT applications that scrutinize not only the environmental conditions but also the mechanical stresses that real fruits endure during the critical post-harvest phase within facility environments.

1.3 Theoretical Frameworks

Internet of Things

The Internet of Things (IoT) is a concept that describes the network of physical devices interconnected through various communication protocols. These devices, equipped with sensors and actuators, enable the collection and exchange of data, facilitating a higher level of interaction between the physical and digital worlds. Key features of IoT include:

- **Interconnectivity:** Devices can communicate with each other and with the Internet, enabling a seamless flow of information;
- **Data Exchange:** Through sensors and actuators, devices can collect data from the environment and exchange it, making them responsive to changes;
- **Remote Control:** Many IoT devices can be controlled remotely, providing convenience and improved manageability;
- **Smart Monitoring:** The data collected can be used to monitor systems and environments, leading to smarter decision-making and automation;
- **Protocol Diversity:** IoT devices utilize a variety of communication protocols, including Bluetooth, Zigbee, and WiFi, to cater to different requirements and scenarios;
- **Energy Efficiency:** A focus on low-power design is typical for IoT devices, to ensure longevity and sustainability, especially in devices that operate on battery power.

IoT technology is rapidly expanding into numerous sectors, including smart homes, industrial automation, healthcare, and beyond. The innovative use of sensors and networked communication opens up vast possibilities for new applications and improved systems.

MQTT

MQTT (Message Queuing Telemetry Transport) is a messaging protocol that is designed for the efficient transmission of data between devices in the Internet of Things. It has gained widespread popularity due to its simplicity and effectiveness in scenarios where network bandwidth is at a premium. Notable features of MQTT include:

- **Lightweight Protocol:** The lightweight nature of MQTT messages makes it suitable for IoT devices with limited processing capabilities and low memory.
- **Publish/Subscribe Model:** Unlike traditional client-server models, MQTT uses a publish/subscribe model that allows for decoupling of devices and applications.
- **Reliable Message Delivery:** MQTT ensures reliable message delivery with Quality of Service (QoS) levels, which can be critical for many IoT applications.
- **Last Will and Testament:** A feature that allows a device to send a final message if it becomes disconnected unexpectedly, useful for monitoring device connectivity.
- **Retained Messages:** MQTT can retain a message on a topic for new subscribers, ensuring that new devices receive the current state immediately upon subscription.
- **Support for Unreliable Networks:** Its design caters to environments with unstable or intermittent connections, which is common in remote IoT applications.

MQTT is particularly beneficial in scenarios where conserving bandwidth and battery life is essential, making it a preferred protocol for various IoT solutions, including home automation, industrial monitoring, and healthcare systems.

Bluetooth Low Energy

Bluetooth Low Energy (BLE), also known as Bluetooth Smart, is a subset of Bluetooth technology and plays a significant role in the Internet of Things (IoT) ecosystem. Some of the defining features of BLE include:

- **Low Energy Consumption:** BLE is specifically designed for short bursts of communications with extremely low power usage, making it ideal for battery-operated IoT devices.
- **Operational Modes:** It supports multiple roles such as Central, Peripheral, Broadcaster, and Observer, which are crucial for various IoT device interactions.
- **Frequency Hopping:** BLE uses frequency hopping in the 2.4 GHz ISM band to minimize interference and enhance communication reliability.

- **Data Transfer:** While optimized for small data packets, BLE ensures efficient data transfer, which is sufficient for many IoT applications.
- **Security:** It incorporates features like encryption and authentication to secure communication between devices.
- **Ubiquity:** The widespread integration of BLE in consumer electronics like smartphones and tablets makes it a readily available platform for IoT connectivity.

These characteristics make BLE an excellent choice for a variety of IoT applications, including wearable technology, health monitoring devices, and smart home automation.

InfluxDB

InfluxDB is an open-source time series database designed to handle high write and query loads. It is a crucial component in the monitoring stack, often used for storing and analyzing real-time metrics and events. Key features of InfluxDB include:

- **Time Series Data Storage:** Optimized for timestamped data, making it ideal for tracking metrics that change over time, such as performance monitoring data.
- **High Performance:** Able to handle large volumes of data writes and queries with high throughput, ensuring data is accessible with minimal latency.
- **Data Retention Policies:** Configurable policies allow for automatic data pruning, which helps manage storage efficiently by discarding old data.
- **Continuous Queries:** Supports running continuous queries to downsample data and write the results to another measurement for efficient storage.
- **Schemaless Design:** Offers flexibility in data modeling, which is beneficial for dynamically changing and evolving data sources.
- **Native HTTP API:** Allows for easy integration with a variety of applications and services using a straightforward HTTP API for data writing and querying.

InfluxDB is commonly paired with visualization tools such as Grafana to create comprehensive monitoring solutions that can provide insights into system performance and behavior.

Grafana

Grafana is an open-source platform for monitoring and observability. It provides powerful and elegant visualizations of complex data and is widely used to create dashboards that track and display metrics. Notable features of Grafana include:

- **Versatile Visualizations:** Offers a wide range of graphing options that can be tailored to suit the data's nature and the user's preferences.
- **Data Source Integration:** Compatible with various data sources, including InfluxDB, Prometheus, MySQL, and more, allowing for a unified view of metrics.
- **Alerting:** Features a robust alerting system that notifies users of significant events or anomalies detected in the data.
- **Customizability:** Dashboards and panels can be customized extensively, enabling users to design views that meet their specific monitoring needs.
- **Collaboration:** Supports sharing of dashboards and data, fostering collaboration among team members.
- **Accessibility:** Provides a web-based interface that is accessible from any browser, making it easy to access dashboards from anywhere.

Grafana is favored in scenarios that require real-time analysis of data, with the ability to drill down into metrics for detailed investigation, making it an indispensable tool for data analysis and system monitoring.

Docker

Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. The service has become a staple in the development and deployment of applications due to its ability to package and run applications in a loosely isolated environment known as a container. Salient features of Docker include:

- **Containerization:** Allows applications to be deployed in containers, which can be run on any compatible system without modification, enhancing portability.
- **Isolation:** Containers are isolated from each other and the host system, ensuring that each application only accesses the resources assigned to it.

- **Microservices:** Facilitates the microservices architecture by allowing each service to be deployed in its own container.
- **Resource Efficiency:** Uses fewer resources than traditional virtual machines as containers share the host system's kernel.
- **Rapid Deployment:** Containers can be created, started, stopped, moved, and deleted quickly and easily, facilitating agile development practices.
- **Consistency Across Environments:** Offers consistency across development, testing, and production environments, reducing "it works on my machine" problems.

Docker is particularly beneficial for developers looking to streamline the process of creating, deploying, and running applications by using containers to segregate applications from the infrastructure.

1.4 Key Concepts

In this section, we delve into fundamental concepts that form the backbone of the study, focusing on the innovative integration of IoT technologies in agricultural monitoring. It is split in *Technical Definitions* and *Statistical Definitions* subsections which will introduce some terms needed for a clear understanding of the covered topics.

Technical Definitions

- **Smart Apple**: A revolutionary approach to agricultural monitoring, the Smart Apple is an actual apple whose core has been replaced by a 3D printed cylinder containing a microcontroller. There are different sensors onboard which are adept at detecting and recording various environmental and physical parameters, such as temperature, humidity, and impact forces, that the fruit encounters throughout the grading process. This data is crucial for identifying potential stress points and improving the handling and processing of apples, ultimately leading to enhanced fruit quality and reduced waste.
- **IoT Gateway**: System which bridges the gap between the sensor-embedded apples and the broader network infrastructure. It collects data transmitted by the Smart Apples, performs preliminary processing, and relays this information to the IoT Server for further analysis. The gateway ensures seamless data flow, supports various communication protocols, and plays a critical role in the system's scalability and security, enabling the efficient management of sensor data across the network.

- **IoT Server**: The centralized computing system that receives, stores, and processes data from IoT devices. It plays a crucial role in data management, analytics, and decision-making processes, enabling users to access and analyze data collected by IoT sensors in real-time or through historical analysis. The IoT Server orchestrates the communication between different devices and applications, ensuring that data flow is seamless and secure across the network.

- **SSH (Secure Shell)**: A cryptographic network protocol for operating network services securely over an unsecured network in a client-server architecture. It enables users to log into another computer over a network, execute commands in a remote machine, and move files from one machine to another. It offers several options for strong authentication and protects the communications security and integrity with advanced encryption.

- **Systemd**: A service manager for Linux operating systems that aims to improve system control and resource management efficiency. It is in charge of initialising essential components at boot and controlling system's processes afterwards. It provides the concept of "units" for managing various resources, which simplifies system management and setup. Systemd also logs events and services, making it easier to detect and fix problems.

Statistical Definitions

In addition to the IoT-specific terms, our study incorporates several key statistical and probability concepts essential for data analysis:

- **Time-Series Data**: Data points indexed in time order, suitable for trends and patterns analysis. This type of data is pivotal for understanding how variables change over time, enabling the identification of consistent patterns, cyclical variations, and potential anomalies within the dataset.

- **Variance**: A measure of the dispersion of a set of data points around their mean value. It quantifies how much the numbers in the distribution are spread out. The variance for a population is defined as $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2$, and for a sample, it's $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$, where μ is the population mean, $\bar{x}$ is the sample mean, and N is the population size.

- **Interquartile Range (IQR)**: The difference between the 75th and 25th percentiles of a data set, represented as $IQR = Q_3 - Q_1$. It provides a measure of variability that is resistant to outliers, offering a clearer picture of the data distribution's central tendency.

- **Z-Score**: A measure that describes a value's relationship to the mean of a group of values, calculated as $z = \frac{(X-\mu)}{\sigma}$ for a population and $z = \frac{(x-\bar{x})}{s}$ for a sample. It indicates how many standard deviations an element is from the mean.

- **Outliers**: Data points that lie an abnormal distance from other values in a random sample from a population. Outliers can be identified using various methods, including the Z-score, where values with a Z-score greater than 3 or less than -3 are often considered outliers.

- **Box-plot**: A graphical representation that summarizes the key characteristics of a dataset through its five-number summary: the minimum, first quartile (Q1), median, third quartile (Q3), and maximum. It provides a visual snapshot of the data's distribution, particularly highlighting the median and the spread of the middle 50% of the data, which is contained within the IQR (Figure 1.2). Outliers, which are data points that fall significantly outside the typical range (typically, any data point more than 1.5 times the IQR above the third quartile or below the first quartile), are also depicted as individual points.

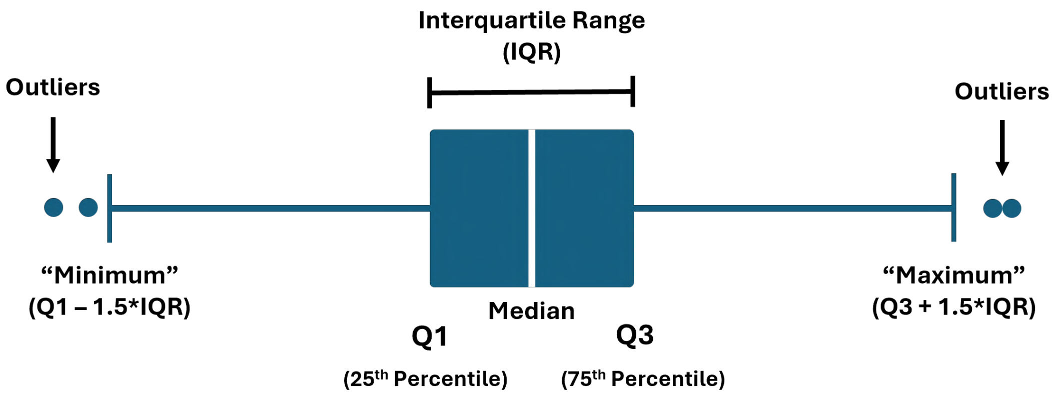


Figure 1.2: A box-plot diagram illustrating the five-number summary and the identification of outliers. The 'whiskers' extend to the furthest points within 1.5 times the interquartile range from the first and third quartiles, respectively, denoting the typical range of the data. Points outside this range are marked as outliers.

- **Cross-correlation**: A statistical measure that calculates the correlation between two sequences as a function of the lag of one relative to the other. It is useful for finding the time-delayed similarity between two time-series data sets, expressed as $R_{xy}(\tau) = E[(X_t - \mu_x)(Y_{t+\tau} - \mu_y)]$, where E is the expected value operator, and τ is the lag.

- **Dynamic Time Warping (DTW)**: An algorithm that measures the similarity between two temporal sequences, which may vary in speed or length. It's particularly useful in time-series analysis where classical methods like Euclidean distance can fail to recognize similar patterns due to differences in phase or time. DTW aligns sequences in time by "warping" their indices to match each other, effectively capturing the intuitive notion of similarity between sequences that are alike but out of phase (Figure 1.3).

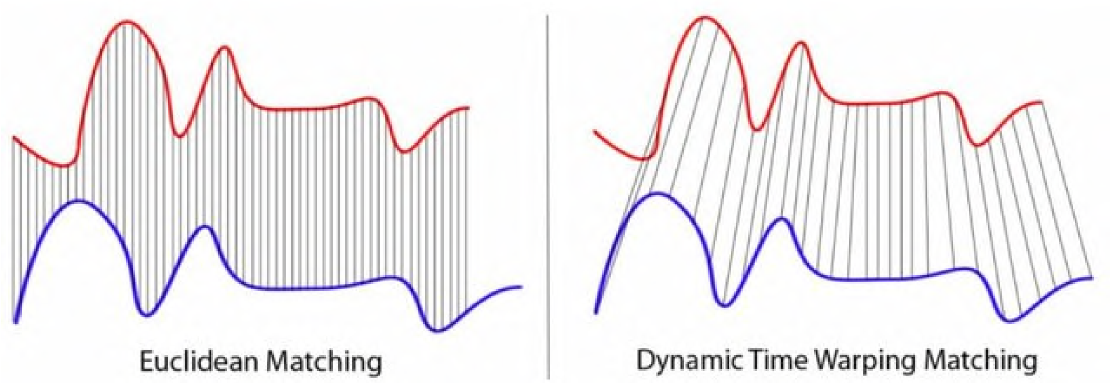


Figure 1.3: Comparison of Euclidean and Dynamic Time Warping matching. The left side illustrates the limitation of Euclidean matching for time-shifted data, where the distance calculation is affected by the misalignment. The right side shows how DTW compensates for this by 'warping' the time axis to align the sequences, providing a more accurate measure of similarity between the two sequences.

Chapter 2

System Design and Architecture

2.1 System Overview

2.1.1 High-Level Architecture

Our objective is to develop a system that is both robust and scalable, enabling comprehensive monitoring of apples across a variety of circumstances and environments. Current monitoring systems within the Melinda facility are limited to conducting quality control through photography in a singular, pre-sorting area. In Figure 2.1, it is shown the sorting phase where apples are separated in pools mainly by appearance. Our system extends this monitoring capability to cover the entire journey of the apples within the facility, including areas where traditional vision systems are impractical or where their effectiveness is greatly diminished. The architecture we have implemented, illustrated in Figure 2.2, facilitates continuous, non-invasive monitoring of apple movements, ensuring their natural flow remains unaffected. The system comprises four main components:

- **Smart Apple**: This component is a real apple in which the core has been replaced by a 3D-printed cylinder housing a microcontroller board. This board is tasked with gathering data from sensors and transmitting it using the BLE protocol to strategically placed gateways throughout the facility.

- **IoT Gateway**: Positioned across various sections of the facility, these devices are designed to receive data collected from the sensor-augmented apples via BLE protocol and forward this data to a central server using WiFi. Powered by power banks, their location can be adjusted to ensure optimal data reception.

- **Central Server**: This device is responsible for the reception, storage, and analysis of data batches. It saves the data in a time-series optimized database and performs analyses, the results of which are then presented through a visual dashboard. Located centrally near a WiFi hot-spot, it ensures efficient data communication within the facility.

- **Router WiFi**: It establishes a wireless network with a range sufficient to cover the areas of interest within the facility. The router is configured to maintain consistent network settings for all devices, ensuring the system's operation is repeatable without the need for additional adjustments.

The Smart Apple is powered by an Arduino Nicla which has a limited form factor containing many powerful IMUs and sensors while a set of Raspberry Pi 4 are used for both the gateways and the central server. The latter's choice is motivated by their cost-effectiveness, versatility, and the ability to function in resource-constrained environments.

Figure 2.1: This image showcases the sorting phase at the Melinda facility in Denno. After inspection through traditional vision-based methods which mainly take into account the fruits appearance, apples are released into pools to then be packaged into bins for subsequent industrial processing.

2.1.2 Component Interaction

All system components communicate with each other using wireless communication protocols, engaging in a streamlined flow of interaction. The apples, which have been modified to incorporate sensors internally, traverse the production line on rollers, slides, and through water currents, continuously gathering data. The nearest IoT gateway detects the presence of these enhanced apples and initiates a data transmission channel utilizing the *notify* feature of the BLE protocol. Upon receiving a data batch, this information is forwarded to the MQTT Broker, which in our setup, is represented by the central server equipped with a Raspberry Pi 4. This central server is responsible for storing all collected data in a database, preparing it for subsequent analysis.

Furthermore, any device connected to the network can be accessed via SSH, offering a secure method for network communication. The results of the analyses, on the other hand, are made available through a dashboard that is accessible via an IP address, ensuring that Melinda's staff can easily access this information.

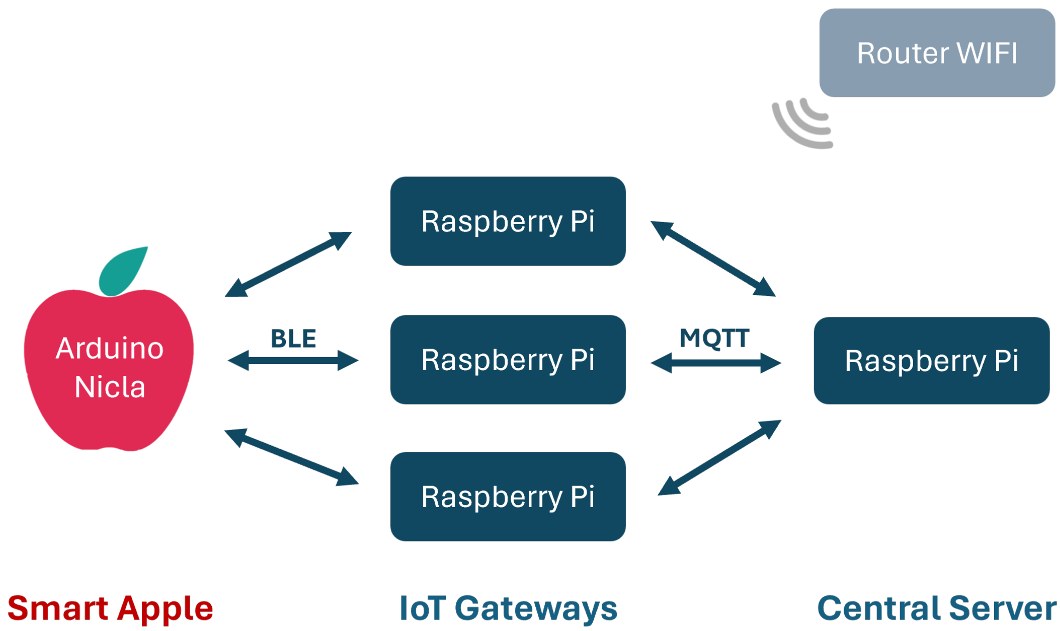


Figure 2.2: This diagram illustrates the interconnected system architecture for apple monitoring at the Melinda facility. At the core is the 'Smart Apple', embedded with an Arduino Nicla for data collection. Data flows via BLE to an array of Raspberry Pi devices serving as IoT Gateways. These gateways then communicate the gathered information through the MQTT protocol to a central Raspberry Pi server, which processes and stores the data. The entire network is seamlessly connected and managed through a WiFi router, ensuring continuous data synchronization and system integrity.

2.1.3 Data Acquisition and Flow

As outlined in Subsection 2.1.1, the Arduino Nicla embedded within the Smart Apples is pivotal for the efficient and swift collection of data relating to impacts and behaviors of apples on the production line, with a critical emphasis on preserving data integrity for subsequent analysis.

The principal sensors deployed for this objective are:

- **Accelerometer**: measures the linear acceleration on the three axes of the apple. It is used to detect the intensity and the direction of the impacts as well as the accelerations along the production line path. It is set so that it compensates for the gravity acceleration vector, removing the constant bias of 1g that would lead to wrong accelerations magnitude;

- **Gyroscope**: used to measure the angular velocity on the three axes which is essential to understand the apples behaviour and identify patterns along the way;

- **Quaternions**: an alternative to Euler angles to orient the apple in space. They are calculated using a Bosch sensor fusion proprietary algorithm which combines the data from the accelerometer, the gyroscope and the magnetometer. They are useful to see the apple movements in simulation after the acquisition in the process line, by avoiding classical problems like gymbal lock.

Figure 2.3 illustrates the data structure handled by each component within the system. Smart Apples dispatch individual packets for each sensor readings to various IoT Gateways. These gateways aggregate the packets into batches of predetermined size and forward them to the central server, which then proceeds to upload them onto the database.

2.1.4 Communication Protocols

In systems with multiple interacting components, defining precise communication protocols is essential to harness their full potential. The integration of the Bluetooth Low Energy protocol for Smart Apples is critical, providing a flexible method for fast, short-range data transfer, thus enabling tailored data transfer techniques for various data types. This adaptability is significant, as it allows devices to request information on an as-needed basis, avoiding communication channel congestion and, crucially, minimizing energy consumption.

Conversely, the implementation of the MQTT protocol on both the gateway and server sides is of equal significance, providing the fundamental features that epitomize an IoT system. Scalability is an imperative consideration in industrial applications, where production processes frequently expand, necessitating the immediate and seamless adaptability of the infrastructure.

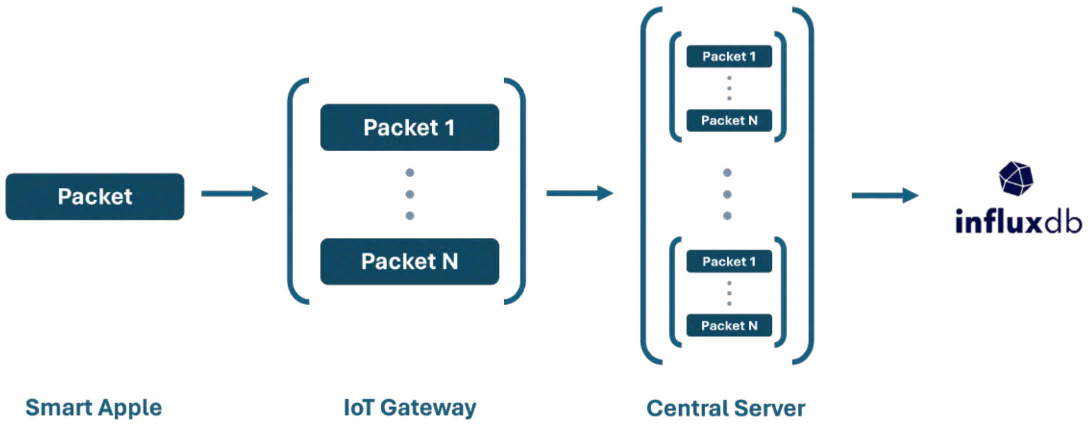


Figure 2.3: The diagram illustrates the data flow within our system, from the acquisition to the database storage. Data packets generated by the Smart Apple are first sent to the IoT Gateway, where they are collected and stored. These packets are then batched together and transmitted to the Central Server. Upon successful reception, the server processes and uploads the data to the InfluxDB database for persistent storage and further analysis.

2.2 System Security and Reliability

2.2.1 Security Measures

Security measures must be taken into consideration when talking about IoT applications for quality control in industrial processes. As outlined in the work "SoK: Investigation of Security and Functional Safety in Industrial IoT" [17], we can make a distinction among the three following groups: Security, Safety and Functional Safety. **Security** in IoT typically refers to the protection of information and systems from malicious attacks. This involves safeguarding data confidentiality, ensuring integrity, and maintaining the availability of the IoT services against unauthorized access and cyber threats. **Safety** relates to the system's ability to operate without causing unacceptable risk of physical injury or damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment. **Functional Safety** is a part of the overall safety that depends on a system or equipment operating correctly in response to its inputs. It is particularly concerned with mitigating the risk of software and hardware errors that could lead to hazards. In the context of Industrial IoT, these aspects are even more important due to the potential for harm in environments like manufacturing plants, critical infrastructure, and other areas where IoT devices are deployed. The integration of safety and security in IoT devices and their operation is essential for creating systems that are not only functionally reliable but also resilient to malicious activities that could disrupt safety functions (see Figure 2.4).

Here's how our system incorporates each of the previously discussed concepts:

- **Security**: Our system operates independently of the internet, with devices interconnected through a local network established by the WiFi Router. Connections to the Raspberry Pis are secured using SSH protocol, which ensures only individuals with verified credentials can gain access. The Nicla sensor data is transmitted exclusively to authenticated devices, and this occurs solely after a handshake message confirms the legitimacy of the connection. Similarly, for MQTT communications, only data packets from authorized devices subscribed to the MQTT topic are permitted to be uploaded to the database. InfluxDB employs secret tokens to authorize the server for uploading sensor data, safeguarding against injection attacks and maintaining controlled access through authentication credentials.

- **Safety and Functional Safety**: In our context, these two notions are somewhat interchangeable since the system's static nature precludes the introduction of any risk to individuals or property due to malfunctioning. The Raspberry Pis are strategically located in areas devoid of personnel traffic or industrial machinery movement. Smart apples are harvested at the final sorting stage to prevent them from entering subsequent processing phases.

2.2.2 System Reliability and Availability

As the Internet of Things (IoT) continues to grow, understanding and ensuring the reliability and availability of its components is crucial. Reliability is the likelihood that an IoT device or system will perform correctly for a certain period under specific conditions. It's vital because if IoT devices frequently fail, they can produce inaccurate data, cause delays, or lose information, undermining the system's usefulness. Availability measures how well a system can be maintained and repaired, incorporating reliability to assess overall performance. For IoT, where constant communication is key, high reliability and good availability are essential to keep everything running smoothly and maintain user trust [15].

In our application, various methods are implemented to ensure the reliability of data acquisition and to prevent the loss of crucial data for analyzing impacts on apples. When packets sent by the Smart Apples reach the IoT Gateways, they are immediately stored in a local file before being packaged and sent in batches to the

MQTT Broker. These packets include a timestamp and an identifier so they can be retrieved in case of a WiFi malfunction or if the Raspberry Pi acting as a server fails. The server also keeps a local copy of all packets received via MQTT before uploading them to the database. Additionally, the gateways send a significant bit to the smart apples to control when to start and stop data transmission. If there is a sudden disconnection, the bit is not reset, allowing the next connected gateway to receive the packets with the correct incremental identifier instead of restarting. A data integrity check is also implemented at the IoT Gateway level, where packets are verified against an established format specification. This ensures that each piece of data is checked for compliance with our predefined structural criteria. In instances where information is missing or a packet deviates from the expected structure, the system responds appropriately to address these discrepancies.

On the availability side, the system itself does not include components that are difficult to set up, and the physical installation of these components does not take much time. Moreover, the system can be quickly scaled by adding more gateways throughout the facility, managing costs and time efficiently.

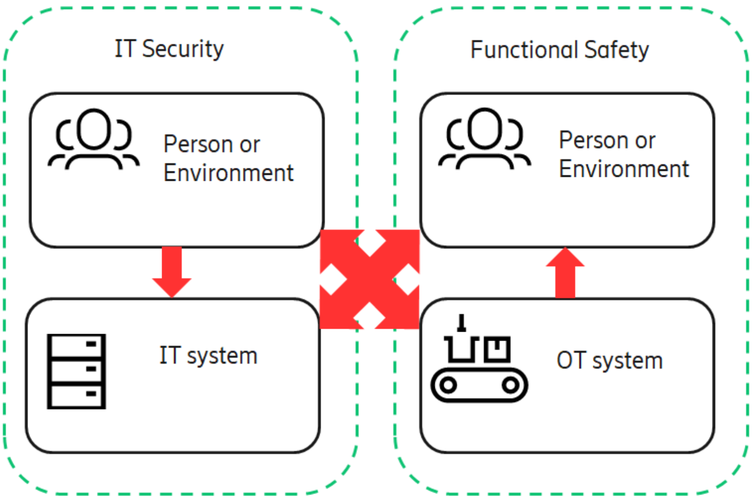


Figure 2.4: This diagram delineates the distinct domains of IT Security and Functional Safety within industrial systems. On the left, IT Security is depicted as a protective measure for information systems, highlighting the flow of potential risk from compromised IT infrastructure towards people or the environment, symbolizing the indirect impacts of cybersecurity breaches. On the right, Functional Safety is represented, focusing on the reliable operation of Operational Technology (OT) systems.

Chapter 3

Smart Apple

With the rise of Industry 4.0 and the Internet of Things, there is an ongoing quest to discover technological solutions that are both efficient and cost-effective for all types of industrial processes. In the agri-food sector, ensuring a high product quality level is paramount, and as a result, the industry is gravitating toward highly informed and data-driven cultivation and production methodologies.

Central to this research is the creation of an IoT device that seamlessly integrates into the post-harvest processes faced by apples in industrial settings. More precisely, this innovation involves an apple whose core has been replaced with a 3D-printed cylinder packed with sensors that track accelerations, angular velocities and other environmental variables. Employing an actual apple ensures the system's authenticity, enabling not just the monitoring of fruit impacts on the production line but also its non-invasive integration into pre-existing facilities.

3.1 Design Concept

3.1.1 Overview

The main idea is to create a system that closely resembles a real apple in both physical properties and behavior, without changing the dynamics of the fruit itself. This section of the study revolves around the developing of a cylinder that can be screwed into a real apple after the core has been removed. This cylinder contains various sensors and is waterproof to prevent water and other materials from entering and damaging the internal components. The cylinder houses an external thread in order to be screwed into the apple, ensuring it does not come out during industrial processes. This design allows the sensor equipped apple to be part of the production flow like any other apple, providing accurate data for analyzing and improving fruit handling processes.

3.1.2 Inspiration

The inspiration behind this project stems from a deep-rooted desire to monitor and enhance fruit quality without intruding on the natural conditions or influencing the behavior of surrounding fruits during the production process. In observing the challenges faced in maintaining apple quality, it became apparent that traditional monitoring methods often fall short, they either lack the precision needed or their presence alters the very conditions they aim to measure. Drawing from the core principle of bio-mimicry [18], where solutions are modeled on biological entities and processes, the idea emerged to develop a device that could not only fit within the physical space of a real apple's core but also match its original properties. This approach ensures that the sensor-equipped apple interacts with the environment and the production machinery just as any real apple would, thereby avoiding any undue influence on its surroundings or on the data collected. Furthermore, this concept takes a leaf out of the coring techniques employed in geology and glaciology, where samples are extracted from the earth or ice to study internal conditions without impacting the overall structure.

Figure 3.1: Smart Apple in the middle, comprising a real apple whose core has been replaced with a mechatronic system for acquiring movements data, such as accelerations and angular velocities. It consists of a 3D-printed cylinder threaded into the apple after the core has been removed, housing an Arduino Nicla powered by a LiPo battery and a mechanical power switch.

3.2 Mechanical Design

The primary component that distinguishes the Smart Apple from a real one is the core, which is replaced by an artificial cylinder created using a 3D printer. It houses a microcontroller board that enables continuous data acquisition necessary for the analysis of apple quality required by Melinda. The main requirements adhered to in designing the device are:

1. **Easy Installation**: Operators in the industrial facility should be able to install the device as easily as possible without complex instructions. This reduces installation times and allows for more focus on data acquisition and analysis.

2. **Ease of Use**: It should be user-friendly without the need for complex training courses, featuring an intuitive system for powering on and off.

3. **Waterproof**: The cylinder containing the sensors must be waterproof to prevent the entry of any material that could compromise the electronic board and battery.

4. **Seamless Integration**: The system design should be non-invasive, ensuring that the artificial apple closely resembles a real one. Once inserted into the production process, it should behave like any other apple to maintain the natural flow and avoid obtaining erroneous acquisitions that do not reflect reality.

5. **Food Safety**: Another important aspect to consider is the type of material used for the device. It must not be toxic or release substances incompatible with food safety criteria.

3.2.1 Cylinder Design

The cylinder has been developed to replace the core of a real apple without altering its mechanical and physical properties. In Figure 3.2, three prototypes developed over time are shown, with improvements applied after field testing.

The cylinder consists of an external threaded part that allows it to be screwed into the apple after the real core has been removed. The thread reflects the characteristics of a male thread, so, in addition to complying with ISO metric thread standards, it has channels that cut the thread vertically. These channels serve to expel the apple material as it is threaded, preventing accumulation that could damage the internal surface of the hole in the apple. This aspect is important because during the production process, we do not want the cylinder to move due

Figure 3.2: The image shows the evolution of the 3D-printed cylinder that replaces the core of a real apple, constituting a Smart Apple. It can be observed how from the first prototype (blue) to the last (red), changes have been made to the position and shape of the threading to improve insertion into the apple. Furthermore, internal modifications have also been made to the microcontroller locking system, not visible in the photo.

to improper threading. Additionally, the use of a high-pitched thread, with a noticeable difference between the inner and outer diameter of the screw, helps to grip the apple pulp firmly, preventing it from loosening during use.

At the circular base of the cylinder, there is a small chamfer that aids in entering the hole, followed by a smooth surface that extends to the threading. The length extension of this part is crucial as it allows for correct positioning inside the apple, ensuring axial alignment and thus serving as a linear guide. If it were too short, the threading could cause incorrect insertion of the cylinder, leading to consequent sensor measurements errors.

Moving to the internal part, given the limited space to accommodate the microcontroller and battery, it was decided to create supports that act as guides for the electronic board to slide on. This is not fastened to the cylinder with screws but held in place by the pressure exerted by the cap that screws onto it, along with a layer of foam to prevent damage. On the upper internal part of the cylinder, there is another threading to screw on a cap, the main purpose of which is to prevent material from entering the cylinder. As shown in Figure 3.3, the cap features a slot with the thickness of a coin allowing for closure and screwing into the apple.

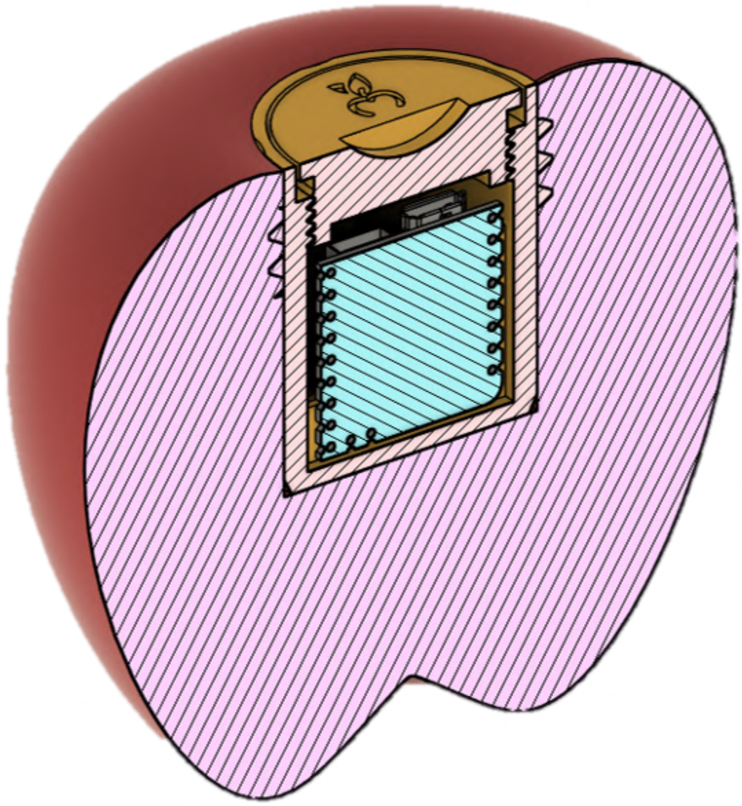


Figure 3.3: A CAD design depicting the cross-section of the Smart Apple that reveals how the device components are organised within the system. The 3D printed cylinder is threaded into a real apple and sealed with a rubber o-ring placed between its body and the cap.

3.2.2 Waterproofing Mechanism

A critical feature of the cylinder is its waterproofing capability, designed to protect the internal sensors and electronic components from moisture and particulate ingress. This is achieved through the use of rubber o-rings and a system similar to that used for shaft-hub coupling, where there a shoulder is present [11]. The cap of the cylinder incorporates a threading mechanism that, when closed, forms a tight seal against the body of the cylinder. This waterproofing mechanism ensures the device's functionality is not compromised by the washing stages or humidity levels prevalent in apple processing environments. The main reason for opting for a cylindrical design is precisely for this purpose, as it would have been difficult to create a waterproof system with a different geometry without the addition of materials or hydrophobic spray on the surface of the cylinder and internal components.

3.2.3 Manufacturing Materials

Considering the cylinder is to be introduced into an industrial setting where food hygiene is very important, it must adhere to specific material composition and form characteristics. It has been fabricated using a 3D printer with PLA (polylactide) which is a biodegradable bioplastic derived from renewable resources such as corn, tapioca, or potatoes. It is extensively employed as a packaging material for food products, particularly those with a shorter shelf life, including fruits, vegetables, meat, and seafood, due to its food-safe properties.

The use of PLA is deemed food safe under certain conditions, which are outlined as follows:

- The printed object should be as smooth as possible to minimize areas where bacteria can accumulate. In our case, the only part exposed to production line is the top part of the cap;
- Only use PLA filaments that are free from any additives;
- Employ a stainless steel nozzle for printing and ensure it is completely clean to avoid contamination from other not suitable materials;
- PLA is not suitable for dishwasher cleaning as the high temperatures can degrade the material;
- For objects intended for frequent use, consider coating with a food-safe epoxy resin to provide a durable, cleanable surface.

Adhering to these guidelines ensures that the material used in manufacturing the cylinder remains safe for contact with food, aligning with the hygiene standards required in the industrial food processing environment.

3.3 Electronic Components

The selection of electronic components is critical in ensuring the sensor-augmented apple meets the rigorous demands of real-world agricultural monitoring while nestled within the fruit. The chosen components must not only be compact and energy-efficient but also capable of precise measurements across a spectrum of environmental conditions.

The main ones we find inside the 3D printed cylinder are:

- **Arduino Nicla**: compact microcontroller board with many high quality BOSCH sensors on-board. It is the heart of the Smart Apple with the aim of gathering data from the environment and send it to the different devices in the infrastructure.

- **LiPo Battery**: lithium-ion polymer battery essential for providing the sufficient energy supply to the circuit.
- **Sliding Mechanical Switch**: used for switching on and off the device.

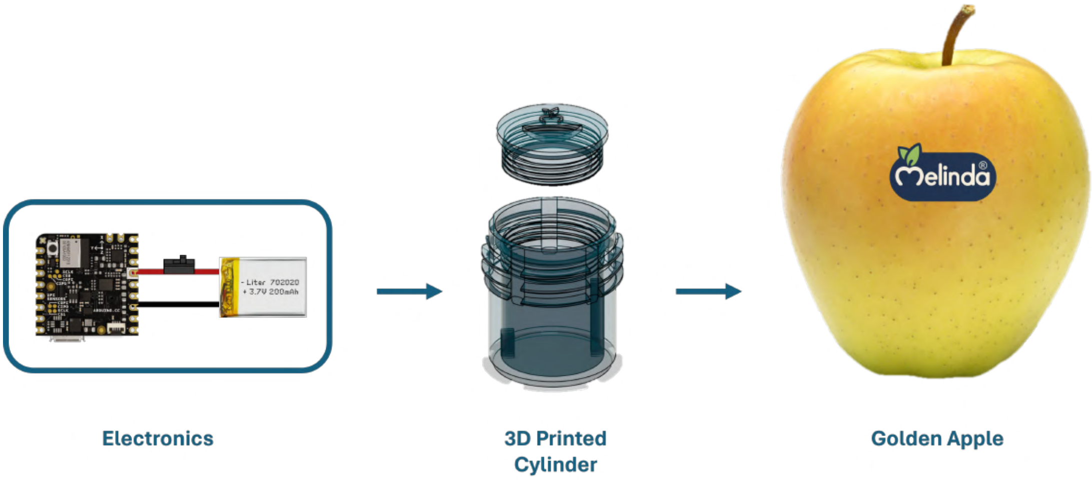


Figure 3.4: Diagram illustrating the Smart Apple's high-level architecture: from the initial arrangement of the Arduino Nicla and battery connected via a mechanical sliding switch, to the integration of these components within the 3D-printed cylinder, and finally to the insertion into a real apple to create a fully functional monitoring device for agricultural analysis.

3.3.1 Arduino Nicla

The Arduino Nicla Sense ME is the chosen microcontroller for this project due to its small form factor, making it suitable for embedding within the envelope of the sensor module. Its low power consumption is pivotal for long-term deployment in the field without frequent battery charges. The inclusion of a BLE antenna allows for efficient wireless communication, a necessity for transmitting data in real time without the constraints of wires. Moreover, the Nicla comes equipped with a suite of Bosch sensors known for their accuracy and reliability. These sensors, supported by proprietary signal filtering algorithms, are capable of capturing a wide range of data points including temperature, humidity, pressure, CO2 levels, acceleration, and gyroscope readings.

Here is an overview of main components of the board, as shown in Figure 3.5 and in Table 3.1 [2]:

- **Microcontroller**: at the heart of the Nicla Sense ME is the nRF52832, a powerful and versatile System-on-Chip (SoC) from Nordic Semiconductor. The nRF52832 is built around a 32-bit Arm Cortex-M4 processor running at 64 MHz.
- **Onboard advanced motion sensors**: the board features the BHI260AP, a smart IMU that includes a 3-axis accelerometer and a 3-axis gyroscope. It is trained with Machine Learning algorithms able to perform step counting, position tracking, and activity recognition. The board also features the BMM150, a compact geomagnetic sensor from Bosch Sensortec including a 3-axis magnetometer.
- **Onboard environment sensors**: the Nicla Sense ME is equipped with the BME688, this is the first gas sensor with Artificial Intelligence (AI) and integrated high-linearity and high-accuracy pressure, humidity and temperature sensors. The gas sensor can detect Volatile Organic Compounds (VOCs), volatile sulfur compounds (VSCs) and other gases, such as carbon monoxide and hydrogen, in the part per billion (ppb) range.
- **Wireless connectivity**: the board supports Bluetooth Low Energy connectivity, provided by the ANNA-B112 module developed by u-blox. This compact, high-performance Bluetooth Low Energy module allows the Nicla Sense ME to communicate wirelessly with other devices and systems.
- **Power management**: the Nicla Sense ME is designed for ultra-low power operations, with efficient power management features that ensure minimal energy consumption even when using always-on motion recognition and environment analysis sensors. The Nicla Sense ME features the BQ25120 from Texas Instruments, a highly integrated battery charge management integrated circuit (IC) designed for wearables and Internet of Things (IoT) devices.

This board versatility makes the Nicla an ideal choice for the diverse monitoring requirements of the agri-food sector. The ready-to-use aspect of the Arduino Nicla, with its wealth of examples and a robust online community, ensures support and resources are readily available. This accessibility simplifies the development process and allows for rapid troubleshooting and enhancement, making it an invaluable asset for this thesis project.

3.3.2 Battery Considerations

The battery selection process was approached with deliberate consideration of size, lifespan, and impact on other components. A small, high-capacity lithium-polymer

Component	Description
BHI260AP	AI smart sensor hub with integrated 6-axis IMU (3-Axis Accelerometer + 3-Axis Gyroscope) for activity detection, powered by a 32 Bit Synopsys DesignWare ARC™ EM4™ CPU
BMM150	Low noise magnetometer with a typical range of ±1300uT in the X,Y axis and ±2500uT in the Z axis
BME688	Environmental sensor that can measure pressure, humidity and temperature. Additionally, the onboard smart gas sensor can also help in determining the air quality index by detecting a broad range of gases, including Volatile Organic Compounds (VOC).
BMP390L	High performance pressure sensor operating between 300 - 1250 hPa with low drift.

Table 3.1: List of sensors incorporated within the Arduino Nicla, accompanied by their identifiers and respective functionalities. It includes an accelerometer, gyroscope, and magnetometer essential for monitoring apple movements and detecting impacts. Additionally, environmental sensors like temperature, pressure, and air quality sensors are included.

(LiPo) battery was chosen to fit snugly within the sensor module while providing several hours of reliable service. Its dimensions were specifically picked to not overshadow the BLE antenna on the Nicla board, avoiding potential interference with the signal transmission. The battery is interfaced with the Nicla using a mechanical switch that allows the system to be turned on and off, conserving power when the device is not in active use. Notably, the Nicla's printed circuit board (PCB) includes a charging stage, which enables the battery to be recharged via the micro USB port. For charging to occur, the system must be switched on to connect the battery poles to the charging integrated circuit (IC), ensuring efficient energy management and device longevity. It is suggested on the datasheet to set the charging current at half the maximum capacity. In our case we used a 100mAh LiPo battery and it is charged at 50mAh when connected to the micro-USB port.

To calculate the energy stored in the battery the following formula can be used:

$$\text{Energy (Wh)} = \text{Capacity (Ah)} \times \text{Voltage (V)} \tag{3.1}$$

While the lasting time of the battery can be calculated with:

$$\text{Time (hours)} = \frac{\text{Battery Capacity (Ah)}}{\text{Current (A)}} \tag{3.2}$$

In Table 3.2 all the different lasting times based on operating voltage and data transfer rate are shown and in Figure 3.7 all the current consumption curves

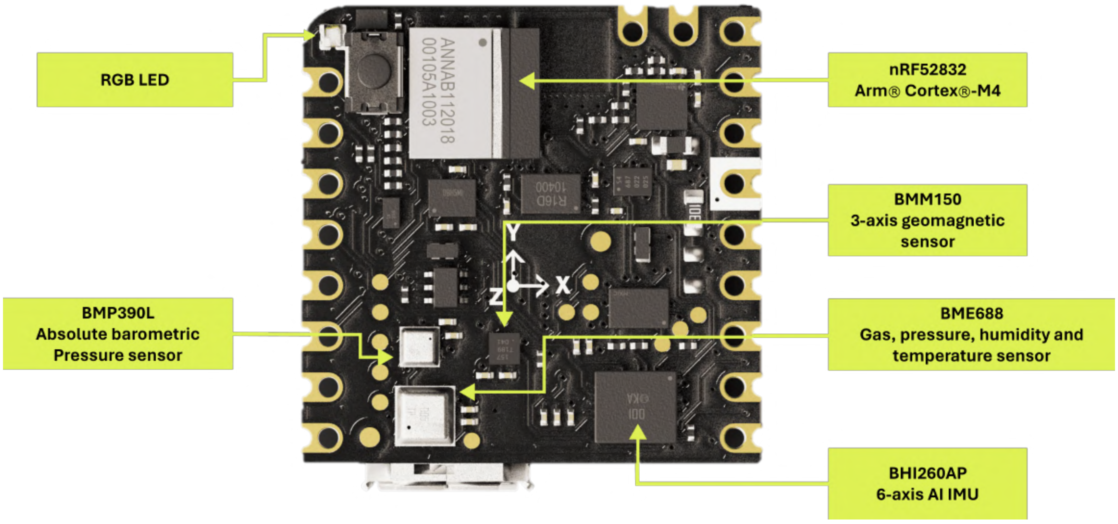


Figure 3.5: Top view of an Arduino Nicla. The various integrated circuits are compactly arranged, providing a useful form factor for many IoT applications. Note the *nRF52832* processor on the upper part of the board, featuring the built-in BLE antenna.

grouped by operating voltage and data transfer rate are present. We notice how currents are naturally placed in three distinct groups, representing the different voltage levels. Each group contains the current consumption over 10 seconds of acquisition using different BLE transfer rate ranging from *10 Hz* to *30 Hz*. The more the transfer rate the more the power consumption of the microcontroller. On the other hand, the higher the operating voltage the higher the battery lasting time.

The energy stored in a 100mAh fully charged 4.2 V battery is approximately 0.42 Wh (Watt-hours). It means that a Smart Apple which consumes an average of 13.6 mA transmitting BLE packets at 30 HZ will last approximately *7.32 hours* before powering off.

3.4 Operational Design

3.4.1 Tool Integration

To facilitate the device's installation and removal within the apple, the cylinder design incorporates a tool integration feature. A specifically designed groove on the cap allows for the use of a common coin or a simple tool to tighten or loosen the device, making it accessible for deployment and retrieval without the need

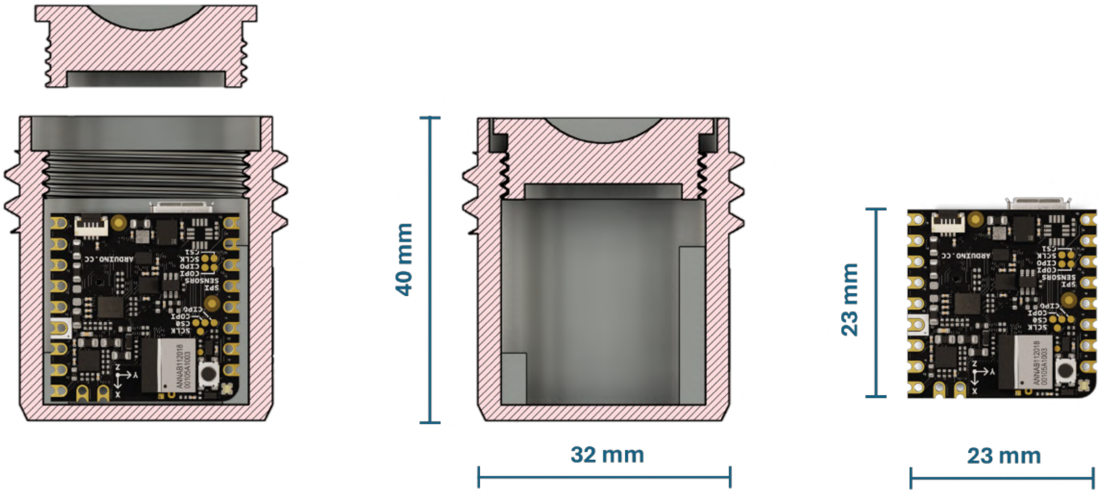


Figure 3.6: Diagram showing the Arduino Nicla integrated into the 3D printed cylinder which will be placed inside a real apple. The cutaway drawing helps in visualising the interior of the cylinder and how the threaded cap is fastened. Also, the dimensions are shown to give an idea of the envelop size.

for specialized equipment. This feature is crucial for practical field applications, allowing workers to easily prepare the apples for monitoring or remove the devices for recharging, data retrieval and maintenance if needed.

3.4.2 Dimensional Considerations

Referring to Figure 3.6, the internal design of the cylinder includes two guiding channels to correctly position the Arduino Nicla board, which cannot be fastened with screws due to spatial constraints. The snug fit of the board is achieved by sliding it down these guides and securing it with the threaded cap. This cap has an additional layer of sponge-like material to dampen vibrations and maintain the stability of the board within the cylinder. The wall thickness of the cylinder is 2mm, providing a structure that is both lightweight and sturdy enough to withstand the operational demands without adding significant weight to the fruit. Moreover, the base of the 3D-printed cylinder features a hollow pattern that not only reduces material use but also cleverly allows for the visibility of the Nicla's status LED from the outside. This design consideration enables operators to quickly verify the device's operational status, ensuring it is streaming data and maintaining a connection to the gateway without the need to open the cylinder. The entire structure is optimized for 3D printing, eliminating the need for support materials and enhancing the manufacturing efficiency. A 20mm nominal diameter o-ring sits within a precisely machined groove on the cylinder, creating a waterproof seal to

	10 Hz	20 Hz	30 Hz
3.7 V	6.81 h	6.80 h	6.76 h
4.0 V	7.19 h	7.16 h	7.10 h
4.2 V	7.40 h	7.36 h	7.32 h

Table 3.2: This table shows the battery life of the Smart Apple during BLE data transmission, varying with different voltage levels and data transfer rates. The battery capacity taken as reference is 100mAh and it can be observed that its life decreases with the increasing of the data transfer rate and decreasing of the working voltage.

prevent the ingress of moisture and debris. To facilitate ease of access, the cap is designed to be operated with a simple coin, allowing for tool-free opening and closing of the sensor module.

3.4.3 Hole Creation

The process of preparing the apple to receive the sensor module begins with the removal of the core using a modified steel pineapple corer, whose diameter is conveniently the same as the external diameter of the sensor cylinder 32mm (Figure 3.8). This coring tool, after detaching its original spiral blade, provides a clean and precise cut, allowing for the smooth insertion of the sensor module. For an apple to be suitable for this process, it should have a minimum diameter of 60mm, accommodating the 40mm height of the cylinder. The thread occupies one-quarter of the cylinder's height, with the remaining three-quarters remaining smooth to act as a guide. This operational design ensures that the sensorized apple can be seamlessly integrated into the production process, mirroring the natural apple's properties and behaviors as closely as possible, which is essential for accurate data acquisition.

3.5 Data Transmission

In the context of sensor-augmented fruits for agricultural monitoring, the data transmission design must be efficient, reliable, and precise. The following sections detail the mechanisms and considerations that underpin the successful transmission of sensor data from the device to the data collection gateway.

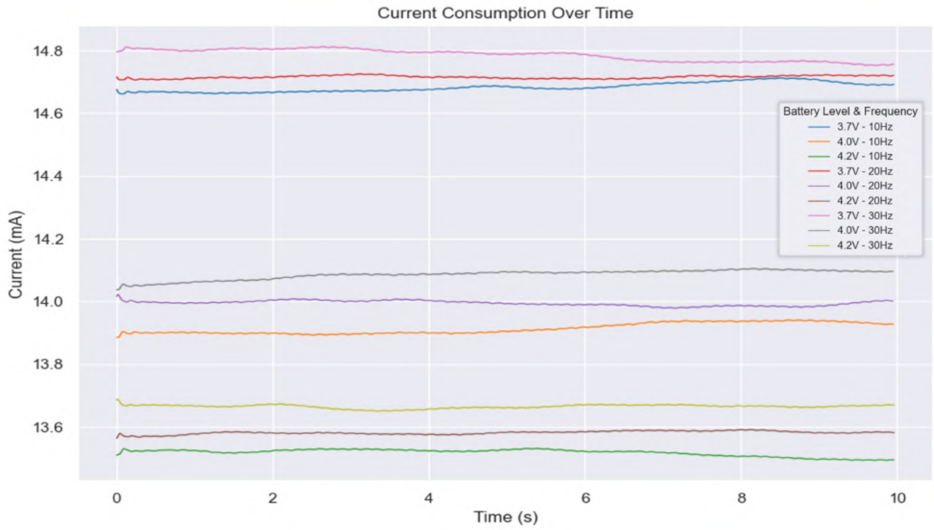


Figure 3.7: The current consumption of the Arduino Nicla while streaming the sensors data over BLE is shown. We can see how 3 groups are formed based on the operating voltage of the system together with its data transfer rate. The measurements are made taking into account 3.7V, 4V and 4.2V over a time span of ten seconds with varying frequency from 10Hz to 30Hz.

3.5.1 Type of Data Collected

The Arduino Nicla Sense ME is equipped with a Bosch BMI160 Inertial Measurement Unit (IMU) which allows to measure different kind of data motion-related. The useful ones for our application are:

- **Acceleration**: Linear accelerations along X, Y and Z axes of the apple are captured. They are fundamental to detect acceleration peaks during the acquisition in the production line and asses impacts;

- **Angular Velocity**: Monitors the apple's change of orientation rate to refine the detection of impact together with linear accelerations;

- **Quaternions**: Calculated using a proprietary sensor fusion algorithm that compiles data from the accelerometer, gyroscope, and magnetometer. They are useful to log the orientation's history of the apple over time during the production line acquisition.

A summary of the sensor data types is provided in Table 3.3

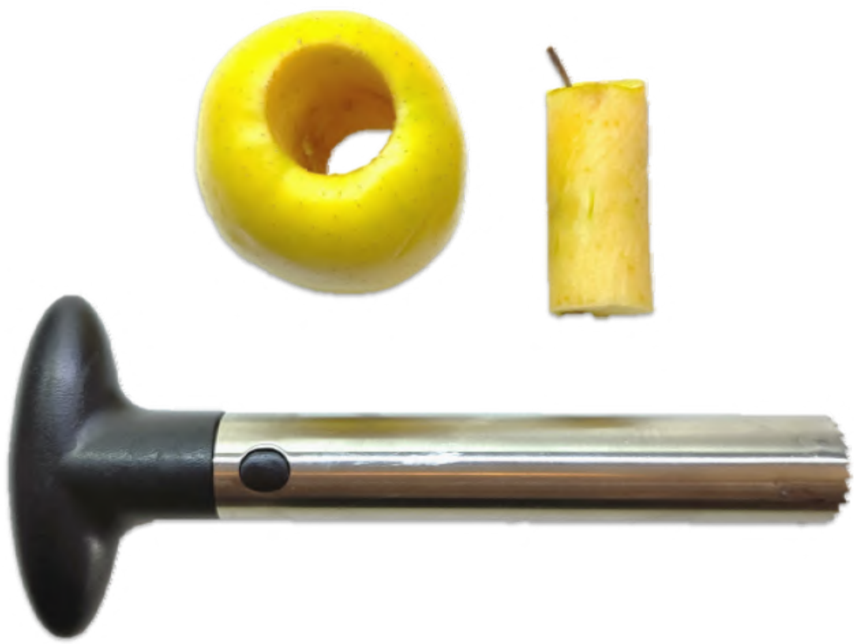


Figure 3.8: An apple corer tool used to precisely remove the core of the fruit, making room for the sensor module. The tool is designed for ease of use and to minimize disruption to the apple's structure.

3.5.2 Frequency of Data Transmission

The frequency at which data is transmitted is determined by the nature of the data itself:

- **Motion Data**: Accelerometer, gyroscope, and magnetometer data is transmitted at a high frequency of 30Hz. This ensures that any transient forces or movements that could affect the apple's quality are captured.
- **Environmental Data**: Temperature and other environmental readings are transmitted less frequently, triggered by significant changes. This strategy reduces power consumption and data traffic, focusing on contextual shifts that might impact the fruit's quality.

3.5.3 Method of Data Transmission

Bluetooth Low Energy (BLE) is employed as the transmission protocol, chosen for its balance of energy efficiency and effective range suitable for the agricultural environment There are multiple ways by which the sensor values can be gathered:

Sensor	Data Type	Format	Sensitivity
Accelerometer	Linear Acceleration	Float (12 bytes)	2%
Gyroscope	Angular Velocity	Float (12 bytes)	3%
Quaternions	Orientation	Float (16 bytes)	

Table 3.3: Detailed Specifications of Onboard Sensors: This table enumerates the sensors integrated into the Arduino Nicla Sense ME module, outlining their respective data types, the precision format in bytes, and the sensitivity each sensor provides. The accelerometer and gyroscope offer granular movement data, while the quaternions give a complete picture of spatial orientation. The latter have 4 bytes more since there is one added component respect to the accelerometer and gyroscope outputs.

- **Continuous streaming**: the data is sent to the gateway as soon as it is collected by the microcontroller. This method is useful to get a high frequency of data points, without any sort of buffering, and allows you to get the data in real time. However, the latter is not strictly necessary for this project since the data can still be analyzed after the apples have been collected.

- **Buffered streaming**: the data is sent to the gateway only when the buffer is full. This method is useful to reduce the amount of data sent to the gateway and to save battery life. However, if the buffer is too small, the data might be lost if the gateway is not reachable when the buffer is full.

- **Hybrid streaming**: the data is sent continuously to the gateway if the BLE communication is working properly, otherwise it is buffered and sent when the connection is restored. This method is useful to get the best of the two previous methods, but still the buffer size limitation applies.

- **Hybrid On-Event**: the data is sent continuously to the gateway if the BLE communication is working properly, otherwise only certain events are buffered and sent when the connection is restored. This method is useful to keep the buffer size small and to save battery life, but still get the most important data points.

- **Offline**: the data is stored in the microcontroller and sent to the gateway only when the apple is collected. This method is useful to save battery life and to reduce the amount of data sent to the gateway but it needs some external storage to be used, such as a SD card.

The transmission strategy accommodates the operational constraints of the agricultural setting, ensuring that data is sent and received reliably, enabling the monitoring system to provide timely and accurate insights into the apple's condition throughout the production process.

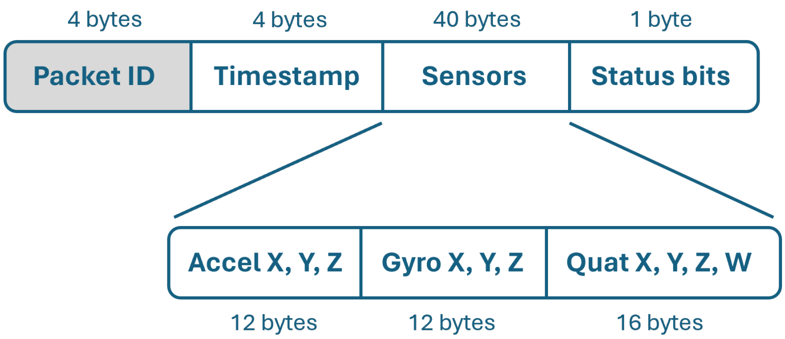


Figure 3.9: This diagram presents the composition of the data frame transmitted via BLE, providing the payload's structure and the specific characteristics of the data transmitted. It includes the packet ID for data sequence integrity and optional timestamps for contextual data analysis. Furthermore, it details the data points from the accelerometer, gyroscope, and quaternions, specifying the precise formatting used for each axis of measurement, ensuring accuracy and consistency in data capture and transmission.

Data is organized within the BLE payload using a structured approach to ensure the integrity and order of the collected information 3.9. Each packet includes a unique packet ID for synchronization and a timestamp when necessary, ensuring data continuity and facilitating accurate post-processing analysis. The BLE communication protocol allows to send up to 255 bytes of payload in a single packet and the different sensors have their own transmission strategy.

- **Packet ID**: the packet ID is a sequential number which is used to identify the sensor readings. This is useful to detect missing packets and to synchronize the data from different sensors. It is sent as a 4 bytes unsigned integer.

- **Timestamp (optional)**: the timestamp is sent only when the data is buffered and sent later. It is useful to determine the relative time between two packets. It is sent as a 4 bytes unsigned integer.

- **Accelerometer**: the accelerometer sends the data as soon as it is available. It is composed of three floating point numbers, one for each axis, which are sent as 4 bytes each.

- **Gyroscope**: the gyroscope sends the data as soon as it is available. It is composed of three floating point numbers, one for each axis, which are sent as 4 bytes each.
- **Quaternions**: the quaternions are sent as soon as they are available. They are composed of four floating point numbers which are sent as 4 bytes each.

In BLE the data can be organized into characteristics, which are identified by a UUID (Universally Unique Identifier) and can be read, written, or notified. The *Temperature* characteristic is read-only, which means that its value is only sent when the client requests it. The *Accelerometer*, *Gyroscope* and *Quaternions* characteristics are notified, which means that their values are sent as soon as they are available.

3.6 Firmware Design

The main purpose of the Arduino Nicla is to collect data from the sensors on the board and send them to any BLE devices that request them. The procedure involves initializing the various sensors and BLE communication and waiting for a device to connect and request data streaming. Once the Nicla receives a confirmation bit, it begins collecting data, creating the frame, and sending it in the most appropriate way. When a connected device sends a stop bit, the Nicla returns to being available for any other devices that may need sensor information.

In Figure 3.10, a detailed flowchart is presented which outlines the operational logic of the firmware embedded within the Arduino Nicla used in the Smart Apple. The diagram systematically illustrates the sequential firmware processes starting from the initialization of the sensors and BLE (Bluetooth Low Energy) communication. It further details the conditional wait for a BLE device's request to begin data transmission, followed by the sending of sensor data packets. Upon receiving a command to cease data streaming, the system halts transmission and reverts to a passive listening state, ready for the next operation.

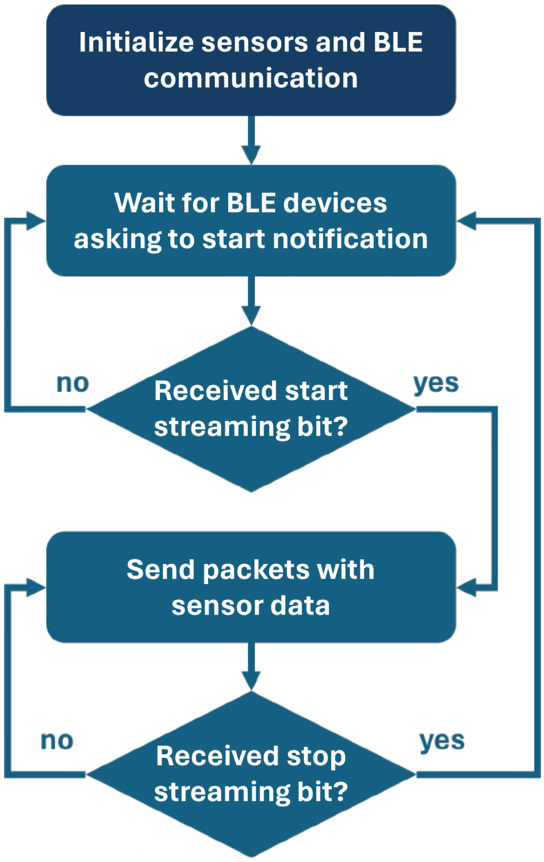


Figure 3.10: Flowchart illustrating the firmware operation within the Arduino Nicla embedded in the Smart Apple. The diagram depicts the initialization of sensors and BLE communication, followed by waiting for a connection request from a device to commence packet transfer. When no longer needed, it disconnects and returns to listening mode.

Chapter 4

Raspberry Pi IoT Gateway

Raspberry Pi gateways serve as crucial nodes within our IoT ecosystem, acting as intermediaries that facilitate communication between Smart Apples and the central data processing server. These gateways are responsible for collecting data transmitted over BLE from the sensor-augmented apples and forwarding it via MQTT to ensure it reaches the server for database storage and further analysis.

4.1 Hardware Configuration

This section outlines the specific hardware components chosen for the Raspberry Pi IoT Gateway, detailing the selection of the Raspberry Pi 4, the integration of an external BLE antenna, some considerations for power supply and mounting solutions within the production plant environment.

4.1.1 Raspberry Pi 4

A Raspberry Pi is a powerful single-board computer (SBC) that is about the size of a credit card and widely finds its use in many applications around the STEM community and industry. All the components needed for a fully functional computer, including the processor, memory, video chipset, storage, and so forth are built onto the printed circuit board which makes it much more compact and less expensive than conventional computers. It has many ports and pins that allow the connection of different devices and components, such as cameras, sensors, LEDs as well as using add-on boards, called HATs (Hardware Attached on Top), to extend its functionalities. It is designed to run any ARM-based Linux distribution as operating system.

The Raspberry Pi 4 Model B was selected for its robust processing capabilities, sufficient to manage the data traffic coming from the BLE communication and

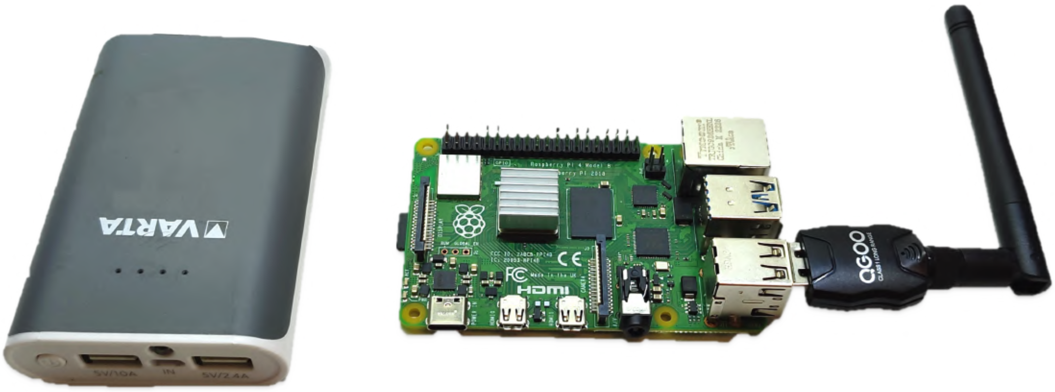


Figure 4.1: Key components of an IoT Gateway. On the left, a power bank is used as the power source, allowing the gateway to be positioned anywhere within the facility. On the right, a Raspberry Pi 4 equipped with a Bluetooth 5.0 dongle provides a stronger signal compared to the integrated antenna, necessary due to machinery in the facility that weakens the BLE signal from the Smart Apple.

forward them through WiFi. The most essential components for our application can be seen in Figure 4.2:

- **Broadcom BCM2711 CPU:** A 64-bit quad-core processor that provides the computing power required for data processing and multitasking;

- **8 GB RAM:** Large memory capacity to handle simultaneous operations and large datasets, ensuring smooth execution of processes;

- **2.4 / 5 GHz WiFi:** Dual-band wireless connectivity that supports high-speed internet access and communication with other network devices;

- **Bluetooth 5.0:** Advanced wireless technology that offers fast and reliable connections with BLE sensors and devices;

- **Micro SD Card Slot:** The primary storage medium for the operating system and any additional data, allowing for easy updates and data management;

- **40 Pin GPIO Header:** A versatile interface for connecting a variety of external sensors, actuators, and expansion boards to extend the Pi's capabilities;

- **USB 3.0 Ports:** Fast data transfer ports that enable the connection of modern peripherals and devices for enhanced functionality;
- **USB-C Port:** The power input port for the Pi, which supports a 5V/3A power supply, crucial for maintaining stable operation under load.

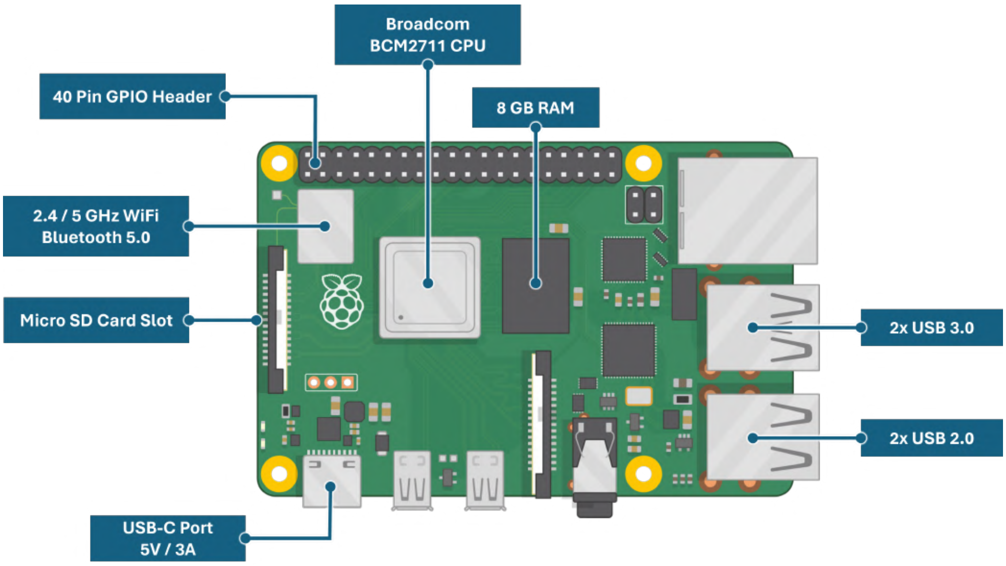


Figure 4.2: Top view of a Raspberry Pi 4, used as a central component in the IoT Gateway setup. This robust single-board computer, comparable in size to a credit card, encompasses various essential features tailored for IoT applications. These include built-in Wi-Fi/Bluetooth antennas, a powerful processor, expanded RAM capacity, and the ability to connect external peripherals via USB ports and GPIO headers.

4.1.2 External BLE Antenna

In environments with high background noise, such as the Melinda facility containing numerous machinery employing motors that may cause interference or have shielding materials hindering wireless signal transmission at certain frequencies, it's crucial for the wireless signal to be robust enough to overcome these obstacles. On the Raspberry Pi 4, the built-in Bluetooth antenna is shared with the WiFi antenna, potentially compromising its performance. Hence, we opted to use an external Bluetooth dongle. To assess its effectiveness, range tests were conducted, with 5 RSSI samples taken every two meters, covering a total distance of 20 meters. A Microsoft Surface Pro 7 and a Raspberry Pi 4 were tested with and without the use of the external Bluetooth adapter, with results shown in Figure 4.3.

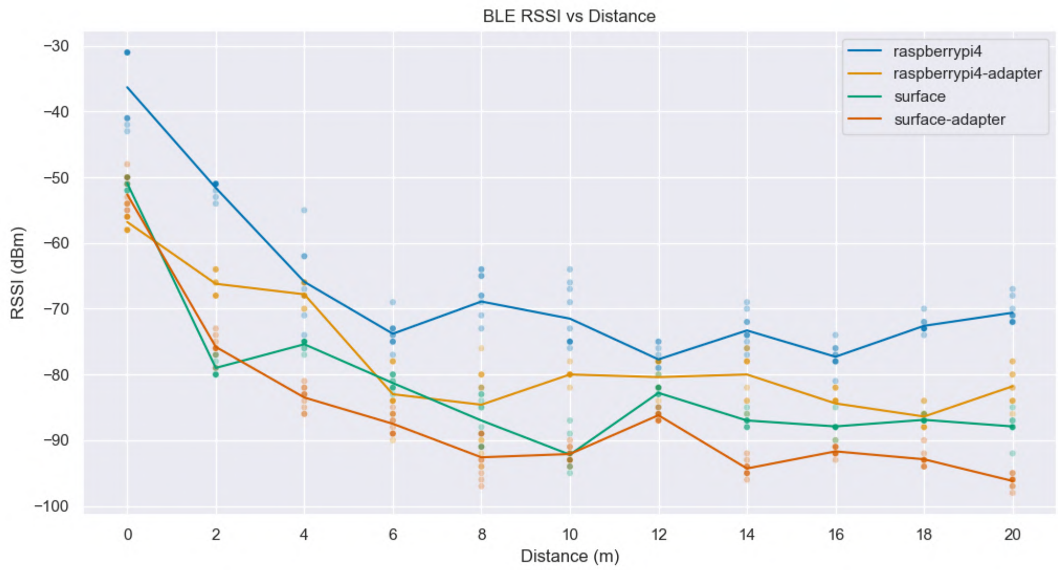


Figure 4.3: Line plots showing the BLE signal strength over a 20-meter range, using different devices with and without an external Bluetooth adapter. Measurements were taken in an open space without obstacles every two meters, with data points indicating the number of samples collected. The Raspberry Pi 4 without the external adapter exhibits slightly better signal strength, although not significantly different from other devices. This is attributed to the outdoor data acquisition setting, which lacks the potential interference from industrial machinery present in an industrial environment.

4.1.3 Power Supply

The uninterrupted operation of the Raspberry Pi gateways is a critical component of the IoT infrastructure within the agri-food production plant. To facilitate continuous operation without dependency on fixed power sources, each gateway is equipped with a USB power bank. The power banks were selected based on their capacity to provide a reliable power supply over extended periods, their proven durability in industrial settings, and the simplicity with which they can be swapped out and replaced as needed. This portable power solution enables the gateways to function effectively even in the absence of proximate electrical outlets.

To further enhance the flexibility of the setup, strategic positioning of the Raspberry Pi gateways was carried out following extensive signal strength tests. This careful placement ensures comprehensive coverage and uninterrupted data acquisition from the smart apples, even in the more remote or obstructed areas of the plant.

The IoT gateways are mounted on telescopic camera tripods placed around the production line. Each tripod is fitted with a custom-designed 3D printed bracket, tailored to securely house the Raspberry Pi 4 with its own power bank unit. This adaptable mounting approach allows for rapid reconfiguration and precise positioning, thereby optimizing the BLE communication range with the smart apples. Furthermore, the elevated positioning provided by the tripods protects the sensitive electronic devices from potential damage due to accidental contact with machinery or staff, as well as from environmental factors such as dust or water spillage.

4.2 Software Setup

This section delves into the software configurations and operational scripts developed, focusing on modifications to the operating system for enhanced stability and the deployment of an automated Python script for BLE device monitoring.

4.2.1 Boot Services

To optimize the Raspberry Pi for its role as an IoT gateway, several adjustments were made to the default configuration Raspbian operating system. WiFi power management features were disabled to prevent unintended disconnections, a crucial modification given the plant's WiFi setup lacks internet access. This is done by creating a bash script that runs on boot using Systemd, which is the suggested daemon program manager to use in Linux operating systems.

4.2.2 Python Script

After the system is booted up and all the boot services have finished running, the main python script is started automatically. It is meant for continuously scanning for Smart Apples, given a certain name pattern and MAC address, requesting a BLE communication for receiving sensors data and send it to the main server after some enhancements. The implementation relies on two key Python libraries: ‘paho-mqtt‘ for handling MQTT communications and ‘bleak‘ for BLE operations. ‘paho-mqtt‘ enables efficient data transmission to the MQTT broker, while ‘bleak‘ offers comprehensive tools for BLE device interaction, including scanning and data exchange. The high level flowchart of the python script is presented in Figure 4.4.

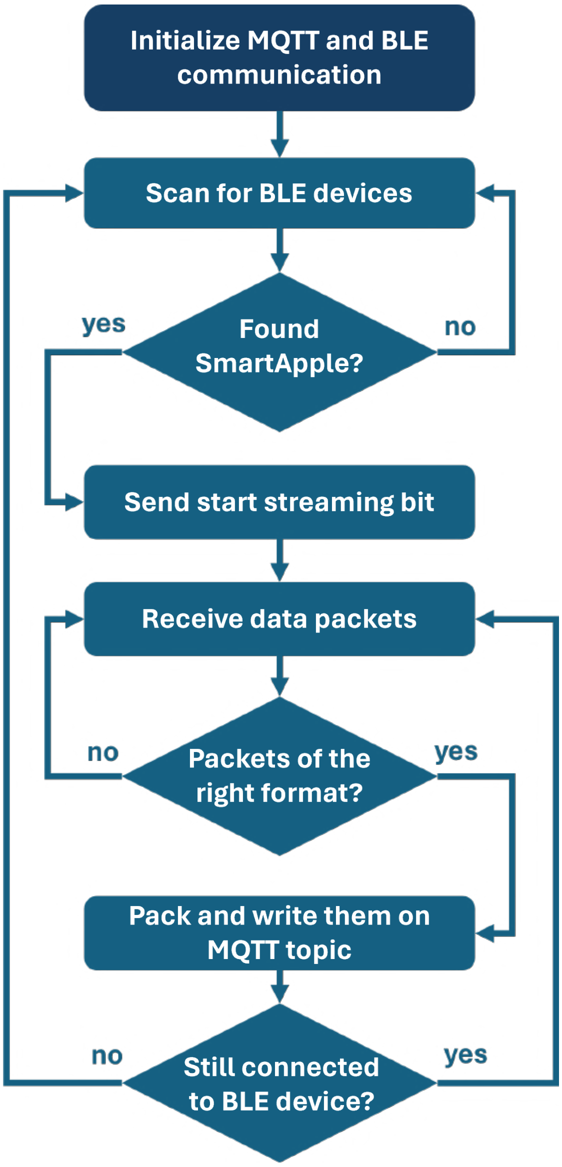


Figure 4.4: The flowchart illustrates the Python script executed on boot of the Raspberry Pi 4 acting as the IoT Gateway. Upon initialization of BLE and MQTT communication, it listens for Smart Apples to receive sensor data. Once data is requested, a communication channel is established to receive BLE packets from the Smart Apple, which are then grouped and sent to the IoT server by writing to an MQTT topic. When the Smart Apple is no longer connected, the gateway returns to listening mode.

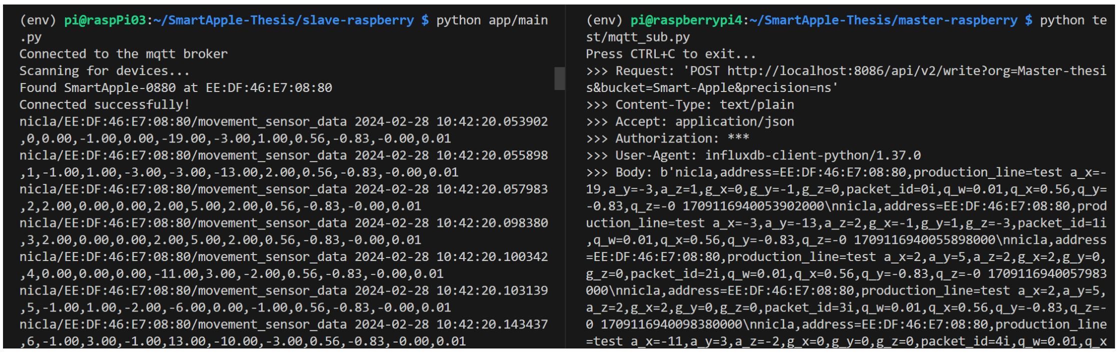


Figure 4.5: The figure depicts two terminal windows side by side, each establishing an SSH connection with an IoT Gateway and the Raspberry Pi 4 server side. The left terminal shows the gateway connecting to the MQTT broker and initiating a scan to detect Smart Apples. Once detected, it establishes a connection and begins receiving individual packets. After collecting a sufficient number of samples, it writes them to the MQTT topic to make them available to the server-side device. In the right terminal, the Raspberry Pi 4 server side creates an MQTT topic and awaits data from connected devices. Upon receiving data, it saves them to a bucket in the InfluxDB database for further analysis.

4.3 Data Handling and MQTT Communication

4.3.1 Data Preparation and Transmission

Upon receiving data from the Arduino Niclas inside the sensor-augmented apples, the Raspberry Pi gateway prepares the data for transmission by batching it into structured payloads. This preparation involves adding a packet timestamp and aggregating data points to optimize network usage when possible. The MQTT protocol is then used to transmit these batches to the designated topic on the 1883 server port of the MQTT broker, ensuring efficient and organized data flow. In Figure 4.5 the exchange of messages between an IoT Gateway and the IoT Server is shown.

4.3.2 Reliability in Data Forwarding

Reliability in data forwarding is fundamental, especially in an IoT ecosystem where decision-making relies on timely and accurate data. The gateway employs several strategies to ensure data reaches the MQTT broker reliably, including QoS settings in MQTT to manage message delivery guarantees and implementing reconnect strategies in case of network interruptions. It also stores the sensor data locally as a backup in case of power loss or operating system corruption.

Chapter 5

Raspberry Pi Server Side

Critical for the proper functioning of the entire IoT architecture of the project and to ensure effective integration into the Melinda facility is the role of the server-side device. This device is responsible for providing a means for other devices to forward data collected from the Smart Apples and channel it into the database. Additionally, it enables the retrieval of this information and performs manipulations and analyses for assessing the impacts experienced by the apples, providing an intuitive dashboard from which results can be observed interactively.

5.1 Hardware and Software Configuration

5.1.1 Hardware Specifications

The device responsible to work as a server for the IoT infrastructure is a Raspberry Pi 4 Model B, chosen for its balance of performance, connectivity, and cost-efficiency. With a quad-core CPU, 8GB of RAM, and ample USB and Ethernet ports, it provides a robust platform for handling MQTT communications and data processing. The components description can be found at Chapter 4.1 since it is the same used for the IoT Gateway implementation. The only hardware difference can be found on the external BLE antenna due to the fact that the Raspberry Pi responsible to handle server side communications doesn't imply the use of Bluetooth. It is placed at the center of the production line in Melinda facility supported by a telescopic tripod.

5.1.2 Python Script

After the system is booted up and all the boot services have finished running, the main python script is started automatically.

The main operational flow consists of:

Figure 5.1: Key components of the IoT Server. On the left, a power bank is used as the power source, allowing the server to be positioned anywhere within the production line due to scarcity of power outlets. On the right, a Raspberry Pi 4 is used for managing the MQTT communication as well as the storing of the data in the InfluxDB database. The black 3D printed part makes it possible to adjust the Raspberry Pi on a telescopic camera tripod.

1. Setting up a MQTT server which devices can discover and publish data into it;
2. verify that sensors data are coming for authorized devices and batch them together;
3. push them to the InfluxDB database for later retrieval and further analysis.

The system utilizes Mosquitto as the MQTT broker software, renowned for its lightweight and high-performance capabilities. In Figure 5.2 the flowchart is shown.

5.1.3 Docker Containerization

To streamline deployment and ensure consistency across environments, we've containerized our MQTT broker, InfluxDB, and Grafana using Docker. This approach simplifies the management of software versions and dependencies, enabling rapid setup and scalability. Our Docker-compose file orchestrates the containers, setting them to launch automatically upon the Raspberry Pi's boot, ensuring our data pipeline is always ready for operation.

As shown in Figure 5.4, each service is deployed in a different Docker container which can be accessed trough the exposed port number.

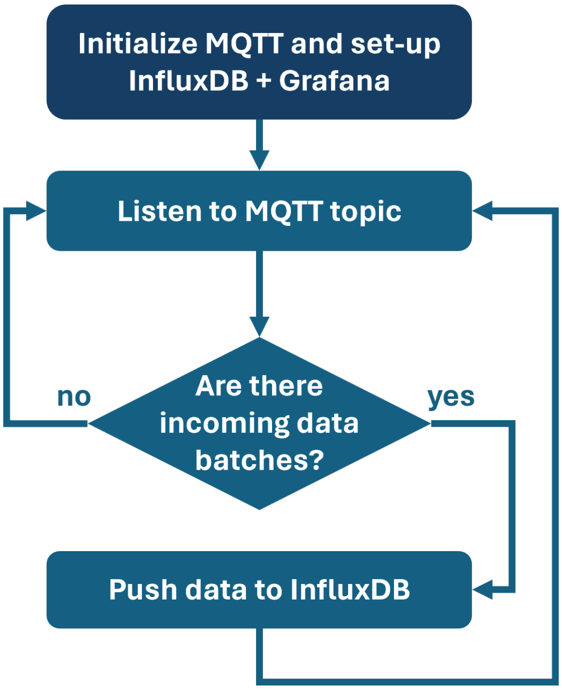
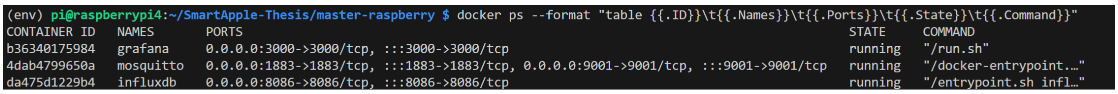


Figure 5.2: The flowchart outlines the data processing operations on the Raspberry Pi 4 IoT Server. Upon server initialization, a Python script is executed to configure MQTT for network communication and to prepare InfluxDB and Grafana for data handling and visualization. The server then continuously monitors the MQTT topic for incoming data from the IoT Gateways, which collect and send information from the sensor-equipped Smart Apples. When data is received, the script pushes it to the InfluxDB database for secure storage. Afterward, it is available for detailed analysis and can be visualized through Grafana. Once the data is stored, the server returns to listening mode to await new data, ensuring a consistent flow of information for ongoing monitoring and analysis.



```
(env) pi@raspberrypi4:~/SmartApple-Thesis/master-raspberry $ docker ps --format "table {{.ID}}\t{{.Names}}\t{{.Ports}}\t{{.State}}\t{{.Command}}"
CONTAINER ID   NAMES       PORTS                                                                     STATE     COMMAND
b36340175984   grafana     0.0.0.0:3000->3000/tcp, :::3000->3000/tcp                                 running   "/run.sh"
4dab4799650a   mosquitto   0.0.0.0:1883->1883/tcp, :::1883->1883/tcp, 0.0.0.0:9001->9001/tcp, :::9001->9001/tcp   running   "/docker-entrypoint.…"
da475d1229b4   influxdb    0.0.0.0:8086->8086/tcp, :::8086->8086/tcp                                 running   "/entrypoint.sh infl…"
```

Figure 5.3: Screenshot of the terminal displaying the active Docker containers in the system, along with their names and network ports for accessing them. It is evident that the *Mosquitto* service is available on multiple ports, allowing for various types of transmission control protocols, such as *websocket*.

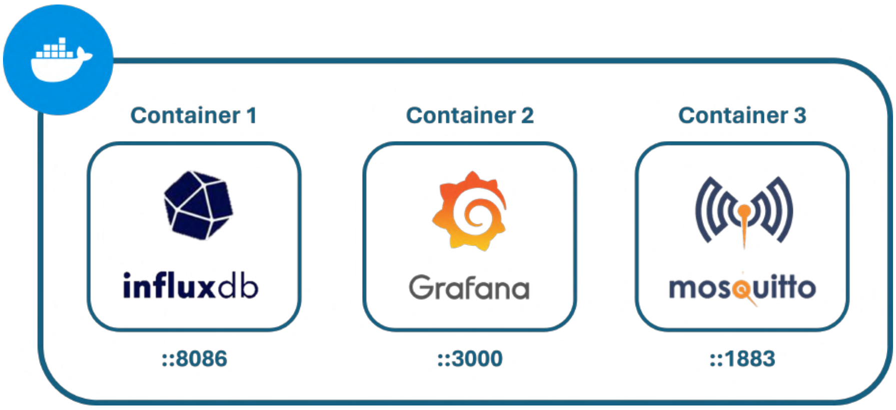


Figure 5.4: To simplify and standardize the deployment process of the IoT Server in the production line, three Docker containers have been created, containing InfluxDB, Grafana, and the Mosquitto service. Upon Raspberry Pi boot, the containers are activated using the *docker-compose up* command, which utilizes official images from Docker Hub to set up the services and make them accessible via network ports as depicted in the image.

In Figure 5.3 the list of all the active containers in the Raspberry Pi is presented showing the containers IDs, entrypoint scripts etc. It is possible to notice that the mosquitto container has multiple network ports exposed due to the possibility to use a websocket communication protocol or more conventional ones. Additionally, the running state of containers such as Grafana and InfluxDB indicates that the system is currently active and ready for tasks like data visualization and database management, which are essential for monitoring and analyzing the data received from IoT devices.

5.2 InfluxDB and Grafana Integration

5.2.1 Database Configuration

The MQTT broker is configured to forward all received data to InfluxDB. This is achieved through a custom Python script that subscribes to the broker's topics, extracting and structuring the data before pushing it to the InfluxDB database, specifically into a "Smart-Apple" bucket designed for this purpose. InfluxQL is used to retrieve the data from the database and it is a SQL-like query language specifically designed for interacting with InfluxDB and querying and analyzing time series data. The following is a query example using InfluxQL which retrieves the last hour of acceleration data from the Nicla device with a certain MAC address and a specific production line id:

```
from(bucket: "Smart-Apple")
  |> range(start: -1h)
  |> filter(fn: (r) => r["_measurement"] == "nicla")
  |> filter(fn: (r) => r["address"] == "EE:DF:46:E7:08:80"
     and r["production_line"] == "plant_1")
  |> filter(fn: (r) => r["_field"] == "a_x" or r["_field"] ==
      "a_y" or r["_field"] == "a_z")
```

In Figure 5.5 the InfluxDB user interface is shown where it is possible to choose among a list of attributes saved in the database. It is quite important to understand if the data is being saved correctly before going to the facility to start the acquisitions. The plots are visually represented using a coarser grid than the actual acquisition sampling rate and that is why it seems there are fewer values than expected.

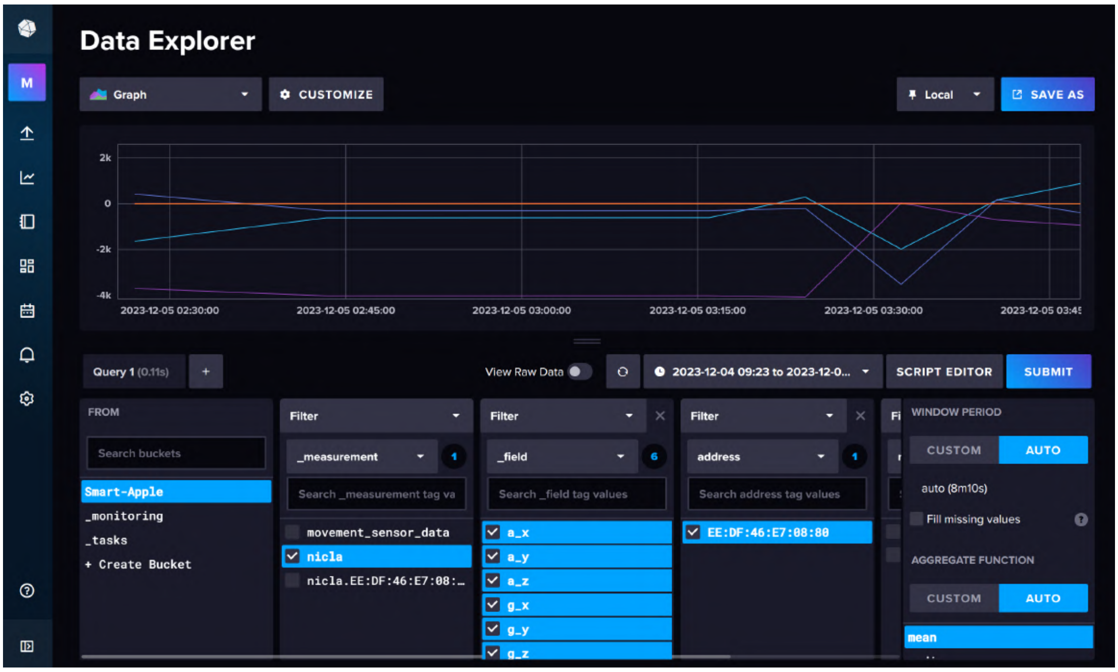


Figure 5.5: The InfluxDB User Interface is accessible via the IoT Server's network port number 8086. This interface is used to ensure that the data from the Smart Apples is correctly saved in the database. A plot with a coarse time base is displayed to allow for faster and more efficient visualisation. Users can also select the type of data to be displayed using checkboxes.

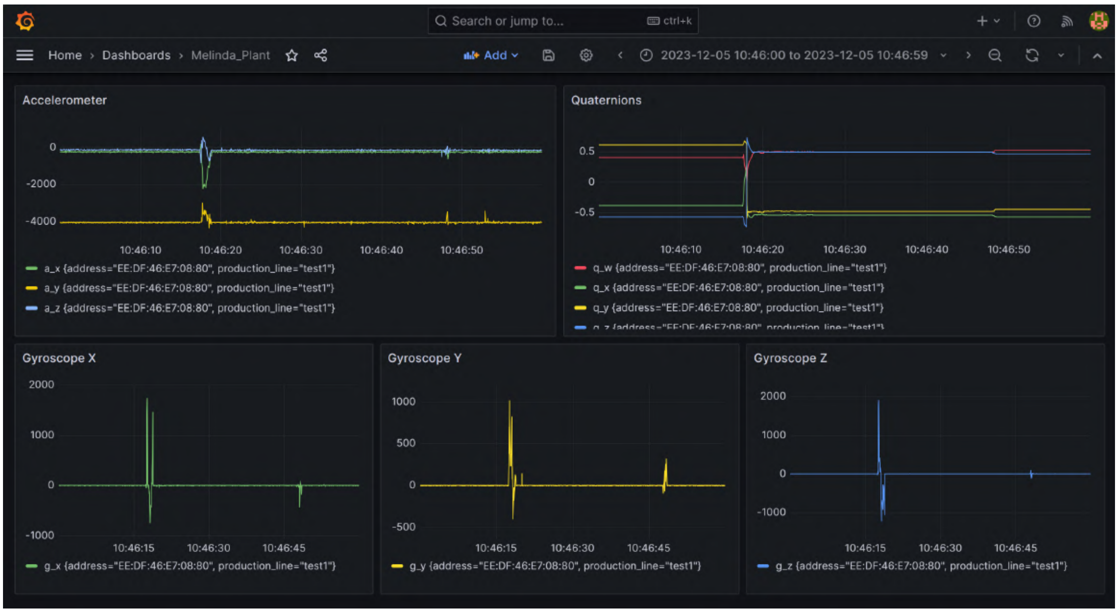


Figure 5.6: The Grafana User Interface is accessible through the IoT Server device's network port number 3000. It communicates with the InfluxDB service to retrieve the most recent batch of Smart Apples data and display it in a user-friendly and interactive dashboard accessible to Melinda operators. The acceleration and angular velocity profiles, as well as the behaviour of quaternions, can all be seen. Just a quick glance reveals a peak in the sensor acquisition.

5.2.2 Grafana Developer Dashboard

Grafana is integrated into the system to provide a powerful and intuitive interface for data visualization. By connecting Grafana to the InfluxDB database, custom dashboards are created to display real-time data and trends from the Smart Apples. This allows for immediate visual feedback to understand whether the system is working properly and quickly assess the accelerations the apples are subjected to.

In Figure 5.6 the Grafana dashboard is shown and refreshed every 5 seconds. Not setting a high refresh rate helps with the computational resources usage while still keeping the dashboard functional and insightful.

Chapter 6

Melinda Production Line

6.1 Facility Overview

6.1.1 Location and Significance

Melinda's plant is situated in Denno within the Val di Non, Trentino, which benefits from a strategic location between the Rocchetta gorge and the Cles plain. This area, nestled on a terrace bordered by the Rio Pleggio valley and the slope descending from Termon, is renowned for its panoramic, sunny positioning conducive to intensive fruit cultivation. Here, apples and pears of the highest quality in Val di Non are produced, with the apples being particularly celebrated for their taste, juiciness, and nutritional value. The climate, terrain exposure, and soil composition significantly enhance the fruit's flavor, marked by its crispness and aroma. The region's exceptional sunlight, exceeding 2500 hours annually, ensures the fruits' perfect ripening, enriching their flavor profile. The facility not only symbolizes the technological advancement in agronomy but also stands as a testament to sustainable and innovative farming practices that are deeply rooted in tradition.

6.1.2 Facility Size and Capacity

The Melinda production line where the Smart Apples were tested is part of the Consorzio Ortofrutticolo Bassa Anaunia - C.O.B.A. (Figure 6.1), which comprises 248 members and spans 565 hectares of cultivated land.

Designed with large-scale apple processing capabilities, the facility boasts the capacity to efficiently manage approximately 400 tons of apples each day. The ample space is well-utilized, housing an extensive selection of specialized machinery and storage solutions that ensure a seamless workflow from the initial intake of apples to their final dispatch. The strategic design and operational throughput

Figure 6.1: Facility of the Consorzio Ortofrutticolo Bassa Anaunia (C.O.B.A.) located in Denno, Val di Non. The location enjoys an ideal climate for the cultivation of various types of apples throughout the year. It comprises 565 hectares of land and houses various industrial processes that involve sorting the apples harvested from the orchard, washing and cleaning them and, after several quality control checks, distributing them into different bins for delivery to customers. The analyses conducted in this research project were carried out within this facility.

enable Melinda to consistently fulfill the demands of both local markets and international customers, further solidifying its reputation as a frontrunner in the fruit processing sector. Through such robust infrastructure, Melinda effectively upholds its commitment to delivering high-quality produce to a global clientele.

6.2 Blueprint Analysis

The blueprint of the Melinda Facility reveals a well-thought-out design that streamlines the flow of apples through the processing stages. Each zone is strategically placed to optimize the transition from one stage to the next, minimizing the time and handling of the fruit, which is crucial in maintaining quality. In Figure 6.2 it is possible to observe the blueprint of the part of the facility where the tests were made. The acquisition of the Smart Apples starts from *Zone 0* to the sorting of the apples in the water canals, after passing through all the other zones with varying speed.

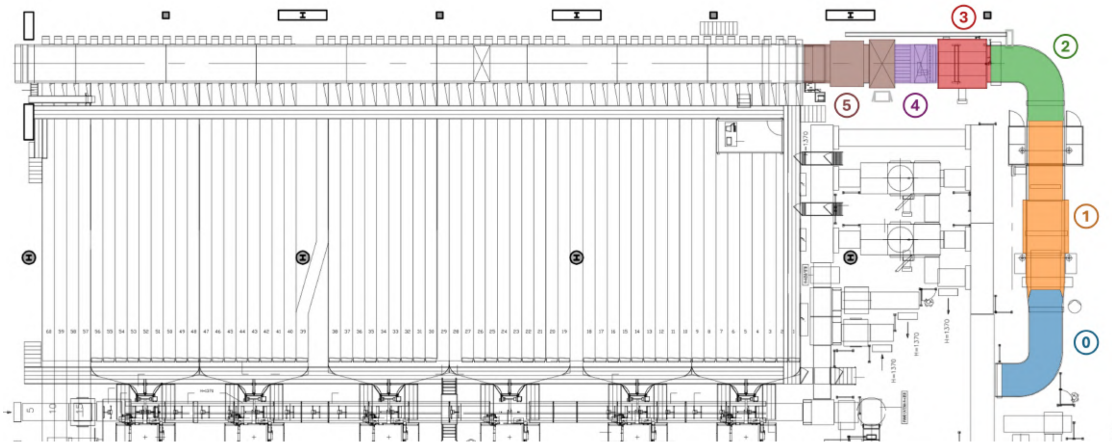


Figure 6.2: Blueprint of the industrial plant section where the Smart Apples were tested and the various analyses requested by Melinda were conducted. The colored zones indicate the areas where apples travel and are processed, highlighting the presence of different machinery worth mentioning. Following zone 5, apples are sorted into water channels based on their external characteristics before being grouped into bins.

6.2.1 Process Zones

Each zone is equipped with specialized machinery tailored for specific tasks. The production line has been split in 5 zones for convenience based on the type of process the apples are involved to:

- **Zone 0**: This is the initial phase where apples transition from the previous production process to the current one under analysis. Operators perform a preliminary visual inspection, discarding apples that do not meet certain standards upon visual examination. This area comprises a small pool where apples float, gradually moving towards the next zone.

- **Zone 1**: In this area, apples undergo a preliminary brushing process to remove any traces of soil and leaves adhering to the surface. They move over rollers that rotate and bounce them, continuously removing external elements from the surface (Figure 6.3). Towards the end of the zone, the rollers allow the apples to float towards the next area.

- **Zone 2**: This area consists of a curved section with water, where apples undergo further washing and float. The movement speed is relatively low, especially when there are many apples, as depicted in Figure 6.4.

Figure 6.3: Zone 1 of the production line at the Melinda plant. Apples are rotated on rollers to remove any dirt and leaves from the surface. Due to being closely packed, they collide with each other, causing bouncing movements.

- **Zone 3**: In this area, apples are sorted using machinery comprising a horizontally rotating cylinder with equidistant pins (Figure 6.5). Apples arrive in bulk and are dispersed to facilitate entry into the next zone.
- **Zone 4**: This is merely a transition zone where apples are sorted into channels and then fed into the next area using a cylindrical brush, as seen in Figure 6.6.
- **Zone 5**: In this part of the facility, there is a sliding platform that transports the apples and rotates them. They pass through an enclosed area where there are light bands and a vision system that takes 70 photos of each apple to categorize them based on size, color, and other parameters (Figure 6.7). Each apple has its place on the platform, as it is necessary for the subsequent part where the apples are sorted based on their characteristics.
- **Sorting pools**: Finally, apples are allowed to slide through water channels and then sorted into different bins, ready for the next phase of the industrial process, which will involve further analysis and possible packaging (Figure 6.8).

Figure 6.4: Zone 2 of the production line at the Melinda plant. Apples move slowly, floating towards the machinery in the next zone. In the photo, the curved section providing guidance to the apples can be observed.

Figure 6.5: Zone 3 of the production line at the Melinda plant. Apples are sorted using a machine consisting of a horizontal cylinder with equidistant pins driven by motors. Here, apples arrive in clusters and are then distributed sparsely to access the next zone.

Figure 6.6: Zone 4 of the production line at the Melinda plant. This is a transition zone where the apples move on linear guides by translating and rotating. They are brushed before entering the next zone which will categorize them.

Figure 6.7: Zone 5 of the production line at the Melinda plant. Apples are moved on a sliding platform and pass through a machine that takes 70 photos of each apple to classify and subsequently sort them into similar groups mainly based on their appearance properties.

Figure 6.8: Last zone of the production line at the Melinda plant where, after being classified in the previous zone, the apples are sorted according to various criteria of shape and surface conformation into water channels. At the end of these channels, there are bins where apples are collected to proceed to the final stages of the production process, which includes further visual quality checks and packaging to send the product to customers.

Chapter 7

Experimental Results

This chapter is crucial for addressing the problem presented by Melinda and giving meaning to the previous chapters. The Smart Apples are IoT nodes designed to record and store data on linear accelerations and angular velocities that apples are subject to throughout the production process within the industrial plant. This data must then be analyzed to understand the impacts received and identify areas within the plant where apples are most stressed, thereby providing Melinda with a detailed report they can use to make any necessary changes to their processes.

The main workflow for data analysis involves initially conducting tests on synthetic data to quantify the stresses due to different types of impacts in a controlled environment. Once significant and consistent results are obtained, the acquisitions and analysis in Melinda's plant are performed.

7.1 Synthetic Data Generation

Synthetic data was generated by embedding a sensor-equipped 3D printed cylinder inside an apple and applying stresses that could validate the approach for later use in the plant. These stresses are divided into behaviors exhibited by the apple, which include:

- **Floating**: moments when the apple is floating undisturbed on its X, Y, Z axes. The apple is released and left floating on the water;
- **Rotation**: moments when the apple is rotating around its axes undisturbed, without impacts. The apple is rotated by hand in different directions;
- **Impact**: moments when the apple is subjected to impacts with other apples;

Figure 7.1: Smart Apple in the setup for conducting synthetic tests. It is important to create a dataset of acceleration and angular velocity sequences that reflect the impacts the apple will receive in production and that could potentially damage it. To achieve this, the apple is struck at various points, simulating impacts both when stationary and when floating. In the image, the 3D-printed cylinder containing the sensors and the bowl filled with water can be seen.

- **Hammer**: moments when the apple is subjected to critical impacts that could damage it. A hammer is used to hit the apple in different points to simulate a critical impact.

This approach will allow for the analysis of the evolution of accelerations over time by classifying them into distinct categories, providing a useful tool not only for identifying behaviors but also for identifying areas within the facility through which the apple is traveling, without resorting to hard-coded or non-repeatable solutions due to the irregularity of the production process.

Tests were conducted by filling a bowl with water and recording data collected by the Smart Apple for each verification class. In Figure 7.1, a setup with a bowl filled with water and a floating Smart Apple is shown. The apple is activated, and sensor data is transmitted via BLE communication to one of the IoT gateways, which saves it in a CSV file. These acquisitions did not utilize the IoT server as it was not necessary.

_time	packet_id	series_id	action	accel_x	...	quat_w
datetime64[ns]	int64	int64	object	float64	float64	float64
y-m-d h:m:s	0	0	floating_x	2.023047	...	0.43

Figure 7.2: Structure of the dataset containing synthetic sequences for detecting apple behavior in the production line. Each row corresponds to a sample of the acquisition, consisting of a timestamp, the ID of the received BLE packet, *series_id* which is a sequential number necessary to distinguish sequences of the same category, *action* representing the category of the sequence, and finally the values of various sensors.

7.1.1 Data Exploration

Now, let's look at the data acquisitions by conducting some data exploration. All tests have been compiled into a single dataframe, the structure of which is visible in Figure 7.2. The first row after the header displays the variable types with which the columns of the pandas dataframe are initialized, and the following row provides an example of the data. The column *series_id* is a sequential number indicating the number of sequences with the same *action*, useful if there are multiple different sequences in the dataset indicating the same action. *packet_id* is the index of samples for each acquisition.

The tests conducted on the apple have varying lengths while maintaining a constant data transfer rate. In Figure 7.3, it is shown that the *floating_x* sequence has the highest number of samples, and *hammer_y* has the fewest.

To understand how the features in the dataset belonging to a single sequence are correlated with each other, the correlation matrix was plotted. In Figure 7.4, the *floating_y* sequence is depicted, and it's observed that the accelerations are strongly correlated with the apple's oscillation around the other axes, represented by the quaternions. This suggests using quaternions to potentially understand the apple's orientation but not for quantifying impacts, as this would complicate the analysis without many benefits, in line with the principle of Occam's razor.

For the analysis and to compare sequences with each other, all acquisitions were resampled at twice the sampling frequency, creating a denser dataset. Since the length of sequences varies regardless of the sampling time, a method to compare them is needed. Ideally, for classifier development using machine learning techniques, a balanced dataset with an adequate number of sequences and similar temporal windows is preferred. However, in our case, this is challenging due to each sequence's unique duration. It's impractical to trim longer sequences to match shorter ones or artificially extend shorter ones through regression, as it would introduce unnecessary noise for any predictive model. The approach taken

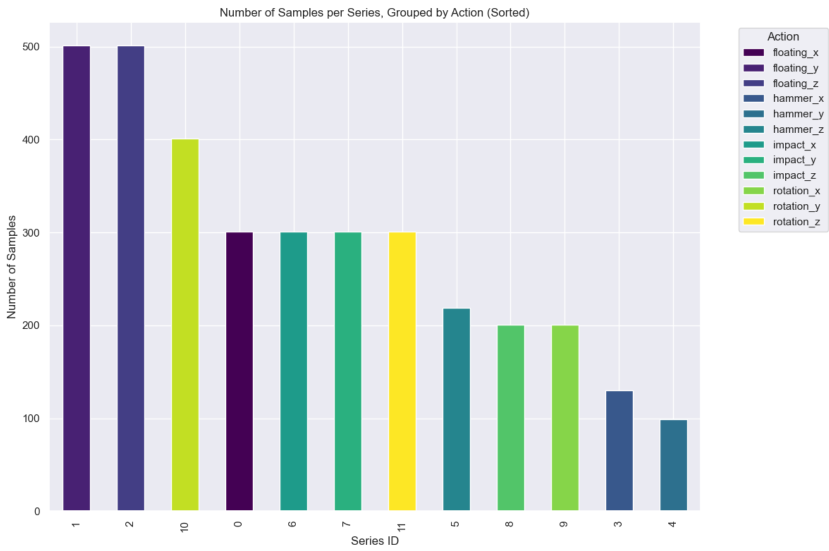


Figure 7.3: The figure illustrates the types of sequences contained in the synthetic dataset, representing the possible stresses that an apple may experience in production. It can be observed that each sequence has a variable number of samples due to the different duration of the sequence; for example, *hammer_y* has the fewest samples as it is a short action representing an impact.

involves analyzing various sequences and comparing them to the main acquisition made in the plant using time-series matching techniques like cross-correlation in the frequency domain or dynamic time warping.

7.1.2 Impact Dataset

To identify significant impacts received by the apple during the production process, the focus is on analyzing the pinpoint stresses classified as *hammer* in the previous dataset. The apple was struck 63 times at different points on its surface with varying intensities to simulate potentially damaging impacts. Figure 7.5 shows peaks representing the impact sequences. Each sequence is 11 samples long, collected at 30Hz, sufficient to represent a pinpoint impact. The composition of accelerations and angular velocity is plotted to better quantify the impact, as recognizing it from the x, y, z components may not be intuitive. One might think setting an acceleration threshold could identify impacts, but sudden direction changes can falsely inflate acceleration readings. For example, shaking an apple in hand generates high accelerations without any surface impact. The analysis thus considers both linear acceleration and angular velocity simultaneously.

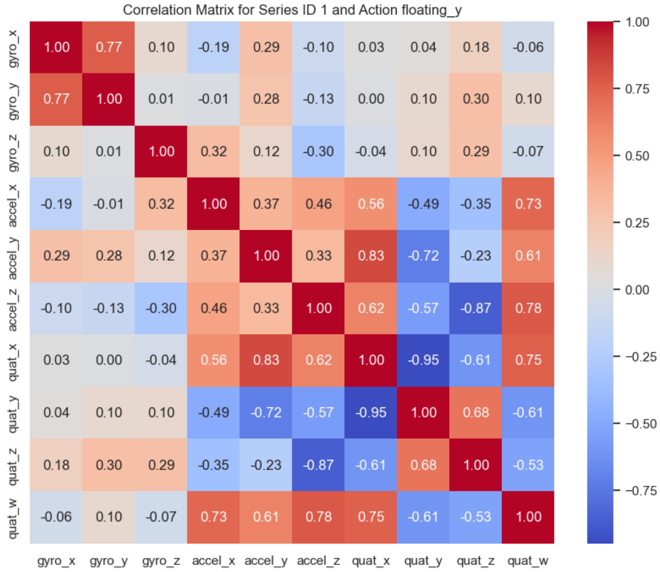


Figure 7.4: Correlation matrix of the *floating_y* sequence, indicating how various features are correlated with each other and identifying any repetitive features that should be removed or not considered. It can be observed that accelerations in the three axes are strongly correlated with quaternions.

7.2 Real Data Analysis

This section covers the analysis of real data collected by Smart Apples in Melinda's plant in Denno. The types of data collected are examined to identify patterns that could be useful for apples' behavioural understanding in the production line, using the generated synthetic sequences. Once the areas with the most significant impacts are identified, suggestions on how to modify the process are provided. Figure 7.6 shows a Smart Apple among other apples in the production process during data acquisition and sensor data streaming.

7.2.1 Data Exploration

The Smart Apple is placed at the beginning of the production line where apples harvested from the field arrive and have already undergone a pre-selection. It moves by floating or rotating on itself, passing from *Zone 0* to *Zone 5*, where it is sorted into water pools based on its shape and surface composition characteristics. The Smart Apple begins transmitting the collected data via Bluetooth and sends it to the nearest IoT Gateway. Figure 7.7 shows the acceleration profile and angular velocity of a Fuji apple across different zones.

It's observed that there are sections where the apple is heavily stressed both

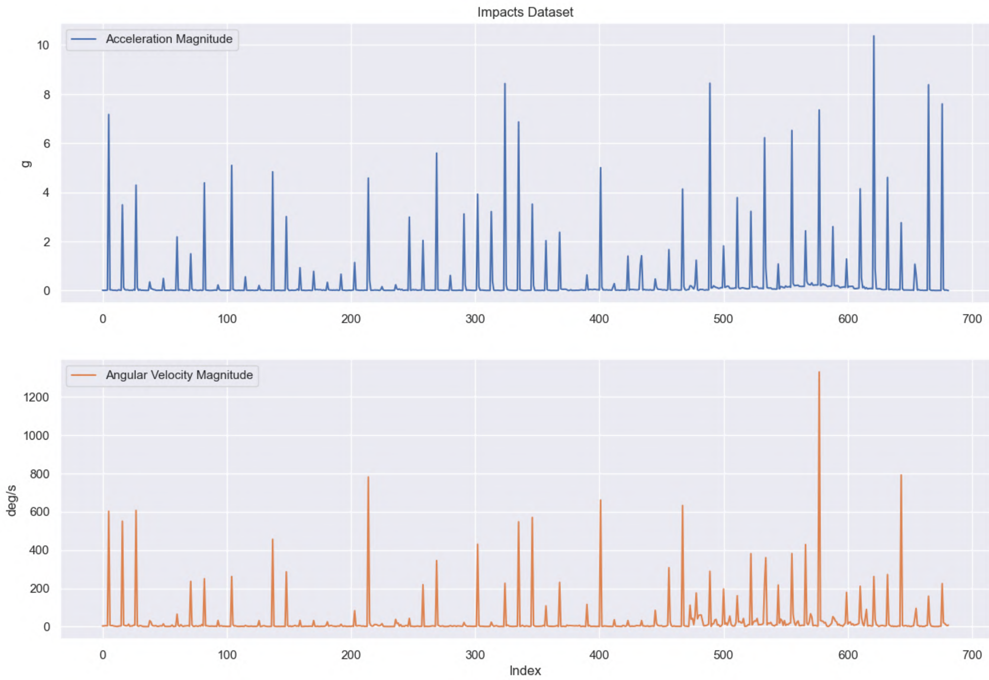


Figure 7.5: The plots display all 63 impact sequences constituting the *impact dataset*, showcasing both the composition of accelerations and angular velocities. It is observed that the dataset encompasses impacts of various natures with different intensities and shapes.

Figure 7.6: Segment of the production line at the Melinda facility featuring a Smart Apple disguised among other apples. It is observed that the Smart Apple behaves like the others, floating with the stem above the water.

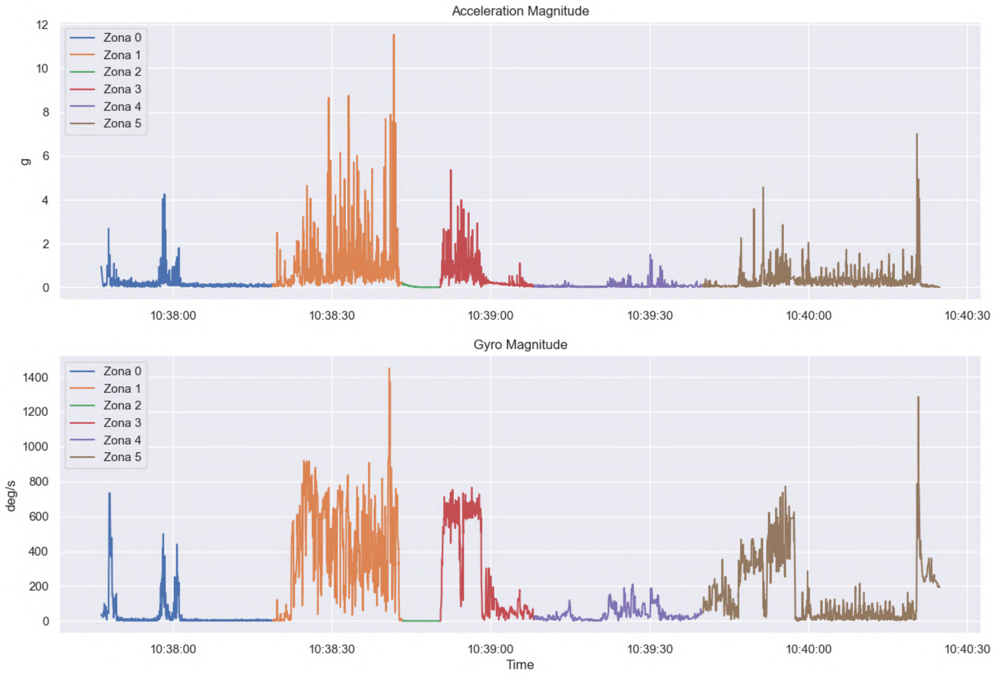


Figure 7.7: The plots depict the acquisition of accelerations and angular velocities from a Smart Apple within the production line at the Melinda facility. Colors denote different zones of the production line, each exerting varying levels of stress on the apple.

by the accelerometer and gyroscope side, and other times when it's more at ease, floating undisturbed or being moved by platforms that do not cause it to rotate.

By analyzing the data transfer rate of the Bluetooth communication, it's found that it does not remain constant across different zones, likely because the apple passes through some machinery which weakens the BLE signal, slowing down the communication. In figure 7.8, the Gaussian distribution for each zone shows that the slowest refresh rate is approximately *40 ms / 25 Hz*, while the fastest is *20 ms / 50 Hz*.

Therefore, all zones were resampled at a frequency double that of the highest received, thus avoiding the loss of information useful for analysis. To determine if there are any patterns in our data in the frequency domain, the spectrogram was plotted in Figure 7.9. It is noticeable that there are zones with a fairly regular distribution of frequencies which spread across the entire spectrum. This could be useful for performing clustering and identifying the zone in which the apple is located. However, since the Smart Apple acquisitions were supervised during the tests in the production line, the zones are identified manually.

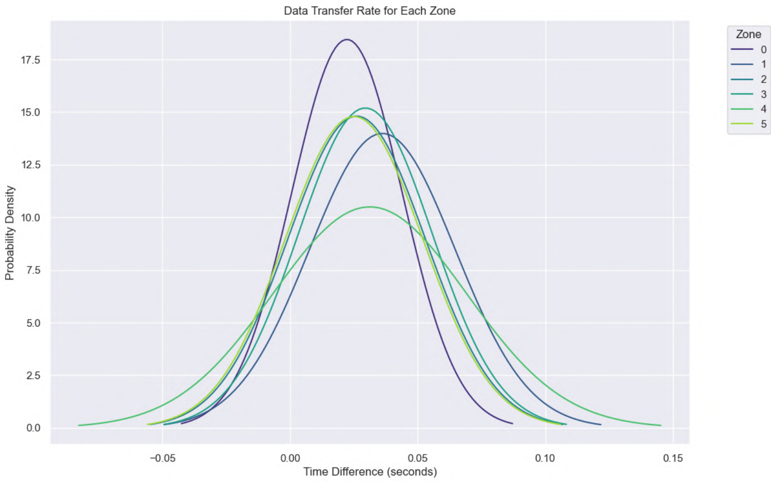


Figure 7.8: The plot represents the Gaussian distribution of the BLE data transfer rate for each zone at the Melinda facility. It can be observed that it is not constant across the three zones, as apples pass through some machinery that weakens the Bluetooth signal, slowing down communication. The slower refresh rate is approximately *25 Hz*, while the faster one is *50 Hz*.

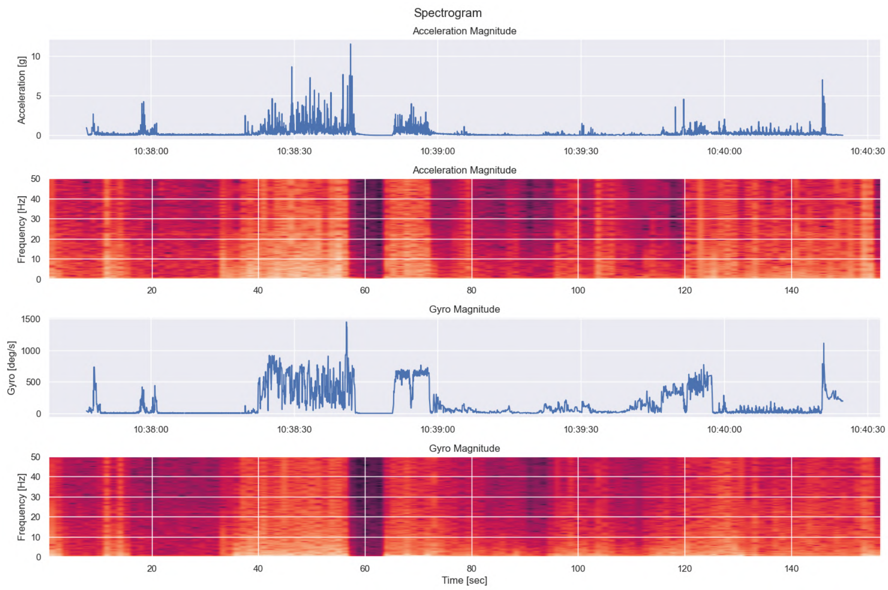


Figure 7.9: In the plots, the acceleration and angular velocity profiles of the Smart Apple in the production line are shown along with their spectrograms, which allow identification of acquisition patterns in the frequency domain. The X-axis represents time, while the Y-axis represents different frequencies.

7.2.2 Impacts Identifications

Statistical Approach

The statistical method aims to identify significant impacts in each zone by defining outliers based on acceleration data. This involves calculating the Z-score, with values exceeding 3 (indicating a deviation more than three standard deviations from the mean) flagged as outliers. Additionally, the Interquartile Range (IQR) is utilized for a dual-method analysis, serving as an ensemble approach to outlier detection. Analysis in Table 7.1 revealed that *Zone 1* exhibits the highest mean acceleration and a substantial number of outliers, suggesting significant activity, while *Zone 3* also presents notable findings worth attention.

Zone	Mean	Median	Std Dev	IQR	Outliers (Z-score/IQR)
0	0.17	0.10	0.29	0.08	28 / 146
1	1.02	0.61	1.29	0.81	17 / 65
2	0.03	0.01	0.04	0.03	4 / 36
3	0.59	0.35	0.63	0.80	13 / 16
4	0.07	0.04	0.10	0.04	19 / 83
5	0.32	0.22	0.39	0.28	30 / 123

Table 7.1: Table presenting a statistical analysis for each zone with detection of outliers using Z-score and IQR methods. *Zone 1*, highlighted in yellow, represents the area with the highest number of outliers and a high mean acceleration, indicating the presence of significant impacts that could damage the apples.

It is possible to plot a boxplot to visually illustrate the distribution of accelerations for each zone, including the various outliers, as depicted in Figure 7.10. Observing the boxplot reveals:

- The line in the middle of the box is the median, which gives you a sense of the central tendency for acceleration magnitudes in each zone.
- The boxes represent the interquartile range (IQR), showing the middle 50% of the data.
- The "whiskers" extend to the furthest points that are not considered outliers, and the dots represent outlier values.
- *Zone 1* appears to have a significantly higher range and variability in acceleration magnitudes compared to other zones, as well as numerous outliers.

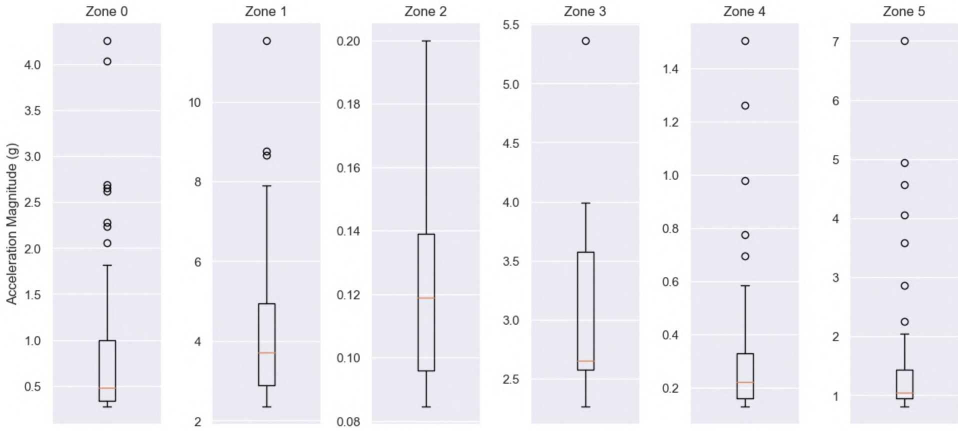


Figure 7.10: The boxplots depict the distribution of data for each zone, where the horizontal orange line represents the median, enclosed within the black-bordered box representing the interquartile range. The dots indicate outliers, and it can be observed that the range of accelerations in zone 1 is very high, with some outliers exceeding 8g. Meanwhile, in zone 5, the distribution is more compressed with the presence of several outliers.

- *Zone 5* has a much tighter distribution, with outliers that appear quite far from the main cluster of data. Probably, it is also due to the longer sequence and to the fact that, at then end of the production line, the Smart Apple is grabbed out from the line generating high accelerations but with no impacts. It is still worthy investigating.

- Zones 0, 2, 3, and 4 show variability with some outliers, but not to the extent of *Zone 1* and *Zone 5*

In Figure 7.11 *Zone 1* and *Zone 5* are zoomed in to understand what it is happening. The previously identified outliers are plotted both for the Z-Score and IQR and we notice how all the peaks are detected. An idea to detect real impacts is by highlighting only the ones in which there are also some outliers in the angular velocity. The outlier at the end of *Zone 1* probably represents an impact received while transitioning to the next zone. At the same way, the first highlighted area in the *Zone 5* corresponds to the transition into the photographic phase of *Zone 5* where apples are categorized. The final highlighted area represent the action of taking the Smart Apple out of the production line generating outliers both in acceleration and angular velocity, but without real impacts.

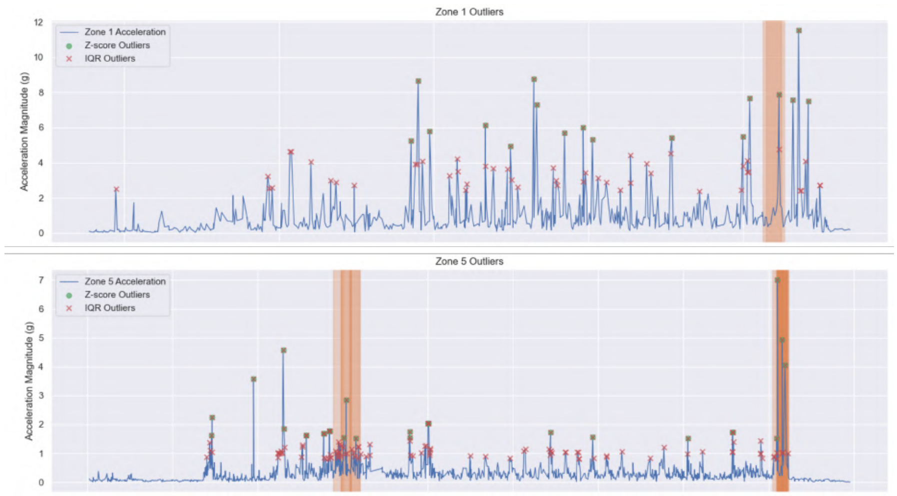


Figure 7.11: The two plots represent the acceleration profiles of *Zone 1* and *Zone 5* acquired by the Smart Apple in the Melinda facility with outliers identified using the Z-score and IQR methods. It can be observed, from the green dots and red crosses, that most of the peaks are identified as outliers. The orange vertical bands indicate points where outliers are present in the angular velocity profiles as well, which are not plotted here.

Multivariate DTW Method

The standard statistical approach conducted in the previous subsection does not take into account the actual correlation between accelerations and angular velocities in a real critical impact. In fact, impacts are treated as samples that deviate by a certain number of standard deviations from the mean. This could lead to false positives and skew the analysis.

An alternative approach involves implementing the Multivariate Dynamic Time Warping (MDTW) algorithm, as outlined in Section 1.4, which aims to identify points in the Smart Apple's acquisition where accelerations and angular velocities resemble a reference impact. The algorithm provides a similarity score that can be used to construct a distribution of significant impacts along the Smart Apple's track.

To accomplish this, we need to utilize the synthetic impact dataset created in Section 7.1.2, which contains 63 samples of critical impacts of varying intensity and shape that could potentially damage the apple. The score of the apple's acquisition has been computed for each of the 63 impact samples, and the 3D plot has been visualized in Figure 7.12. On the X-axis, there is the index representing the temporal instances in the acquisition at the Melinda facility, on the Y-axis are the IDs of the synthetically generated impacts, and on the Z-axis is the score provided by the MDTW algorithm. It can be observed that the plot shows impacts at points similar to the results of the statistical approach, in zones *1* and *5*, but it provides additional insights into which impacts are truly similar to critical impacts, without considering outliers/statistical aspects.

Now, we can compute the average along each acquisition index for each impact and visualize the original acceleration profile at the Melinda facility with a gradient reflecting the similarity of the impacts calculated with MDTW. As shown in Figure 7.13, it can be observed that true impacts are not those with the highest acceleration but the ones with a higher similarity score. The highest impact corresponds to index *2052* with an acceleration of *1.06 g* and an angular velocity of *1148.98 deg/s*.

In Figura 7.14, it is possible to see the number of impacts happening on each zone based on a specified interval. Acceleration samples with a low similarity score are given the color green while high similarity score represents high risk impacts. We conclude that *Zone 1* has the highest risk for the apples to receive damage while in *Zone 3* and *Zone 5* there is low risk but still being cautious.

7.3 Acquisition Dashboard

To ensure that operators within the Melinda facility have a preview of the data collected immediately after acquisition, a dashboard has been created and made available through an IoT Server port. Figure 7.15 displays the dashboard, which was developed using the Python library *Plotly*.

We proceed by loading a CSV file containing the acquisition data of the Smart Apple, which includes accelerations and angular velocities. This file is processed, and four interactive plots are displayed:

1. **Top Left**: 3D plot of an apple approximated to a sphere with the three axes X, Y, Z. It is possible to adjust the resolution of the sphere using the slider below and visualize the direction of impacts received by the apple. The extent of the red part is determined by the impact intensity. The apple can be rotated and zoomed in for further analysis.

2. **Top Right**: plot of accelerations along the three axes during the acquisition on the production line. The slider allows selecting the temporal instance to analyze, subsequently modifying the other plots accordingly.

3. **Bottom Left**: Classification of impacts in different zones of the facility.

4. **Bottom Right**: Estimation of the apple's surface deformation profile from impact, addressed in "Deformation behaviour simulation of an apple under

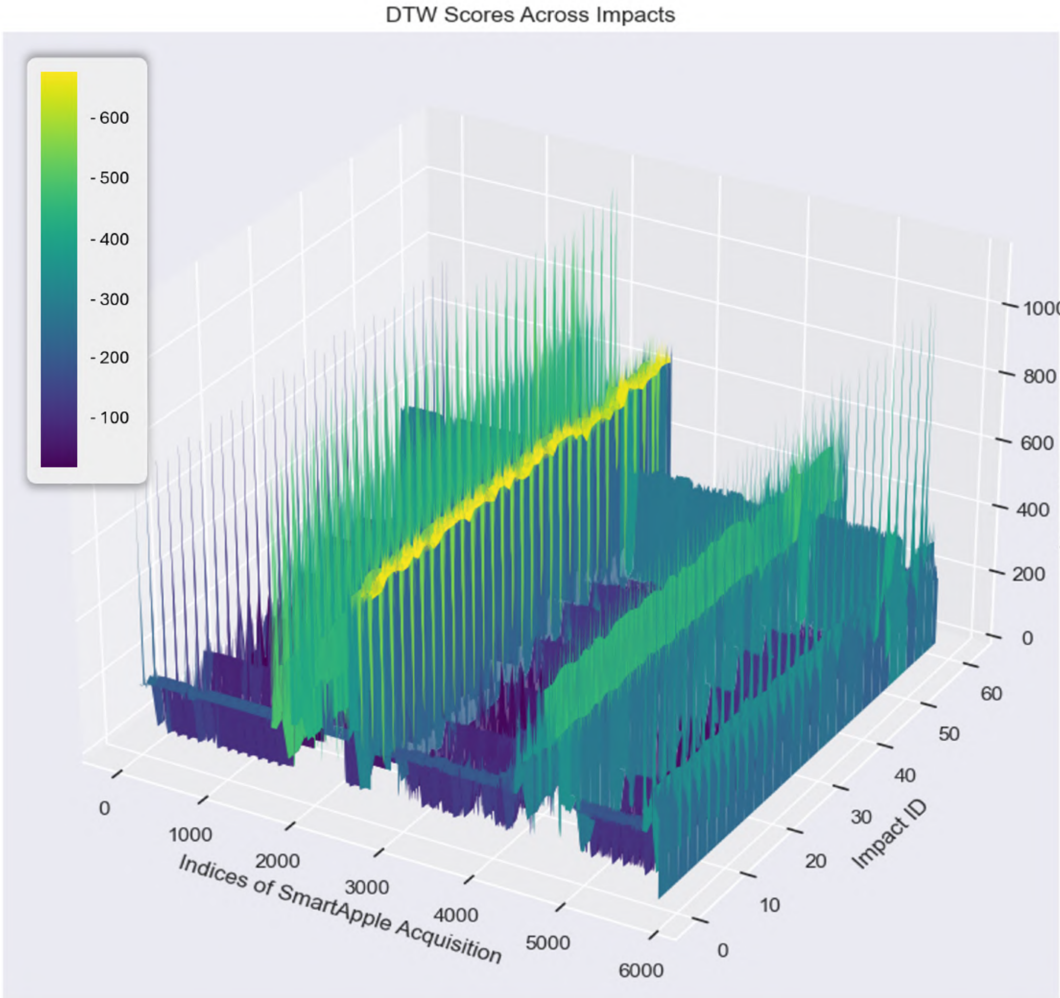


Figure 7.12: A 3D plot representing the distribution of impacts detected using the Multivariate Dynamic Time Warping method. For each type of impact contained in the synthetically created impact dataset (63 impacts), a similarity score is calculated with respect to the acceleration and angular velocity acquisition of the Smart Apple at the Melinda facility. Points represented in yellow indicate a higher probability of a critical impact, while purple indicates the absence of impacts. On the X-axis there are the indices of the acquisition, where zero indicates the start of data transmission and movement of the Smart Apple along the production line. On the Y-axis there are the IDs of each impact sequence present in the synthetic dataset, and on the Z-axis the similarity score is shown. Noticeably, around 2000 samples a yellow strip is present indicating a high probability of critical impact spanning across the entire impact dataset.

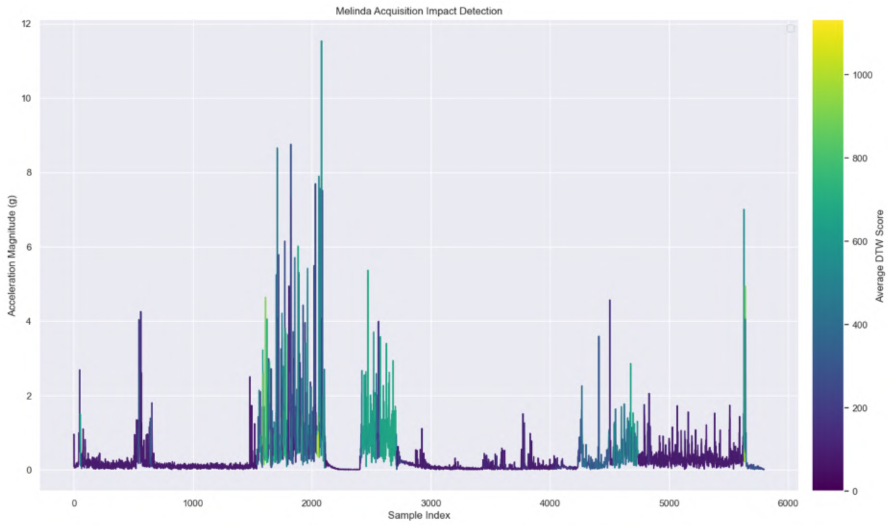


Figure 7.13: Plot showing the probability distribution of a critical impact occurring in the acceleration acquisition of the Smart Apple. This is calculated by averaging the similarity scores for each synthetic sequence present in the impact dataset obtained through the DTW algorithm. It can be observed that *Zone 1* is the most stressed and presents some critical impacts.

Figure 7.14: Histogram grouped by production zones at Melinda. It provides a visual representation of impact distribution across each zone. The Y-axis indicates the number of samples where impacts were detected by the algorithm, with colors indicating different ranges of similarity scores.

Figure 7.15: Interactive dashboard providing an overview of accelerations and other relevant information post-acquisition. It features a yellow sphere approximating the geometry of an apple, with impact intensity represented on its surface at a specific moment in time in the correct direction. Additionally, the dashboard displays the acceleration profile in its three components and a plot indicating the quantity of impacts in each production zone. In the bottom right one the surface deformation profile of the apple can be observed.

drop case by finite element method" [5]. This paper mathematically models a Golden apple and uses finite element method (FEM) simulations to assess the effects of dropping the apple onto a rigid surface from various heights. The study provides a stress-strain curve, which can be utilized to convert measured linear accelerations (in units of g) into impact force using the formula:

$$F = m_{[\mathrm{kg}]} \cdot \mathrm{accel}_{[\mathrm{g}]} \cdot 9.806 \left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$$

where *m[kg]* is the mass of the apple while *accel[g]* is the magnitude of the acceleration vector. Thus, we can calculate the distribution of deformations along the trajectory during the industrial processes in the plant.

7.4 Final Considerations

There are several factors to consider in providing an accurate analysis of the issue requested by Melinda. The area of the facility where apples are most stressed is *Zone 1*. In this part of the production line, apples move on rollers with some bouncing and collision among them, which could damage them depending on the operating speed of the production line. One solution could be to reduce the speed at which apples travel by reducing the angular velocity of the rollers or by modifying their size to achieve a smoother transition. Another possible measure could be to arrange the apples more sparsely to cause fewer collisions between them and therefore fewer uncontrolled bounces. The latter can be done by adding a new machine which divides the apples before entering the zone.

In *Zone 3* instead, apples are sorted into channels using a rotating cylinder with staggered pins. Impacts could be due to the fast rotation of the cylinder or more likely to the impact with the pins when apples are accumulated together. Again, a solution could be generating distance among apples which can be done by directly applying *Zone 1* solution since the production line presents sequential processes.

Conclusions

In the agri-food sector, it is crucial to ensure product quality from field harvesting to packaging and distribution to customers. Throughout this process, maintaining a high level of quality is essential. While many industries have been implementing IoT systems for several years to ensure continuous monitoring of the supply chain, the agri-food sector still lags behind in this regard. However, in recent times, many initiatives have emerged with the aim of adopting these changes. Among them is Melinda, a consortium located in Val di Non, Trentino, specializing in apple production. Melinda has sought collaboration with the University of Trento to enhance the quality control process of their apples within their facilities.

Melinda noticed that certain apples were developing bruises days after being packaged for shipping to customers. They hypothesized they might be occurring due to significant impacts during the production process and aimed to identify the specific points along the production line where the apples were experiencing the most stress.

To address this issue and better understand the position of important impacts, an IoT infrastructure was created that allows apples to be monitored without disrupting their natural flow or the production line. The "Smart Apple" is the heart of the system and was designed by replacing the core of a real apple with a 3D-printed cylinder holding a microcontroller and onboard sensors like an accelerometer and gyroscope. This Smart Apple is placed among conventional apples and subjected to all standard procedures. Data collected by the Smart Apple is relayed via BLE communication to IoT gateways made up of Raspberry Pi 4 devices distributed along the production line. These gateways collect and organise data before sending it in batches to a server device, in our case another Raspberry Pi 4, using MQTT communication. The server-side device serves as a broker, receiving data written to MQTT topics and then pushing it to an InfluxDB database specifically designed for time-series storage. This data can be visualized using Grafana or via a custom interactive dashboard, enabling detailed examination of accelerations, impact direction and intensity on the apple's surface, affected zones, and an estimation of surface deformation at the stress point.

The analysis entails creating a synthetic dataset of accelerations due to impacts on the surface of an apple in a controlled environment. This dataset is used to identify points in the Smart Apple's acquisition with a high likelihood of being critical impacts, employing various algorithms including Dynamic Time Warping in multiple variables. In addition to this method, impacts are classified as outliers using classical statistical analysis techniques such as Z-Score and Interquartile Range.

From the conducted analyses, it was observed that there is a particular zone in the production line where apples experience significant accelerations and angular velocities. This stress, occurring as apples move and rotate on rollers, can potentially cause internal damage, increasing the risk of future bruising. The high volume of apples causes them to compact and collide, resulting in additional stress compared to the rest of the production process. To mitigate this, the roller dimensions could be adjusted for a smoother transition and spacing between apples to prevent collisions.

This innovative solution presents potential for diverse applications beyond industrial settings. It could be deployed in orchards, during apple transportation between facilities, and throughout the supply chain from facility to customer. Moreover, its adaptability suggests feasibility for integration with other fruit types, such as oranges. From a business perspective, there's an opportunity for further refinement and commercialization, making it accessible to a broader spectrum of companies with tailored data analysis packages. It's crucial to acknowledge that while this system represents significant progress, there are ongoing areas for enhancement. Future endeavors may involve scaling its application to larger facilities and different fruit varieties, as well as broadening monitoring capabilities to encompass various stages of the supply chain beyond production processes.

Bibliography

[1] Zhao De-An et al. "Design and control of an apple harvesting robot". In: *Biosystems Engineering* 110.2 (2011), pp. 112–122. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2011.07.005. URL: https://www.sciencedirect.com/science/article/pii/S1537511011001206.

[2] Arduino. *Nicla Sense ME User Manual*. 2021. URL: https://docs.arduino.cc/tutorials/nicla-sense-me/user-manual/ (visited on 02/24/2024).

[3] Chinaza Godswill Awuchi. "HACCP, quality, and food safety management in food and agricultural systems". In: *Cogent Food & Agriculture* 9.1 (2023), p. 2176280. DOI: 10.1080/23311932.2023.2176280. eprint: https://doi.org/10.1080/23311932.2023.2176280. URL: https://doi.org/10.1080/23311932.2023.2176280.

[4] D. Bini, D. Pamela, and Shajin Prince. "Machine Vision and Machine Learning for Intelligent Agrobots: A review". In: *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*. 2020, pp. 12–16. DOI: 10.1109/ICDCS48716.2020.243538.

[5] H. Kursat Celik, Allan E.W. Rennie, and Ibrahim Akinci. "Deformation behaviour simulation of an apple under drop case by finite element method". In: *Journal of Food Engineering* 104.2 (2011), pp. 293–298. ISSN: 0260-8774. DOI: https://doi.org/10.1016/j.jfoodeng.2010.12.020. URL: https://www.sciencedirect.com/science/article/pii/S0260877410006126.

[6] Marcos David Ferreira et al. "EVALUATION OF POTE NTIA L MECHANICAL DAMAGE IN APPLE PACKING LINES IN THE MAIN PRODUCING REGIONS OF BRAZIL". In: URL: https://api.semanticscholar.org/CorpusID:252047581.

[7] Wei Ji et al. "Contact force modeling and variable damping impedance control of apple harvesting robot". In: *Computers and Electronics in Agriculture* 198 (2022), p. 107026. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.107026. URL: https://www.sciencedirect.com/science/article/pii/S016816992200343X.

[8] Kai Junge and Josie Hughes. "Soft Sensorized Physical Twin for Harvesting Raspberries". In: *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. 2022, pp. 601–606. DOI: 10.1109/RoboSoft54090.2022.9762135.

[9] Önder Kabaş and Valentin Vladut. "Determination of Drop Test Behavior of a Sample Peach Using Finite Element Method". In: *International Journal of Food Properties* 18 (Apr. 2015). DOI: 10.1080/10942912.2014.994069.

[10] Hanwen Kang et al. "Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting". In: *Sensors* 20.19 (2020). ISSN: 1424-8220. DOI: 10.3390/s20195670. URL: https://www.mdpi.com/1424-8220/20/19/5670.

[11] Zhen Li. "Study on the Sealing Performance of O-ring under High-Pressure Environment". In: *Journal of Physics: Conference Series* 2419 (Jan. 2023), p. 012005. DOI: 10.1088/1742-6596/2419/1/012005.

[12] Wei Luo, Hailiang Zhang, and Xuemei Liu. "Hyperspectral/Multispectral Reflectance Imaging Combining with Watershed Segmentation Algorithm for Detection of Early Bruises on Apples with Different Peel Colors". In: *Food Analytical Methods* 12 (May 2019). DOI: 10.1007/s12161-019-01456-0.

[13] Naveen Kumar Mahanti et al. "Emerging non-destructive imaging techniques for fruit damage detection: Image processing and analysis". In: *Trends in Food Science & Technology* 120 (2022), pp. 418–438. ISSN: 0924-2244. DOI: https://doi.org/10.1016/j.tifs.2021.12.021. URL: https://www.sciencedirect.com/science/article/pii/S0924224421006786.

[14] Charles Okpala and Malgorzata Korzeniowska. "Understanding the Relevance of Quality Management in Agro-food Product Industry: From Ethical Considerations to Assuring Food Hygiene Quality Safety Standards and Its Associated Processes". In: *Food Reviews International* 39 (Sept. 2021), pp. 1–74. DOI: 10.1080/87559129.2021.1938600.

[15] Slavko Pokorni. "Reliability and availability of the Internet of things". In: *Military Technical Courier* 67 (June 2019), p. 588. DOI: 10.5937/vojtehg67-21363.

[16] Roman Stopa et al. "Determining the resistance to mechanical damage of apples under impact loads". In: *Postharvest Biology and Technology* 146 (2018), pp. 79–89. ISSN: 0925-5214. DOI: https://doi.org/10.1016/j.postharvbio.2018.08.016. URL: https://www.sciencedirect.com/science/article/pii/S0925521418305726.

[17] Emrah Tomur et al. "SoK: Investigation of Security and Functional Safety in Industrial IoT". In: *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. 2021, pp. 226–233. DOI: 10.1109/CSR51186.2021.9527921.

[18] Nathalie Verbrugghe, Eleonora Rubinacci, and Ahmed Z. Khan. "Biomimicry in Architecture: A Review of Definitions, Case Studies, and Design Methods". In: *Biomimetics* 8.1 (2023). ISSN: 2313-7673. DOI: 10.3390/biomimetics8010107. URL: https://www.mdpi.com/2313-7673/8/1/107.

[19] Kubilay Vursavuş and Faruk Ozguven. "Determining the Effects of Vibration Parameters and Packaging Method on Mechanical Damage in Golden Delicious Apples". In: *Turkish Journal of Agriculture and Forestry* 28 (Jan. 2004), pp. 311–320.

[20] Jie Xia et al. "Flexible sensing technology for fruit quality control in the cold chain: Characterization, application, and improvement". In: *Food Control* 154 (2023), p. 109976. ISSN: 0956-7135. DOI: https://doi.org/10.1016/j.foodcont.2023.109976. URL: https://www.sciencedirect.com/science/article/pii/S0956713523003766.

[21] Zou Xiao-bo et al. "In-line detection of apple defects using three color cameras system". In: *Computers and Electronics in Agriculture* 70.1 (2010), pp. 129–134. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2009.09.014. URL: https://www.sciencedirect.com/science/article/pii/S0168169909002051.

List of Figures

List of Tables