

## Repositório ISCTE-IUL

---

Deposited in *Repositório ISCTE-IUL*:

2023-02-06

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Filipe, P., Mataloto, B. & Coutinho, C. (2022). IoT system for the validation of conditions in shipping couriers. In 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI). (pp. 103-108). Shanghai: IEEE.

Further information on publisher's website:

[10.1109/ISSI55442.2022.9963374](https://doi.org/10.1109/ISSI55442.2022.9963374)

Publisher's copyright statement:

This is the peer reviewed version of the following article: Filipe, P., Mataloto, B. & Coutinho, C. (2022). IoT system for the validation of conditions in shipping couriers. In 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI). (pp. 103-108). Shanghai: IEEE., which has been published in final form at <https://dx.doi.org/10.1109/ISSI55442.2022.9963374>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

---

### Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

---

# IoT system for the validation of conditions in shipping couriers

Pedro Filipe  
ISTA  
Instituto Universitário de Lisboa  
(ISCTE-IUL)  
Lisbon, Portugal  
pmmfe@iscte-iul.pt

Bruno Mataloto  
ISTA  
Instituto Universitário de Lisboa  
(ISCTE-IUL), ISTAR  
Lisbon, Portugal  
bruno\_mataloto@iscte-iul.pt

Carlos Coutinho  
ISTA  
Instituto Universitário de Lisboa  
(ISCTE-IUL), ISTAR  
Lisbon, Portugal  
carlos.eduardo.coutinho@iscte-iul.pt

**Abstract**—Online activity has increased over the years and has become a staple in modern society. With it, online shopping businesses have become more common, as well as the number of Internet of Things (IoT) devices. As the shopping experience becomes less personal, some people try to exploit the system causing online businesses to increasingly suffer from return frauds. This paper proposes a solution to this problem, integrating IoT and image processing. The proposed system uses IoT to verify package conditions in courier services, and a database to monitor shipments. The methodology followed for development is described, including the research models, research questions and hypothesis. Development was based on a literature review conducted in order to understand what has been done in the area of logistics services, and what equipment and tools were used for these projects. In addition, the system’s design and architecture are described, giving insight to each components’ function in the system. The system is then implemented to simulate a real-world scenario and tests are carried out. The results of the tests are gathered and analyzed to fine-tune the system. Finally, conclusions are drawn out on the system’s capability of achieving the goals and requirements established before development.

**Index Terms**—IoT, Logistics Services, Package Monitoring, Package Integrity, Industry 4.0

## I. INTRODUCTION

In today’s era, online engagement has grown significantly [1] and has become a staple in modern society. E-commerce has also grown gradually over the years [2], making online shopping increasingly common [3]. Online shopping businesses rely on delivery services to distribute goods to their customers. A good shipping service has a positive effect on consumers, as evidenced by research [4], which found that shipping has a positive influence in the customer satisfaction of online shopping. Within the delivery process, package quality and condition is closely related to customer satisfaction, as confirmed by Vasić et al. [5]. Furthermore, study [6] of users’ review comments revealed that inadequate handling of goods is the most frequent topic.

Online engagement has driven the number of Internet of Things (IoT) devices up as well, a growth of 150% from 2017 to 2020 [7]. IoT has various applications in everyday life and a lot of potential to create solutions to problems never fixed before, like poor handling of delivered goods and improving quality of life.

## A. Motivation

When a product delivered is reported damaged by a customer and the customer wishes to return it, it triggers a multistep process of handling that product, also known as Reverse Logistics (RL) [8] in the Supply Chain Management (SCM). Damage caused to a product can occur anywhere along the line of Forwards Logistics (FL), however for courier services, their responsibility for the package lies within the distribution stage. For a delivery service to be liable for package damage, it means that the package was damaged during delivery.

Considering that the package wasn’t damaged before distribution, it leaves only two possibilities: The delivery driver damaged it, or the customer did. This becomes a case of good faith, and without any compromisable proof to settle the dispute, return policies cannot be enforced, leaving the business with a tough decision: deny the return and leave the customer dissatisfied, or allow it and lose out on revenue. This is a global issue, research studies demonstrate that in the clothing industry 50% of returns are fraudulent, and 8% of all returns in North America are fraudulent, rendering return policies useless [9].

A system based on IoT could be crucial to support delivery businesses with fraud returns, or at very least validate the transportation conditions under which products are delivered. This system could prove beneficial for both consumers and businesses.

From a business standpoint, courier services are the ones that stand to benefit the most from the system previously described. The system can offer non-repudiation as it validates the state of the product before it is delivered to the customer. This means that if the customer attempts to return a product due to mishandling, the system can provide proof to enforce return policies. With this improvement, courier businesses may reduce revenue lost on returns while maintaining customer satisfaction. Aside from that, data collected by the system can serve as a monitoring tool for drivers.

The system’s non-repudiation also extends to customers. For example, if a product is delivered damaged, the system will recognize it as such, meaning that the customer would be

in their right to return the item. The monitoring of drivers also benefits the client indirectly since it assures proper delivery by serving as a deterrent to improper driving.

### B. Context

Delivery companies' success relies on their customers. Developing ideas that improve delivery, correlates directly with customer satisfaction, which correlates with the success of the company.

Study [10] employed REDTags, an IoT-enabled device network that allows delivery service personnel to verify the status of a product at each step of the delivery process. A framework was built to track and monitor the items, with back-end features for smart data transmission, administration, storage, and analytics. Machine learning was used to predict potential damage of items packaged, yielding good results.

This study highlights the importance of IoT for the development of new solutions in the logistics services industry. These two topics, IoT and delivery services, are the ones covered in this paper.

### C. Objectives

The main goal of this study is to implement a system that can validate the conditions of packages into courier services, using IoT. A complementary goal for this study is to use the data acquired by the IoT system as a monitoring tool for driving during transportation.

### D. Research Questions

It's critical to detect whether a package has been mishandled before delivering it to a customer to ensure that all parties are treated fairly. During the development of an IoT system that validates conditions in shipping couriers, the following questions are to be answered:

- How can the system verify the quality of the package before delivery?
- How effective is the system at detecting package damage?
- What further applications are there for the data acquired by the system?

### E. Hypothesis

Hypothesis is defined to predict the answers to the research questions written down previously. Regarding the first research question, to achieve verification of package conditions before delivering to clients, the system configuration should include a camera so it can take pictures before delivery. The camera is connected to a microcontroller that can activate/deactivate it to save battery life. This microcontroller is paired to a network module to transfer the information collected to a database. The database is necessary to store the data and monitor the deliveries. Besides that, the microcontroller should include sensors for tracking.

For the second question, the system aims to correctly identify and verify the conditions of all packages transported, which translates to an efficiency of 100%.

Finally, concerning the third question, the data acquired is going to be utilized as an evaluation tool for the transport trucks drivers' ability. If the system detects too many occurrences of damage, it could indicate poor driving by the person behind the wheel.

### F. Research Model

The research model "Designated Science Research Methodology (DSRM)" by Peffers at al. [11] was followed for the development of the suggested work. This model is comprised of six stages as is shown in Fig. 1.

The first stage of the process was already complete in the *Motivation* section, where problems were identified, and motivation was given. The formulation of objectives for a solution, which makes up the second stage, is declared in sections *Research questions*, *Hypothesis* and *Objectives*.

The next two stages consist of designing, developing and testing the artifact. This is achieved in chapters III and IV. Following that, the artifact's efficiency is evaluated. In order to do this, results are analyzed considering the objectives established. The analysis is carried out in chapters IV and V.

The final stage consists of communicating the problem and its importance, as well as the artifact via academic publications. Stages two and three can be iterated through to implement a new solution in case of sub-par results obtained in stage five, or new additions for publications in stage six.

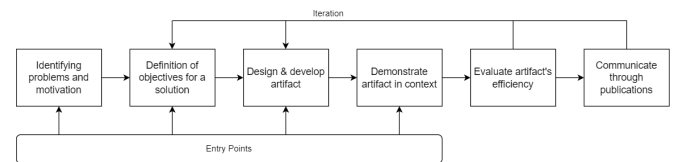


Fig. 1: DSRM Process [11]

## II. EXISTING THEORIES & PREVIOUS WORK

This chapter describes the three most relevant studies linked to the paper's theme, which is the usage of IoT for the improvement of cargo conditions in logistics services. The studies were encountered during the systematic literature review. Highlighting these studies shows that there is few attempts at verifying package integrity.

### A. REDTag: A Predictive Maintenance Framework for Parcel Delivery Services

REDTag Service [10] presents an IoT-based, big data analytics prediction framework adapted to logistics services, to track the condition of transported goods and to early predict potential damages. It uses REDTags, which are smart ad-hoc hardware tags, attached to each package.

The tag is an IoT device containing sensors, batteries, memories, a processor, and a networking module to communicate with external mobile devices, using Near Field Communication (NFC). The framework consists of front-end and back-end services to store, manage, and analyze REDTag

data. It relies on Big Data architecture to store data, as well as a machine learning element to predict potential breaks.

The data and predicted results are shown on dynamic and customizable dashboards, allowing monitorization of packages and quick intervention. Clients can also be alerted of where, when, and who caused damage to packages, motivating drivers to be careful.

### *B. Real-Time Monitoring via Patch-Type Piezoelectric Force Sensors for Internet of Things Based Logistics*

Chuang et al. [12] suggest the usage of patch-type piezoelectric force sensors that can determine if a parcel has been opened and/or damaged. These sensors release electric charges when mechanical tension is applied. If the parcel is opened, tensile stress is produced, and compressive stress if the parcel collides.

An application developed, iTape, analyzes and monitors the data so that it can be uploaded to a cloud database via 3G/4G communication. The database keeps note of the occurrence's time, type and location of the occurrence, so that it may be monitored in real time.

## III. SYSTEM DESIGN AND ARCHITECTURE

The solution proposed in this paper aims to assist logistics companies in preventing return fraud. This chapter explains the IoT system's design and architecture, which is based on the objectives proposed in I-C. The chapter is broken down into two sections: System Requirements, System Design and Architecture

### *A. System Requirements*

The requirements to achieve the first objective are:

- Guarantee that all packages are identified and information is stored for future reference;
- Information retrieval about a parcel must be simple and straightforward;
- It must be possible to check whether a delivery was damaged during the delivery process.

For the second objective, the following requirements were devised:

- Package data must be paired with the driver responsible for delivering it;
- Delivery information must be able to be categorized into damaged or not damaged.

1) *Santos e Vale - Logistics, Distribution and Transportation*: Santos e Vale [13] brought up the issue of customers claiming that packages were damaged during shipping when they weren't. The project developed in this paper was based on solving this problem with the intention of implementing it in the corporation's vehicles. Given the enterprises' involvement in the development of the system, and the goal to integrate it in Santos e Vale's vehicles, it was required that the system's operation must be autonomous, meaning it cannot interfere with employees' activities, and that package identification is accomplished through QR codes.

### *B. System Design and Architecture*

With the requirements established, the design and architecture of the IoT system can now be defined. The architecture of the system is split into four layers, as shown in Fig. 2: Hardware, Network, Data and Application Layer.

- **Hardware Layer** - Two sets of components are used as hardware: an ESP32 with a motion sensor and a buzzer, and a Raspberry Pi with a camera module. The motion sensor can detect when the truck cargo door is opened, which means that cargo is going to be loaded/unloaded. Knowing this, the Raspberry Pi activates the camera so it can detect QR codes and record the cargo. When a QR code is identified, the buzzer sounds and a video is recorded, allowing the driver to display the package as a whole. The end of the recording is signaled by the buzzer and the driver may proceed. The Raspberry Pi and ESP32 were picked based on three factors: Space occupation, configuration simpleness and connectivity. The Raspberry is a pocket sized computer with WiFi and Bluetooth that is simple to set up. Being a tiny computer it has access to most tools computers have, such as Node-RED that enables access and data storage in a SQLite database easily. Its modularity design enables the use of a camera, which is needed to identify QR codes and capture video. These two operations are carried out via Python scripts and the use of the OpenCV library. The ESP32 is a small, powerful microcontroller with WiFi and Bluetooth modules appropriate for low power applications. The motion sensor choice was between the HC-SR501 and SB00422A-1, where the HC-SR501 was picked due to its better performance evidenced on study [14]. The buzzer was a late addition due to necessity shown by the results in IV-B.
- **Network Layer** - The Raspberry Pi hosts the main constituent of the network layer, the Node-RED server. Node-RED was chosen because of the simplicity of its flow based programming and compatibility with other hardware and software. The server acts as a mediator for the exchange of information between the ESP32 and the Raspberry Pi. It executes the Python scripts necessary for the system to function, and retrieves and stores information in a database. This information is presented through dashboards with nodes available in Node-RED, although HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) is also used for a more complex display. The flow of information is guaranteed by the Internet Protocol (IP), and in the case of the ESP32, the Message Queuing Telemetry Transport (MQTT) protocol as well. MQTT is a standard messaging protocol for IoT projects, popular due to its small, minimal data packets and ease of implementation. Its ideal for connecting remote devices with low bandwidth and power, high latency and connection costs [15]. It uses a publish/subscribe architecture where clients subscribe to topics, and when messages are published

- to said topic, they are relayed to all clients subscribed.
- **Data Layer** - Two elements incorporate the data layer, an SQLite database, and two Python scripts that employ the OpenCV library. The scripts are programmed to execute without human input and instead, when the necessary conditions are met. The first script is responsible for identifying QR codes, and is executed when the cargo door is open. When a QR code is successfully validated, the second script records a ten second video for the driver to display the package. This video is then stored in the Raspberry. Afterward, the first script is executed again for a new cycle, and is only interrupted if the cargo door is closed. All the information gathered by the QR codes is stored in a SQLite database, in a table with five columns: "Unique ID", "Package ID", "Date", "Operation Type", and "Classification". Information can be ordered by package id or date, specific package id's can be searched, and each package can be classified whether it was damaged or not.
  - **Application Layer** - This layer represents the front-end of the system. Node-RED is equipped with tools capable of creating web pages to show all types of information. In this case, two pages are available. The first displays the information in the database table, such as package ID, date, operation type, classification and a hyperlink that downloads the video recording related to that entry. The second page shows information regarding the state of the cargo door and its evolution over time through a graph.

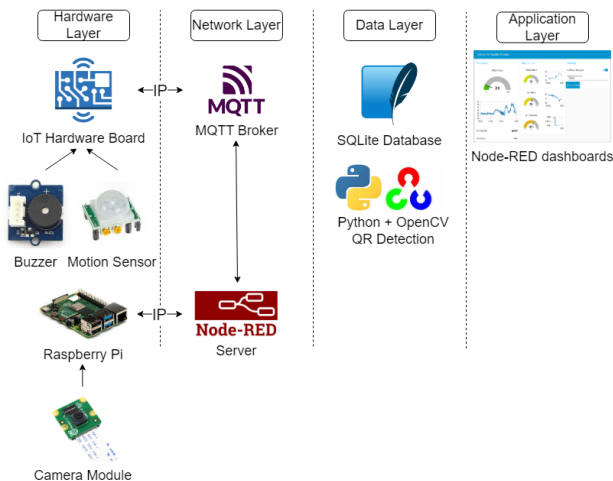


Fig. 2: System's architecture and layers

#### IV. IMPLEMENTATION AND RESULTS

This chapter covers the implementation and the results obtained by the testing procedures performed on the system developed. The goal is to simulate a real-world scenario of package delivery, so that the results obtained from implementing the system and testing are as comparable to an implementation inside of a transport truck.

A description of the real-world scenario simulated and the system's implementation inside the transport truck is provided in section IV-A. The following section includes all the tests performed, their reasoning, and an analysis and discussion of the results obtained.

##### A. Implementation

To replicate the real-world scenario, the system was implemented inside a classroom that represents the cargo space inside a transport truck. Packages of different sizes with QR codes were used as deliveries to be made. The packages were brought in and out of the classroom to simulate the loading and unloading process. The motion sensing ESP32 was positioned right next to the entrance to ensure that the Python scripts were running by the time the QR codes crossed the Raspberry Pi's camera field of view. Initially, the Raspberry was mounted halfway and high above inside the room to get a top-down view of the cargo, but the results in section II reveal that this arrangement is incompatible with the achieving the end goal, thus the Raspberry Pi was instead positioned on a table at waist level.

##### B. QR code detection range

This test was performed to measure the maximum detection range of the QR code detection script. Using printed QR codes with different sizes, this trial studied if the size difference affected QR detection. In order to work in real life, the detection range has to be enough that it can identify the QR code and record the entirety of the package in its field of view. For this test, three QR codes were printed on A3, A4, and A5-sized sheets of paper, and the farthest distance at which the QR code be read was measured. The results gathered are presented in table I.

TABLE I: Maximum detection range per QR code size

| Size | Maximum Distance [cm] |
|------|-----------------------|
| A3   | 181                   |
| A4   | 128                   |
| A5   | 96                    |

These findings imply that the bigger the QR code size, the bigger the range. With a bigger distance, the area recorded also increases. However, longer detection range has its drawbacks. One, the bigger the code, the harder, or even impossible, it is to apply on package. One of the packages used during test could only fit an A5 QR code. Second, bigger QR codes result in more expensive deployment. A slight reduction in deployment costs through the use of smaller QR codes results in significant long-term cost savings, since so many packages are handled everyday.

There are also two factors that may hinder the maximum detection range: QR Code intersection angle and the camera. When tested, the camera's line of sight intersected the QR code at the perfect angle of 90°, which means that the distance measured is the best case scenario. In real life, the angle formed by the intersection may be less than 90° due to the QR

code being slightly tilted, shortening the detection distance. This situation is explained visually in Fig. 3. The camera’s specifications can also influence the detection range. Higher quality cameras have better resolution, providing better results and more detailed video recordings. Given these shortcomings, the A5 QR code size was adopted as standard, and since it has a small detection range, a buzzer was included to indicate to the driver when a QR code was successfully identified.

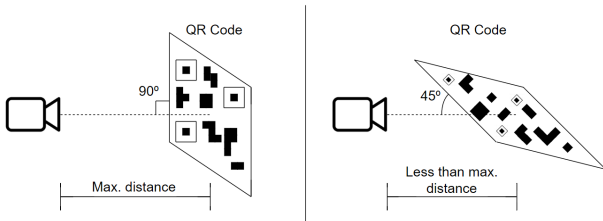


Fig. 3: Influence of QR code angle in detection

### C. QR code detection fidelity

The goal of the second test is to assess the fidelity of the two Python scripts executed by the Raspberry Pi. To achieve this, the loading and unloading procedures were simulated using five packages with distinct QR codes. As a result, 10 QR codes should be identified and 10 videos recorded, which is going to be cross-checked with the database and the Raspberry Pi folder where videos are kept. This process is going to be repeated 10 times to ensure reliable results. As a consequence to the increase of the recording duration resulting from the test in section IV-D, a second attempt was done where the process was repeated five times. The reasoning for the redo is to analyze the changes in the values gathered.

If any entries or videos are missing, it’s indicative of script or system error, and therefore should be addressed. Other than this, the test also allows to determine the extra time needed for package identification and video recording by accessing the system’s logs, and to evaluate the visibility quality of the videos recorded. The results from running this test can be found in table II.

TABLE II: QR code detection fidelity results

|                                    | 1st Attempt | 2nd Attempt |
|------------------------------------|-------------|-------------|
| Codes Scanned                      | 100         | 50          |
| Codes Missed [%]                   | 0           | 0           |
| Videos Missed [%]                  | 0           | 0           |
| Average Loading Time [s]           | 87.8        | 99.8        |
| Average Unloading Time [s]         | 119.1       | 119         |
| Average Time Spent Per Package [s] | 7.71        | 12.6        |

No codes or videos were missed means that the scripts are running correctly. These results demonstrate that the distance between the ESP32 and the Raspberry is enough to detect QR codes on time, and that the Node-RED configuration is correct. The difference between unloading and loading is due to the different way the simulation were performed. For the

first attempt, taking away the five second period needed to record the video from the "Average Time Spent Per Package" value results in obtaining the time needed to authenticate the QR code and save the video file. This means that after the recording ends, the system can’t identify another QR code for 2.71 seconds. For the second attempt, the same time needed is 2.6 which means that increasing the recording duration didn’t affect this value.

Playing back the videos recorded highlights a flaw within the setup. The top-down view captures the entire loading/unloading procedure, however, the package is not entirely visible making it difficult to inspect for damages. To fix this, the Raspberry Pi was positioned at waist level so workers can scan the QR code and display the package to the camera. To further aid drivers with package display, the buzzer also sounds off at the end of the video recording.

### D. Video recording quality

The purpose of this test is to determine if the camera module can record videos capable of detecting damages to packages that can occur during delivery. The camera used can record up to 1080p, but higher quality video means larger file size. To find the optimal relationship between file size and video resolution, a small resolution will be tested first, and increased if necessary. Small holes and dents were inflicted in packages, and these were put through the system in order to be recorded. The videos recorded are going to be examined for damages, and if they cannot be identified, the system fails in achieving its objectives, so a higher quality is needed.

Since the results were not quantifiable, a scale was defined to convey them. The scale has three classifications based on whether the damages can be seen through the videos that are: "Visible", "Barely Visible", and "Not Visible". The packages are labeled one through five, ordered from smallest to biggest. Table III displays the results obtained from the tests.

TABLE III: Video recording quality results

| Packages | Not Visible | Barely Visible | Visible |
|----------|-------------|----------------|---------|
| 1        | -           | -              | ✓       |
| 2        | -           | -              | ✓       |
| 3        | -           | -              | ✓       |
| 4        | -           | ✓              | -       |
| 5        | ✓           | -              | -       |

Packages four and five show unsatisfactory results, however, it isn’t because of the video quality, but the recording duration. These packages are bigger than the others, so when it’s time to display them, the duration of the video is insufficient to show all sides clearly. To resolve this problem, the recording period of the second script was extended to 10 seconds, allowing drivers to show all sides without worrying about time constraints. As a result, inspecting package integrity is now possible, although, this comes with a trade-off of a bigger file size and longer "Average Time Spent Per Package" calculated previously.

## V. CONCLUSION AND FUTURE WORK

The system proposed and developed for this research was capable of inspecting the integrity of packages before delivery. This was accomplished through QR code identification and video recordings of the loading and unloading process. The system is 100% accurate, as it identified all the packages, and the front-end dashboards built allowed easy access to the information for the users. This conclusion proves the first two hypotheses stated in section I-E. Even though it was successful, the system isn't fully autonomous which was a requirement that was asked by Santos e Vale. For the system to fully work the driver must display the package for the video recording, and the video also has to be examined manually in order to determine damages.

Concerning the third hypothesis, the system falters. In spite of being able to determine whether a package was damaged, since the system is assigned to a specific vehicle, and drivers may use different vehicles each day, the data gathered collected by each system must be cross-referenced with data of who's responsible for each vehicle in order to monitor driving.

Overall, the system is successful in achieving its main objective of verifying package integrity with differences regarding initial expectations. The project also serves as proof of concept, since there isn't a lot of systems in place to solve the issues stated in this paper, and this project shows there is much more room for innovation and further development.

### A. Future Work

Addressing the limitations faced during the development reveals what improvements can be made to the system. First off, the camera used can be improved upon. Replacing it with a higher quality may provide a longer detection range and better video quality. Also, the inclusion of an extra camera fixes the issue of not being able to capture the entire package during loading/unloading. If placed correctly, both cameras could record the entire procedure without the need for the driver to display each package. With this change the system's autonomy increases and the only manual intervention needed is analyzing the footage gathered. Another improvement that could be tested is the use of machine learning to classify whether a package has been damaged. If successful, the change would make the system fully autonomous, removing the need of a manual reviewer to examine package damages.

### ACKNOWLEDGMENT

This work was supported by Fundação para a Ciência e a Tecnologia, I.P. (FCT) [ISTAR Projects: UIDB/04466/2020 and UIDP/04466/2020].

### REFERENCES

- [1] S. Setti and A. Wanto, "Analysis of backpropagation algorithm in predicting the most number of internet users in the world," *Jurnal Online Informatika*, vol. 3, no. 2, pp. 110–115, 2019.
- [2] U. C. Bureau, *Quarterly retail e-commerce sales 3rd quarter 2021*, 2021.
- [3] C. Chen, D. Zhang, X. Ma, *et al.*, "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1478–1496, 2016.
- [4] Y. Pasharibu, E. L. Paramita, and S. Febrianto, "Price, service quality and trust on online transportation towards customer satisfaction," *Jurnal Ekonomi dan Bisnis*, vol. 21, no. 2, pp. 241–266, 2018.
- [5] N. Vasić, M. Kilibarda, M. Andrejić, and S. Jović, "Satisfaction is a function of users of logistics services in e-commerce," *Technology Analysis & Strategic Management*, vol. 33, no. 7, pp. 813–828, 2021.
- [6] A. Chopra, M. Arora, and S. Pandey, "Delivery issues identification from customer feedback data," *arXiv preprint arXiv:2112.13372*, 2021.
- [7] R. Doshi, N. Aphorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2018, pp. 29–35.
- [8] S. Agrawal, R. K. Singh, and Q. Murtaza, "A literature review and perspectives in reverse logistics," *Resources, Conservation and Recycling*, vol. 97, pp. 76–92, 2015.
- [9] D. H. Shih, F. C. Huang, C. Y. Chieh, M. H. Shih, and T. W. Wu, "Preventing return fraud in reverse logistics—a case study of espres solution by ethereum," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 16, no. 6, pp. 2170–2191, 2021.
- [10] S. Proto, E. Di Corso, D. Apiletti, *et al.*, "Redtag: A predictive maintenance framework for parcel delivery services," *IEEE Access*, vol. 8, pp. 14 953–14 964, 2020.
- [11] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [12] C. H. Chuang, D. H. Lee, W. J. Chang, W. C. Weng, M. O. Shaikh, and C. L. Huang, "Real-time monitoring via patch-type piezoelectric force sensors for internet of things based logistics," *IEEE Sensors Journal*, vol. 17, no. 8, pp. 2498–2506, 2017.
- [13] Santos e Vale - *Historia*. [Online]. Available: [https://www.santosevale.pt/pt\\_pt/history](https://www.santosevale.pt/pt_pt/history).
- [14] B. M. G. Mataloto, "Iot\*(ambisense): Smart environment monitoring using lora," Ph.D. dissertation, 2019.
- [15] *Oasis MQTT - Overview*. [Online]. Available: [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=mqtt#overview](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt#overview).