IoT Smart Collect - routing process and driver guidance

[3]Maryam Abbasi
[1]Bruno Nascimento, [1]Rui Santos,
[3]Filipe Sá, [4,5]Filipe Cardoso, [2]José Silva, [2]Pedro Martins

[1] Polytechnic of Viseu, Portugal
[2] CISeD - Research Centre in Digital Services, Polytechnic of Viseu, Portugal
[3] Polytechnic of Coimbra, Portugal
[4] INESC Coimbra, Portugal
[5] Polytechnic of Santarem, Portugal

[3]maryam@dei.uc.pt
[1]estgv7494@alunos.estgv.ipv.pt,[1]estgv16816@alunos.estgv.ipv.pt,
[2,3]filipe.sa@isec.pt, [4,5]filipe.cardoso@esg.ipsantarem.pt, [2]jsilva@estgv.ipv.pt, [2]pedromom@estgv.ipv.pt

***Abstract*—Waste collection is a traditional process that involves a driver collecting waste from a set of designated deposits based on a pre-determined route. The authors present a new approach that utilizes Artificial Intelligence to define the route based on the occupancy volume of the deposits. The new process involves the use of a mobile application to assist the driver during the journey. The application communicates with the central system to receive information on the next route, calculates the best possible route considering traffic laws and road conditions, and guides the driver throughout the journey. The application also provides real-time updates on the driver's progress and allows the driver to provide feedback. All collected data is stored and can be consulted and explored through lists, graphs, and filtering options. The authors believe that the new approach will improve the efficiency of waste collection and provide a better experience for the driver.**

***Index Terms*—smart cities, routing process, driver guidance, artificial intelligence**

I. Introduction

Waste collection is a critical issue in many parts of the world, where inadequate services for waste treatment and disposal pose risks to health, the environment, and the economy. The cost of waste collection, including the allocation of millions of dollars for collection and up to 40% of the cost of garbage trucks, can consume a significant portion ofmunicipal budgets [1].

The proliferation of embedded systems, particularly those facilitated by the Internet of Things (IoT), has resulted in an exponential increase in data generation that holds intrinsic value [2] [3]. To address this challenge, this article leverages the system developed in [4] to create a mobile app that supports waste collection drivers and provides feedback on their routes. The feedback received from the app about the status of the collection points and issues on the route will help the AI model improve and will know how to avoid those scenarios. The app integrates Google Maps API [5] to present the best path between collection points. The implementation of this solution involves an initial phase of learning and data collection, during which each driver will use a Near Field Communication (NFC) tag for authentication. The mobile app offers real-time monitoring of route progress, feedback, and issue reporting (e.g., missing IoT device at a collection point, overflowing or empty collection point). Drivers can also access past routes to evaluate their performance. The navigation system employs geolocation to accurately locate each collection point and track the presence of drivers, using Google Maps API to provide real-time information on traffic and road conditions and optimize collection routes.

The use of this mobile app has demonstrated improved efficiency and reduced costs in waste collection, with decreased errors during driving and optimized routes based on real-time traffic and road information.

This document is organized into five sections. Section II, presents the related work review. Section III give an overview about the architecture applied. Section IV, explain how the prototype is deployed and the initial configuration. Section V, shows the obtained results. Finally, in Section VI conclusions are drawn, followed by the introduction of future work guidelines.

II. Related works

Waste container monitoring has been a topic of interest in various scientific articles. A recent study by Meisi et al. [4] proposes a solution for optimizing waste collection routes using an AI-based algorithm, considering the current volume and short-term forecast of waste generation. The study outlines the development of a mobile application to aid drivers in following the optimal collection route, taking into account traffic rules and conditions.

2023 18th Iberian Conference on Information Systems and Technologies (CISTI)
20 – 23 June 2023, Aveiro, Portugal
ISBN: 978-989-33-4792-8

Advances in navigation technologies have been the focus of several studies, including Boyce [6], who examines the use of technologies to provide road information, such as traffic conditions and potential hazards, to drivers. Another study by Fernandez et al. [7] explores the potential of utilizing social platforms for obtaining and processing traffic information to aid in traffic management.

The impact of navigation systems and driving aids on driver performance and accident rates have been studied, including the comprehensive research by Khan [8], who proposes the Driving Monitoring and Assistance Systems (DMAS) model. Route guidance strategies have also been investigated in their effect on traffic emissions in intelligent transportation systems. In a study by Cui et al. [9], six route-guiding algorithms were tested using a Nagel-Schreckenberg cellular automaton model. The results showed that the congestion coefficient strategy (CCS) and mean velocity route guidance strategy (MVS) were the most effective in reducing emissions and improving traffic flow.

Srinivasan [10] investigates the effects of route guidance features on cognitive load and driving productivity. The study suggests that auditory route guidance systems may cause less distraction, as most driving information is acquired visually.

In conclusion, the monitoring of waste containers and the implementation of driving aids and navigation systems have been the focus of several studies, presenting solutions for optimizing waste collection, reducing emissions, improving driver performance, and reducing cognitive load.

III. Architecture

The increasing use of mobile devices in daily life is due to their convenience and accessibility [11]. With high processing power and ample storage, these devices are utilized in both professional and leisure settings. The two dominant mobile ecosystems, Android and iOS, prioritize the user by offering solutions that improve performance, monitor health, manage schedules, access financial assets, control other equipment, and more. This focus on mobile devices is referred to as ”Mobile First” approach, where all interactions occur via the device's resources such as camera, geolocation, orientation, and short distance communication. The support infrastructure for these solutions is typically hosted in the cloud for greater availability and scalability at a reasonable cost.

In this section, we present the architecture of a system for managing waste collection routes. The system architecture is illustrated in Figure 1.

An IoT device was designed to be installed in each container, with the function of reading the sensors, and sending this data to the cloud, more specifically to a Broker service.

The Broker service is based on MQTT protocol, and provides the communication channels for IoT devices, in addition to managing equipment inventory, and provides a security layer for access control. Each device is declared and managed individually, having pairs of unique authentication keys, with the possibility of remote control. [12] [13]

The Workers have the mission of reading and processing the data present in the Broker, and then save it in the database. In the meantime, rules, filters or other types of processing can be applied to adjust and treat these data.

The Database follows the basic principle of its existence, it aims to store generated data, and make them available to the API. These data are stored in document form, and are organized by containers, each dedicated to a type of data [14]. Due to the volume of data in this type of solution, the most suitable and performant is to use non-relational databases, NoSQL. [15] [16]

The REST API is designed to be deployed as a container, which enhances its scalability and performance, and allows it to respond to various clients such as mobile applications. Moreover, the containerized REST API facilitates integration and communication with other systems and solutions. [17] [18]

The artificial intelligence component of the system is responsible for training the algorithm on historical data and generating the corresponding model. This model generates collection routes based on the data from each waste container.

In Figure 2 it is possible to see in perspective the architecture of the mobile app, on the internal side we see the relationship with the sensors and storage, and on the external side its relationship with the chosen navigation service, and the system's API.

The mobile application is designed to provide a user-friendly interface by communicating with the REST API. The main features of the application include authentication, navigation, and exploration. Authentication verifies the identity and access rights of each user. Navigation provides the collection route with the corresponding waypoints, and calculates the best route by combining the route information with an external map service. Additionally, the navigation function allows the user to send feedback at each collection point. The exploration feature enables the user to access their history and apply different types of filters to the data, which can be presented as a list or graphs.

The navigation function is a critical aspect of the application. To perform navigation, the system requires at least two types of content. The first content is geographical information about the environment, which provides data about the existing routes and their characteristics based on reliable and up-to-date sources [19]. The second content is geolocation information, which provides information about the location and movement of the user [20]. By combining these two types of information with a defined collection route, the system guides the user along the entire route and enables the accomplishment of the mission, which in this case is visiting the necessary collection points.

The user authentication process is designed to be simple and efficient. Given these requirements, and considering the capabilities of mobile devices, we have chosen to use Near Field Communication (NFC) technology for authentication. NFC is a widely available technology on mobile devices, enabling quick interaction with systems and solutions while avoiding the need to type sensitive information. Additionally,

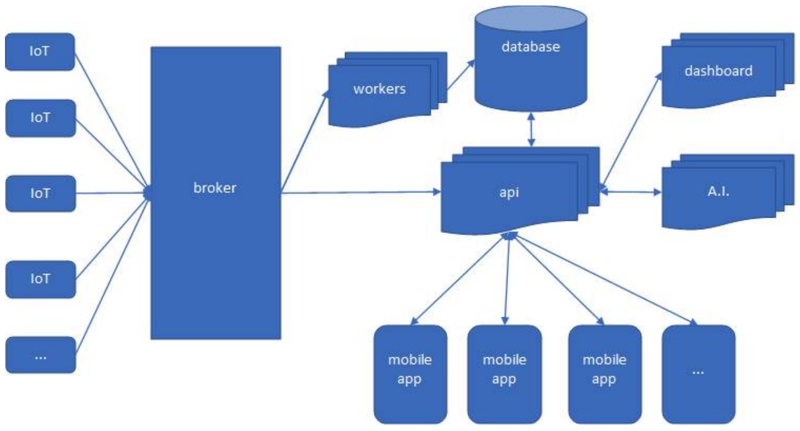


Fig. 1. Architecture illustrated for the

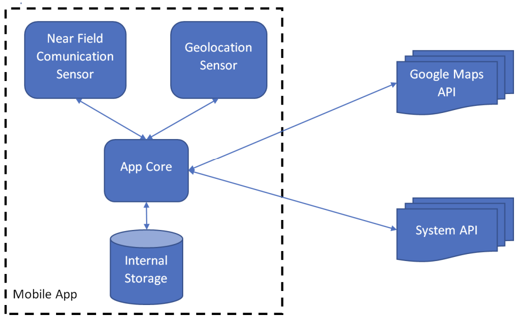


Fig. 2. Mobile App inside Architecture

NFC eliminates common problems such as forgetting access data or blocked accounts. [21] [22]

A. Mobile application Workflow

The mobile application's operating flow was designed based on the drivers' work sequence. The first step is user identification and validation. If everything is OK, we move on to the second stage, where the welcome screen appears with the next route to be taken, and the respective vehicle to be used.

To start, just click on the "Start" button, and we move on to the third stage, where navigation is carried out between all the collection points. To finish, just click on the "Finish" button, the route is completed, and we return to the welcome page. Finally, there is a fourth stage, which enables data exploration and application of filters.

B. Online and Offline

Increasingly, 4G and 5G networks are expanding geographically, with increased signal quality and respective speeds. But there are still cases, or situations, in which the coverage is weak and makes it impossible to access the internet, that is. For this reason, a synchronization system was developed that allows the temporary storage of information. When the Internet connection is reestablished, this information is sent to the central system, via the Rest API. This type of technique is transparent to the user.

Regarding the navigation service, map information, routes and directions are also stored locally and synchronized in the background. If there is a loss of internet connection, the application continues to function normally. In this way we managed to make the application work in different states, and not be 100% dependent on the internet connection.

C. Server side versus Mobile side

From the point of view of the relationship between the mobile application and the rest of the system, the "Divide and Conquer" methodology was adopted. Explaining this concept better, the mobile application will only make requests to the Rest API when it is really necessary, and obtain the information strictly necessary to carry out the tasks in question. For example, in the welcome screen, the Rest API is asked for information on the next route, and this same information is later shared with the navigation function, thus avoiding a new request.

During navigation, all progress is processed on the device, location updates have a reasonable time interval, and requests to the navigation service are only made when necessary, thus avoiding high use of the network and internal sensors, which causes a side effect on battery consumption, and consequently the reduction of device autonomy.

IV. Experimental setup

The REST API component was developed in C#, to provide easy access to data. It also applies business rules and functions such as data collection and driver login. To improve scalability, it was converted into a container and published in the Container Registry. A web app was declared using the App Service, hosting the REST API and generating a public URL.

To train the algorithm, data was generated through a python script, using normal distribution with defined mean and standard deviation for volume (mean 200, std 100), temperature (mean 30, std 10), and humidity (mean 50, std 25) for 50 collection points, 24 records per day for 2022.

An autoregressive integrated moving average(ARIMA) model [23] was trained using the generated data. ARIMA is a statistical analysis model that uses historical values to predict values for the future. With the ARIMA model a process was created to select the collection points to be collected. The process will firstly select all the collection points with an occupation above 80%, after that the process will train the ARIMA model for each point and predict the occupation for the next day, if the current occupation is above 90% or the next day occupation is above or equal to 95% the collection point will be selected to be collected.

After selecting the collection points to be collected the process sends a file to the API with collection Points, that will be added to a table in Cosmos DB. With that information the mobile app with the Google Maps API will create routes with the collection point closer to each other, and with real time information from the Google Maps API it will also select the fastest and safest way to get to each collection point.

For the mobile app development, Ionic framework [24] was chosen, as it is widely used and offers updates, native app generation for Android/iOS, smooth learning curve (based on JS/TypeScript), web components, and easy access to device features (Camera, Geolocation, NFC). Its style is based on Material Design, with quick development and good documentation.

The Fig. 3 overviews the prototype's main functions. It's based on 3 tabs using the Ion-Tabs component. Tab 1 (screen with Home title) has the reception function displaying a welcome message, the next route and assigned vehicle via Google Maps and Ion-Card. Tab 2 (screen with Navigation title) has the navigation function with crossing points listed using Ion-List, a Progress Bar, and a map with an anomaly reporting option and a button to finish route. Tab 3 (screen with History title) has the history function using Ion-List, Ion-DateTime, and Ion-Modal to view routes, filter, and view the same data in graph form using Ng2-Charts. Google Maps is chosen for its worldwide coverage, detailed maps, compatibility, and API [25] [26]. The pickup routes are georeferenced and the navigation tab uses Geolocation and adjusts viewing angle for better UX. Data is updated in real-time with Geolocation from the Capacitor/Geolocation component. User authentication uses NFC tags and the Awesome-Cordova-Plugins/NFC component to read the tag and verify user info.

V. Results and analysis

The first testing phase was carried out in a pre-production environment, and focused on validating the various system functionalities, on the common tasks of Create Read Update Delete (CRUD) on all types of data. At the beginning with little data, but over time this volume gradually increased.

In the second phase of testing, focused on performance, response times and system overload were analyzed, this time in the production environment. There were some mishaps in training the Artificial Intelligence model, since the volume of data was considerable and training the model took many hours. This difficulty was overcome through refinement and selection of the most appropriate data. As for the mobile application, some problems were found in the management of services, which hindered user interaction and provided a bad user experience. The solution involved refactoring some components and reorganizing access to services.

TABLE I
Google Play Store performance

Version	Stability	Performance	Accessibility	Security & Reliability
1	0	2	206	0
2	0	1	7	0
3	0	2	10	0
4	0	2	0	0
5	0	2	15	0
6	0	1	5	0

After overcoming all obstacles, the third phase of testing was reached, this one aimed at end users, and in a production environment. The feedback obtained was very useful because it came from people who will use the solution on a daily basis, and are more process-oriented than technology-oriented. At the beginning, the reported problems were more focused on the perception of information, and later on, they moved on to improvements in interaction, and suggestion of new features, such as voice commands.

It is worth noting that in order to speed up the testing of the mobile application, it was submitted and approved in the Google Play Store, which proves to the user that he is not installing a virus or something similar, discloses the actual accesses to the components, and demonstrates the veracity of the application. This procedure contributed to a relationship of trust between test users and researchers.

The Google Play Store performs a series of tests, considering the main axes of a mobile application: Stability, Performance, Accessibility, Security and Reliability. In table I we can check these results, and their progression with each released version. It is natural that there is an increase in anomalies in some aspects, because each version contains new features, evolution's, adjustments and corrections. In our case, the main trend is a decline in all aspects, indicating positive progression, but it is worth mentioning that the aspects of Security and Reliability have always been at zero.

The mobile application assists drivers during their route by displaying a list of all pickup points and the pickup route. By integrating with Google Maps API, the route can be shown on the map with starting and ending points, as well as the pickup points. This service determines the optimal route to each point following the local road map.

Testing showed that the app benefits new drivers by reducing their route learning curve compared to more experienced drivers who previously relied on their memory or prior knowledge. This solution levels the playing field, enabling all drivers to complete the route effectively.

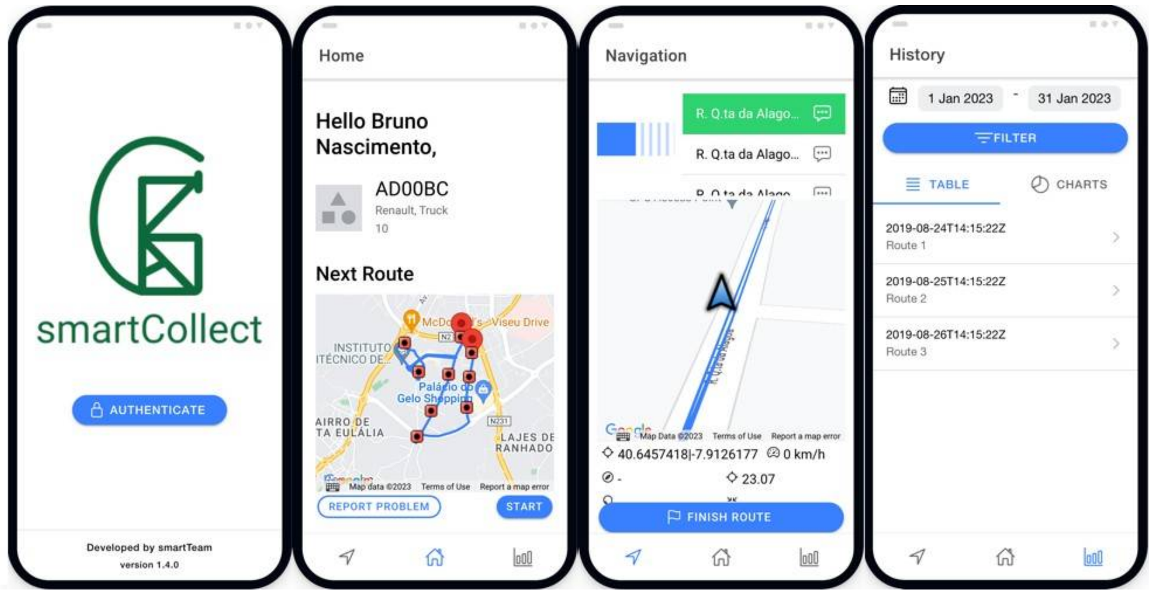


Fig. 3. Mobile App main screens

For the training of Arima, there is the need to define values for the variables p,d and q. The p value refers to the Auto Regressive term. It denotes the number of Y lags to be utilized as predictors. The d value is the smallest number of difference that is required to render the series stationary. The q value refers to the Moving Average term. It is the amount of lag prediction errors that should be included in the ARIMA Model.

To determine the best values for each variable, a grid search algorithm was used to test which combination of parameters returned the lower root mean squared error(RMSE). After the execution the algorithm returned the values P equal to 4, D equal to 0 and Q equal to 2, with an RMSE = 81.634% as shown on the Figure 4.

In the Table II, a big improvement is observed after applying the app in July 2022.
The routes became more efficient as it passes through less collection points per route and the number of routes per month also decreased, the duration of each route also decreased, which means that with use of the app it managed to make waste collection more efficient and cleaner, as fuel consumption and carbon dioxide emissions also went down.

In the Table II, a decrease on driver errors is also observed. Through dynamic collection routes routine and monotony were prevented, resulting in better driving performance. Additionally, the anomaly reporting functions improved communication among employees and generated formal documentation of issues found at collection points, leading to improved community well-being.

After the application of the app, a reduction on costs was observed, as shown in 5. With the reduction of the number of routes per month and the number of collection point per route, the total number of kilometers covered also decreased which means that it managed to save money on fuel and also save time. With the time saved on each route, the drivers will have a more flexible schedule, as they will not need to do as many routes. The drivers working less hours will also help on reduce driving errors.

VI. Conclusions and Future Work

With this Article, the objective was to reconstruct the way the waste collection process is done nowadays. With the application of IoT Sensors and the use of an AI Model, a new process was created to select the collection points that are in need to be collected.

Then a Mobile Application was developed to make use of that information and assist the driver on the route and provide a way to get more data to improve the AI model.

With this solution, waste collection expenses were reduced, making routes more efficient and optimized, resulting in fewer unnecessary collections.

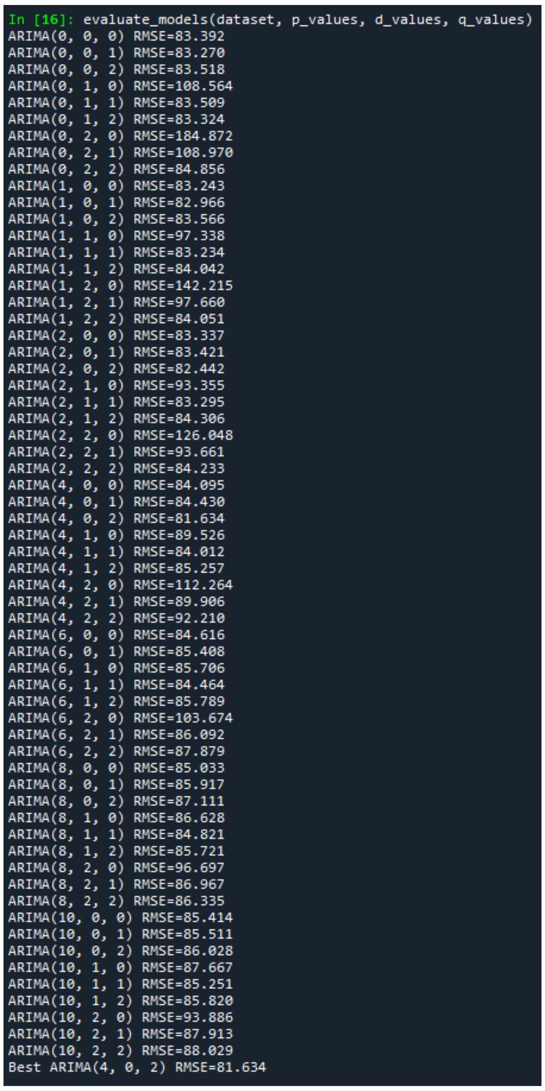
AI model accuracy improves with increasing collection points. Solution is scalable and considers cost reduction, CO2 emissions, and ecological footprint, making companies more environmentally friendly.

Driver feedback about which collection points were overflowing or empty will also help the AI model predict and avoid these events. Navigation system boosts driver confidence and reduces driving errors, and considers current traffic to avoid prolonged stops. Also helps new drivers during the learning phase.

Problem handling improved as it is documented in real-time with better location information, leading to better data quality. This kind of information is always welcome in the quality departments, which are inputs to the Objectives and Key Results (OKRs).

TABLE II
REDUCTION OF DRIVING ERRORS

Date	Avg. Collection Points	Routes	Avg. Time (Mins)	Avg. Errors
Jan-2022	10	25	100	5
Feb-2022	8	24	83	4
Mar-2022	7.5	25	77	3
Apr-2022	9	22	87	3
May-2022	8	23	80	2
Jun-2022	9	20	88	3
Jul-2022	6	15	67	2
Aug-2022	4	13	44	1
Sep-2022	5	11	50	1
Oct-2022	4	12	45	1
Nov-2022	4	12	45	0
Dec-2022	5	14	50	1



```
In [16]: evaluate_models(dataset, p_values, d_values, q_values)
ARIMA(0, 0, 0) RMSE=83.392
ARIMA(0, 0, 1) RMSE=83.270
ARIMA(0, 0, 2) RMSE=83.518
ARIMA(0, 1, 0) RMSE=108.564
ARIMA(0, 1, 1) RMSE=83.509
ARIMA(0, 1, 2) RMSE=83.324
ARIMA(0, 2, 0) RMSE=184.872
ARIMA(0, 2, 1) RMSE=108.970
ARIMA(0, 2, 2) RMSE=84.856
ARIMA(1, 0, 0) RMSE=83.243
ARIMA(1, 0, 1) RMSE=82.966
ARIMA(1, 0, 2) RMSE=83.566
ARIMA(1, 1, 0) RMSE=97.338
ARIMA(1, 1, 1) RMSE=83.234
ARIMA(1, 1, 2) RMSE=84.042
ARIMA(1, 2, 0) RMSE=142.215
ARIMA(1, 2, 1) RMSE=97.660
ARIMA(1, 2, 2) RMSE=84.051
ARIMA(2, 0, 0) RMSE=83.337
ARIMA(2, 0, 1) RMSE=83.421
ARIMA(2, 0, 2) RMSE=82.442
ARIMA(2, 1, 0) RMSE=93.355
ARIMA(2, 1, 1) RMSE=83.295
ARIMA(2, 1, 2) RMSE=84.306
ARIMA(2, 2, 0) RMSE=126.048
ARIMA(2, 2, 1) RMSE=93.661
ARIMA(2, 2, 2) RMSE=84.233
ARIMA(4, 0, 0) RMSE=84.095
ARIMA(4, 0, 1) RMSE=84.430
ARIMA(4, 0, 2) RMSE=81.634
ARIMA(4, 1, 0) RMSE=89.526
ARIMA(4, 1, 1) RMSE=84.012
ARIMA(4, 1, 2) RMSE=85.257
ARIMA(4, 2, 0) RMSE=112.264
ARIMA(4, 2, 1) RMSE=89.906
ARIMA(4, 2, 2) RMSE=92.210
ARIMA(6, 0, 0) RMSE=84.616
ARIMA(6, 0, 1) RMSE=85.408
ARIMA(6, 1, 0) RMSE=85.706
ARIMA(6, 1, 1) RMSE=84.464
ARIMA(6, 1, 2) RMSE=85.789
ARIMA(6, 2, 0) RMSE=103.674
ARIMA(6, 2, 1) RMSE=86.092
ARIMA(6, 2, 2) RMSE=87.879
ARIMA(8, 0, 0) RMSE=85.033
ARIMA(8, 0, 1) RMSE=85.917
ARIMA(8, 0, 2) RMSE=87.111
ARIMA(8, 1, 0) RMSE=86.628
ARIMA(8, 1, 1) RMSE=84.821
ARIMA(8, 1, 2) RMSE=85.721
ARIMA(8, 2, 0) RMSE=96.697
ARIMA(8, 2, 1) RMSE=86.967
ARIMA(8, 2, 2) RMSE=86.335
ARIMA(10, 0, 0) RMSE=85.414
ARIMA(10, 0, 1) RMSE=85.511
ARIMA(10, 0, 2) RMSE=86.028
ARIMA(10, 1, 0) RMSE=87.667
ARIMA(10, 1, 1) RMSE=85.251
ARIMA(10, 1, 2) RMSE=85.820
ARIMA(10, 2, 0) RMSE=93.886
ARIMA(10, 2, 1) RMSE=87.913
ARIMA(10, 2, 2) RMSE=88.029
Best ARIMA(4, 0, 2) RMSE=81.634
```

Fig. 4. Arima search for best Parameters

The testing and validation phases with end users proved to be very useful and fruitful, in the sense that the solution gained maturity in a short period of time. The constant feedback received, both corrections and small improvements, helped to reduce the barrier to adoption and adaptation of drivers, in addition to breaking resistance to changes, since the mode of operation is portrayed in a simple and effective way.

Still in the field of the mobile application, the fact that we are using a hybrid development framework is an added value of the solution, because it offers independence from the mobile ecosystem. This makes it possible to generate native Android and iOS applications with a single source code base. It also contributes to ensuring that in future stages of evolution, in case of need to allocate human resources, there will be a greater supply of people with these skills.

The publication of the mobile application in the respective application stores, such as the Google Play Store, has a great contribution in the testing and validation phase, because it conveys confidence to the testers and credibility of the solution. In addition, stores provide more data and provide important analysis tools, such as crash reports, usage and quality indicators, compatibility analysis with a wide range of devices, among others.

A. Future Work

In the future, voice command functions will be added for improved driver interaction and device use during driving. A scoring methodology will be evaluated to reward drivers for good performance, considering factors such as speed, braking, and changes in direction.

Integration with the vehicle will also be studied for fuel consumption and emission data, to reduce the ecological footprint and improve Carbon Emission indicators.

Development of an algorithm to reorganize the routes with parameters of proximity and time interval, that is, to group the points that need to be collected due to their geolocation. The other parameter, time interval, will balance the number

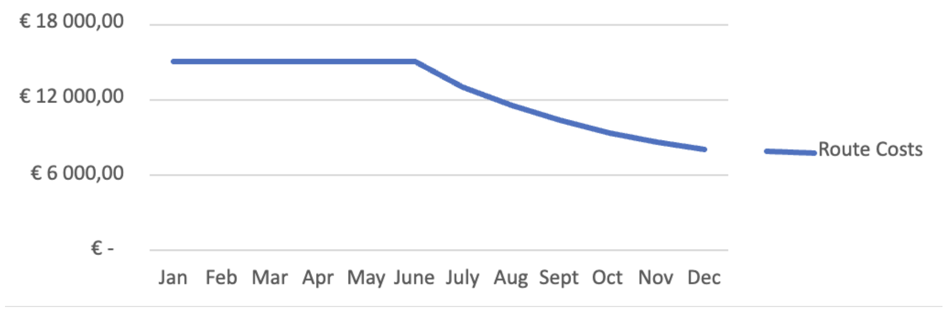


Fig. 5. Routes Costs reduction

of routes by day intervals, to avoid one day being overloaded, and the following days with few routes.

Acknowledgements

"This work is funded by National Funds through the FCT – Foundation for Science and Technology, I.P., within the scope of the project Ref. UIDB/05583/2020. Furthermore, we would like to thank the Research centre in Digital Services (CISeD) and the Instituto Politécnico de Viseu for their support." Maryam Abbasi thanks the National funding by FCT - Foundation for Science and Technology, P.I., through the institutional scientific employment program-contract (CEECINST/00077/2021)

References

[1] Paulo Victor Sousa and Macelo Costa. Prot ótipo de lixeira inteligente no contexto das smart cities e da internet das coisas. *Revista Sistemas e M´ıdias Digitais (RSMD)*, 2019.

[2] Ricciotti Cardoso. *Desenvolvimento De Uma Plataforma Inteligente Para Coleta De Lixo Reciclável Aplicando o Conceito IoT*. PhD thesis, Universidade Santa Cec´ılia, 2017.

[3] Igor Lima, Aldo Sales, Marco Domingues, and Anderson Moreira. Uso de iot para auxiliar na coleta de lixo urbano. In *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação*, pages 334–341, Porto Alegre, RS, Brasil, 2022. SBC.

[4] Bruno Nascimento and Rui Santos. Smart cities - prediction and prioritization of residues collection points. *CISeD - Inside One Publication*, 2023.

[5] Timothy John Pattiasina, Eddy Triswanto Setyoadi, and David Wijayanto. Saving matrix method for efficient distribution route based on google maps api. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2-3):183–188, 2018.

[6] David E Boyce. Route guidance systems for improving urban travel and location choices. *Transportation Research Part A: General*, 22(4):275–281, 1988.

[7] Alberto Fern ández-Isabel and Rubén Fuentes-Fernández. Social-aware driver assistance systems for city traffic in shared spaces. *Sensors*, 19(2):221, 2019.

[8] Muhammad Qasim Khan and Sukhan Lee. A comprehensive survey of driving monitoring and assistance systems. *Sensors*, 19(11):2574, 2019.

[9] Nan Cui, Bokui Chen, Kai Zhang, Yi Zhang, Xiaotong Liu, and Jun Zhou. Effects of route guidance strategies on traffic emissions in intelligent transportation systems. *Physica A: Statistical Mechanics and its Applications*, 513:32–44, 2019.

[10] Raghavan Srinivasan and Paul P Jovanis. Effect of in-vehicle route guidance systems on driver workload and choice of vehicle speed: Findings from a driving simulator experiment. In *Ergonomics and safety of intelligent driver interfaces*, pages 97–114. CRC Press, 2020.

[11] Hoi Yin Yu, Yan Yung Tsoi, Anthony Hae Ryong Rhim, Dickson KW Chiu, and Mavis Man-Wai Lung. Changes in habits of electronic news usage on mobile devices in university students: a comparative survey. *Library Hi Tech*, 40(5):1322–1336, 2022.

[12] Silvio Quincozes, Tubino Emilio, and Juliano Kazienko. Mqtt protocol: fundamentals, tools and future directions. *IEEE Latin America Transactions*, 17(09):1439–1448, 2019.

[13] Dan Dinculean ă and Xiaochun Cheng. Vulnerabilities and limitations of mqtt protocol used between iot devices. *Applied Sciences*, 9(5):848, 2019.

[14] Suprio Ray, Bogdan Simion, and Angela Demke Brown. Jackpine: A benchmark to evaluate spatial database performance. In *2011 IEEE 27th International Conference on Data Engineering*, pages 1139–1150. IEEE, 2011.

[15] Andreas Meier and Michael Kaufmann. *SQL & NoSQL databases*. Springer, 2019.

[16] MAM MUS. Comparison between sql and nosql databases and their relationship with big data analytics. *Asian Journal of Research in Computer Science*, 2019.

[17] Bokolo Anthony Jnr, Sobah Abbas Petersen, Dirk Ahlers, and John Krogstie. Api deployment for big data management towards sustainable energy prosumption in smart cities-a layered architecture perspective. *International Journal of Sustainable Energy*, 39(3):263–289, 2020.

[18] Harihara Subramanian and Pethuru Raj. *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Packt Publishing Ltd, 2019.

[19] Guonian L ü, Michael Batty, Josef Strobl, Hui Lin, A-Xing Zhu, and Min Chen. Reflections and speculations on the progress in geographic information systems (gis): a geographic perspective. *International journal of geographical information science*, 33(2):346–367, 2019.

[20] Hardy Pundt. Field data collection with mobile gis: Dependencies between semantics and data quality. *GeoInformatica*, 6(4):363, 2002.

[21] Roland Schl öglhofer and Johannes Sametinger. Secure and usable authentication on mobile devices. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, pages 257–262, 2012.

[22] Vabrice T Wilder, Yujing Gao, Shuangbao Paul Wang, and Alfredo J Perez. Multi-factor stateful authentication using nfc, and mobile phones. In *2019 SoutheastCon*, pages 1–6. IEEE, 2019.

[23] Domenico Benvenuto, Marta Giovanetti, Lazzaro Vassallo, Silvia Angeletti, and Massimo Ciccozzi. Application of the arima model on the covid-2019 epidemic dataset. *Data in Brief*, 29:105340, 2020.

[24] Letizia Bollini. Beautiful interfaces. from user experience to user interface design. *The Design Journal*, 20(sup1):S89–S101, 2017.

[25] A Alsobky and R Mousa. Estimating free flow speed using google maps api: accuracy, limitations, and applications. *Advances in Transportation Studies*, 50, 2020.

[26] A Mu ñoz-Villamizar, EL Solano-Charris, Mojdeh AzadDisfany, and L Reyes-Rubiano. Study of urban-traffic congestion based on google maps api: the case of boston. *IFAC-PapersOnLine*, 54(1):211–216, 2021.