An Intelligent Machine Learning-Based Real-Time Public Transport System

Menzi Skhosana[1], Absalom E. Ezugwu[2(✉)], Nadim Rana[3], and Shafi'i M. Abdulhamid[4]

[1] School of Mathematics, Statistics and Computer Science, University of Kwazulu-Natal, Westville Campus, Private Bag X54001, Durban 4000, South Africa
216032734@stu.ukzn.ac.za

[2] School of Computer Science, University of KwaZulu-Natal, Pietermaritzburg Campus, King Edward Avenue, Pietermaritzburg 3201, KwaZulu-Natal, South Africa
ezugwua@ukzn.ac.za

[3] College of Computer Science and Information Technology, Jazan University, Jizan, Kingdom of Saudi Arabia
nadimranal7@gmail.com

[4] Department of Cyber Security Science, Federal University of Technology, Minna, Niger, Nigeria
shafii.abdulhamid@futminna.edu.ng

Abstract. More often than not, commuters are left stranded at pick-up spots – clueless about the availability and proximity of public transport vehicles hence the stigma of public transport being unreliable, especially in developing countries. This is a result of poorly managed fleets, caused by varying demands and rigid schedules. In this paper, we present an intelligent real-time transport information system to keep commuters informed about the status of buses currently in transit, and also provide an insight to bus managers based on ridership data and commuter behavior. The system is composed of three subsystems designed to cater for commuters, bus-drivers and bus managers respectively. This system is developed on the Backend-as-a-Service (BaaS) platform Firebase. Furthermore, a neural network is trained to provide predictions to bus managers on the expected ridership numbers per route. The trained model is integrated with a web application for bus managers. An Android application used by bus drivers collects the ridership data being fed to the network. The proposed system was evaluated with a real-world data set that contains the daily ridership on a per-route basis dating back to 2001. Evaluation results confirm the effectiveness of the new system in reducing the total mileage used to deliver commuters, reducing fuel costs, increasing the profit of bus operators, and increasing the percentage of satisfied ridership requests.

Keywords: Smart Irenbus · Bus location tracking · Mobile tracking · Short-term forecasting · Long-term forecasting

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 649–665, 2020.
https://doi.org/10.1007/978-3-030-58817-5_47

1 Introduction

Public transport plays an integral role in the economy and sustainability of urban areas. Yet there are very little to no advancements being made in this sector in developing countries. Amongst many other benefits, public transportation is significantly affordable, reduces traffic congestions, increases mobility and can lessen the carbon footprint of urban areas. The sizable capacity of public transport vehicles results in lesser trips and reduced fuel consumption. Statics South Africa (SSA) stated that more 76.7% of the South African population relies on public transport for daily commutes [1]. This again highlights the importance of public transport and why problems in this sector need to be addressed. With the recent trend and drive for more viable and eco-friendly environments through the construction of smart cities in developed countries around the world, modernising public transportation by integrating it with the latest technologies could be an excellent way for developing countries like South Africa to also get started on this trend and potentially improve its economic state. Furthermore, it is noteworthy that approximately 55% of the world's population resides in urban areas, and this figure is expected to increase to 68% by the year 2050 [9]. The given statistics make it evident that the development of smart cities is needed to ensure sustainability and efficiency of urban areas, through the deployment and use of intelligent technologies to render services to citizens and to manage and refurbish existing infrastructures such as public transportation.

According to the Independent Communications Authority of South Africa (ICASA), smartphone usage has increased drastically over the last five years. As of 2018 September, 81.7% of the South African population owned a smartphone, which is nearly double the percentage of 2016. Also, according to ICASA's report on the state of the ICT sector in SA - March 2019, the national population coverage for 3G remained stable at 99.5% in 2018, and the coverage for 4G/LTE increased from 76.7% in 2017 to 85.7% in 2018 [10]. Given the widespread use of smartphones and good network coverage - the use of smartphones as a platform in implementing the public transport management system will allow for easy access to transit information, and lower implementation costs as almost all smartphones are equipped with Global Positioning System (GPS) technology and other position and orientation sensors.

The ability to predict future public transport demand could be revolutionary as it will allow for public transport vehicles to be scheduled based on forecasted demand. And this will, in turn, save the authorities a lot of money by reducing unnecessary trips, saving fuel, and mitigating the effects of global warming while ensuring that commuters are transported efficiently.

This paper proposes the development of a holistic, dynamic real-time information system that keeps passengers informed about buses currently in transit and other relevant information such as vehicle arrival time. The system is composed of three subsystems that cater to commuters, bus drivers and bus managers, respectively. More so, a machine learning-based model is employed to predict ridership numbers that can be expected per route based on past ridership data, so that the buses are assigned by the bus managers accordingly. Summarily, the main goal of this paper is to build a dynamic transportation system which helps the traditional bus companies to become

more competitive and productive in a typically challenging business environment. The system is called *Smart Irenbus*: an intelligent real-time public transport system and it is a software solution that assists bus companies in carrying out their logistic tasks by providing accessible communication between controllers and drivers and also serves as a data management tool to intelligently store and organize data collected on daily commutes and offer predictions and useful insights based on it. The proposed system differs from the previous Irenbus implementation presented in [29] because of the added intelligence and predictive capability that comes with the current system.

2 Related Work

Using GPS modules and embedded mini-computer systems, Sungur et al., developed a bus monitoring system that provided real-time information such as the estimated bus arrival time and the route name to commuters. This information was displayed on an LCD screen mounted at the bus stop [2]. Although this system is pretty solid and provides useful information to commuters it has a few shortfalls, such as being fixed at one place, i.e. it's not able to provide information to commuters who are not at the bus stop, and also implementing this system will be a costly operation since every bus stop has to have an LCD screen mounted on it.

In [23], the authors discussed how a machine learning approach could be used to implement and assess predictive services for the users of a bike-sharing system. The models used in this study were trained on real-world historical usage data comprising of more than 280 000 entries covering all hires in Pisa for two years. Seasonality manifests sharp changes in the usage patterns (e.g. bikes tend to be used more in spring compared to winter). These seasonal patterns were captured by the learning models through the appropriate encoding of the bike usage time, which explicitly models cyclic information such as weekday and holiday.

Manikandan et al. [12] used the Global System for Mobile Communication (GSM) Query Response System with GPS trackers to develop a real-time bus information system. Where a user has to send a request to the central server system using SMS, and a response is sent back to the user as an SMS with the information requested [12]. Implementing this system will be much easier as it uses mobile phones to provide real-time information to commuters. But the major drawback with this system is that the user has to send an SMS to request information of a specific bus - no information about all buses currently in transit is provided.

Shirisha et al. [5] designed a system that equipped a bus with GSM-based processor and a bus stop with a time data transmitter; this then allowed the commuters to identify at what time the bus reached its previous bus stop, and based on this information, the commuter can estimate the position of the bus [5]. The time data is transmitted continuously through IR led. And whenever the bus reaches a bus stop, the time data will be acquired through TSOP (IR sensor package) and it will be stored into the ROM of the main processing unit. The GSM module interfaced with the main processor can then send the time information along with the bus stop name to the caller (commuter) in the form of an SMS.

Othman and Tan [25] proposed a simulator supplied with predictive travel times through congestion prediction to evaluate and improve bus utilization through effective scheduling. Their model predicted correctly predicted the exact travel times 13% more accurately than the expected arrival times estimated by the Land Transport Authority (LTA) of Singapore.

In [26], Toqué et al. investigated the use of smart card data to forecast multimodal transport passenger flow with both long- and short-term forecasting time horizons. The study was deemed being challenging as it involved a major business district in the Paris Metropolitan Area (La Défense). Their results demonstrated the effectiveness of machine learning methods for such prediction tasks, as they obtained reliable results for all transport modes (train, tram, and bus). They aim at improving the results by investigating how anomalies could be considered in the prediction process.

Liu et al. in [27] proposed a system that uses a single inexpensive camera mounted overhead to count passengers by combining a Convolutional Neural Network detection model and a spatiotemporal context model to address the counting problem in the scenes of low resolution and with a variation of illumination, pose and scale. Experimental results showed better performance, and they plan to extend the current method with more deep learning algorithms. The only drawback to their system is that it may not work in a very dense and crowded scene, and for this case, they plan to explore crowd density map estimation for their future work.

A Web-based system was developed by Kumbhar et al. for tracking buses in transit using a GPS tracker installed on the bus. Users get real-time information straight to their mobile phones. Using Google Maps user can see the bus on the map as it moves [6]. Their system ensures punctuality - which has been proven to be an essential factor in making buses more reliable [4]. Their system also seems to be less expensive to implement compared to other previously done work but still lacks the ease of use - as it only offers a web application with no native mobile application. And doesn't give insight to public transport authorities about future demand and ridership patterns.

In [28], they explored options for using smart card data for performing simple analyses by using transport planning software. The data was converted to represent passengers per line and matrixes between stops. This matrix was then taken into the network to produce the measured passenger flows. Their method turned out to be valuable to operators to gain insights into small changes but was not able to give accurate insights into long-term changes.

According to our knowledge, there haven't been any published work that discusses how a holistic public transportation system can be designed and developed to relay useful transit information amongst commuters, drivers and bus operators while at the same time collecting and studying commuter boarding data to enable the prediction of future ridership patterns, hence allowing the development of better and more efficient bus schedules.

3 Smart Irenbus Design and Implementation

The primary objective of smart Irenbus is to provide transit information to commuters, but for commuters to get the bus transit information we have to be able to track the bus location, this then led to the incorporation of the driver component into Smart Irenbus. Again, the drivers need to be managed and assigned to specific buses; this then pointed out that a manager component will be needed for the system. After gathering all the system requirements, it was clear that the Smart Irenbus system will have to be divided into three subsystems. The Agile Software Development Methodology was employed as it is an adaptive approach that responds to changes favourably and allows for iterative development and continuous testing.

To take full advantage of the cloud, the system was mainly developed using Google's Firebase, which is Backend-as-a-Service platform. Firebase was chosen over other similar platforms such as the Amazon Web Service's EC2 because it allows both mobile and web applications access to shared data and computing infrastructures. Any changes made to the data are automatically synchronised with the Firebase cloud and with other clients within milliseconds [16]. As the mobile application was developed for Android devices, the integration of Firebase was seamless as both platforms are from Google.

A. Database Design

To handle frequent location updates and keep all devices connected in synchronization, a robust and dynamic database infrastructure is required. The Firebase Realtime Database is a non-structured query language (NoSQL) cloud-based database that synchronizes data across all clients in real-time and provides offline functionality. Data is stored in the Realtime database as JSON, and all connected clients share one instance, automatically receiving updates with the newest data [7]. In conventional database systems, the information is stored in tabular format. But for the NoSQL database, data or records are stored in a tree-like structure. Records are additionally prearranged in collections [11]. After several iterations of requirements gathering, the resulting database structure consisted of five nodes:

- *Buses* which stores basic bus information uniquely identified by a 16 character randomly generated id.
- *OnlineBus* which stores buses that are currently in-transit with both current location and current driver.
- *BusFares* which stores bus prices for different stages for both peak and off-peak times.
- *BusLines* which stores the bus line information like scheduled times and more.
- *Users* which stores basic user information.

The use of a NoSQL database was mainly chosen because of its dynamic schema - it provides the flexibility to change the data schema without modifying any of your existing data. While the traditional relational database management system relies on a static data structure, the best practices stipulate the establishment of a database schema before any coding even begins [15]. A lot of traditional databases use a locking

mechanism to ensure data integrity, i.e. before a transaction starts, the relational database management system (RDMS) marks the data so that no other process can modify it until the transaction either succeeds or fails. This then can have a severe impact on the database performance when serving thousands of users concurrently. NoSQL makes a trade-off between consistency and performance by shunning the use of locked transactions. After taking all of these facts into consideration, the NoSQL database was an obvious choice to efficiently handle the dynamic data model of Smart Irenbus.

B. User Authentication

Mobile and web applications need to provide some form of authentication to identify users and control and protect user data. Without a way to differentiate one user from another, it would also be impossible for the application to know which data and settings belong to which user [7]. Furthermore, authentication allows users to set preferences, store data and helps provide personalized experiences that are consistent across all of the user's devices [8].

The Smart Irenbus mobile application can only be used by the commuter and the driver, and the web application (online dashboard) by the bus manager. The Smart Irenbus mobile application is mainly divided into two, based on the login credentials submitted the user interface for a commuter or a driver will be shown. When users are signing up, drivers are required to provide a bus code, which is a 16-character alphanumeric code, and this code can only be obtained by and from bus managers. The bus code ensures the integrity of the system and prevents bogus driver accounts from being created.

Commuter App

The following screenshots show some of the functionality of the Smart Irenbus commuter mobile application (Figs. 1, 2 and 3).

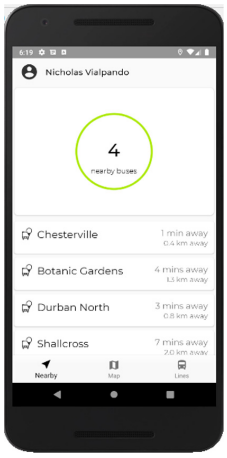

Fig. 1. Nearby fragment

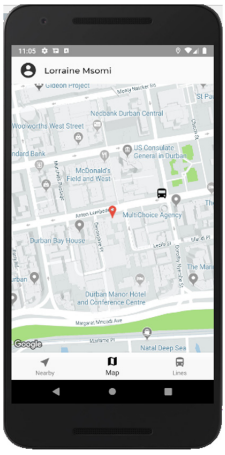

Fig. 2. Map fragment

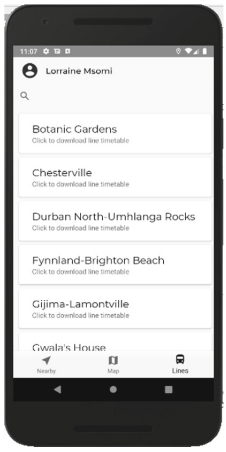

Fig. 3. Lines fragment

Nearby Fragment shows all nearby buses - which are buses within a 3 km radius from where the user (commuter) is currently located. The estimated distance and time it will take the bus to get to the user's location is shown along with the corresponding bus route/name. These estimates are calculated using the Google's Directions API, which is a service that calculates directions between locations, and it is accessed through an HTTP interface, with requests constructed as a URL string, and latitude/longitude coordinates to identify the locations [3]. Google also provides a similar API called the Distance Matrix API which provides access to travel distance and time for a matrix of origins and destinations, but the Directions API was chosen over this API because it offers more flexibility as it doesn't only calculate the shortest route but other alternate routes.

Map Fragment has the full map view which has all online buses' current locations, and these are updated in real-time as the buses move. The user can pan around, zoom in and zoom out to see the online buses. The map was created using the Maps SDK for Android API, which handles access to Google Maps servers, data downloading, map display, and response to map gestures.

Lines Fragment lists all bus lines schedule (timetables) - these are in a Portable Document Format. To minimize the amount of storage requirement of the mobile application on the user's device, these time-tables were uploaded to Cloud Storage for Firebase which is a powerful, simple, and a cost-effective object storage service that allows for file downloads regardless of the network quality. The PDF timetable of a bus line is then only downloaded when the user clicks on that bus line on the list.

Driver App

The following screenshots show some of the Smart Irenbus driver mobile application's functionality (Figs. 4, 5 and 6).

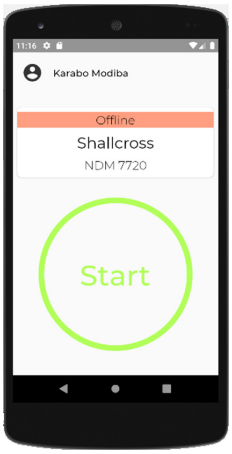

Fig. 4. Offline state

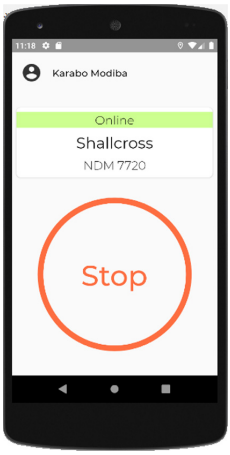

Fig. 5. Online state

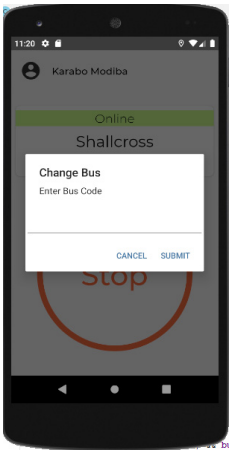

Fig. 6. Bus change

Offline State has the bus driver as offline, which means that the current location of the bus that is driven by this driver will not be shown to Smart Irenbus users.

Online State has the bus driver as online, and the current bus location is updated in real-time as it moves around. And these location updates are set as visible Smart Irenbus users.

Bus Change option allows the currently logged in bus driver to change the bus they are presently driving. This is done by asking the bus driver to input the bus code, which is a 16-character alphanumeric code that uniquely identifies each bus in the Smart Irenbus system.

Online Dashboard

The online dashboard was developed to provide the bus manager with an overall view of the buses and drivers registered on the Smart Irenbus system. This dashboard is hosted using Firebase Hosting, which offers fast and secure hosting for web applications, static and dynamic content, and microservices.

Map Tab has a full map view of all buses currently in transit. The bus manager can interact with the map and click on any bus marker to see detailed information about that bus. The map also has Google Street View integrated into it, which enables users to view and navigate through 360 degrees horizontal and 290 degrees vertical panoramic street-level images of various cities.

Buses Tab shows a table of all registered buses. The bus manager can add more buses, edit, and update information about buses that already exist and search for buses using bus code, number plate, bus number and route name. And when a new bus is added the 16-character alphanumeric code is automatically generated. In this tab, the bus manager will see the predicted number of boardings per selected route, so that they can assign buses accordingly.

Drivers Tab shows a table of all registered bus drivers with the bus code of the bus they are currently driving. More than one driver can be assigned to one bus, but the system will not allow to online at the same time, because a bus only has one driver.

Data Collection and Further Training

The driver mobile application will be used to collect new data about daily ridership. The collected data will then be synchronized with the bus manager's web application and used to train the model further. The model will be retrained when enough data has been collected, every month or two.

Summary of Tools and Technologies Used

As mentioned in [14], the choice of tools used in the software development process can make or break a project. Hence, it is important to be aware of the types of tools that are available for use, and the benefits each can provide as well as the implications for using them. Given Smart Irenbu's requirements, the following were selected (Table 1 and Fig. 7).

Table 1. Selected tools and technologies

	Mobile application	Web application
Languages	Java	JavaScript
APIs	Firebase Storage	Firebase Storage
	Firebase Authentication	Firebase Authentication
	Firebase Database	Firebase Database
	Token Service API	Firebase Hosting
	Android Maps SDK	Token Service API
	Directions API	Maps JavaScript API
	Identity Toolkit API	Identity Toolkit API
Frameworks	Glide v4.8.0	Bootstrap v4.3.1
IDEs	Android Studio v3.4.2	Sublime Text v3.2.2
Learning Models		
Languages	Python, JavaScript	
IDEs	PyCharm, Google Collab	
Frameworks & Libraries	TensorFlow + Keras, Pandas, Seaborn, Matplotlib	

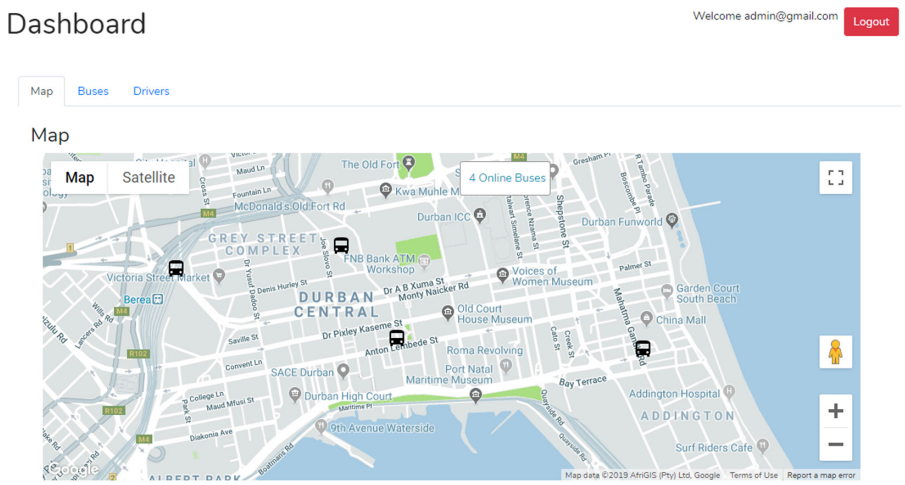

Fig. 7. Map tab

Multiple tools were used to assess the quality of both the Smart Irenbus mobile and web applications. Firebase BaaS provides several testing tools, and we used three of those viz. Firebase Test Lab which is a cloud-based app-testing environment that allows you to test an Android or iOS app across a wide variety of devices and device configurations, and see the results which also include logs. Firebase Performance Monitoring helps you to gain insight into the performance characteristics of mobile and

web applications. And Firebase Crashlytics, which is a crash reporter that enables you to track and fix stability issues that affect a mobile or web application's quality in real-time.

4 Machine Learning Integration

As mentioned earlier, the overall objective of this work is to demonstrate how a holistic real-time transport system can be assembled in the real world with technologies that are currently available to solve problems on the public transport sector within the South African context. Shown in Fig. 8 is how machine learning will be integrated into the system described in the previous sections through the use of an adaptive machine learning model. The model evolves as a result of being retrained periodically with both newly obtained and historical data collected through the Smart Irenbus mobile subsystem.

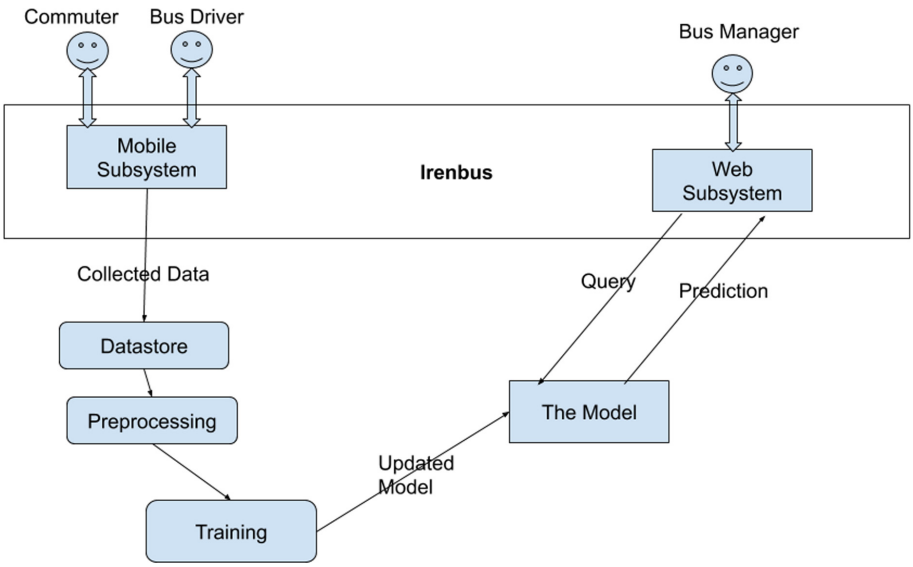

Fig. 8. Machine learning integration overview on the smart Irenbus system architecture

4.1 Dataset

The dataset for this study was obtained from the Chicago Data Portal, which is the City of Chicago's open data portal that lets you find city data and helps you find facts about the city's neighborhoods. Specifically, we used the Chicago Transit Authority's dataset [20], which is the operator of mass transit in Chicago, Illinois and some of its surrounding suburbs with a fleet of 1,879 buses and a 242.36 million annual bus ridership [21]. The dataset shows total daily ridership on a per-route basis dating back to 2001.

We filtered the dataset and only left relevant columns, namely boardings (daily), day type (working day or not), day of the month, day of week and route (number), which are self-explanatory. Shown in Fig. 9 is the distribution of data among these columns.

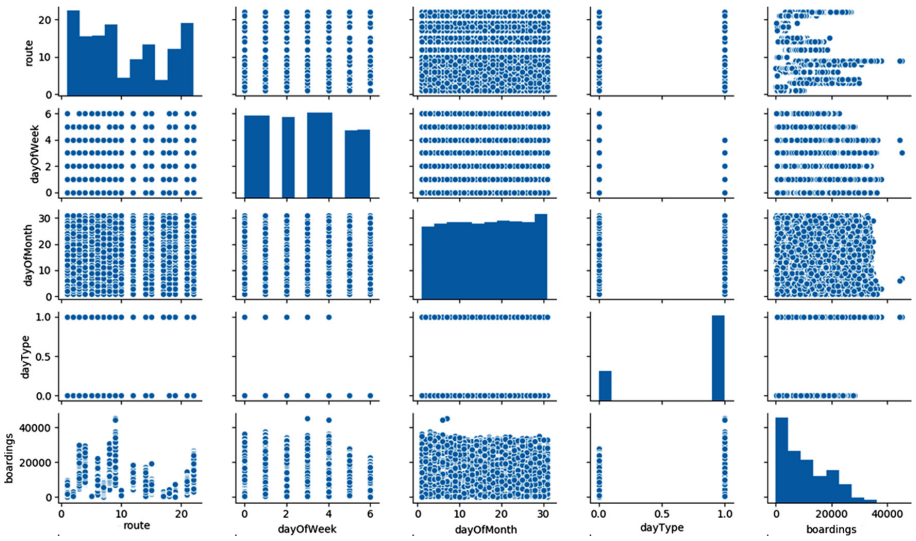

Fig. 9. The joint distribution of columns in the dataset

4.2 Machine Learning Model

Neural Networks can be trained to approximate virtually any nonlinear function to a required degree of accuracy [24]. The learning algorithm used in this study to adjust weights in the network is the Adaptive moment estimation (Adam) optimization algorithm, chosen because of its computational efficiency, little memory requirements and straightforward implementation.

A. Preprocessing

Because of the varying scale of values in the dataset in question, it is best practice to prepare the data before modelling it using a neural network model. Two methods were employed to rescale the attributes of the dataset viz, Normalization and Standardization. As mentioned in [22], unscaled input variables can result in slow or unstable learning, and unscaled target variables on regression problems can result in exploding gradients, causing the learning process to fail.

In normalization, the data is rescaled from the original range in such a way that all attributes have values within the range of 0 and 1. A normalized value of an attribute is calculated as follows: $y = \frac{x - min}{\max - \min}$.

Standardization is rescaling the distribution of values so that the mean of observed values is 0 with a standard deviation of 1. A standardized value is obtained using: $y = \frac{x-mean}{\text{standard_deviation}}$.

B. Model Architecture

The model was built using Keras Sequential model with 4 densely connected hidden layers and a single output layer that returns one continuous value.

- The model expects rows of data with 4 variables
- The first hidden layer has 50 nodes and uses the ReLU activation function.
- The second hidden layer has 100 nodes and uses the ReLU activation function.
- The second hidden layer has 50 nodes and uses the ReLU activation function.
- The output layer has one node and uses the linear activation function (Fig. 10).

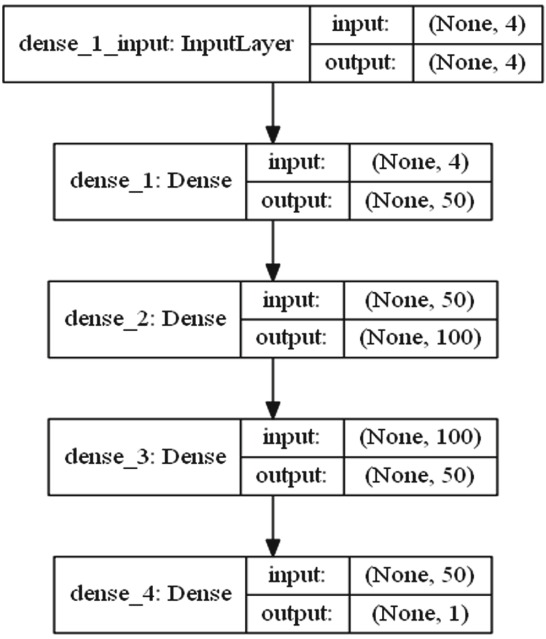

Fig. 10. Model Summary

C. Training

The training was done using three different alterations (standardized, normalized, and unscaled) of the dataset with 78 463 unique entries. This was done so we can observe the effect of preprocessing the dataset on the model's ability to learn. So ultimately there were three different resulting models. These models were trained concurrently on a Dell Personal Computer with 8 GB of Memory, Intel(R) Core (TM) i5-4310U CPU

@ 2.00 GHz, 2601 MHz, 2 Core(s), 4 Logical Processors. On average, each model took 2 h and 12 min to train. The training carried on for 500 epochs, which was found to be optimal. Each model used the Adam optimization algorithm, which automatically tunes itself to provide better results.

Validation loss is the same metric as training loss – it is calculated the same way, but it is not used to adjust the weights of the neural network. In neural networks, the validation set is typically used on every epoch, because training too long can cause overfitting, of which models do not recover from. And to prevent overfitting in a model the training curve in a loss graph should be similar to the validation curve (for most cases). The lower the loss, the better the model, unless the model has overfit the training data. But that will not be the case here, as the model is tested using unseen data.

To determine to the optimal number of epochs to be used in the training process, for each model we experimented with the number of epochs starting from 100 and increasing by 100 in each try while observing the amount of loss from each model. When we reached 400 the models trained with preprocessed dataset features showed some stability, we then increased the number of epochs by 100 again to make sure that the models were genuinely stable. As a result of the final training process (500 epochs), the loss calculated using the Mean Squared Error for each model was as shown in the Fig. 11.

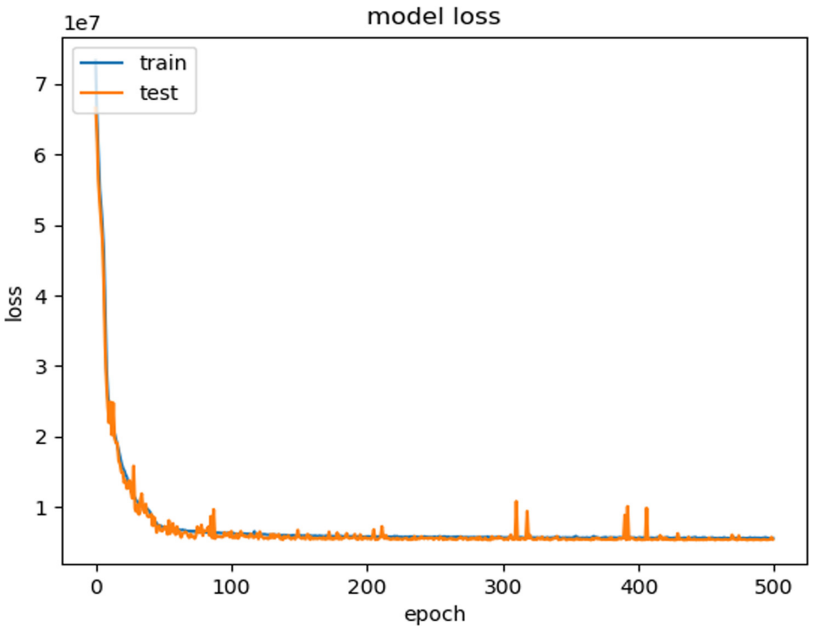

Fig. 11. MSE for a model trained with unscaled data

Training using raw dataset features results in a somewhat unstable model, as the model still has spikes of loss around the 400th epoch (Fig. 12).

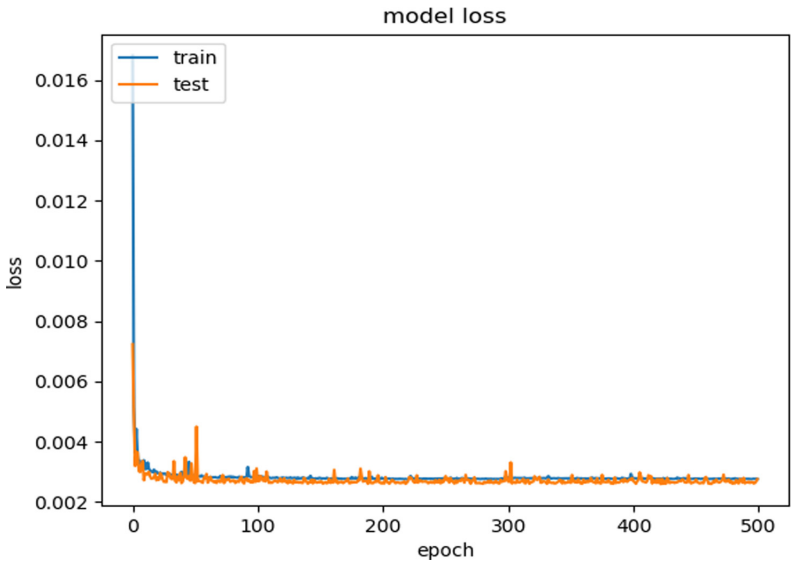

Fig. 12. MSE for a model trained with normalized data

Normalized dataset features significantly reduced the loss experienced when using raw features, as the last noticeable loss spike is around the 300th epoch (Fig. 13).

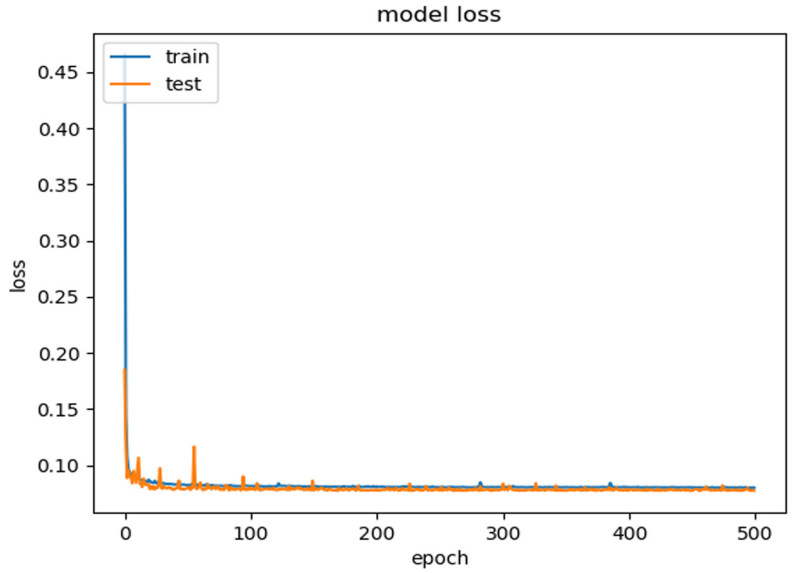

Fig. 13. MSE for a model trained with standardized data

With the standardized dataset features, the last noticeable model loss spike was around the 100th epoch, which is very good for a model trained on more than 70 000 instances and just 500 epochs. The standardization of the dataset features has dramatically improved the learning ability of the model in comparison with normalization. While training with raw dataset features shows greater loss compared to normalized dataset features, this then goes to show the importance of dataset preprocessing and how much it can affect the accuracy of a given model.

We can then save the model trained with standardized data as our best model. This model is then imported into the bus manager's web application using JavaScript and predicts the number of bus boardings that can be expected on a given day per route. This is an excellent deal for bus managers as they can now assign buses to routes based on demand (predicted by the model) and not stick to a static schedule. This will also help cut fuel costs and save the bus operator huge sums of money.

4.3 Discussion and Limitations

As the Smart Irenbus system mainly provides information based on the current user location, high accuracy is needed when tracking the device location. Android mainly offers two types of location permissions viz. `ACCESS_COARSE_LOCATION`, which report the device's current location with an accuracy equivalent to one city block, and `ACCESS_FINE_LOCATION`, which provides the highest possible location accuracy [13]. But the higher the accuracy, the higher the battery drain. This is as a result of the fact that the GPS receiver - a small chip and antennae located inside a smartphone - is always listening to cell towers, which give it a rough estimate of where the device is situated always geographically [17]. A lot of location tracking applications face the battery drainage problem, e.g. the popular cab-hailing app - Uber explains in [18] how they are attempting to solve this problem. The same issue will be faced by drivers when using the Smart Irenbus driver mobile application. But commuters with the Smart Irenbus commuter mobile application will not be affected as much since they will typically only use the app when they want to ride or track a bus. The Smart Irenbus mobile application only works on Android, for now, as the Android operating system makes up approximately 88 per cent of the mobile phone global market share [19].

5 Conclusion and Future Direction

With more than half of the world's population residing in urban areas, the need for reliable public transportation is more evident now than ever before. This solely depends on the availability of real-time information to commuters and useful in insights to bus operators. We developed an intelligent machine learning-based real-time transport system to address this. The system is composed of three subsystems to cater to passengers, drivers and bus managers. The new system mobile application is very lightweight, and after being subjected to several tests, it has proven to be efficient in terms of device resource consumption. And in comparison, with previous work done on this topic, this project took full advantage of cloud computing services while only requiring a smartphone to work - eliminating the need and costs of installing and maintaining

separate GPS trackers on the vehicles. While additionally collecting daily ridership data and using it to offer useful insights to bus managers.

The proposed system can be improved by the incorporation of other modes of public transportation other than buses; automating the assignment of drivers through feeding the model's predictions to a driver assignment subsystem; and by additionally catering for iOS devices. Also, there have been discussions with the eThekwini municipality stakeholders about a possible implementation of the proposed system.

Appendix

Smart Irenbus GitHub repository with source code including raw and processed training data, https://github.com/m3n2ie/Irenbus.

References

1. Statistics South Africa. Measuring household expenditure on public transport (2015). http://www.statssa.gov.za/?p=5943
2. Sungur, C., Sungur, I., Babaoglu, A.: Smart bus station-passenger information system. In: 2015 2nd International Conference on Information Science and Control Engineering (2015)
3. Google Developers. Get started — directions api. https://developers.google.com/maps/documentation/directions/start?hl=enUS
4. Monchambert, G., De Palma, A.: Public transport reliability and commuter strategy. J. Urban Econ. **81**, 14–29 (2014)
5. Shirisha, K., Sivaprasad, T.: Acquire bus information using GSM technology. Int. J. Adv. Technol. (2016)
6. Kumbhar, M., Survase, M., Mastud, P., Salunke, A., Sirdeshpande, S.: Real time web based bus tracking system. Int. Res. J. Eng. Technol. (IRJET) **3**(02), 632–635 (2016)
7. Moroney, L.: The firebase realtime database (2017)
8. Moroney, L.: Using authentication in firebase. In: The Definitive Guide to Firebase. Apress, Berkeley (2017)
9. UN DESA — United Nations Department of Economic and Social Affairs. 68 per cent of the world population projected to live in urban areas by 2050, says un, 2018. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html
10. Independent Communications Authority of South Africa. State of ICT sector in South Africa - 2019 report (2019). https://www.icasa.org.za/legislation-and-regulations/state-of-ict-sectorin-south-africa-2019-report
11. Lahudkar, P., Sawale, S., Deshmane, V., Bharambe, K.: NoSQL database-Google's firebase: a review. Int. J. Innov. Res. Sci. Eng. Technol. **7**(3), 243–250 (2018)
12. Manikandan, R., Niranjani, S.: Implementation on real time public transportation information using GSM query response system. Contemp. Eng. Sci. **7**(05), 2014 (2014)
13. Smaato. How to optimize your location data's accuracy (2019). https://www.smaato.com/blog/optimize-your-location-data-accuracy/
14. Smartdraw. Selecting the right software development tools for your developers. https://www.smartdraw.com/technology/right-softwaredevelopment-tools-for-developers.htm

15. WhiteSource Software. When to consider a NoSQL vs relational database. https://resources.whitesourcesoftware.com/blogwhitesource/when-to-consider-a-nosql-vs-relational-database
16. StackShare. Amazon EC2 vs firebase — what are the differences? https://stackshare.io/stackups/amazon-ec2-vs-firebase
17. The Verge. Why GPS-dependent apps deplete your smartphone battery (2018). https://www.theverge.com/2018/8/17/17630872/smartphonebattery-gps-location-services
18. Hameed, S., Chida, J.: Mastering Android Wear Application Development. Packt Publishing Ltd (2016). https://eng.uber.com/activity-service-dependency-android-architecture/
19. The Statistics Portal. Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018. Statistics Portal (2018). https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems
20. CTA - Ridership - Bus Routes - Daily Totals by Route. https://data.cityofchicago.org/Transportation/CTA-Ridership-Bus-Routes-Daily-Totals-by-Route/jyb9-n7fm
21. Annual Ridership Report: Calendar Year 2018 (PDF). Transitchicago.com. https://www.transitchicago.com/assets/1/6/2018_Annual_Report_-_v3_04.03.2019.pdf
22. Machine Learning Mastery https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/
23. Bacciu, D., Carta, A., Gnesi, S., Semini, L.: An experience in using machine learning for short-term predictions in smart transportation systems. J. Log. Algebraic Methods Program. **1**(87), 52–66 (2017)
24. Agarwal, P.K., Gurjar, J., Agarwal, A.K., Birla, R.: Application of artificial intelligence for development of intelligent transport system in smart cities. J. Traffic Transp. Eng. **1**(1), 20–30 (2015)
25. Othman, M.S., Tan, G.: Machine learning aided simulation of public transport utilization. In: 2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), 15 October 2018, pp. 1–2. IEEE (2018)
26. Toqué, F., Khouadjia, M., Come, E., Trepanier, M., Oukhellou, L.: Short & long term forecasting of multimodal transport passenger flows with machine learning methods. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 16 October 2017, pp. 560–566. IEEE (2017)
27. Liu, G., Yin, Z., Jia, Y., Xie, Y.: Passenger flow estimation based on convolutional neural network in public transportation system. Knowl.-Based Syst. **123**, 102–115 (2017)
28. Van Oort, N., Brands, T., de Romph, E.: Short term ridership prediction in public transport by processing smart card data. Transp. Res. Rec. (2015)
29. Skhosana, M., Ezugwu, A.E.: Irenbus: A real-time public transport management system. In: 2020 Conference on Information Communications Technology and Society (ICTAS), 11 March 2020, pp. 1–7. IEEE (2020). https://doi.org/10.1109/ictas47918.2020.234000