Real Time Predictive Monitoring System for Urban Transport

Nauman Ahmad Khan

PhD

March 2017

Real Time Predictive Monitoring System for Urban Transport

A dissertation submitted by

Nauman Ahmad Khan

In fulfilment of the requirements for the Doctor of Philosophy degree in Computer Science

Faculty of Science, Engineering and Computing,
Kingston University London, United Kingdom

March 2017

Supervisory Team: Professor Vesna Brujic-Okretic, Professor Souheil Khaddaj

Acknowledgements

I would like to express my special appreciation and thanks to my supervisors Professor Vesna Brujic-Okretic and Professor Souheil Khaddaj; you have both been wonderful mentors for me. I would like to thank you for encouraging my research and for always being helpful. Your advice on both research, as well as on my career, have been priceless. A warm thank you note goes to Mads Hansen, mermaid technology, who supported me endlessly in this research and provided me all the resources and data I needed to complete this research.

My most valuable appreciation and thanks go to my late father, Muhammad Sharif Khan, and my mother, Hameeda Khanam, who always dreamed big for me and supported me beyond their resources and energies to accomplish the best in Education and Career. You were, are and will always be my inspiration to excel in life. I wish my father was with me today to see my accomplishment, but I know his blessings will always be with me. I thank my dearest sister and my brothers (Dr. Muhammad Khan, Mubashar Khan, Mudassar Khan) for standing by me and for being a support without having to ask for it. Especially my sister, Moazma Khan, who has always offered unconditional friendship and mentoring all these years.

I am blessed to have friends like Muhammad Shakeel, Muhammad Zubair, Khurram Imtiaz, Madiha Shafqat, and Sandeela Waseem, who have helped me throughout my office responsibilities and studies, and were always welcoming to hear my to irrational ideas and plans about work and offer their valuable opinion when I needed.

Completing this research would not have been possible, if I hadn’t received tremendous help and support from my wife, Guljana Syed, who not only sacrificed her time, but also offered extensive patience and dedication to bring up the children remarkably. My adorable daughter Khadeejah Khan, my sons Habib Ullah Khan and Abdullah Khan have been so patient during the last four years, that I can’t be thankful enough for their cooperation and sacrifices for their father’s time. I am also thankful to my nephew Muhammad Ibrahim and all of the children in the family – thank you for your love and prayers.

TABLE OF CONTENTS

List of Figures

List of Tables

ABSRTACT

Ubiquitous access to mobile and internet technology has influenced a significant increase in the amount of data produced, communicated and stored by corporations as well as by individual users, in recent years. The research presented in this thesis proposes an architectural framework to acquire, store, manipulate and integrate data and information within an urban transport environment, to optimise its operations in real-time. The deployed architecture is based on the integration of a number of technologies and tailor-made algorithms implemented to provide a management tool to aid traffic monitoring, using intelligent decision-making processes. A creative combination of Data Mining techniques and Machine Learning algorithms was used to implement predictive analytics, as a key component in the process of addressing challenges in monitoring and managing an urban transport network operation in real-time. The proposed solution has then been applied to an actual urban transport management system, within a partner company, Mermaid Technology, Copenhagen to test and evaluate the proposed algorithms and the architectural integration principles used. Various visualization methods have been employed, at numerous stages of the project to dynamically interpret the large volume and diversity of data to effectively aid the monitoring and decision-making process. The deliverables on this project include: the system architecture design, as well as software solutions, which facilitate predictive analytics and effective visualisation strategies to aid real-time monitoring of a large system, in the context of urban transport. The proposed solutions have been implemented, tested and evaluated in a Case Study in collaboration with Mermaid Technology. Using live data from their network operations, it has aided in evaluating the efficiency of the proposed system.

CHAPER 1: INTRODUCTION

1.1 Motivation

The industrial revolution focused on the development of technology to create efficient vehicles, such as steam powered trains and internal combustion cars. The rapid growth of technology and its integration with the huge road, rail and water infrastructure helped in making efficient and mobile transport services. The development of efficient transport services went through an evolutionary process of adopting environmental and safety considerations in the policy design. The technology innovation hubs worked on developing devices that could help collect information from the vehicles to identify the impact of growth in the transport industry on the environment. With the creation of computer and information technology, it allowed the innovative inventions to integrate immensely by making the vehicles information-aware, allowing the vehicles to share this information with external systems. This awareness and decision making based on shared information complemented the efficiency objectives of: safety, information technology integration, adaptive intelligence, energy consumption and environmental impact. Public transport serves to increase people's mobility, and it is a key part of the futuristic concept of smart cities and a connected digital ecosystem. Safer and better transport services can be ensured to provide reliable mobility for commuters, with the development and use of data-driven intelligence embedded in Intelligent Transport Systems (ITS).

ITSs cover a wider domain of transport services including: public transport, commercial vehicles for freight and logistics, private cars and car services, such as taxi. For example, *Uber* and *Lyft*, record useful data about their operation, environment, traffic and road situation and density of other vehicles in a geographical location. Many other car-hailing services support the idea of sharing private vehicles, so that fewer vehicles are used in commuting. These services can be facilitated if data is recorded from these vehicles and then shared through integrated platforms, so that different transport services can be designed. This data contains information about current position of each vehicle; the area or zone in which the vehicle is traveling, and the destination. This comprehensive information about vehicles, and the environment it is interacting with, defines its context, allowing us to create a definition of the vehicle as an information entity. Consequently, from the extensive growth in technology, the amount and volume of information being generated every day is unprecedented, and it is increasing further with the advancements in internet technology, the surge in the consumption of smart devices, the availability of high bandwidth, and the exponential growth of social media phenomenon [38], [39], [40]. Voluminous amounts of uncertain sensor data are also coming from sensors and smart devices installed in the infrastructure, which is used in intelligent transport management systems [3]. This variety of data sources opens new opportunities for testing the validity of information through multiple sources, resulting in self-adapting transport services [4]. Effective analysis of these patterns can help in improving the services like public transport systems [5], [6].

There are many legacy systems where data is collected, but never used, and transport systems are no exception. The only information that is generally shared with external agencies is the location of the vehicles. This results in a limited support from external agencies and systems because of the lack of information available to help in decision making. The growth of technology demands that the public transport system should either implement the data analytics platforms themselves or integrate with external systems so that the vehicles are more information-aware, which can help in making the service reliable and dependable. Developing architectures that can provide a basis for intelligent information processing and generation of efficient and effective decision parameters, is critical for efficient and reliable public transport. Although there are many high-level architectures that exist in different parts of the world, the operational situation of different routes on which bus operators providing essential services advocate for specific implementation architectures. These architectures should reflect the inclusion of data from the system, it is being applied to, for a potentially accurate realization of the problems and suggesting solutions. In this thesis, we are proposing an intelligent architecture that provides an implementation blueprint for developing small-to-large scale public transport systems, where the buses are information aware and the sharing of information is done in real-time. The complete life cycle of the data is discussed and predictive analytical techniques are applied to generate short-term and long-term predictions for accurate forecasting of the bus arrivals. The architecture can be easily integrated with existing systems and provide on-the-fly analysis in the form of useful visualizations to facilitate productive decision making.

1.2 Background

The convergence of communication and computing for smart consumer devices is on the evolutionary course to bring interoperability and leverage the services and functions from every industry [13], [14]. The advancement in smart devices has resulted in creating more sensor solutions that can give information about not only the device but also its operating environment. Installation of these mobility sensors on vehicles has increased the amount of information available about the status of the vehicles, their location and the environment in which they are being driven. Smart devices provide integrated services for communication, computing and mobile sectors, including: voice communication, messaging, personal information management (PIM) applications and wireless communication capability [15], [16], [17]. The development of sensors has helped building many smart devices that can be used on moving objects like vehicles and help in the collection of useful information. These smart devices are equipped with the capabilities to assist in navigation, built-in camera, audio/video playback and recording, record temperature and light, and much more [18]. Their prevalence is indicative from the fact the Smart Devices were initially intended for businesses, but have now become a common choice for consumers [19]. The increase in the use of smart devices by consumers and businesses, such as bus companies, generates demand for the development of intelligent transport systems where consumers can access information about the traveling preferences they have. It is expected that the smartphone adoption will continue a fast-paced trajectory through 2017 [20], [22] that pushes the bus companies to develop systems that integrate sensors from the buses and provide interfaces for mobile users. All of which can be used for efficient and reliable information sharing.

As mentioned earlier, the information about an entity characterizes the situation of an entity (e.g. vehicle, passenger, driver, route, weather) is its context [13], [23]. Context-aware systems use information communication technology to provide a greater awareness of relevant information about the physical worlds to assist the information recipient in the decision-making process [24]. This information is transmitted using many different file protocols like XML, JSON and raw format that is schema-free [48], [49]. The context of these systems delivers relevant information to stakeholders and other systems users [25], [26], [27]. The relevance of the information is relative to a specific circumstance or context. Context-awareness can be used in route planning to resolve conflicts between different vehicles for accurate selection of the route to travel [28], [29]. Context-awareness can also be used in suggesting better routes if there is congestion and presentation of alternate routes based on commuter's preferences [30], [31]. Context-awareness can help the vehicles find nearest locations for emergency help, garages and fuelling en-route [32], [33]. Context awareness systems have become a growing area of study for pervasive and ubiquitous research communities [34], [35]. Unfortunately, smart transport system research is too often conducted without interactions with the pervasive/ubiquitous research communities [36].

The intelligence provided in context-aware systems like public transport system is implemented through channelling the contextual data being collected from the transport infrastructure into prediction systems so that behaviour of the transport system can be forecasted. The predictions, projections and surveys [37-40] envisage a highly efficient information management architecture that can handle that unprecedented volume of data. Significant efforts have been made in last few years to contribute in context-aware applications regarding collaborative workspaces [41], context-aware computing [42], [43] and context-aware social computing [44]. The ITSs are not only context-aware systems, but they need the collected data to be processed in real-time because the vehicles are moving objects and so information from sensors are continuously changing. This adds an element of complexity since the information is to be processed in real time to reach its end audience, the commuters [21]. If the information that is being presented to the user is not valid anymore, then there is no relevance of giving that information to the user [45], [46], [47] regardless of how good or relevant it is in sync with the context requirements of the user. The real-time information management system coupled with context handling [55] should have the ability to quickly process the information. Processed results should be delivered to target applications and users efficiently. In future-smart applications like intelligent transport systems, the context awareness is considered a central aspect to perform tasks based on the ambient context conditions. The context-aware systems such as intelligent transport systems can assist both transport companies and commuters in building reliable commuting opportunities [50], [51]. To some researchers, context awareness and what is supposed to be considered part of the context awareness is an open discussion [17], [18], [41].

However, handling of data coming from variety of sensors becomes a challenge because of the diverse structure and data attributes from different sources should be joined to establish relationships between them [56]. The studies on architecture, privacy issues, data handling platforms and data acquisition from smart devices present a case for data-driven decision making [2], [3], [4], [13], [15]. There is a need to process the data collected from smart devices so that detailed analytics are available, which are necessary for making data-driven decisions. Effective data analytics exposes in-depth information about the data; patterns can be deduced that reflect behaviour of the system, thus producing data which facilitates the integration of these patterns with visualisation systems for strategic decision-making. Data Analytics is the use of advanced techniques, mostly data mining and statistical, to find (hidden) patterns in data [6], [7]. A significant number of these techniques relies on commercial tools such as relational DBMS, data warehousing, ETL, OLAP, and business analytics tools. Top ten data mining algorithms identified, based on the number of times they are cited [8], survey results and expert opinion are C4.5, K-means, SVM (support vector machine), Apriori, EM (expectation maximization), PageRank, AdaBoost, KNN (K-nearest neighbors), Naïve Bayes, and CART. They cover classification, clustering, regression, association analysis, and network analysis [170], [171], [172]. The data mining techniques can help in detection of patterns in the data and provision of prediction and forecasting based monitoring platform that can facilitate acquisition, storage and processing of collected data for strategic decision making [9], [10]. These techniques, combined with data produced from smart sensors on-board vehicles, can automate the process of dynamic intelligence development in the transport system of future.

1.3 Research Questions

The research questions that emerge from the issues laid out in the Introduction section center around the following problem areas:

Data acquisition and processing in real-time using smart sensor devices and a dedicated system architecture to store, manage and dynamically display information, with application in the transport domain.

Data mining, using intelligent data mining techniques and the pool of information dynamically acquired from the urban transport environment – in an automated fashion and in real time. This includes non-direct inputs from the vehicles, such as buses and trends obtained from journey behaviors that can be captured by the system sensors.

Information integration and prediction in real time – selecting automatically relevant data and information from various parts of the system and from the sensors, such as GPS, accelerometer – and integrating it with the data coming from external travelling data providers with an instant, dynamically compiled short-term and long-term arrival time prediction.

ITS architecture – presenting a new ITS architecture that can help in the implementation of large-scale intelligent transport systems, as well as small customised ITS, with emphasis on quality of service attributes and the provision of reliable and scalable prediction service.

Predictive monitoring – implementation of a comprehensive real-time predictive monitoring system integrated with detailed visualisation of current and projected situation of the public transport system, applicable to buses, trains and trams.

1.4Aims and Objectives

The aim of the project is to design and implement a novel system architecture to integrate and visualise data and information in real-time on a predictive monitoring system, particularly in the context of urban vehicle navigation and transport. This would require effective data acquisition so that real-time predictive analytics can be achieved. The novelty lies in the intelligent retrieval of information using context parameters of buses and profiling of completed journeys, and then a seamless integration and display of data from a variety of different sources. The data sources include sensors, historical (archived) data, outputs of real-time, data analytics and others. The used methods will include context awareness and contemporary methods of storing and managing data, particularly data mining techniques, and visualisation strategies for monitoring.

To achieve these aims, the following set of objectives have been set up:

- Analyze and explore the existing system architectures that deal with intelligent transport systems to acquire context data and with limited data integration, in real-time, to establish state-of-the-art technology.
- Research into methodologies pertinent to the research problem, including: data mining techniques and predictive analytics, all in the context of the defined research problems.

- Review state-of-the-art technologies used for managing and storing data with a view to using an appropriate selection in the context of this project.
- Design a system architecture that will enable data acquisition, manipulation and intelligent retrieval, and display of information/data in real-time; the design output is envisaged as a generic tool applicable in a variety of application areas, but focusing the case study on the urban transport and navigation scenario.
- Implement the design, in conjunction with a targeted industrial partner, building on an existing information system in the context of urban transport.
- Evaluate the design and implementation using the industrial partners' infrastructure, eight months of data history and data being collected in real-time.
- Compare the prediction results with existing prediction system being used by the industrial partner and critically reflect upon the accuracy of new prediction system.
- Implement a real-time predictive monitoring system using state-of-the-art maps and analytics based visualization strategies.
- Draw conclusions and suggest directions for future work.

1.5 Contribution

There are a set of contributions made and they have been categorized as primary and secondary contributions as below.

1.5.1 Primary Contributions

1- Design implementation and evaluation of a self-adaptive intelligent transport system architecture equipped with real-time predictive monitoring engine for next generation transport systems.

2- Adaptive architectural framework for intelligent transport systems with emphasis on buses

3- Automated data acquisition framework for contextual data coming from smart sensors and transformation of this data for predictive analytics.

4- Flexible data mining application framework for a real-time predictive monitoring system for urban transport.

1.5.2 Secondary Contributions

1- Comprehensive visualisation strategies for urban transport systems combining the historical and real-time data streams for dynamically updating displays.

2- Off-line prediction system for vehicles.

3- Two Papers published on Intelligent Environments and Intelligent Architectures and Frameworks.

1.6Thesis Outline

This chapter presents an introduction to the topic and the motivation behind the work. The rest of the thesis is structured as follows:

Chapter 2 outlines the key elements of intelligent transport systems including context derivation from sensors data, different representation of the context and algorithms, techniques to acquire and to process the data from contextual sensors. It also presents a comprehensive review of intelligent transport systems with special emphasis on the supporting architectures and infrastructures. In addition, the chapter highlights the research on how transport systems have evolved over time taking into account changing preferences of commuters. For example, choosing public transport over private vehicles due to environmental concerns. Chapter 2 starts with an in-depth insight into the context definition and its relevance to the system being implemented using the context data. Different applications of the context are discussed, followed by techniques and algorithms for handling context data lifecycle. Next, several middleware architectures are presented which can be used to handle the context of sensors data, like the one coming from buses in a public transport system. Then, a comprehensive discussion on Intelligent Transport System Architectures (ITS) outlines the different type of ITS that exist globally and evaluates their advantages against each other in detail.

Chapter 3 discusses the challenges faced in handling large volumes of data being collected from different sensors and reviews data mining algorithms that can be used for contextual data processing. It describes how data mining algorithms can help in processing the context data and the discovery of patterns that are helpful for designing prediction frameworks. The ability of the data mining algorithms is highlighted in producing prediction data for arrival time of buses on stops based on forecasting patterns identified in the context data coming from buses. This chapter further reviews some existing arrival time prediction systems using data mining algorithms and provides a critical analysis of their effectiveness.

Chapter 4 emphasizes the development of a new architecture for intelligent transport systems based on the reviewed ITS architectures. The justification and evolution of a new ITS architecture is elaborated on and its impact on performance of ITS is discussed, using quality of service attributes. The quality of service attributes such as scalability, sustainability and performance in real-time are related to high performing systems and their relevance to ITS are expanded on. This chapter presents the evolution of the new proposed system architecture and how it can solve the challenges being faced by transport services. It also elaborates on the deployment considerations for the proposed architecture and importance of data flows in an ITS for a public transport system, especially bus management systems.

Chapter 5 provides details of the implementation of the proposed system architecture presented in chapter 4. The novel architecture solution is applied to an existing and operational transport system, run by the partner organization on this project, Mermaid Technology, using a subset of their transport data for buses. It aims to improve the reliability and performance of the transport system, in real-time, by applying predictive analysis methods. The implemented system is evaluated for its performance and compared with the existing system for its ability to address challenges pointed out in previous chapters. This chapter starts with an insight into the existing data structure of the Mermaid Technology system. The data, in its current form, is visualized to enable comparisons to be made between the performance of the existing system and the proposed one. Various data mining algorithms have been applied to experiment with the system to obtain an optimal reliability and performance.

Chapter 6 presents a detailed case study and full automation of a prediction and monitoring system that uses state-of-the-art technologies to implement each stage of the automation process. It starts with adaptive data acquisition with capacity to accommodate data coming in real-time and the process to transform this data into algorithm-ready format. An auto-learning process of applying data mining techniques is elaborated on so that it is continuously updating prediction probabilities that are applied for prediction of bus arrivals. A detailed evaluation of prediction is conducted with a comparison to an existing prediction engine. It also presents and elaborates on the monitoring platform that showcases detailed visualizations of the transport system that are populated and updated with latest data as it arrives.

Chapter 7 concludes the thesis by specifying the main contributions of the work. It gives a summary of the work done as well as its evaluation results. It also outlines possible research areas that can be carried out in future work and identifies some guidelines and directions of future intelligent transport management research.

CHAPER 2: LITERATURE REVIEW: INTELLIGENT TRANSPORT SYSTEMS

2.1 Introduction

This chapter outlines the key elements of intelligent transport systems, including: context derivation from sensors data; different representation of the context and algorithms; techniques to acquire, and to process the data from contextual sensors. It also presents a comprehensive review of intelligent transport systems with special emphasis on the supporting architectures and infrastructures. This chapter reviews significant intelligent transport systems and ITS standards from the literature that make basis of how the transport systems are designed and what actors are important to consider in improving services of a transport system. In addition, the chapter highlights the research on how transport systems have evolved over time taking into account changing preferences of commuters. For example, choosing public transport over private vehicles due to environmental concerns [153]. Commuters would make choices to travel on public transport, such as buses, based on their income and car ownership; availability and convenience of public transport operating in their areas. Therefore, the challenge faced by transport authorities is to create a balance between the use of private cars and public transport [156]. The increasing problems of congestion and the environmental impact of using private cars, make it essential for transport such as buses to become available and reliable. This awareness of potential transport problems helped bus companies to develop the context of the buses in the form of sensors on-board, so that information about the bus and its operation is available, which can be reviewed for performance and reliability and can be used to improve its working, as well as integration with other external systems.

The above factors started defining the context (choice factors) of the people and would help the transport authorities to identify the demand and supply of transport services in an area. Moreover, the management of public transport, including buses, requires the ability to plan adequately arrival and departure times as part of overall transport management. Before the development and installation of sensors, the latest smart equipment, and proliferation of the smart personal devices on buses, buses had no way of communicating to people waiting on stops of their arrival times other than the schedule print displayed on the stops. The information about buses such as location, the direction they are heading, the speed they are traveling on and their ability to meet the timetable are the parameters that define the context of the buses, because they are directly linked with the context of the travelers. Thus, transport systems have gradually enhanced their context development abilities, so that information about them is available and reliable.

Earlier studies discussed the problems that public transport systems faced in a pre-sensor area with growing population and demand for increased infrastructure [153], [154]. With no infrastructure for buses to be able to communicate to each other and to the management office, the efficiency and effectiveness of the use of buses had no understandable evidence. Installing buses with technology is one of the key innovations that can bring the urban transport system in integration with larger systems to help build sustainable smart cities [155]. This integration is needed because until public transport systems, especially in urban settings, can share information about the vehicles and their operations with other external systems, a common platform cannot be built to achieve a sustainable transport infrastructure.

This chapter starts with an in-depth insight into the context definition and its relevance to the system being implemented using the context data. Different applications of the context are discussed, followed by techniques and algorithms for handling context data lifecycle. The GPS location of a bus is a key part of the context, particularly in the identification of bus location and its relative and physical distance to the immediate destination (next stop) and destination (where the journey ends). Next, many middleware architectures are presented which can be used to handle the context of sensors data, like the one coming from buses in a public transport system. This is followed by a comprehensive discussion on Intelligent Transport System Architectures (ITS) which outlines the different type of ITS that exist globally and evaluates their advantages against each other in detail. The section also elaborates on how different ITS systems vary from one continent to other based on constraints applied by local transport conditions. Finally, the chapter concludes with a critical analysis and summary.

2.2 Smart Sensors – A context perspective

This section presents a review of how context has grown over a period of time and its importance in systems where data from sensors plays an important role. The use of smart devices has increased dramatically in the last few years and is now covering many applications and services such as banking, healthcare and transport services, to name but a few. [32], [41]. The advancements of internet (including high bandwidths, lower prices, and increase in mediums of connectivity from wired connections to wireless, satellites) have helped to develop highly distributed systems which also applies to public transport domain. The role of contextual sensors and users have become important in terms of how data is retrieved, processed and then stored because smart devices' users are increasing in number, resulting in a variety of context data types. There are many definitions for context that exist in the literature. The Future of Context-Aware Computing [20], Context Aware Search Augmentation [173] define context awareness as the properties of a device that help in between different devices and between devices and the sorrounding environment. The device can be any entity like smart devices on-board vehicles, WIFI and wireless devices, satellite, another mobile, the internet, servers, and any other device that may exist for collaboration with connected services or can use the sensors to collect and supply information. Buses, being moving vehicles, can also take advantage of these technological developments and integrate these sensors to constitute its context. Having more and more information available, and with smart devices enabling users to ask for information from many different places and situations, has resulted in new research fields being introduced like ubiquitous computing and context aware computing [20]. Context-aware computing started developing around 1990's when smart computing devices were introduced [59]. A simple model of context acquisition is presented in Figure 2.1.

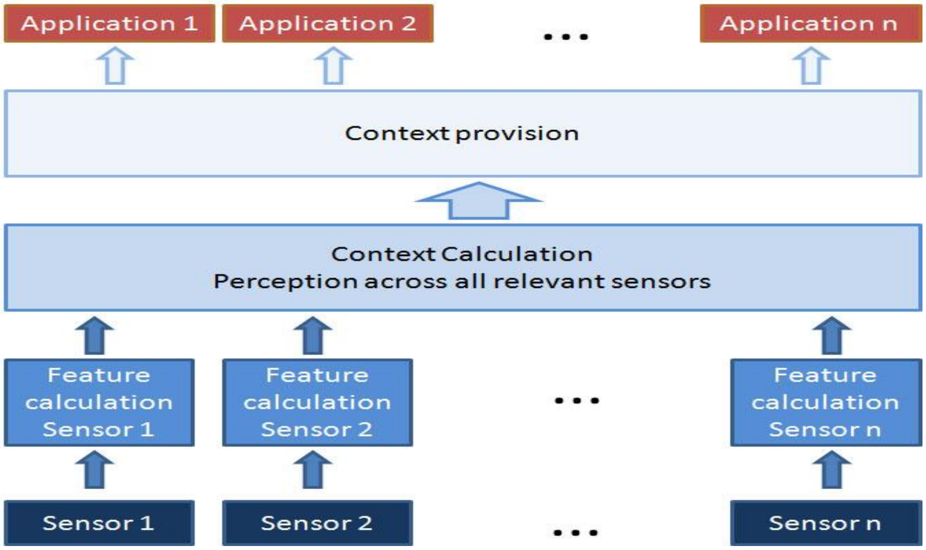

Application 1
Application 2
...
Application n
Context provision
Context Calculation
Perception across all relevant sensors
Feature calculation Sensor 1
Feature calculation Sensor 2
...
Feature calculation Sensor n
Sensor 1
Sensor 2
...
Sensor n

Figure 2-1: Context acquisition, reasoning and action handling [20]

However, there is a need to standardize the interface between smart devices and external services. Recent studies on context monitoring platforms suggest that there is no standard approach to designing the model of the smart device context. Consequently, many of the studies have offered different methods for context modeling [1], [49], [52]. Having different context models will make it comparatively difficult for a common model that can work as an interface for the flow of context information between devices as well as services. Therefore, there is a need to define and design an architecture to standardize the integration of smart device context, which indirectly defines user context, with the process of data acquisition, data storage, data manipulation, and dispensing the data to target users and devices. Such an architecture is also needed to study if the widening scope of context parameters attached with a device are helping to make things better when it comes to data exchange; or, adding unnecessary complications.

The representation and integration of context in future systems of public transport highlights the connectivity and collaboration challenges because each type of vehicle will produce a different set of data. Thus, processing of the data for collaboration between the different types will need very sophisticated public transport systems. A representation of context data collection and coordination between vehicles framework from US department of Transportation is shown in Figure 2.2.

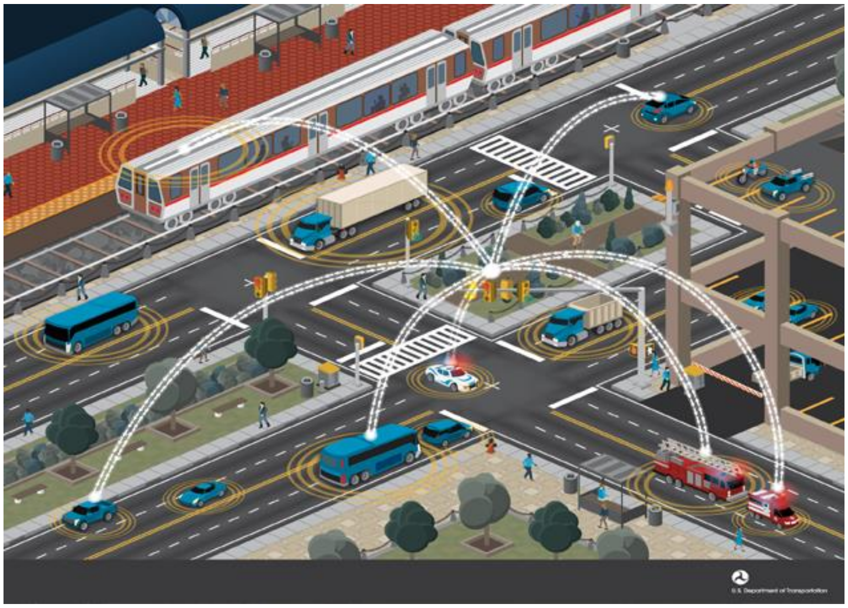


Figure 2-2: Context Data Collection and Coordination (US Department of Transport)

2.2.1 Use/History of Sensor based Context Development

This section presents literature review of different techniques and strategies used to process different stages of the data retrieved from the context devices. According to Dey and Abowd [53], Context is any information that can be used to characterize the situation of an entity [54, 55]. It is important for entities that come in interaction with other entities or the objects, in the environment with which they interact, to provide or consume some services to/from each other. The concept of entity can be associated with a person or place, or an external device which will identify a set of information that they can share with each other about their environment and work. According to Dey, this sharing of information and the objective relevance of this information for all the entities is what makes it context aware. The researchers noted in [56], [57], [58] that time, place, events, the persons and devices are different entities that produce information for context-aware computing to provide services that can handle entity's specific requests. A discussion about the process life cycle of converting context information from context data acquisition to reasoning is presented in the following sections.

2.2.2 Context data life cycle

The context of any entity including a transport vehicle like bus is collected through internal and external sensors installed that continuously generate data about dynamic context. The nature of the context is dynamic because different sensors like location, direction, speed, orientation, temperature, light and presence of other objects around keep changing. It is more evident on a bus because it interacts with many external entities: passengers, roads, other vehicles, and time and space. All this information is valid for only one minute for buses when they are completing their scheduled trips. This one-minute validity is because the buses are moving and their location, orientation and other trip related context information, changes on the go and their arrival/departure times are updated every minute. It means lifecycle of the data collected from different sensors can be different and can produce different results when combined. A formal context life cycle should be planned for effective integration of the context data, so that each data element captured and collected from different sensors that make the context of the vehicle, can be processed. This will allow different sources to be combined to give an expression of the context. This flow of contextual data from generation to processing is being presented as steps of a data life cycle, which are presented in the following section.

2.2.2.1 Context classification

Different types of sensors generate different types of data, and they can have different structures respectively. To be able to combine the data from different sensors to generate combined context, the data should be categorized, so that the application designers can integrate the data for extraction of relevant aspects of data and how different categories are related to each other. A categorization approach was proposed by (Figueiredo, Jesus et al), which considers the location, identity, activity and time of a context to differentiate between different context types [59]. Such an approach answers the questions of where, who, what and when and allows other sources of information from within the context to be indicated. The study elaborated on more specific examples based on the location of the object. The location of the object can provide information about: its physical presence in an environment; other objects and entities in the environment; different events or incidents that potentially are happening in that environment, and their impact on the object itself.

In another contexts, identification for a user who is trying to login to a system, a social media application can extract secondary context from just the login information that belongs to the person's friends or contact information that is not being provided in the login process, but can easily be derived based on it. Similarly, a vehicle has a primary context and a secondary context where the primary context may represent the route it is driving. The secondary context would indicate what other vehicles are driving along the same route, and would determine whether anyone of them were causing any delays for the whole route. So, classification of different types of context data can provide details of possible interactions, as well as the impact of those interactions on other objects in the environment.

2.2.2.2 Context Acquisition

With a variety of sensors available that constitute the context of a device or application, acquisition of data from these sensors is an important design and development decision [60], and usual source of this context information are the sensors themselves [61]. As mentioned in the context classification section, the context can be either the primary/ direct context that it is collected from. Devices that look at: location, orientation, temperature, and light; or, it can be a secondary/ derived context that is extracted from the primary context [59]. The type and category of each sensor will require a specific method to acquire the data. Numerous contextual data acquisition strategies have been proposed [61] which are discussed below:

Direct sensor access: The contextual sensors are usually hardware devices that have a software layer to read data from them. This approach suggests using software directly accessing sensor information without the use of any middleware or processing of that information. Each hardware device that supports software interfaces comes with driver software that can be used by the application designers to connect to these devices and collect the data. Once the sensor data is retrieved, it needs to be published to the central server that processes this data for extracting patterns and relationships between different data points.

Middleware infrastructure: This is based on the concept of encapsulation which is used to separate business logic and the user interface. In this approach, low-level sensing details are hidden from the application. This is usually achieved either through third party software development kits (SDKs) that help in the collection of the data; or, a custom-written component for a context-aware application to apply specific transformation before data is provided to the application itself. In doing this, it helps with building extendable applications because the code to fetch the data does not change even if the driver software for a specific hardware sensor device is changed. It also makes reusing hardware much simpler, unlike direct sensor access [62]. The buses are equipped with both direct and middleware infrastructures to acquire the data from installed sensors.

Context server: This is a hybrid approach where middleware-based architecture is extended by using distributed context acquisition systems. It is distributed implementation of middleware architecture where different data acquisition layers are integrated to provide access to the context across different platforms. This integration involves the utilization of an access managing remote component. To access the context information, a cloud type access system allows multiple subscribers.

2.2.2.3 Context Modelling

A model represents the relationship between different attributes of the data collected from a sensor. It also highlights the interaction relationship between data coming from different sensors. An example is that a bus's location information is related to direction sensor to conclude its progress on the road and know about the environment around it. The context data modeling is needed for establishment and gathering of the context information that is produced by sensors. Tare a number of different context modeling techniques available, which are outlined below:

Key-Value models: This is the simplest model to represent context information as key-value pairs. Key-value models show the minimum key data structure in context modeling. They are often employed in a wide range of service frameworks, where simple values define system attributes and service values. These may include: the name, time and location. Although this type of model is easy to use, it does not allow for advanced structuring characterized by a retrieval algorithm for an efficient context.

Mark-up schemes models: Hierarchical data structure are best for these models because they represent an organization of the attributes where each child attribute belongs to its parent attribute as well. For example, a location attribute of a bus-stop, along with name and bus lines visiting that stop, can define the context for that stop.

Graphical models: Unified Modelling Language (UML) is an example of powerful graphical modeling technique that helps in modeling generic systems. It can be used to model different types of context and can also assist in modeling the relationships between different context entities and their attributes. Specifically, this approach to modeling is suited for creating an Entity Relationship (ER) model, which clearly illustrates a relational schema in a system that is structured around an architecture of context management. The sequence diagrams can help model the life cycle of context data.

Object oriented models: This approach to modeling considers a context that utilizes the full extent of object orientation, which includes reusability, encapsulation, and inheritance. It is similar to graphical modelling technique, but it is more focused on specifics of the context objects and their attributes. Reusability and encapsulation are employed to resolve problems that are related to context dynamism. In the object-oriented world, different objects are used to represent context variables, such as place and location, and the approaches encapsulates information related to context presentation and processing. The interface defined on top of these context modeling and implementations allow access to this information without knowing how this context data is being handled internally.

Logic-based models: Rules are an important part of the context because they define contexts by interaction and relationship of different context variables. There are different action items for each observation made from the context, which varies from one application domain to another. This is because the rules can be different on how a context variable should be handled in the specific application area. For example, the change in location has a different impact in a public transport domain compared to a road planning system. The reasoning process involves deriving new facts based on existing system rules. Contextual information is represented formally as facts.

Ontology-based models: An ontology-based approach provides a formal description of the ideas and relationships between different attributes of a context. A classic example of an ontological approach is the Context Broker Architecture (CoBrA) system [61], which uses a set of concepts to characterize entities which may include people and places. With increasing number of sensors being installed on buses, formal modeling and specification of the relationship found in different contexts attached to a bus can help in making effective use of the data collected.

2.2.2.4 Context Reasoning

The reasoning stage of an application is an important part of the process to convert context information to a doable action because it has a lot of variance from domain to domain and application to application. Moreover, a major part of context reasoning depends on the context of users and application usage being targeted. Different context models mentioned earlier, focus more on modelling the objects and their relationships than context reasoning. Therefore, reasoning and rules have less representation in the modeling process [64]. The context-aware systems need to derive a combined context from different attributes collected from a variety of sensors [65]. An overview of these context reasoning mechanisms discussed in [65], [66] presented different strategies that can process context data. Some of these techniques are later applied to the data to produce prediction probabilities. For example, *Fuzzy Logic* is a type of data processing used to find the likelihood of an event as either true or false. This approach is comparatively more useful in systems that require multi-sensor fusion to identify a conflict between different context variables. More complex problems can be addressed using two or more fuzzy sets to create a new fuzzy set. Probabilistic logic can also be applied to multi-sensor fusion by applying logical reasoning based on probability. The rules of probabilistic logic do not give sufficient expressive power to capture dependencies and uncertainties of context variables [67], [68].

A public transport system can be represented on a graph where different stops are the nodes and link between them can be represented effectively using Bayesian Networks (BNs) because such networks are effective for the representation and storing of probabilities. This technique can derive reason behind derivation of contextual information. Although it can effectively represent context data of a transport system, it lacks the ability to monitor and update the consistency of the continuously changing data [64], [69]. Other techniques discussed include: Neural Networks, but its capability to predict is not as accurate as other reasoning techniques. Worse still, the training of the network is slow [66].

2.3 Importance of Middleware Architecture for Context-Aware Systems

The Middleware architectures are used to play the role of a bridge to integrate different systems intended to work together. This section briefly reviews some important Middleware architectures from literature that provide guidelines for considerations to integrate one system with another. Intelligent transport systems are large-scale systems that exist in the integration of different systems working together. Although each of these systems has an architecture on which they are based on, they need to have another middleware architecture that can define the relationship and dependencies and interactions between those systems. Different applications of Middleware architecture need to have standardized components with segmentation of interaction between them [86]. Numerous middleware infrastructures are available that implement context-aware systems [86]. One such example is GAIA Meta-Operating System (GAIA, named after the old Greek Earth-Goddess). This is based on distributed components (sensors and actuators) and applications responsible for management of all GAIA components [87] where information about different hardware and software components are stored in a central repository [88], so that it can provide fault tolerance. It allows an easy access management of the context so that data is available to users and applications on request. Furthermore, any updates to the context are updated automatically [89]. This architecture allows plug and play interface for both software and hardware components and therefore, various types of networked devices and sensors can be attached to the residential gateway regardless of their physical location [90]. The context of the connected device is obtained from its local execution context periodically [91]. The interaction and flow of information between different components or systems is controlled by the middleware architectures.

The information between different components of middleware architecture is exchanged through XML documents that contain service requests to be matched with a resource or service [88], [96]. These documents are then needed for storing and exchanging configuration information too [88]. This flexibility helps in maintaining a consistent fault tolerance when different packages fail because they are replaced by another working service or device [92]. The major problem with such architectures is the non-scalable integration, which is a very important aspect of a public transport system, because more and more buses and devices are being connected. Consequently, integration of data coming from these buses should scale for effective analysis. Such architectures also rely heavily on the network for their performance [93], [94] and different representation of context, may exist, making the conflict resolution challenging [95], [97].

2.4 Intelligent Transport System Architectures

The concept of Intelligent Transport Systems (ITS) emerged in late 1970s before the growth of digital devices and availability of platforms to share the data produced or consumed by these devices. ITS is a wide domain that covers both private and commercial transportation systems and is defined as a group of technologies, systems and services for better and secure transport services. This section presents a detailed review based discussion from the literature about existing ITS architectures; their comparison with each other, and discussion of the gap between these architectures and the practical implementation of intelligent transport systems. The ITS systems have provided a new domain of intelligent technology integration, resulting in many new opportunities for building safe, reliable and scalable service infrastructures for transport. ITS is a complex system that is a combination of many development areas associated with relevant user services attached with it [143]. Some of the important categories for the application of ITS are:

- Public Transport Management
- Management Systems for Private and Commercial Vehicle Fleets
- Intelligent Technology Integration
- Electronic Payment Systems
- Development of Intelligent Vehicles
- Safety Management

ITS architecture is comprised of different software and hardware components and the connectivity between them. It is important to identify the interfaces between these ITS components so that communication between them can be standardized [98]. The communication and interaction between different ITS components provide a framework, which is used for planning, defining, deploying, and integrating intelligent transportation systems. Such architecture typically defines the functional entities: work flows to connect these entities and the services offered by ITS systems and applications for the user [99]. The user services include information about: traveling, management of different modes of traffic, collection of data for performance improvement, improving driving and monitoring driving behavior, incident response handling, and managing vehicle fleets in public and private settings [100]. If we take the example of a bus system, these user services are performed with the help of different entities [101] like passengers, buses, garages, bus management operators, transport companies and different hardware and software devices the system is interacting with.

Many of the entities, such as passengers, bus companies, and operators are direct users of the system and they are connected to one of the services offered by the ITS in their context. The external entities ITS interacts with, like data providers (weather, roadworks etc.), and different sensors (GPS, Speed, Direction, temperature, light) installed on vehicles are also part of the system definition [104]. A promising architecture considers all these parameters in designing a system that can perform according to the expectation. The information and data flows define functional details of the system [101], [102]. Each user service offered by ITS is represented as a workflow where each step is bound with required input and output of the process. ITS, being a distributed system where different entities of the system may exist remotely to each other, should design workflows for its services formally to represent how different functions are formed so that their compliance with standards can be verified.

2.5 Characteristics of Successful Large Systems

Like any large software system, ITS is a very complex system that needs formal specification so that the relationship between different components is clear, and so the behavior of these components is predictable [103]. This is why the architecture of such a system plays a key role in the identification of the complexities and implementing their solutions. Depending on the environment or region in which ITS are being implemented, different stages of deployment should be planned [87], so that the system grows on the knowledge of the context it belongs to. It is a continuously improving system because the demand for services is changing with new functions introduced to meet those requirements. This ongoing process involves the evolution of existing functions and integration with new features, while maintaining existing functionality [87]. Different characteristics that are common in large-scale systems that have the capacity to evolve and expand on services are: Compatibility, Expandability or Extendibility, Interoperability, Integration and Standardization [101], [104]. Other important attributes are: system performance and accuracy of the information to enhance reliability. Scalability, which is closely linked with expandability and performance, enables the system to perform gracefully under heavy load. Some of these attributes are described next, and their link with intelligent transport architecture is further explained in Chapter 4.

2.5.1 Compatibility

Compatibility defines the ability of the system to continue working when a software or hardware component is replaced or upgraded [112]. It is employed by writing clear and consistent specifications of functionality where boundaries for components are specified and interaction between them is well documented. When a new software component is designed to be deployed; either as a replacement or a new addition. The compliance of this new component with existing interfaces will increase the chances of its compatibility in the environment.

2.5.2 Expandability

Expandability defines the flexibility of a system to be expandable in the functions it offers for the environment where it is deployed. This may be needed because the system must support more users; or, the data handling in the system has new challenges because the volume of data has increased due to the installation of new users (buses in the transport system and installation of new sensors on the buses for data collection). It has many common points with compatibility [111], [113] because the lack of compatibility can drastically impact the ability of the system to be expandable. Some of the examples of expandability of a bus system are:

- There are more buses added to the system. Different buses have a different type of internal hardware and varying ability to interact with external software and hardware components.
- New sensors installed on the buses so that more data can be collected about the context of the bus for the overall improvement of the bus system.
- New data source or destination added, which results in increasing load on the system for its data processing and distribution.

It becomes easier if the system architecture has formally specified the requirements for integration of new resources into the system and expectation of the data handling layer.

2.5.3 Interoperability

Interoperability defines the ability of the system to be able to work with external systems without interfering with the functionality or performance of the other systems [113]. It is one of the key challenges that need to be addressed with an increase in technology integration into intelligent transport systems, because many different platforms are being integrated and they need to work together. Different cities and modes of transport are using different payment systems and different prediction systems that don’t talk to each other. Also, the ability of a bus system to be able to work with other transport systems like trains, trams, and planes will reflect its interoperability, because different systems are designed under different architectures. A standardized architecture that has formal specifications for all these interfaces and treats internal and external entities as part of the system can help in building interoperable platforms.

2.5.4 Integration

Once the components are designed and their interoperability is verified in their specification, integration of these components to work together in the deployment environment is the next parameter to reflect robustness of the system. Integration is not only about bringing new components to work in the deployment, but also the ability of the system to work with external systems [114] where information is exchanged through APIs. One key element of integration is the ability of system architecture to help to integrate with existing legacy systems. A good architecture structures the components to facilitate integration at component level, as well as system level so that any component of a smart architecture can be easily integrated with an external platform even if it is the old legacy environment. A bus system may need to work with different bus arrival prediction systems for bus arrival times in different routes. An architecture that has agreement on interfaces and the workflows can make integration much more reliable [131].

2.5.5 Standardization

The above mentioned desired characteristics of a large-scale system like ITS are easier to achieve if the architecture is designed based on available standards [105], [113] and technical development is monitored for compliance with agreed standards. These standards help in making the data models, interfaces, and functionality consistent. The architecture makes it a formal process to create standards for different entities and components that add to their flexibility to grow into large-scale systems [106]. A variety of devices and data sources can be interchangeably integrated into the system with a predictable behavior. These software and hardware devices, data sources, data distribution endpoints and integration with external systems can be simplified if the interface rules for their characteristics are specified according to agreed standards [130]. A commercial system, especially an ITS system working in real-time environment, that must perform in production environment is therefore required to exhibit desired characteristic standards as part of its implementation, because it will face functional, as well as non-functional performance challenges. The desired characteristics devise the standards that should be implemented in a commercial product because they ensure not only the required functionality of the system, but also present the promises. A commercial system should be: compatible and flexible; interoperable with internal and external systems; accurate in the data it produces, and should be expandable to increase the scalability with optimal use of resources.

2.6 Advantages of ITS System Architecture

The past segment described how a framework design can help advance the incorporation of desirable attributes in a complex, developing framework like ITS [104], [105]. There are numerous extra advantages, which an ITS framework can offer. ITS architectures offer the following advantages [107]:

- Provide a basis for vision development to develop ITS for national and regional frameworks [108], [109], [115].
- Provide guideline for planning of new components or extension of existing components and formal specification of the interface between them [124].
- Developing a framework that meets requirements of a compatible, expandable and interoperable system that supports future development and extension [107], [129].

2.7Review of different ITS System Architectures

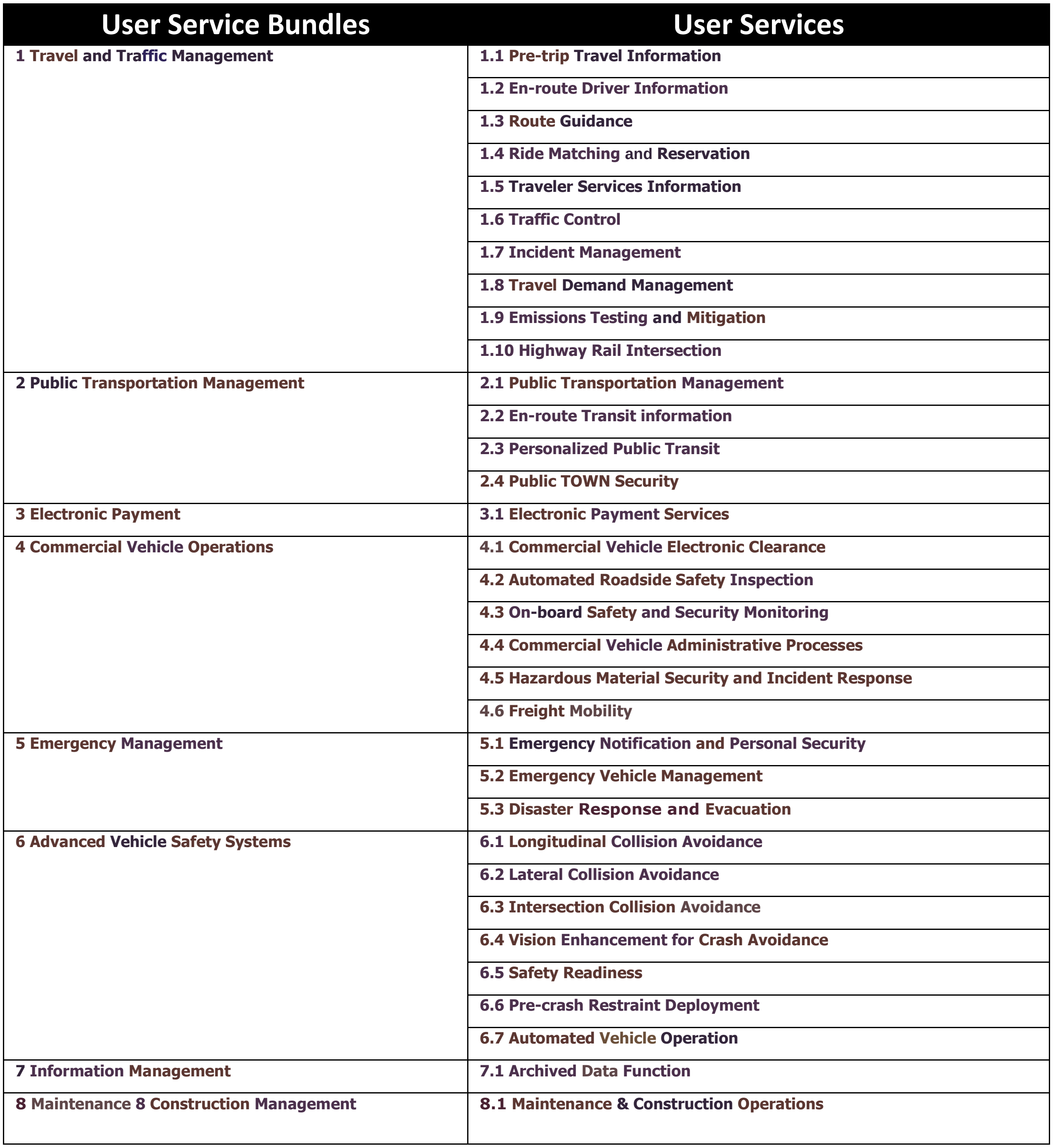
Many developed countries, such as USA, Japan, and European countries have developed ITS architectures [125] and transport systems like bus management system, which implement their functionality based on these developed architecture standards. Other countries have taken inspiration from these pioneer implementations and designed ITS for their regional environment. In addition, the International Organization for Standardization (ISO) incorporates a working group on ITS architecture (WG1) in its technical committee on ITS (TC204). These standards are useful for enhancement of the existing transport systems and the development of new systems in countries like Pakistan where no formal intelligent transport system exists.

2.7.1 U.S. National ITS Architecture

The first ITS system starting in the early 1990s was designed in the USA [108], [109]. With no legislative support for an ITS architecture, ITS America worked hard to win support for a system architecture, that was right, and that would be applicable from the onset. This helped them win Federal government’s support for the initiative. The process started with taking input from different stakeholders that generated a lot of interest for ITS legislation. Eventually, it was discussed and approved in the Transportation Equity Act for the 21st Century, 1997. This legislation encouraged the continuing development and use of the USA ITS architecture [120], [128]. It provided a detailed guideline on how regional transport systems can be deigned in conformance with the US ITS architecture, if the system used any Federal funds to develop the infrastructure [114], [118], [125]. The architecture presented standards for a set of user services based on their individual requirements, as well as a guide for the development of future standards.

2.7.1.1User Services

The Table 2.1 lists 33 user services offered by the US ITS architecture based system. These services are bundled together based on the type of user. Great importance is given to user services in public agencies and the operation of the infrastructure [110], [111]. Different user services are bundled in groups with each bundle addressing a specific part of the transport system. One of the key user services referred in the bundle: "Public Transport Management" is Personalized Public Transit, which mentions taking the context of the transit in consideration when designing a system to offer this service. It covers different desirable characteristics discussed before like "Electronic Payment" for integration of different payment gateways, "Travel and Traffic Management" for information retrieval and response management for different events and incidents while a vehicle is driving on the road.



User Service Bundles	User Services
1 Travel and Traffic Management	1.1 Pre-trip Travel Information
	1.2 En-route Driver Information
	1.3 Route Guidance
	1.4 Ride Matching and Reservation
	1.5 Traveler Services Information
	1.6 Traffic Control
	1.7 Incident Management
	1.8 Travel Demand Management
	1.9 Emissions Testing and Mitigation
	1.10 Highway Rail Intersection
2 Public Transportation Management	2.1 Public Transportation Management
	2.2 En-route Transit information
	2.3 Personalized Public Transit
	2.4 Public TOWN Security
3 Electronic Payment	3.1 Electronic Payment Services
4 Commercial Vehicle Operations	4.1 Commercial Vehicle Electronic Clearance
	4.2 Automated Roadside Safety Inspection
	4.3 On-board Safety and Security Monitoring
	4.4 Commercial Vehicle Administrative Processes
	4.5 Hazardous Material Security and Incident Response
	4.6 Freight Mobility
5 Emergency Management	5.1 Emergency Notification and Personal Security
	5.2 Emergency Vehicle Management
	5.3 Disaster Response and Evacuation
6 Advanced Vehicle Safety Systems	6.1 Longitudinal Collision Avoidance
	6.2 Lateral Collision Avoidance
	6.3 Intersection Collision Avoidance
	6.4 Vision Enhancement for Crash Avoidance
	6.5 Safety Readiness
	6.6 Pre-crash Restraint Deployment
	6.7 Automated Vehicle Operation
7 Information Management	7.1 Archived Data Function
8 Maintenance 8 Construction Management	8.1 Maintenance & Construction Operations

Table 2-1: US ITS Architecture user services [110, 111]

2.7.1.2 Logical Architecture

The logical architecture represents details of the system's function without worrying about technical details of their implementation to ensure a common standard is available for different implementations across a selection of technology used for the development [116], [126]. The details discuss the data elements; the sequence of events that are supported; the behavior of different components based on their configuration; data and information flows, and specification of logical processes that control the request and response framework. It also advises on different techniques and strategies used for storage and processing of data in different stages and levels of the system. An abstract data flow diagram (DFD) of a logical architecture is shown in Figure 2.3. It describes on flow and sequence of different action and information elements that are activated when a specific stimulus is started from a vehicle or management endpoint.

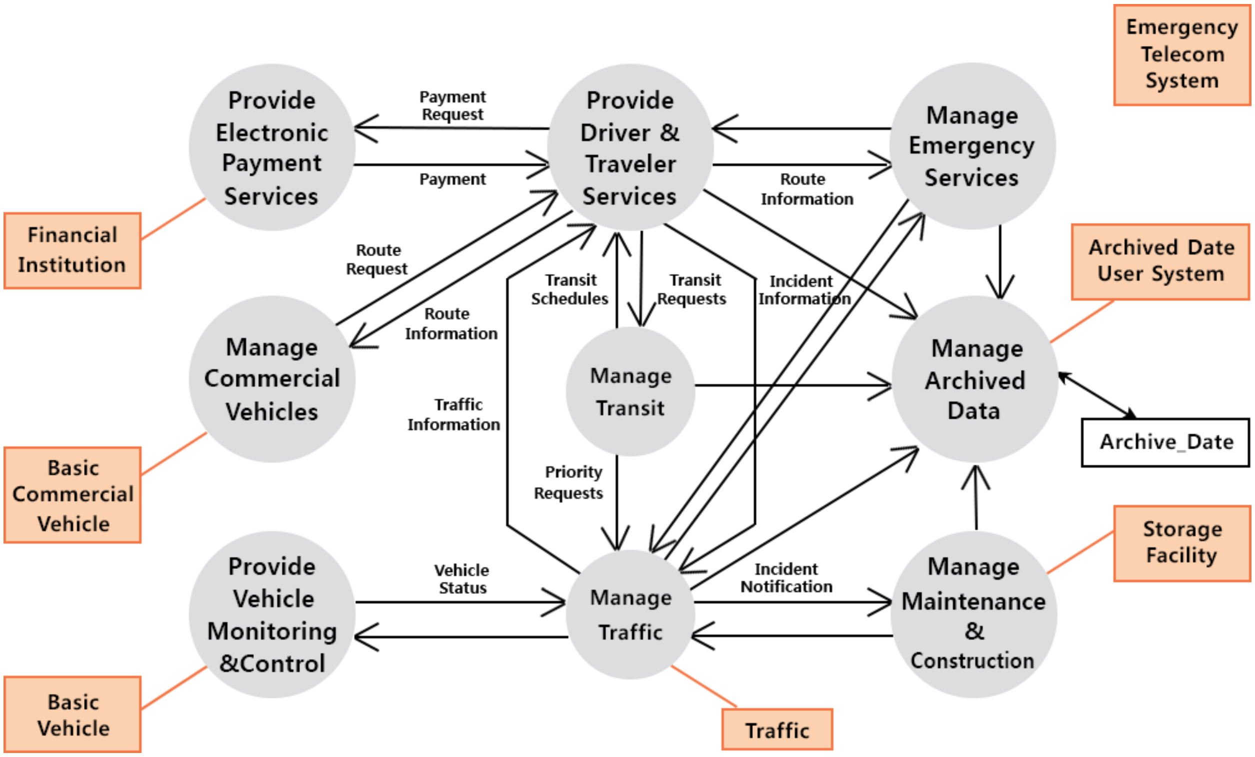


Figure 2-3: Logical Architecture of a US ITS Architecture

2.7.1.3 Physical Architecture

The physical architecture is a realization of the logical architecture when applied to a specific requirement and identifies the important ITS interfaces and system components implementing those interfaces [117]. The logical flows are converted into architecture flows and concepts from logical architecture are given a representation of entities from the real environment with detailed interface specification of communication between those entities [125]. Figure 2.4 shows the data flows and the corresponding communication requirements that define the interfaces between entities of physical architecture.

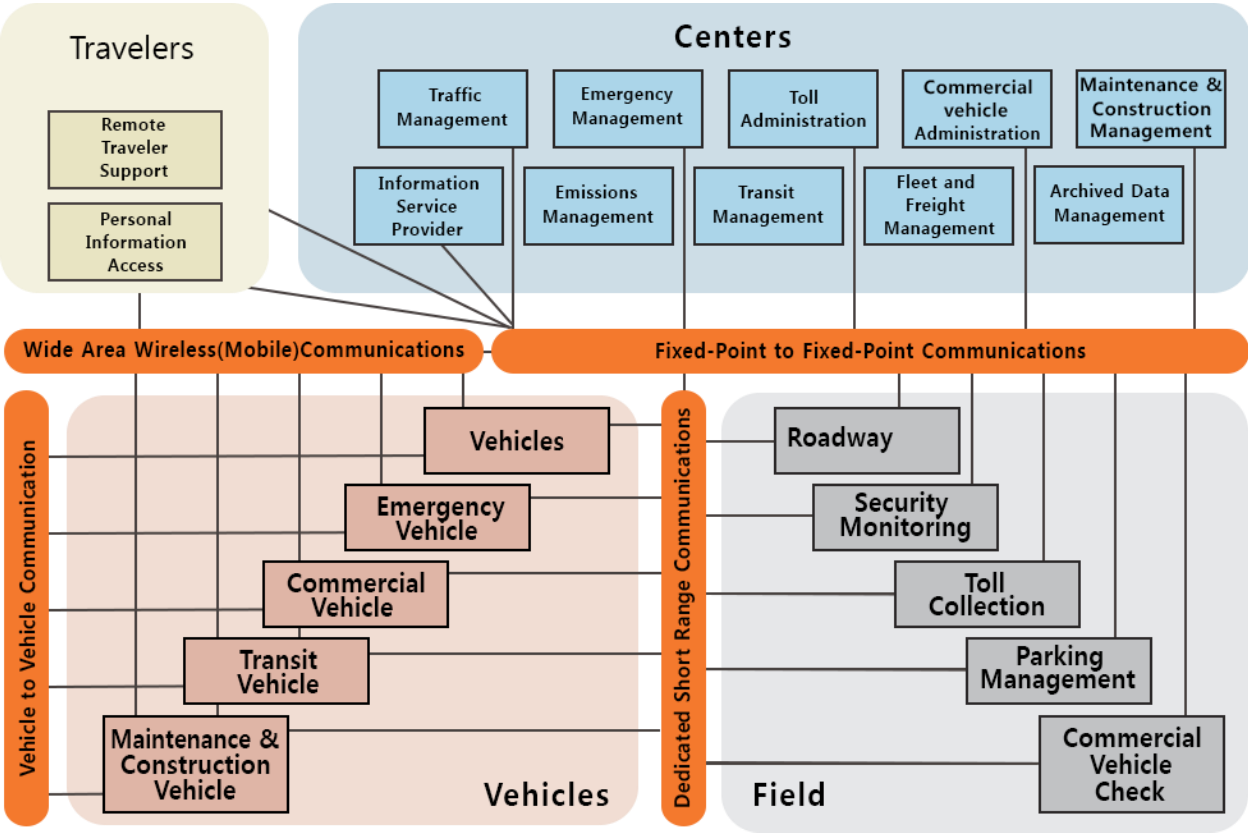

Travelers
Remote Traveler Support
Personal Information Access
Centers
Traffic Management
Emergency Management
Toll Administration
Commercial vehicle Administration
Maintenance & Construction Management
Information Service Provider
Emissions Management
Transit Management
Fleet and Freight Management
Archived Data Management
Wide Area Wireless(Mobile)Communications
Fixed-Point to Fixed-Point Communications
Vehicle to Vehicle Communication
Vehicles
Emergency Vehicle
Commercial Vehicle
Transit Vehicle
Maintenance & Construction Vehicle
Vehicles
Dedicated Short Range Communications
Roadway
Security Monitoring
Toll Collection
Parking Management
Commercial Vehicle Check
Field

Figure 2-4: High-Level Physical Architecture of US ITS Architecture

The USA ITS architecture presents a very comprehensive structure and guideline for building coherent transport systems across different domains of transport services. It identifies different types of vehicles and the set of actions associated with each vehicle for effective planning and management. Different stakeholders are identified and corresponding services are described to enable the stakeholders to interact with the system. However, the USA architecture is at a very high level and emphasis is on the government policies and guidelines. There is a need to focus more on different public transport systems separately to improve their performance and integration with the complete ecosystem of transport. Specifically, the buses operating in a public transport system (and other public transport mediums like trains, trams) are not operated by the government itself and are subcontracted to bus management companies. These companies focus only on the operation of buses that are part of their network to follow the timetable provided to them. This results in many subsystems working under different bus companies, and these systems are not connected with each other directly. It means that; although the standards exist, there is hardly any formal process to enforce them, especially if the small companies (subsystems) are the actual service providers. These architectures should reflect their application at small level so that all subsystems can work in integration with each other.

2.7.2 European ITS Framework Architecture

The Fifth Framework research program [132] supported a project, European ITS Framework Architecture (FRAME) in EU Directorate on Information Society Technology. It was developed based on an existing EU ITS architecture called Keystone Architecture Required for European Networks (KAREN) [133]. KAREN was developed around the same times when US ITS architecture in the late 1990s by the Fourth Framework research program for Transport Telematics [134]. FRAME is not an architecture but a specification based guideline to help member EU countries design their regional ITS architectures based on this common standard. Different viewpoints of FRAME [134] are briefly described here:

- User-centric services (comparable to user service bundles in US ITS architecture).
- Functional Architecture to represent different system functions, data flows involved and the relationship between different functions and the outside world (see Table 2.2).
- Physical Architecture to group the functions from functional architecture into physical locations so that system implementation considers this for deployment.
- Communication Architecture for standards specification of communication requirements between different physical architecture elements.
- Deployment compliance to FRAME.
- Standardization of requirements.

Functional Areas	Functions
1 General	**1.1 Architectural Properties**
	1.2 Data Exchange
	1.3 Adaptability

	1.4 Constraints
	1.5 Continuity
	1.6 Cost/Benefit
	1.7 Expandability
	1.8 Maintainability
	1.9 Quality of Data Content
	1.10 Robustness
	1.11 Safety
	1.12 Security
	1.13 User friendliness
	1.14 Special Needs
2 Infrastructure Planning & Maintenance	**2.1 Transport Planning Support**
	2.2 Infrastructure Maintenance Management
3 Law Enforcement	**3.1 Policing/Enforcing Traffic Regulations**
4 Financial Transactions	**3.1 Electronic Financial Transactions**
5 Emergency Services	**5.1 Emergency Notification & Personal Security**
	5.2 Emergency Vehicle Management
	5.3 Hazardous Material & Incident Notification
5 Emergency Management	**5.1 Emergency Notification and Personal Security**
	5.2 Emergency Vehicle Management
	5.3 Disaster Response and Evacuation
6 Travel Information & Guidance	**6.1 Pre-trip information**
	6.2 On-trip driver information
	6.3 Personal information services
	6.4 Route guidance & Navigation
7 Traffic, Incidents & Demand Management	**7.1 Traffic Control**
	7.2 Incident Management
	7.3 Demand Management
	7.4 Safety enhancements for Vulnerable road users
	7.5 Intelligent junctions & links
8 Intelligent Vehicle Systems	**8.1 Vision enhancement**
	8.2 Automated Vehicle operation
	8.3 Longitudinal Collision avoidance
	8.4 Lateral Collision avoidance
	8.5 Safety readiness
	8.6 Pre-crash Restraint Deployment
9 Freight & Fleet Management	**9.1 Commercial vehicle Pre-clearance**
	9.2 Commercial vehicle administrative processes
	9.3 Automated roadside safety inspection
	9.4 Commercial vehicle on-board safety monitoring
	9.5 Commercial fleet management
10 Public Transport Management	**10.1 Public transport management**
	10.2 Demand responsive public transport
	10.3 Shared transport management
	10.4 On-trip public transport information
	10.5 Public travel security

Table 2-2: Functional areas of FRAME project

There are many guidelines available on how EU ITS can be developed and the future vision for enabling it for better service and information integration [144], [145], [146], [147]. The original EU ITS architecture was inspired from the US ITS model and that is why the primary focus of this was to clarify some of the services and improve their categorization so that it is easier to understand which domain the user service belongs to and who are the stakeholders expected to use that service. One major difference in the design of user services in EU ITA architecture was that they focus a lot on quality of service attributes and explicitly mentioned the quality attributes like Adaptability, Expandability and Maintainability, to name but a few, which are part of the desired attributes of large scale systems. The EU ITS architecture also emphasized the security of the transport systems and the identification of vulnerable road users so that security of the transport service is considered vital for adding public transport to the preferred way of traveling. The intelligence implementation of vehicle systems was incorporated so that data sharing between different components of the transport infrastructure is facilitated. However, it shared the abstractions of its counter USA ITS architecture in explicit segregation of public transport systems and presented an overall view of the transport infrastructure including all types of road infrastructure and commercial transport services. There were no clear guidelines on how different transport companies (bus companies to be more specific) would implement a system based on this architecture and communicate to each other. No information was incorporated for the individual companies to follow the same standard in implementation of their IT systems. Therefore, there are many vendors, each implementing the small level ITS for their network according to its preferred technologies and architectures suitable under its localized constraints.

2.7.3 Japanese ITS System Architecture

Completed in 1999, ITS architecture in Japan was completed by a collaborative effort of five government ministries working with VERTIS [136]. They worked out the integration of technology at the very early stage so that the intelligence part of the architecture for decision making, played the key role to help different subsystems work together. The agreed design aimed at promoting [136], [138]:

- Building efficient infrastructure of an integrated intelligent transportation system;
- Keeping the architecture expandable and maintainable so that systems built on it can accommodate new changes and challenges;
- Contribution to domestic and international intelligent transport system designs;
- Accommodate changing social needs and integrate evolving technology;
- Interoperable and Interconnect-able.

Like US and European ITS architectures [135], 137], it also focused on user services, logical and physical architecture, but it also focused on the creation of ITS standards to be followed internationally. The user services supported by this architecture are listed in Table 2.3.

Development Areas	User Services
1. Advances in navigation systems	**1. Provision of route guidance traffic information**
	2. Provision of destination-related information
2. Electronic Toll Collection Systems	**3. Electronic toll collection**
3. Assistance for safe driving	**4. Provision of driving and road conditions information**
	5. Danger warning
	6. Assistance for driving
	7. Automated highway systems
4. Optimization of traffic management	**8. Optimization of traffic flow**
	9. Provision of traffic restriction information in case of incident
5. Increasing efficiency in road management	**10. Improvement of maintenance operations**
	11. Management of specially permitted commercial vehicles
	12. Provision of roadway hazard information
6. Support for public transport	**13. Provision for public transport information**
	14. Assistance for public transport operations and operations management
7. Increasing efficiency in commercial vehicle operations	**15. Assistance for commercial vehicle operations management**
	16. Automated platooning of commercial vehicles
8. Support for pedestrians	**17. Pedestrian route guidance**
	18. Vehicle-pedestrian accident avoidance
9. Support for emergency vehicle operations	**19. Automated emergency notification**
	20. Route guidance for emergency vehicles and support for relief activities
10. General	**21. Utilization of information in the advanced information and telecommunications society**

Table 2-3: Japanese ITS Architecture

The user services list development areas that target a specific user or a group of users. There are development areas identified for public transport and advances in navigation systems, so that passengers of public transport systems like buses have access to reliable information about the operation of the buses. Figure 2.5 shows the connection of different components of the ITS and how they are communicating with each other to make the complete system. It can help the application developers to implement these relationships clearly and independently so that the technology integration can contribute to a maintainable and expandable ITS.

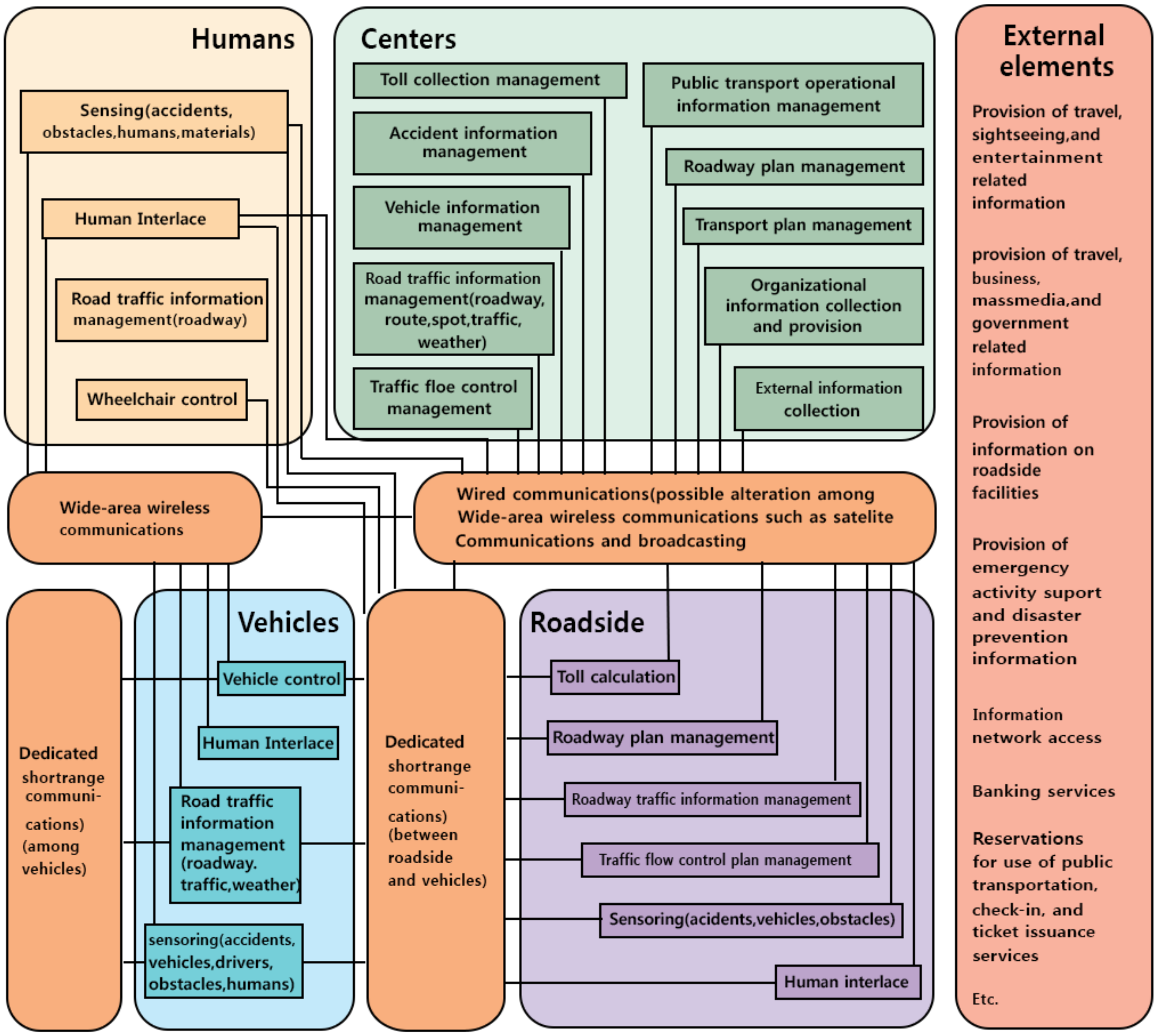

Humans
Sensing(accidents, obstacles,humans,materials)
Human Interlace
Road traffic information management(roadway)
Wheelchair control
Centers
Toll collection management
Public transport operational information management
Accident information management
Roadway plan management
Vehicle information management
Transport plan management
Road traffic information management(roadway, route,spot,traffic, weather)
Organizational information collection and provision
Traffic floe control management
External information collection
External elements
Provision of travel, sightseeing,and entertainment related information
provision of travel, business, massmedia,and government related information
Provision of information on roadside facilities
Provision of emergency activity suport and disaster prevention information
Information network access
Banking services
Reservations for use of public transportation, check-in, and ticket issuance services
Etc.
Wide-area wireless communications
Wired communications(possible alteration among Wide-area wireless communications such as satelite Communications and broadcasting
Dedicated shortrange communi- cations) (among vehicles)
Vehicles
Vehicle control
Human Interlace
Road traffic information management (roadway. traffic,weather)
sensoring(accidents, vehicles,drivers, obstacles,humans)
Dedicated shortrange communi- cations) (between roadside and vehicles)
Roadside
Toll calculation
Roadway plan management
Roadway traffic information management
Traffic flow control plan management
Sensoring(acidents,vehicles,obstacles)
Human interlace

Figure 2-5: Components interconnection diagram from Japanese ITS Architecture

The Japanese ITS architecture improved the organization of its user services with increased focus on the public transport and the commuters using these services. Contrary to USA and EU ITS architectures, Japanese ITS architecture provided a concrete list of components and services directly linked with public transport and sharing of information between transport services and commuters for effective consumption of the service. It suggested better traffic control with smart sensors implementation on the roads and vehicles so that accidents and incidents can be identified quickly, and obstacles can be removed for smooth operation of the transport services. Integration of external data sources like weather data and other traffic data was also considered in design of the architecture. Although this architecture encouraged use of latest sensors for enabling the vehicles to become a data source of the context in which they are running, collection of data from these vehicles and the extended dimension of the data collection possibilities were not included in the architecture specification. The aspect to address the situation of different bus companies implementing a solution for part of the public transport network is missing in its specification. It does not specifically include quality of service attributes like the EU ITS architecture and provision of these attributes is left to the implementation team, which will end up in each solution having different quality measurements and expectations.

2.7.4 ISO ITS Reference Architecture

As shown in different tables and diagrams, and described in explaining the three major architectures of ITS from USA, Europe and Japan, standardization of the architecture and interfaces between different subsystems is important for effectiveness and usability of it to work as a guiding principle. The International Organization for Standards (ISO) has developed ITS architecture standards so that collaboration between different architectures can be facilitated. This architecture is kept relatively simple so that it can be used more as a base for developing more complex systems rather than restricting other applications through constraints based architecture implementation [119], [139]. It is available as an ISO standard ISO 14813 that provides a reference model and collection of user services. A high-level depiction of this architecture is presented in Figure 2.6 that shows different components of the system and the type of users interacting with each component. These components are subsystems themselves that work together like components of an architecture and provide an integrated interface to the whole ITS.

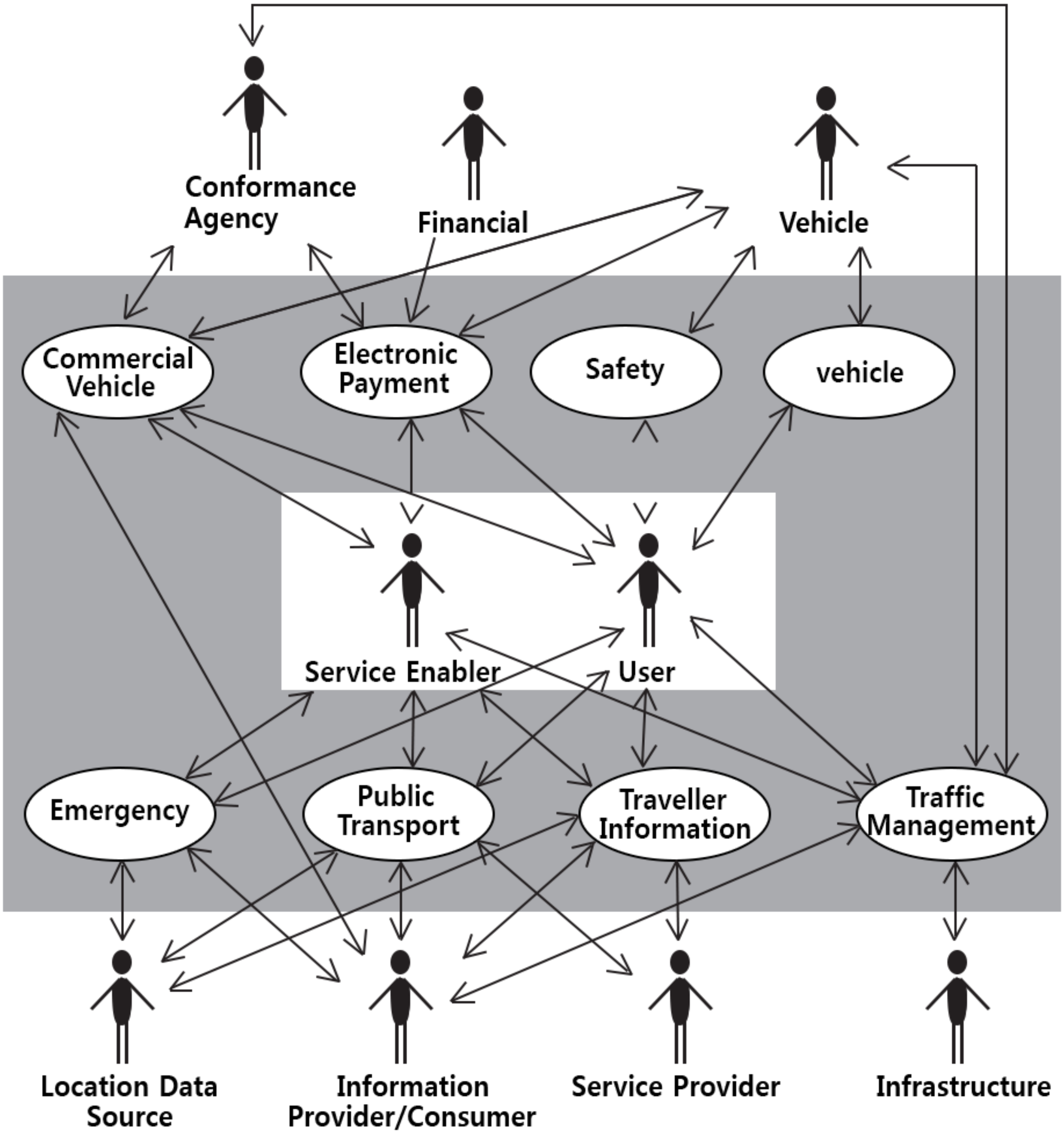


Figure 2-6: High-Level ISO Core ITS Architecture

The fundamental user services covered by this architecture are shown in Table 2.4. These different service domains are just like service bundles of US architecture and development areas in Japanese ITS. This similarity indicates that the ISO technical committee took existing system architectures into consideration when designing the standard so that the compatibility and integration aspects are taken care of.

Service Domains	Service Groups
1. Traveler Information	**1.1 Pre-trip Information**
	1.2 On-trip Information
	1.3 Travel services information
	1.4 Route Guidance & Navigation pre-trip
	1.5 Route guidance & navigation-On trip
	1.6 Trip planning support
2. Traffic management and operations	**2.1 Traffic control**
	2.2 Transport-related incident management
	2.3 Demand management
	2.4 Transport in infrastructure maintenance management
3. Vehicle	**3.1 Transport-related vision enhancement**
	3.2 Automated vehicle operation
	3.3 Collision avoidance
	3.4 Safety readiness
	3.5 Pre-crash restraint deployment
4. Freight transport	**4.1 Commercial vehicle pre-clearance**
	4.2 Commercial vehicle administrative processes
	4.3 Automated roadside safety inspection
	4.4 Commercial vehicle on-board safety monitoring
	4.5 Freight transport fleet management
	4.6 Intermodal information management
	4.7 Management and control of intermodal centers
	4.8 Management of dangerous freight
5. Public Transport	**5.1 Public transport management**
	5.2 Demand responsive and shared public transport
6. Emergency	**6.1 Transport related emergency notification and personal security**
	6.2 Emergency vehicle management
	6.3 Hazardous materials & incident notification
7. Transport-related electronic payment	**7.1 Transport-related electronic financial transactions**
	7.2 integration of transport related electronic payment services
8. Road transport-related personal safety	**8.1 Public travel security**
	8.2 Safety enhancement for vulnerable road users
	8.3 Safety enhancements for disabled road users
	8.4 Intelligent junctions and links
9. Weather and environmental conditions monitoring	**9.1 Weather monitoring**
	9.2 Environmental conditions monitoring
10. Disaster response management and coordination	**10.1 Disaster data management**
	10.2 Disaster response management
	10.3 Coordination with emergency agencies
11. National Security	**11.1 Monitoring and control of suspicious vehicles**
	11.2 Utility or pipeline monitoring

Table 2-4: Fundamental user services in ISO Architecture

The user services offered by ISO ITS architecture are more abstract than all three major (USA, EU, Japanese) ITS architectures discussed earlier. That probably is expected because ISO is a standards organization and cannot be specific for suggesting guidelines at micro level implementation of the systems. It has described general domains that should be covered in the implementation of intelligent transport systems without going into detail on how much and to what extent technology should be implemented to enhance the transport services. No focus on public transport system and no user services suggesting possible interaction between difference services and components of a public transport system. The data flow in figure 6 mentions the type of user interacting with each service domain but does not elaborate on the type of user service being offered.

2.8 Other Intelligent Transport Architectures

As we have elaborated in the review for high level ITS architectures in the previous sections, one of the common issue that none of them have emphasized is the actual implementation and construction of a public transport system. In this section, we review a small-scale intelligent transport system that was designed for shuttle service of University buses [164]. The shuttle service (UniShuttle) runs between different campuses of the University of Wollongong Australia (UOW) and other key destinations like city center, city beach and other attractions in the city of Wollongong. The purpose of this project was to design and develop an intelligent transport system for the shuttle service so that over 20,000 students and around 2000 staff members use it as primary preference to travel to/from the university. That would ease the pressure on university to accommodate increasing demand of parking spaces for students and staff members. This ITS is very close to the work in this thesis and therefore, a more detailed understanding and critical review for UniShuttle ITS is being presented.

2.8.1 System Architectural and Design

The architecture for UniShuttle system is designed based on IETF presence model [165] which is a subscription and observer model that allows subscribing for some information, which is shared when there is any change. There are three key elements of the UniShuttle architecture as *presentity*, *presence service* and *watcher*. The bus represents the concept of presentity (in the meaning of entity for presence model), server side represents the presence service and watcher is represented by the mobile application that was built part of the project. The architectural design for UniShuttle ITS is presented in Figure 2.7.

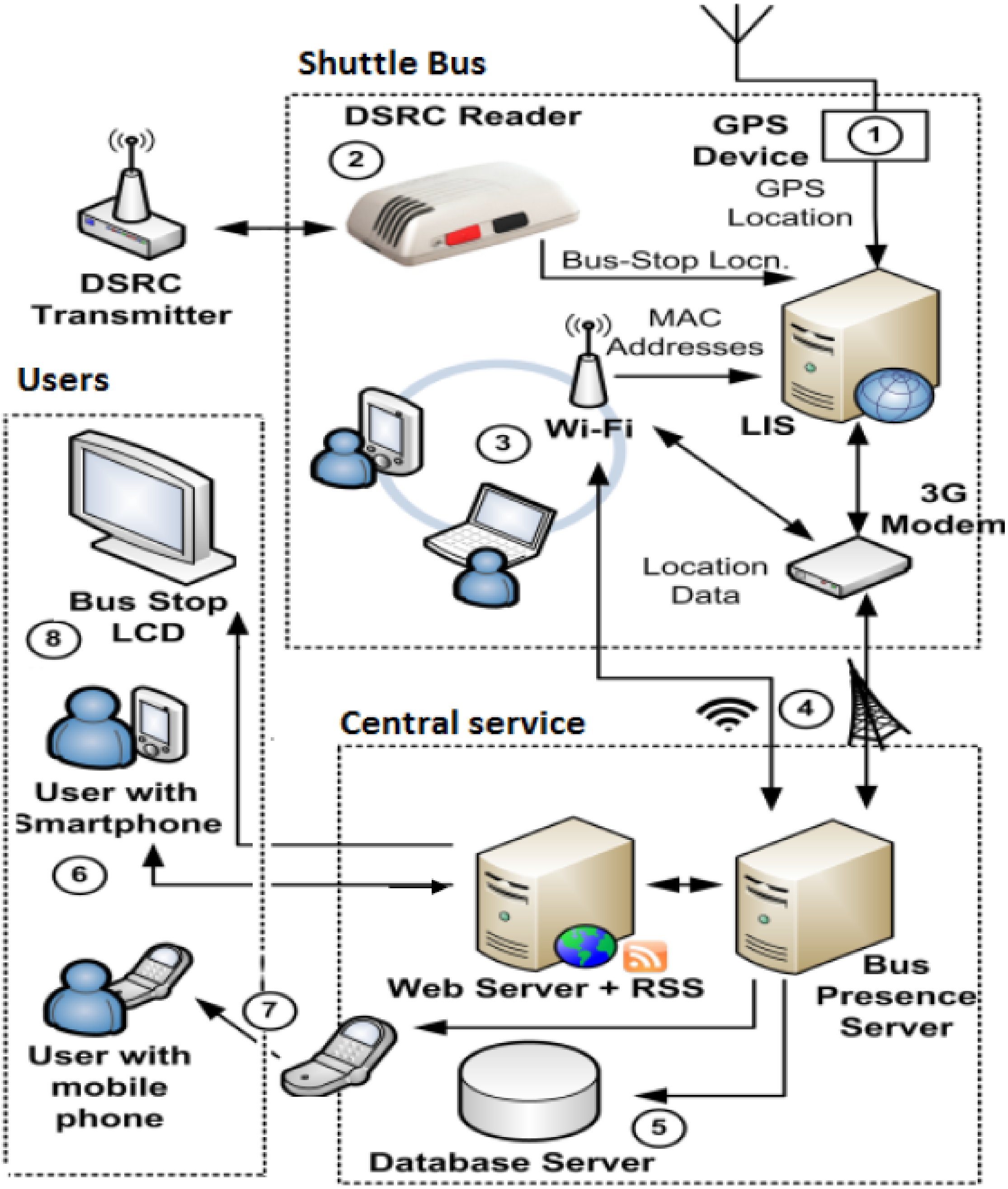


Figure 2-7: Architectural Design of UniShuttle ITS

The diagram in Figure 2.7 elaborates the flow of information between different nodes of the system and sharing of information once it has been processed on server side. It presents two different ways of interacting with the commuters. One is to communicate to them through local WIFI on-board the bus and second is through the internet for the commuters who are not on the bus and waiting on stops for the bus to arrive. A component diagram and interaction between different components of UniShuttle ITS are shown in Figure 2.8.

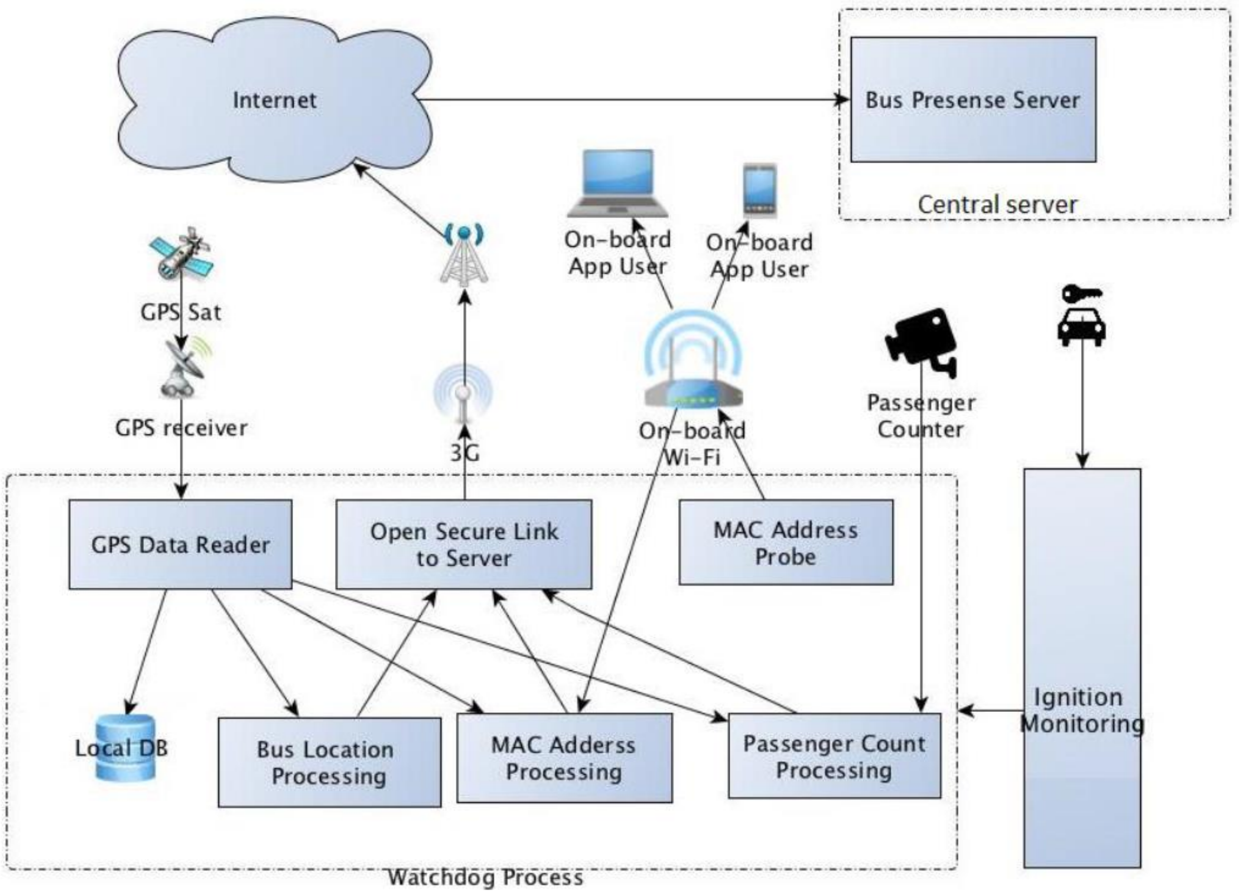


Figure 2-8: Component Diagram for Unishuttle ITS

Location of the bus is the key to start all the processing and sharing of information and therefore, the system implements GPS device as the primary source of location and Dedicated Short Range Communication (DSRC) transmitter/receiver as a backup in case the GPS device fails. This location is continuously shared with the presence service so that server side is always aware of the most recent reported location of the bus. The system installed inside the bus monitors the connected users through free WIFI service on the bus to know how many connected commuters are on-board. Once the server receives the data about location of the bus, it applies data mining technique to update the trends data for bus arrival to stops it will reach in future. This information is made accessible through web services to the devices like a web application or the mobile devices to display latest location of the bus and the time it will reach the next stop.

UniShuttle ITS is an interesting piece of work but it has many fundamental issues that create gap between its working and acceptance for a real-time intelligent transport system. There was no focus on the quality of service attributes and the system was not tested for performance when number of buses increase in the network. As mentioned in its name, it was designed for a small-scale but benchmarking it for the potential to be applied for large systems could have given more insight into its capacity to be expandable. The process of data mining is integrated into the real-time operation because it triggers application of data mining techniques when location for a bus is reported to the server. As the amount of data increases, the processing as part of data mining techniques will take more time and as a result delay propagation of trends information to commuters waiting on the website or the mobile application. This potential limitation of integrating data mining on the fly should have been covered in the architecture design so that the process of data mining is working in parallel and the information about location of the bus is available to commuters as soon as server has received it. The information collected through WIFI router for number of passengers is also not accurate because not everyone is connected to WIFI and it apparently seems unnecessary function when the bus already has a video camera for counting passengers.

In addition, the system does not integrate with any external data sources and no work has been done to make the architecture interoperable and compatible with external systems. Although the buses in UniShuttle system serve a limited audience of students and staff members, the buses still share the road with other private, commercial and public transport vehicles but the impact of external factors has not been studied. The prediction model used in this project to predict arrival of bus at a specific stop is based on history data joined with real-time information about distance being covered on every segment of the route. The problem with this model is that it does not consider a situation where the driver can actually makeup for the delay that may have occurred at previous stops or on the trip between previous stop and the next stop bus will reach. It means delay at any point in the journey from first stop to last stop will be added in the predicted arrival time for next stops until the bus actually reaches a stop where the latest arrival time will be used for predicting next stops. Also, the prediction model considers real-time data from only the segment the bus is driving through. No information from previous stops is used and delays at previous stops can have impact the prediction for future. These issues in the prediction model and the architectural design will also be addressed in this thesis in Chapter 5.

2.9 Critical Analysis and Summary

This chapter started with description of the context of an object, its relevance to the sensors and the process of converting the context into useful information using different context handling techniques of acquisition, modelling and establish rules to understand context based on reasoning techniques. The concept of vehicle in a public transport is changing because more and more services are being added to serve the needs of the commuters in addition to the buses, trains and trams. The popularity of Uber and other services like Lyft globally indicates that public transport systems are not smart enough to meet the capacity requirements. Effective and domain specific context handling techniques in a public transport system can help in improving understanding and integration of the context attributes in decision making.

The review of different ITS architectures presented in this chapter indicates that technology integration is vital to improve public transport systems, enabling them to achieve the goals of a reliable and scalable infrastructure. The Intelligent Transportation Systems (ITS) technology of today is going through a revolution to standardize integration of context sensors and has contributed to enhanced transportation e.g., buses [149], [150], [151], [152]. There are many standards designed for developing ITS in developing countries based on the understanding that they either don't have any existing ITS or the ITS is not well integrated with technology and needs improvement [121-123], [139], [140-142]. In spite of all these standards in place, the implementation of these architectures is not straightforward [148], [149] and working with difference service providers makes it a challenge. In London alone, there are more than 12 different bus companies providing the commuting service to millions of passengers and each company has its own internal solutions designed to facilitate achievement of their operational goals. These ITS architectures are very high level because they intend to consider all types of transports and involve the integration of different user services for each type of transport. Different bus or train operators are using different solutions for their network and that makes it hard for them to work together.

A typical example of this is a bus arrival time prediction system being used by different companies and the software systems installed on the bus to facilitate the driver. These different systems produce a different structure of the data and then share information about the buses with a central agency like Traffic for London (TfL) while using their own prediction and bus management systems. That indicates that the focus of these national and international ITS systems has not made its way down to the individual bus companies and they don't have a standard architecture to follow. Also, the bus companies that are not using any prediction system and rely completely on agencies like TfL for fetching the information can risk reliability of the information because TfL data would not consider complete contextual information of the bus system. There is a clear need for an intelligent bus system architecture that considers context information of the system, data being generated from its member buses through software/hardware sensors and users, and integrates this data for decision making. Such architecture should focus on improving the service based on the history of the system itself and integrating with real-time data as it arrives. It should also support easy integration of external data sources like weather and roadworks to enhance the system's ability to develop and provide predictive analytics for intelligent decision making. Therefore, in this work, a new architecture is presented in Chapter 4 that can address challenges of existing as well as new system to handle heterogeneous data coming from different context sources. It will enhance the public transport systems for context-driven decision making to improve the service and infrastructure for different stakeholders. A comprehensive evaluation of this proposed and implemented architecture is discussed in Chapter 5.

The next chapter focuses more on the data, techniques to acquire, store and process it and the ability of the machine learning algorithms to produce predictive analytics from it. Different data mining techniques will be discussed and existing bus arrival prediction systems based on some of these data mining techniques will also be elaborated.

CHAPER 3: DATA AND TECHNIQUES

The acquisition, storage, processing and delivery of large volumes of data in the era of internet has changed with high bandwidths available to the common user. An increase in adoption of smart commerical and personal devices and advancement in number and type of sensor devices (such as location, temperature, light, accelerometer, orientation, GPS, RFID, etc.) needs specialised infrastructures and frameworks to handle this data generation. The automotive industry is enhancing its capacity to support smart devices and sensors they intend to install on vehicles to make them connected for better planning and completion of the journeys. Public transport is no exception, and modern buses are installed with many sensors like GPS, accelerometer, temperature, video (CCTV) and operational sensors like opening/closing of doors, driving behaviour and fuel consumption. Different techniques of processing this context information have been discussed in Chapter 2 which elaborated on Intelligent Transport Systems and current applications of it in different regional settings. Chapter 2 also emphasized that handling context needs special techniques so that the context information can be converted into intelligent workflows for a smart bus network.

This chapter discusses challenges faced in handling large volumes of data being collected from different sensors and reviews data mining algorithms that can be used for contextual data processing. It will describe how data mining algorithms can help in processing the context data and the discovery of patterns that are helpful for designing prediction frameworks. The ability of the data mining algorithms is highlighted in producing prediction data for the arrival time of buses on stops based on forecasting patterns identified in the context data coming from buses. This chapter further reviews some existing arrival time prediction systems using data mining algorithms and provides a critical analysis of their effectiveness.

3.1New Data Challenges

A combined study and implementation by representing academia research and technical lead at Twitter [54] shares some remarkable findings of the latest developments of data growth, compatibility of existing mining techniques. The study also elaborates about algorithms in this new data paradigm of large-scale, heterogeneous variety and sources of data, data production velocity, industrial perspective and needs of the organisations regarding data analytics and large volumes of data. Data Analytics is in use since 1990s where data mining algorithms were used to generate statistics on the data and to find patterns of information that could be converted to knowledge. It is, however, interesting to understand the impact of data challenges of size, variety and velocity of data generation on these data mining techniques and the scale on which they are expected to be potentially used in meeting these challenges. Also there is increased use of real-time data analytics and its manipulation to improve services when it comes to healthcare, transport, education, social media. Other services such as banking (such as checking banking frauds based on analysis of the huge amount of existing data combined with real-time transaction data), real estate (checking real-time information about ownership, disputes and any loans on the property) can benefit from application of real-time analytics for improving the quality of service. Considerable changes in the trends and expectation of analytics have been discussed based on recent era of data and business analytics, data challenges and handling of real-time data streams for effective and efficient knowledge discovery as well as service delivery to potential end users [54]. The data acquisition and storage is discussed next with reference to data mining techniques and the potential use of these techniques is elaborated.

3.1.1 Data Acquisition

The Data acquisition has drastically changed from gathering only valuable data as per perception of the organisation based on well-defined business objectives to gathering a lot of other data that represents not only predefined valuable information but also the interactions different system stakeholders have with the business system of the organisation. The data about user behaviour, trends of user interaction with the system, user feedback on the service, user's discussions with other users regarding the services and products, exploiting user profiles to understand implicit decision-making paradigms of users and user experience is also gathered to link all these information elements to generate knowledge that can be used for strategic decisions regarding services, products and an enhanced user experience. That obviously results in ample amount of data being generated and gathered for analysis. So the data mining techniques and algorithms that were used to handle gathered data have to be scaled up so that it can reasonably process the large datasets being produced every day. The efficient transformation of data before it can be used by some analytical tools is only possible if the algorithms are scalable and can handle variety and velocity challenges associated with data coming from heterogeneous sources. There are many different techniques being used to acquire data from dynamic system entities like buses where the context of the vehicle changes every second. The data is collected through hardware and software components installed on the vehicle, along with the route and at different locations where the buses either start or end their journey.

3.1.2 Data Storage

Once the data has been acquired, it should be stored in a storage system so that it can be used for processing. This section discusses different storage platforms presented in the literature and presents a comparative elaboration of what each storage platform provides and how this may benefit in implementation a system like ITS. There is a variety of database management systems available like MySQL, MS SQL Server, Oracle that are examples of a relational database. The data growth has not seen only the volume and frequency of new data being produced, but it has also witnessed data coming in different structures, and it becomes a challenge to store this non-structure or structure-free data into relational system mentioned above. This is needed only if the platform we use is schema oriented because relational database systems need a fixed structure of the schema with proper relationships pre-defined so that data search and query process is fast. Although the growth of these relational databases from server machines to cloud setups has increased their access to a wider audience, but their lack of ability to handle always changing data structure needs options to explore new storage platforms. NOSQL (known as No SQL databases or structure free databases) is a new generation of storage systems where each instance of the data is considered a document and there is no limit on the number of attributes each document can have. One such commonly used option is MongoDB, which allows storing data without defining a schema. The data is indexed on every attribute, and therefore the access to the information is relatively faster than relational database systems. Another promising database system is Kognitio [157] that provides in-memory processing of the data and still maintains the structure of the data. It does not use the standard Transaction SQL (TSQL) language to store or access data from the database, and it has a purpose-designed script, which is very similar to SQL but has been optimised for better performance.

As mentioned earlier, the volume of data is increasing at high speed, and the database systems have to support the storage for this data. No data is useful if it is not retrieved back for processing. So, the storage system chosen for data must support a scalable access to the data as well to ensure its processing. The relational databases like MS SQL Server provide good performance through indexes on the schema attributes but using more indexes makes the insertion process slower. The authors (Fojtík et al.) proposed a technique to store large volumes of data in SQL Server by splitting it into one primary database that keeps data for last one month [158]. The data older than one month is moved to dynamically created databases. This can keep the query performance better by maintaining the size of the primary database to be not more than 2GB (this varies from schema to schema and type of data being saved). This option is suitable only for the systems where the query is always made on the primary data, and the older data is never needed for processing. Although the use of indexes on relational databases can increase the performance of the queries [160], the faults and failures of SQL Server were elaborated in [159] where different deadlock situations were pointed out that can cause severe performance issues with SQL Server. An example is to apply an index on a column of a large table that already has a lot of data that takes very long time and can potentially stop access to the table for other queries.

A comparative study of relational databases (MySQL and Oracle) [161], [162] was performed with NOSQL database like MongoDB to identify what competitive advantage is offered by aggressively developing NOSQL growth of schema-free database systems. MongoDB (and another popular NOSQL database called Couchbase) is a document based database compared to fixed-schema options like MySQL and Oracle. MongoDB supports conceptually different data types: JSON, BSON, XML and BLOBs with uniform access to all data with one interface. Every document that is stored in the system is converted into a deep implementation of key-value access so that joining data from different sources/types is easier without explicitly defining the relationships between them. In contrast to the relational databases to have a mandatory requirement of a fixed-schema, the MongoDB's one-point access to all types of data and with much faster speed than counterpart SQL databases makes it a good choice for the dynamically varying structure of data.

The context data, like the data coming from sensors of a bus system, is more suitable for storage in NOSQL systems where possible. The reason is the development of new sensors and enhancement of existing sensors to collect more and more data about the context of the vehicles, and that needs a schema-free access so that information collected from any type of source can be stored without worrying about how its types and attributes will be handled. An example is the data about the location of a bus. If the sensor installed on the bus provides latitude and longitude about the location of the bus, if a new sensor provides extra information like position in multiple dimensions and height of the bus relative to the road; we will need a new schema to store it if we are not using a NOSQL database. Storing dynamically changing information in relational databases will always need to change the schema of the database, and that increases the time to implement a feature or function in the system. The choices like Kognitio is also an option to consider because it maintains the structure and still has implemented techniques for fast access even when the data is coming for storage very frequently. Kognitio has very flexible integration with data mining platforms like R, so that processing large volumes of data are possible for pattern recognition and prediction without using different mediums for storage and processing. Kognitio provides a feature of "external script" that allows integration with R script and combines high performance computing power of Kognitio with extensive data mining ability of R. This helps in working on the data without loading into memory like platforms like R do when they make an ODBC connection to SQL Server, which fails on large volume of data because the process need to load all the data in memory.

3.2 Data Challenges in an Intelligent Transport System

Different data storage platforms have been discussed in the previous Section 3.1. The processing of data depends on how fast it can be accessed from the storage but it also depends on the technique or the algorithm that is used to process the fetched data. The data generated from an ITS comes from many different sensors and there are many challenges that need to be addressed before a formal analytical technique can be applied. Some of the challenges are mentioned below:

1. The data needs to be cleaned for wrong entries in the dataset so that these wrong entries do not influence the results.
2. The data from different sources needs to be integrated so that impact of different attributes can be evaluated for consideration of relationship between different attributes within the same data source or across different data sources.

3. The number of combinations of the values in a complex system like ITS can be exponential because number of incidents such as arrival events for a bus can be in millions.
4. The volume and frequency of data coming from thousands of vehicles driving on hundreds of routes resulting in hundreds of thousands of arrival, departure and location events make the first three points even more complicated to handle.
5. The processing of this data, in real-time, once it has been made ready for analysis.
6. Real-time visualisation of the results from predictive analytics so that the data can be effectively used for predictive analytics.

Data mining techniques offer diverse number and types of algorithms based techniques to process the data and they can be very helpful because most of these algorithms have already been tested for the function they perform and lot of algorithms, as explained in the next Section 3.3, can help address the challenges mentioned above. A review of data mining techniques that can be potentially used for context data of a public transport system is presented in following sections.

3.3 Data Mining Techniques

This section presents a review of data mining techniques found in the literature and focuses on elaboration of significant techniques that are relevant to the work done in this Thesis. Data mining is the extraction of implicit, previously unknown, and potentially useful information from data [153], [154], [157]. Data Mining Techniques are a set of algorithms (such as clustering and classifiers etc.) that help in the identification of information patterns that exist in the data being analysed and then classify them so that the data elements can be put in groups for a better understanding. This segregation of data instances helps in knowing the unknown and unusual piece of information that exists in the knowledge and converts that into knowledge so that it can be used for decision making based on the domain of data and the knowledge that is produced as a result of the data mining process. Different data mining techniques serve the purpose of analysing the data and present their output based on the algorithms working behind the technique. Machine learning is the technical basis for data mining. It is a set of algorithms for acquiring structural descriptions [157] such as structural descriptions that represent patterns explicitly. Machine learning algorithms can be used to predict outcome in a new situation; can be used to understand and explain how prediction is derived (maybe even more important). These methods originate from artificial intelligence, statistics, and research on databases.

There are primarily two types of data mining techniques known as Supervised and Unsupervised. The unsupervised learning includes a set of machine learning algorithms that can draw inferences from datasets without already labelled responses. The most commonly used unsupervised method is called cluster analysis used to detect patterns in the data through exploratory data analysis. Depending on the number of attributes for each entry in the dataset, clusters are modelled using many similarity measuring techniques such as Euclidean or probabilistic distance. Some of the commonly used clustering algorithms are Hierarchical Clustering, K-means Clustering, Gaussian Mixture Models, Self-organizing Maps and Hidden Markov Models. Another unsupervised method is association that helps to discover the rules for description of large amount of data to indicate association between different rules created. A typical example is in the retail market where association techniques can help in identification that a customer will also buy Y product if he bought X product. Apriori algorithm is an association rule technique that can help derive rules from the dataset along with relationship between different rules so that a combination of rules can be used to establish decision making based on those rules. Supervised learning, on the other hand, has input variable(s) and the output variable based on the values of the input variable(s). The supervised learning algorithms learn the mapping between input and output variables and derive a function that represent the mapping. The objective of applying supervised learning is to approximate the mapping function so that output can be predicted for data with any new inputs, even if the inputs were not included when the mapping function was created. This is achieved through splitting of the dataset into training data, which is the data with input variables and their expected output value, and test or case data that is used to evaluate the accuracy of the produced prediction. If the mapping function does not produce the correct results, there are optimisation techniques to tweak the mapping function so that error in the function can be minimized. The supervised algorithms can be further grouped into classification and regression algorithms. The classification problems are where the output value of a variable is a category whereas regression problems have a real value as an output. There are some algorithms such as Linear Regression that handle only regression problems only, Support Vector Machines that address classification problems only and algorithms like Random Forest that address both classification and regression problems. Neural

Networks is another set of supervised algorithms that can be used for both classification and regression problems.

There are many data mining techniques, which help in the analysis of data and finding patterns to produce useful knowledge out of the data. It is helpful when the data has many dimensions and it is important to either join different dimensions into one or discard the ones that have least impact on the outcome. Principal component analysis is useful for identification of important dependent variables so that unnecessary data can be removed [163]. It can be used in collaboration with dimension reducing algorithm for the cleanup of unwanted data instances. Collaborative filtering is used to convert the raw data into recommendation dataset using the relationship between profiles of different entities in the dataset [164]. Association rule, like collaborative filtering, and Link prediction identify association between different data instances and then produces grouping of related entities and events related to them. Different probabilities are mentioned and derived from data that show comparative correctness or relevance of the association [165], [166], [171]. Data Summarisation algorithm can work as pre-processing stage processing for Association Rule and Link Prediction to produce summary of the data for basic understanding of the data patterns [175].

Latent Dirichlet Allocation (LDA) is a generative algorithm used for sentiment analysis and text mining that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar and how can observations be used for making predictions [168], [169]. Another algorithm similar to LDA is Text Mining algorithm helpful for extracting information, which is comparatively high quality than the redundant data it is derived from [178], [179], [182]. Text Mining can be used in conjunction with clustering algorithms to cluster the information for a higher level abstraction/visualization of the data being processed [180], [181]. Dijkstra's Shortest Path is useful for the data where instances can be represented as graphs and the analysis needs to identify shortest paths based on decided criteria factors such as time and cost [170]. Anomaly detection algorithm identifies inconsistent patterns in the data that standout from expected information and find those patterns in data that do not comply with known data patterns considered normal [176], [177]. Each of these techniques discussed above can process the data as per the logic behind the algorithm and many times combination of more than one algorithms, such as clustering followed by regression, are used to understand the data being processed. Algorithms like Clustering, Regression, Decision Tree and Neural Networks are discussed in more detail because of their relevance with some existing bus arrival time prediction systems and their use in the work of this thesis. The choice of algorithms implemented in this Thesis could be more advanced where better sampling and modelling algorithms can be used but we implemented a proof of concept strategy to indicate the application of these classification techniques. The design of the predictive analysis, as described in detail in Chapter 4 and 5, is kept independent of the choice of algorithms or techniques and therefore, any advanced algorithm can replace the ones being used in this work and improve the quality as well as functionality of the produced prediction data.

3.3.1 Classification algorithms

Classification tries to predict, for each individual in a population, which set of classes this individual belongs to. A classification task, given a new individual, determines which class that individual belongs to [175], [181]. It may also assign a probability to this association. Few of the classification algorithms are elaborated in this section followed by description of regression algorithms that can potentially be used for data processing in intelligent transport systems.

3.3.1.1 K Nearest Neighbour (KNN)

KNN is one of the fundamental classification methods used to perform discriminant analysis when little or not information is available about distribution of the data and parametric estimates of probability density are unknown or hard to calculate [184]. It uses Euclidean distance as a method to calculate distance between the test sample and the specified training samples. It performs better if the features of the dataset are transformed before they are input to the classifier. Two commonly used techniques to transform the features are standardization and fuzzification. The Standardization technique removes scale effects so that features with different scaling units can be standardized to avoid the bias of scale values whereas fuzzification transforms by replacing original features and creating three fuzzy sets from each of the original value.

3.3.1.2 K-Means

K-means groups the data instances on the mean distance calculated on the similarity criteria and this similarity is based on closeness of the attributes with mean of a cluster [158]. K in K-means represents the number of clusters that the algorithm will try to create based on the configuration applied to the technique. Clustering is a classification technique, which helps in dividing data objects into different clusters where objects in one cluster are more similar to each other than the objects in another cluster. The criteria for similarity between members of the cluster is the driving force in the design of different algorithms because each criterion can address a specific problem domain and identify the similarity index specified for the algorithm [159]. The similarity in a public transport system can be based on zone in which buses have similar delay behaviour, buses going to the same destination, buses going through similar route and buses driving in a specific timeslot. Few of the techniques based on the clustering criteria are Connectivity based clustering, Centroid-based clustering, Distribution-based clustering and Density-based clustering. Some of the criteria types used by these algorithms are like density (grouped based on the density value or density value in range), connectivity (buses having similar delays on a set or range of stops items can be a connectivity factor between different buses so they should be clustered accordingly).

Clustering models identify relationships in the dataset that are not visible through casual observation. It becomes complicated when there is lot of data and there may be many data points for similar attribute values. The clustering models will identify the relationship between these occurrences and look at each attribute that is being input to see which cluster a data point should belong to. After the initial definition of clusters, the clustering algorithm validates ability of each cluster for its representation of the data points it contains. This is an iterative process through which the algorithm keeps checking each point for its relationship with the cluster it belongs to and the possibility of attaching this data point to another cluster that it may have a closer relationship with. The data required for clustering models generally contains a single key column, input columns and an optional predictable column. The techniques like K-means and Expectation Maximization (EM) have a basic difference in the way they work where K-means is a hard clustering technique as it puts one data point strictly in one cluster where as EM can put same data point in different clusters if it discovers that data point has attributes belonging to more than one cluster.

The clustering models can be customised with several parameters like clustering method, clusters count, cluster seed and minimum support that can change the behaviour, performance and accuracy of the model. For example, selection of EM over K-means offers some key advantages like it requires one database scan, works in limited memory and its sampling approaches perform better than K-means' sampling approaches. EM has strong statistical basis, it is robust to the noisy data, can handle high dimensionality and it has the potential to converge fast if initialized properly [183]. The number of clusters has a major impact on the behaviour, performance and accuracy of the model because less their clusters are, more data points will belong to each cluster and because there is limit on the number of clusters that can be made, some irrelevant data points can also exist in the cluster. This relationship can exist on how less different (rather than more similar) data points are with the cluster they belong to compared to other clusters it was not picked to be put in. The cluster seed can change the initial behaviour of the way clusters are generated. The default is zero, but we can pass different seed values so that we can compare different models generated with different seed values. If the clusters do not change much with changing seed values, this is an indication of relatively stable clustering model. The minimum support attribute can help in cleaning up the clusters that are not important. We can mention a minimum number of data points that a cluster should have. Any cluster that does not have that many data points is considered an empty or discarded cluster. However, it is very important to understand that the data points that exist in empty or discarded clusters may reflect outliers or rare events and therefore if the data points in empty or discarded luster should be checked further for their relevance separately to avoid ignorance of important data points.

3.3.1.3 Hierarchical Clustering

Hierarchical Clustering, also called hierarchihcal cluster analysis creates a hierarchy of clusters using primarily two strategies [172]. The bottom-up approach, Agglomerative, where clusters are observed and merged with the cluster moving upwards in the hierarchy and top-down approach, divisive, where a cluster is observed and then split into further clusters with a cluster moving downwards in the hierarchy. The clusters are joined together or divided into sub-clusters based on the similarity found between different clusters. In the bottom-up approach, it starts with putting each data point in a separate cluster and then measuring the similarity between them. Once two items are put together to create a cluster, then its similarity is checked with other data points and clusters. The distance or similarity between two clusters is measured based on similarity between one of the items in one cluster with any item in the other clusters and they will be joined together if the similarity is higher than other clusters available in the system. Similarly, in top-down approach, the clusters are split into further levels of clusters based on the similarity and the clusters with closer similarity are kept on the same level and are further split based on similarity between data points of each cluster. This algorithm can only be used on the large dataset in Hadoop like systems if the size of data in each node of the cluster is within the range of this algorithm handling it gracefully. Also, application of this algorithm in evolving data streams needs to be evaluated as the metrics used by this algorithm like Euclidian distance, Manhattan distance and maximum distance for clustering of the information may not be applied to data packets arriving in data streamIt greatly depends on the number of attributes in the data and can be very slow in processing the data if there are many levels of attributes and each attribute linked with huge number of data instances. The data from a public transport system can be processed using this algorithm if different attributes being collected from sensors on the vehicles can be prep-processed into less dimensions of data.

3.3.1.4 Neural Networks

Neural Networks (NN) creates a network that is composed of up to three layers of nodes, also called neurons, named as an input layer, hidden layer and output layer. NN algorithm is based on human understanding of the problems and it creates data neurons from the sample being studied. The relationship between different neurons is established based on output expected from the algorithm because generally output is already known [163]. NN works by checking each state of the input attribute against each possible state of predictable attribute and calculates probability for each of these combinations for all input variables. These probabilities can be used for both classification and regression. NN can also have multiple output attributes and that will generate one network for each output attribute. It means that the probability of each input attribute with each of the output attribute will generate a network and for example, three output attributes will produce three neural networks from the same set of input attributes. Weights are allocated to each attribute based on the perception and each of these training data instances is called perceptron with each of them being considered relevant based on the weight given. When training data is applied to these perceptrons, relevance of weight allocated is checked and if it is not accurate then the weights are changed for the attributes again. By repeating this process, best weights or perceptions are identified that can produce the required result.

There are many types of techniques to train a neural network and one common technique used is Multilayer Perceptron network, also called Back-Propagated Delta Rule network, that uses input and output layers of the network with optionally using hidden layer when asked to. Each neuron, in the Multilayer Perceptron network, receives one or more inputs and integrates with the hidden layer to produce one or more identical outputs. Each input neuron is connected with nodes in the hidden layer and the nodes in hidden layer are connected with output neurons with no connection between neurons in the same layer. As hidden layer is optional and when it is not used, the inputs pass forward directly to the nodes in output layer. After the algorithm extracts training data from the sample based on the percentage specified, it reserves a percentage of the training data for assessment of the network accuracy. This reserved percentage is called holdout data and it is used for accuracy assessment after each iteration through the training data. Multilayer Perceptron network has many parameters that control behaviour, performance and accuracy of the algorithm. Holdout data size and seed can change the accuracy assessment by determining how many cases are placed in the holdout data and which cases are put in the holdout data.

Each input is assigned a value called weight that indicates relevance or importance of the input variable and its impact on the relationship with hidden or output neuron it is linked with. The value of weight assigned to the input can be positive or negative with positive value indicating importance and negative value indicating the lack of importance. These weight values can completely change the way a neural network is created for a set of inputs and outputs. They can be used to give importance to some specific inputs more than other inputs even when the default data may be supporting different set of inputs. For example, when we have factors like day, hour , location and distance attributes and we want to build the network around distance more than the hour attribute, we can increase weight for the distance attribute and negative weight value for hour – that will increase impact of distance and reduce impact of hour attribute, which is achieved by multiplying weight of each attribute with its value. Each neuron in the network is attached with a non-linear function (activation function) that describes importance of the neuron to layer of neural network it belongs to. Hidden layer neurons use *hyperbolic tangent* function and output layer neurons use *sigmoid* function for their respective activation. These activation functions help the neural networks to model non-linear relationships between input and output attributes. There are many feature selection methods used in building the neural networks like Interestingness score, Shannon's Entropy, Bayesian with K2 Prior and Bayesian Dirichlet with uniform prior and scoring methods like Z-score are used for encoding the attributes so that weighted sums (based on weight assigned to each input attribute) so that input value is more evenly distributed on a uniform scale.

Other commonly used techniques to train a neural network are Gradient Descent, Newton's method, Conjugate Gradient, Quasi Newton and Levenberg Marquardt. Gradient descent, which is also called steepest descent, is one of the simplest neural network training algorithm that is a first order method because it uses information from the gradient vector. The error function is optimised at each successive step by settings the training rate to either a fixed value or using one-dimensional optimisation. Each step computes training direction of gradient descent and suitable training rate is found, which can be a fixed value as well. This method can be slow because it may need many iterations for long and narrow valley structures and it does not necessarily produce a faster convergence. It is however recommended for very big neural networks having thousands of parameters because it stores only gradient vector and not Hessian matrix. The Newton's method uses Hessian matrix and calculates second derivatives of the loss function to optimise the training direction. The training rate, just like gradient descent, can be wither fixed or calculated through line optimisation but measurement at each step may result in maximum value and not the minimum as intended. The reason for the higher value is because of the Hessian matrix being not positive definite. Although Newton's method takes less steps compared to the gradient descent to find minimum values of the loss function, it is computationally very expensive to evaluate Hessian matrix and its inverse.

The Conjugate gradient algorithm is an optimisation for slowness issue of the Gradient descent algorithm and computational complexity and expensiveness of the Newton's method. The motivation behind Conjugate gradient is to avoid the excessive information requirement by Newton's method for storage, evaluation and inversion of the Hessian matrix as well as accelerate Gradient descent's slow convergence. This is achieved by searching along conjugate directions to accelerate the convergence and conjugating the training directions with respect to the Hessian matrix for reducing the information requirement. These optimisations make Conjugate gradient a better algorithm than Gradient descent to train neural networks and it is recommended for bigger neural networks because it does not need Hessian matrix inversion like needed in Newton's method. The Quasi-Newton method is an improvement on the Newton's method in terms of its computational expensiveness, and is also called variable matrix method. At each iteration of the algorithm, an approximation of the Hessian inverse is measured rather than calculating direct Hessian calculation followed by inverse calculation. The approximation of the Hessian inverse is build using first partial derivative of the loss function and that is why exact computation of Hessian matrix and its inverse are not needed and yet, it provides better performance than Gradient descent and Conjugate gradient.

Levenberg-Marquardt algorithm, also called damped least-square method is a neural network training algorithm that is designed to work with optimisation of loss functions that are measured as sum of squared errors. It does not compute the exact Hessian matrix and works with a gradient vector and the Jacobian matrix. The Jacobian matrix contains derivatives of the errors linked with the parameters resulting in the error. For these specific types of errors, measured from sum of squared errors, Levenberg-Marquardt algorithm is fast at training the neural network compared to the Gradient descent and Conjugate gradient buts its problem is that it is applicable to very specific type of loss functions. Another problem with this algorithm is that its requirement of memory grows with the size of the neural network and therefore, it is not suitable for bigger neural networks. Each of these algorithms has its advantages and disadvantages depending on the size of the neural network and the available resources. The Gradient descent and Conjugate gradient are suitable when the neural network has many thousands of parameters and they work in less memory compared to Levenberg-Marquardt algorithm that works best with neural networks when number there are a few thousands of instances with a few hundred of parameters.

3.3.2 Regression Algorithms

Prediction algorithms are used for predicting the value of a certain value and regression is one such algorithm [167]. For example: How much delay will a given bus have on a specific stop? The amount to be predicted here is service usage, and a model could be generated by looking at other, similar arrival events in the population and their historical usage. There are primarily two types of regression techniques used to process the data. Single regression is applied where there is only one independent variable and prediction of the dependent variable depends solely on the occurrences of one independent variable. Multiple regression is used when there are more than one independent variables that are not only associated with each other but they have a combined impact on the value of the dependent variable. The value to be predicted can be of any type from finding a true/false value for dependent variable, prediction of the percentage or value for the dependent variable or prediction of an expression like percentage of the occurrences. When more than one independent variables are used, the coefficient for each variable is identified so that data instances from the test data can be combined with these coefficients to produce prediction probabilities for dependent variable.

By default, the algorithm picks the most suitable input variables as regressors and generates prediction coefficients for each input variables. However, it is important to note that increasing the number of input parameters does not mean the regression algorithm picks all of them while training the model. The algorithm can potentially ignore an important input attribute and therefore, the prediction coefficients should be investigated against the data and input attributes so that the results reflect maximum population of data. The *minimum leaf cases* argument of the algorithm differentiates from trees behaviour of splitting the training data into more than data elements like decision trees algorithm does, so the value of *minimum leaf cases* should be more than the number of cases used for training the model. A regression line is derived from the input variables depending on the number if input attributes, the equation is of the form as shown in the Equation 1.

$$y = B + Ax$$

Equation 3-1: Linear regression equation

B is the regression intercept, x is the input variable (in this case only one), A is regression coefficient for input variable x and y is a function of adding intercept to summation of the product of input variable values with their respective coefficient. The coefficients are adjustable to help in giving higher or lower importance value to a specific input variable to achieve desired results. One important parameter of regression algorithm is *force regressor*. It can be used when regression line is to be predicted for some specific aspect of the data or when the algorithm does not pick the correct set of input variables and we can enforce the algorithm to consider specific set of input variables.

3.3.3 Time Series Algorithm

A thesis on data mining, optimization and simulation tools for the design of Intelligent Transport Systems (ITS) presents a comprehensive review and implementation of data mining techniques for improving its services and showcased application of Time Series algorithm to enhance dynamic optimisation of traffic signals [169]. The dynamic variation of traffic for public transport has a pattern attached with time of the day along with occasional association of unavoidable incidents like emergency and accidents. The thesis uses an iterative and incremental model to convert the data into different timeslots using road intersections as centroids and representation of roads as connectors that join different intersections. The flow of traffic and operational data of the signals was modelled and a forecast of signal times was produced that optimises operation of signals for better flow of the vehicles with minimized waiting time on the signals.

3.4 Bus Arrival Time Prediction Systems

This section reviews existing bus arrival time prediction techniques presented in the literature and critically analyses their relevance and shortcomings when applied to a larger scale or real-time ITS. There are two key elements of a public transport system, arrival time and departure time, that uses buses and trains to transport passengers from one place to another [58]. The people waiting on a stop need to know when is the next bus coming to a stop and how long will it stay on the stop to depart for its next destination. The accuracy of arrival and departure times represent the reliability of this information and therefore, can encourage passengers to use the service because they can plan their journey based on this information and can expect that they will reach their destination in expected time [59], [74]. The primary source of this information comes from the planning of the routes and the time of different buses expected to drive on those routes. The bus companies make annual or biannual plans to indicate the number of buses serving a route with expected time of arrival on stops of the route for each bus throughout the day. The density of the traffic, as well as usage of the service, is considered in planning these routes and bus times so that different type of traffic and consumption at different times of the day can be handled by the bus system [60], [73]. These are all considerations made for factors that are either internal to the system like how many buses are available for a route and the number of stops that need to be served; or the external factors that are already known to peak and off-peak times, weekday and weekends and any planned holidays or events.

Although there are bus lanes designed for smooth operation of the buses on roads, so that impact of congestion and heavy traffic load is minimized on the buses, but it is not always the case especially in Metropolitan cities where roads are shared between public transport and other traffic on the road [61]. There are many external factors that have a close impact on the way buses run on the road and their ability to meet the expected arrival times as indicated in planning their routes. The density of the traffic considered at the start of the planning cannot be a true indication of the actual traffic density because it is just an estimate based on the history data and observations. The incidents that can happen anytime on the road are also not part of the consideration because such incidents, especially emergencies, can potentially bring the traffic to a standstill for a reasonable amount of time. There are many events that are not planned up front, like a walk or sports event, and they attract a lot of pedestrian and automobile traffic to a specific place [75]. The buses operating in that area are automatically effected and their timetable is not followed as expected because of the delays incurred by this extra traffic and pedestrian considerations [62]. The extra pedestrians contribute to the users who may be using the buses from going to/from the location where the event or incident is taking place. This extra usage can also cause the buses to stay on the stops more than the planned time, to accommodate extra passengers boarding or taking-off the buses, and therefore causing delays on the stops to come in the route.

The GPS coordinates reflect the physical location of any object on the ground, and they are widely used aboard vehicles, and buses are no exception to it [63], [64]. This is the only major indicator of moving bus and eventually making progress with the journey or not. All the stops on the route also have GPS coordinates attached to it, so in theory, this is always possible to calculate the distance between the bus and next target stop it is driving to. Although there are many different map projections to convert the spherical or spheroid shape of the earth into a flat surface to perceive the roads as we see, each map projection strategy provides features to calculate the distance between any two points on the map. The bus systems can use these features to calculate the distance between the bus and the target stops and calculate the distance to use it as parameters in measuring the progress. Correlating this distance with the bus speed on the road can give an indication of the expected arrival of time on the next stop [72]. The speed of the bus can also be derived from GPS coordinates of two different points the bus has gone through and then measure the distance covered against the times taken between those two points. The amount of progress that a bus makes contributes how much the bus will comply to the planned arrival time on the stops. Other external factors that can potentially impact the progress of a journey being made include weather, roadworks and connection with other modes of transport to facilitate the passengers to make the best use of their time. An effective planning of routes and times of the buses on stops should help passengers spend less time on waiting on stops and completing their journeys [65], [70, [71].

Because of all these internal and external parameters identified, there should be an adaptive system that can consider all these parameters and forecast arrival times on the stops so that the passengers on the stops are aware of any possible changes if the buses are not able to meet the time mentioned in planned routes. This information is shown on different type of signs displayed on the stops to communicate to the passengers about progress being made by the bus travelling to that point [66]. This information is also integrated with many different services and applications that may be used by the bus companies as well as passengers to stay aware of the bus times and plan their journeys according to the updated information [67]. There is a lot of research into designing effective techniques and algorithms so that an accurate and adaptive prediction system is in place to help both the bus companies and the passengers. An important internal factor that provides deep insight into the behaviour of a bus system is the history data, which shows arrival times against planned times for the stops for a longer period of time. The techniques can be based on this history data as well so that forecasting is possible for the situation, when there is no information available about external factors and even GPS position of the bus, is also not available. There are many approaches to predict information about transport systems presented in [80], [81], [82], [83] and some of those techniques to make a prediction of arrival times are discussed in the next sections [68], [69], [79].

3.4.1 Clustering based Prediction

A dataset of trips covering a period of 83 days from London Underground was analysed using different models based on clustering to propose personalised intelligent transport systems [167]. The trips were aggregated using number of ongoing trips over time and mean trip time between two nodes. The commuter characteristics were also used for grouping them in the same categories if their pattern of traveling is similar. These patterns were derived based on different commuter characteristics like repeating the same trip over time and their traveling preferences on a weekday or weekend. They applied agglomerative hierarchical clustering and used its implementation version of dendrogram clustering [168] and controlled hierarchies of clustering by limiting the hours of the day with dividing a day into 5 segments. Prediction was designed using Self-Similarity (similarity between different trips of the same user) and Trip-Familiarity (similarity between user trips and overall mean of all trips) models. The self-similarity model produced better results for making predictions with least Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). These models can easily be applied to buses because the stations can be represented by bus stops and trip time can be the time taken by a user from one bus stop to another.

3.4.2 Timetable Adherence and Dwell Time based Prediction

The study [76] applied an artificial neural network model to automated vehicle location (AVL) data to develop a prediction model for bus arrival times. Two aspects of the data were considered in this as adherence to the planned timetable along with stay time (dwell time) spent on stops to understand its impact on arrival time on the stops to come. They analysed the data collected from Southbound Route 60 in Houston where the GPS location of the buses is reported every five seconds. Three input parameters of: arrival time, dwell time and the schedule adherence (calculated as the difference between actual arrival time and planned arrival time) were used in the study to build and apply the model to the prediction of the estimated arrival time. The data was clustered to weekday and the time of day to cover different variations for peak and off-peak time and special consideration for weekends because even peak times on weekends are not as busy as they are on weekdays because of offices and the schools. Although the study [76] concluded that the artificial neural networks model is more effective and accurate compared to two other models, a statistical model based on history data and a regression model, it did not consider the external factors of traffic derived these from the schedule adherence of buses. It also did not consider the schedule adherence when it comes to connecting the passengers to other modes of transport. Their technique was purely estimation without its integration with the real-time system to integrate on the fly changes in the input parameters.

3.4.3 Historical data based Prediction

The data about external factors like weather, roadworks and real-time data about congestion of traffic on a specific route or part of the route is not always available. Also, there are many bus services, like provided by the industries to commute their staff members from different parts of the city, that do not have access to separate bus lanes. Therefore, buses on these specific routes have to be part of the normal flow because they are very specific application of a public transport system serving a limited community of passengers. The study [77] designed a lightweight model to predict arrival time of buses for a bus service being driven by the industry for transporting its staff members to and from their homes to the offices. They employed historical data based model to make the prediction and then evaluated its performance against the application of artificial neural networks (ANN) and support vector machine (SVM) algorithm. Contrary to the study [76] that concluded ANN to be performing better than historical approach, the results from [77] supported the historical approach as performing better. This new study is still in use (73.8 Km) and the route used by the industrial bus service and collected data through devices installed on the bus. Although [77] does not mention the frequency of the data recorded by the onboard devices like was the case in [76], it does perform clean-up of the data to ensure that anomalies and errors in the data are removed before applying the model. The input parameters to this model are only the timestamp, and the GPS location reported and then everything else such as speed is derived from the two input parameters. That limits the applicability of this model in a real-time scenario where the connection between different stops is indicative of the delays being projected from one to the stops to come.

The quality of the dataset collected in the study [77] is not good for making an effective prediction system for many reasons. The study itself pointed out that the devices used for reporting the data were not accurate always and recorded wrong GPS coordinates. Also, the algorithm employed to pick the correct timestamps closest to the stops returned values from a good distance to stops sometimes because of invalid data reports, so a lot of anomalies were detected and then removed in the clean-up process. The removal of data instances from the data based on their inaccuracy also decreases the data on which the model was trained and therefore, potentially missing out scenarios where there could be a better match of data instances with the instances being predicted. With all these constraints, the technique calculates speed, and the distance covered from between two stops and then tried to predict it based on that for the future stops or points. The median speed calculated is for every fifteen minutes which is not a good indicator of the speed calculation because it covers a larger interval of time and the bus may have undergone various speeds based on the traffic and median could be biased towards lower or higher range depending on some reported occurrences in each.

3.4.4 Social Sensing Enhanced Arrival Time Prediction

Different social events can have an impact on the way buses operate in the area where the event is happening. With the increase in products and services in the form of apps and websites for social media and trend of using these platforms like Twitter, Facebook and Meetup, these services have become a source of information that needs to be integrated with public transport systems. This will materialise the impact of any such events on buses driving in that area and can potentially increase efficient of a bus system by being more aware. The study [78] applied the principle of social sensing including attributes of physical road conditions and the ability of the buses to commute the passengers to the destination point where the event is happening. The high ridership of passengers to a specific point is taken as an indication that there is some event happening because this riding behaviour is not visible on normal days. The study [78] found that in addition to the increased ridership of the buses in a specific area, extra data is needed to find out the time and duration of events so that the arrival time prediction system can incorporate this expected change in ridership of the buses.

The researchers [78] collected events from different social platforms and converted each event to a localised event that is associated with time, ridership and situation of the road use for different type of users (based on disability, age groups, ethnicity, etc.). They applied an artificial neural network model to this data where the events are used as input variables to determine the arrival time of the bus on a stop. The study presented a very good concept where every external event can be considered a physical road attribute in a specific zone and then its impact on the real-time arrivals in that zone is calculated based on data mining techniques like ANN. However, the study is very vague in the formal identification of the impact of each event because not every event (like social events happening in an area or multiple events happening in an area involving a diversity of passengers) can be measured for their expected impact on the calculation. Also, these events are not frequent, and there is not enough data available for measuring the role of the event in delaying the buses running in that area. If there is a validated source of these events with clear information about the time and duration, type of attendees, some passengers expected to use public transport and the diversity of passengers using the buses; the bus prediction system can incorporate these parameters more effectively.

Another interesting work used information about the events to predict the number of bus arrivals expected closest the venue of the event [166]. The data is collected from event websites through established APIs and then it is categorised for processing through the data mining algorithms. The implemented model was not designed to make any arrival time predictions, but estimate the number of buses expected to arrive in that area with an extension to possible delays in the arrival of the buses. They used the typical architecture of a Neural Network containing one hidden layer, activation functions, back-propagation learning and regression output. The association of neurons in the hidden layer is shown in Figure 3.1 that indicates different set of neurons representing venue, time identifier, weekday/weekend and event categories. Neural Network was chosen after preliminary application of many techniques like K-Nearest neighbour, Gaussian processes and regression trees etc. but Neural Networks performed better based on Pearson's correlation coefficient (CC), mean absolute error (MAE), and root mean squared error (RMSE) calculated for each algorithm.

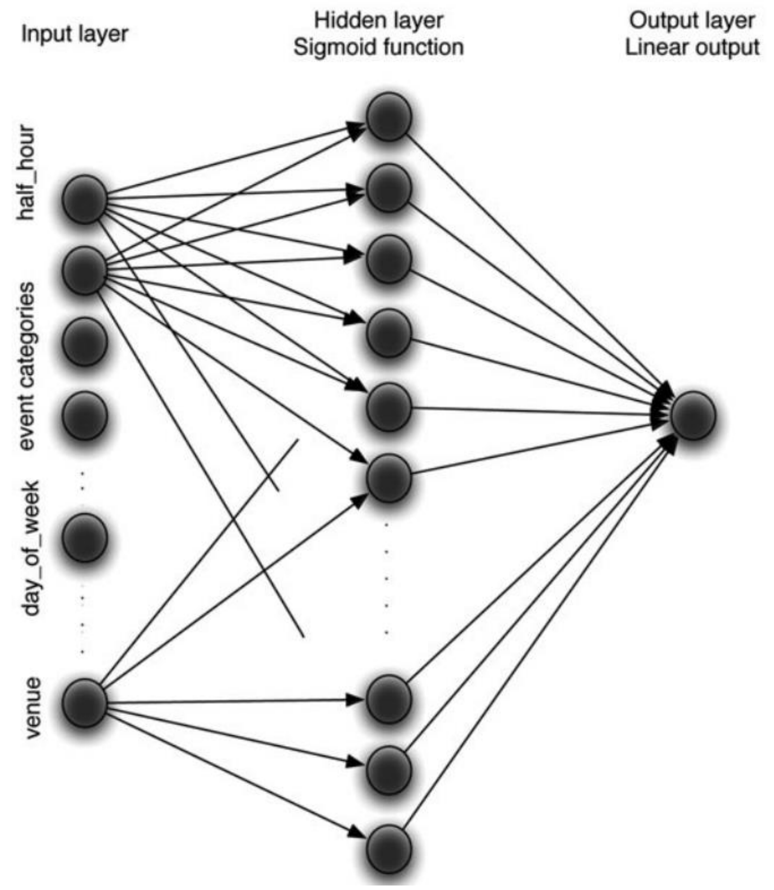


Figure 3-1: Typical model architecture of Neural Networks algorithm

3.5 Summary

This chapter presented different challenges in handling the data along with a variety of data acquisition and storage techniques that can facilitate better manipulation of data before it is stored. It also enabled efficient post-storage processing. The understanding of context data based on the explanation in Chapter 2 requires that the maximum number and types of sensors to be installed on buses, and these sensors produce data that should be stored for effective analytics. The data coming from all these sensors is closely linked together because the combination of all the data attributes constitutes the context of the bus. A variety of database management systems were briefly discussed and comparison between them particularly in terms of suitability for contextual data was elaborated. Tools like Microsoft SQL Server, Oracle, MySQL and what they offer in handling context data along with potential limitation they offer were discussed. The growth of NOSQL databases and their potential to store and manipulate schema-free data coming from all variety of sources was also briefly presented. Kognitio, a new promising database solution, was discussed that provides very good performance even with the schema-based data models.

Many different data mining techniques were discussed that could be used to process the collected data and generate prediction models based on different dimensions of the data. The data is information, and it is converted to knowledge after manipulating through the application of intelligent machine learning techniques. If the data is collected from a system, and then the knowledge derived from this data is applied to strategic decision making and automation of the same system, it is bound to help in improving the service and operational excellence. Out of all the techniques mentioned and major area of their application, techniques like Neural Networks, Regression and Clustering were discussed in more detail because of their use in existing bus arrival time prediction systems. The existing bus arrival time prediction systems were reviewed and compared in order to understand what work has already been done.

The public transport systems around the world collect data from buses, trains and trams and then develop prediction and forecasting systems to improve the customer confidence and reliability of the service. An automated monitoring of patterns in the data can be useful as it can potentially change behaviour of the transport system. It is possible through intelligent integration of data mining techniques, in the data acquisition and storage process, that important data can be extracted for decision making. The data mining techniques help not only in prediction of bus arrival times on stops but we can use them to identify behaviour and situation of the network based on arrival information available for different stops on a route or across different routes. This highlights that data-driven decision making can help design and improve public transport systems of the future. The next chapter will focus on justification of the new proposed architecture, based on a review conducted for intelligent transport systems in Chapter 2, which also integrates and evaluates a suitable selection the data mining techniques discussed in this chatper.

CHAPER 4: DESIGN

After reviewing and discussing many different architectures and intelligent bus transport systems in the previous chapter, it is noticeable that focus on all these architectures and systems revolves around the concept of how the system should be designed. Less consideration is given to the impact of system architecture on the performance of the system and how designing a smart architecture can help achieve the quality of service attributes. Designing a smart and intelligent transport system is not all about integrating data from different sensors and adds the ability to decide on their base, it is also an architecturally driven decision that defines how will the system behave in the production environment. The architecture also guides on benchmarking and fall-back strategies so that the system scalability, sustainability and performance in real-time can be tracked and improved. The existing architectures lack the discussion about specific design considerations like coordination of different components in real-time data processing and transformation. They also do not consider the limitations of existing transport systems and how the architecture design can help those legacy systems achieve the smart decision-making.

Another important aspect of existing reviewed architectures and intelligent transport systems, is that the existing architectures do not address the key challenges faced by the bus management systems in this era. The list of challenges include an early congestion detection, congestion patterns and future behavior of congestion, automatic incident detection and relay of impact on connected vehicles, rerouting and route optimization, handling and collaboration of vehicle arrival with other modes of transport. It is also a challenge for the bus management systems to be able to perform short-term and long-term prediction for arrival times, projection of impact of delays across stops and zones, adaptive decision making automatically by the bus system and effective resource management for cost and environmental implications. Smart bus management systems are part of the future concept of smart cities and therefore designing very reliable systems that work in active coordination with other units in the transport system and have interfaces to work with other systems in the environment is needed. Such challenges should be addressed by the architectures designed for such systems so that expectations are set and quality of service attributes are measurable to evaluate the systems for their capabilities.

Therefore, a novel architecture design is needed that is equally capable of not only building a new generation of smart and intelligent transport systems but also it helps the legacy systems connect their data to a bridge for this architecture. Although the concepts of Big Data, Streaming Data, and Data Mining are getting popular and the solutions are being implemented to handle the requirements, it is important to evaluate how it can benefit the existing systems and what are the choices. Transport companies cannot scrap their existing implementations as their solutions are widely deployed at customer sites so their interest would be how to integrate their systems with new frameworks and architectures. The proposed novel architecture is designed to consider these limitations and provides help on building components that interact with each other even though a component is representing an existing system; or, a completely new concept. It provides a solution for the specific challenges addressed and has guidelines to implement these components. It also provides complete support for integration with latest Big Data technologies like NOSQL databases, parallel processing of segregated data and efficient storage and retrieval of the data. In addition, the architecture supports short-term and long-term prediction of how the system will perform and the prediction engine is designed to provide a highly sustainable and reliable forecasting of the transport network.

4.1Justification of a new architecture – the Proposed Architecture

With experience of designing and implementing bus and train management systems with one of the largest digital signage and transport systems provider company in the Scandinavia, and after having reviewed the existing systems and architectures in the previous chapter; we have the ability and opportunity to analyze the turnkey transportation systems on the following grounds

- Existing system architectures/Challenges
- Quality of service attributes for transport system architectures
- System Intelligence Objectives
- Data Integration Challenges
- Environmental perspective

4.1.1 The Existing architecture/Challenges

A typical architecture of an existing transport system is shown in Figure 4.1 presenting server side implementation. The client connects to the server on REST services to exchange the data in XML format. The name REST stands for Representational State Transfer, which is resource-based web services model that works with six constraints to provide the interface to the function it is providing. It has six constraints that it meets when implementing an interface for the function and they are uniform interface, statelessness, client-server architecture, cacheable, layered system and implemented with code on demand. It is a traditional 2 tier (Server-Client) architecture based system where the client has a request/response driven connection with the server, and the server application handles all the storage, processing and dispensing of the data that is collected from clients. The same REST interface is used to publish the information back to the clients and devices indicating that a load on the data collection can potentially delay the data export activity. This reflects many of the architectures discussed in Chapter 2 and 3 where existing ITS architectures are discussed and potential architectures for a transport system is described. The number of components can be different from one system to another based on the number of features provided in the system, but the design is around an architecture shown in the Figure 4.1.

There are many problems in such an architecture for a time critical system like a smart transport system where information is outdated every minute about arrival of the buses on a stop. The quality of service attributes are the measures that speak of the ability of a software system to perform under all circumstances. This real-time requirement of updating information to all places, and sharing of same information among different buses connected, highlights required focus on quality of service attributes of a system and capacity of the system architecture to control the quality of service attributes like performance, response time, accuracy and extendibility. This must be spread across different components of the system architecture so that each component contributes to achieving a high quality of service in the functions provided by the system. This is one of major things that the existing system architecture lacks because different tiers in the architectures are not well connected and coordination between them is not aligned for combined performance. Only making different tiers in an architecture and then putting different components in those tiers cannot offer the level of independence in components implementation, as they still need to make sure the use of shared resources like databases tables is controlled to avoid performance deadlocks.

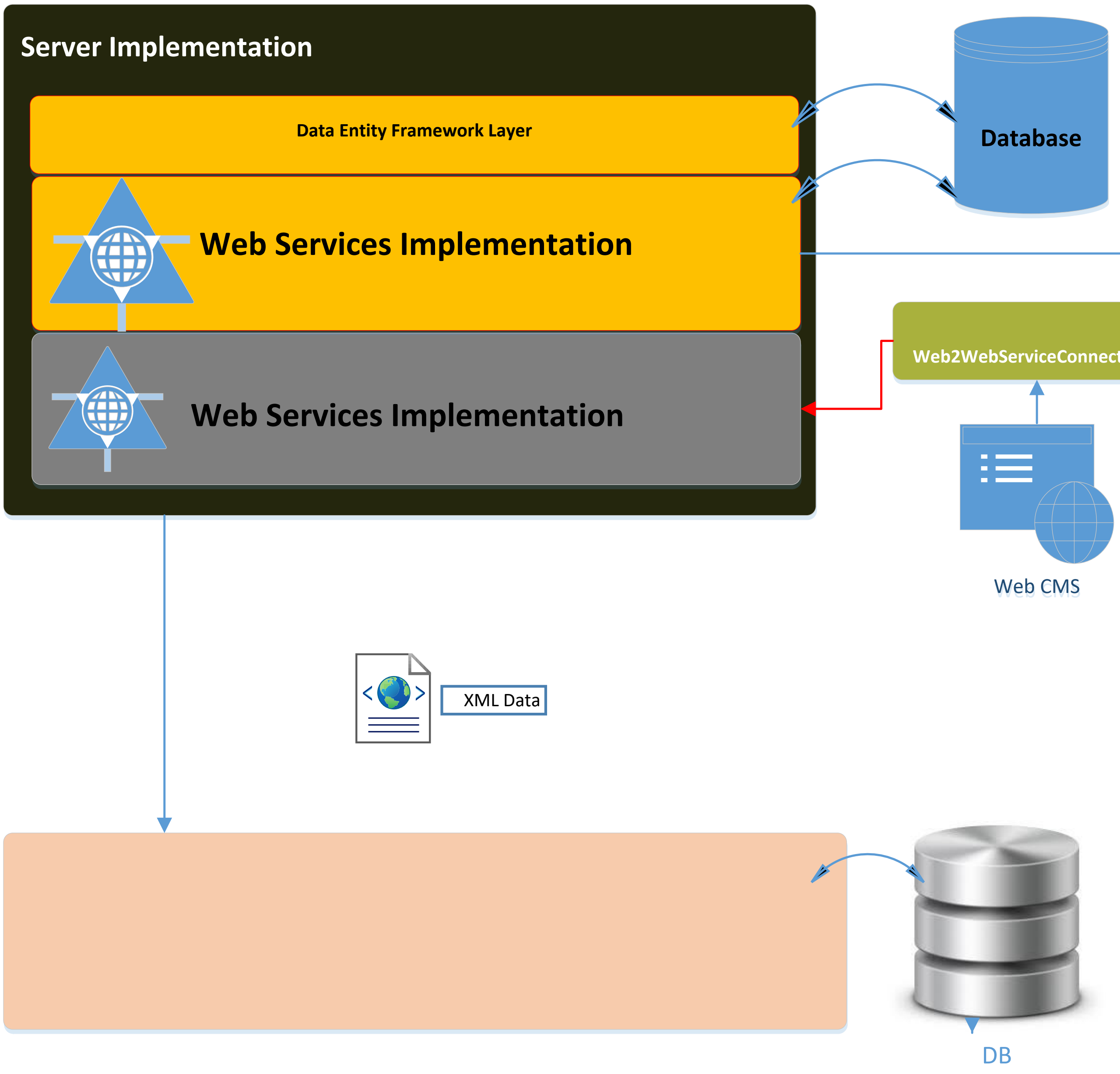

Server Implementation
Data Entity Framework Layer
Web Services Implementation
Web Services Implementation
Database
Web2WebServiceConnector
Web CMS
XML Data
DB

Figure 4-1: Typical server side architecture

One such example is that the access to the database is managed by the same component or service for both data capturing and processing layers. It means there are potentially the same database and set of tables being used for these activities. The databases used are also mostly relational database management systems (RDBMS). The databases are used to store information about the planned schedules the buses should be running along with information being recorded for actual arrival and departure times, and data coming from different sources like 3rd party expected arrival and departure times. The size of such databases grows to hundreds of Gigabytes of data and conducting any analysis on these databases for projection or prediction is not only impossible because of the infrastructure limitations of these database systems; but also, because it will interfere with the live operational system. Any such activity will undermine the provision of basic arrival and departure information and result in database deadlocks and delayed dispensing of information. The amount of time it takes to transfer this data to another platform for analysis can be a problem for real-time analysis of the information because the analysed data cannot be integrated with the real database to provide this information back to buses and other applications being used by the bus company and the commuters. Using only one database for the application, means the prediction information will be stored in the same database, increasing the load of the prediction engine if that is implemented.

The service implementation is two-fold. One part handles the data coming from external data providers for a different type of data like arrivals and departures, and other part deals with the requests coming from the same network through WCF interface. Usually, the WCF are serving the data received from 3rd party providers to the buses and optionally sending data about buses like location data back to 3rd party listeners that may use this information for adjusting the data they provide. The web service that is used to send data from client to server is the same used for fetching the response back as well. The response time is an important factor in it because the bus is waiting for the response and therefore this architecture has a natural limit of the amount of processing that can be integrated for on-the-fly analysis before it is sent back to the client. The components that are implemented in the service to support features like 3rd party integration are also not easily replaceable because every 3rd party source has a specific data format and needs an implementation to handle that format to import data into the system. The dependence and tight coupling of these services with implementation make them less flexible and therefore stops their integration with other services or data providers.

The service components are implemented in the same layer that can reduce the flexibility of implementation, as well as to minimize chances of extending the function on a generic basis. A typical transport system has components for supporting schedules, routes, and journeys that reflect the trips buses make during the day. The structure of the data is fixed when storing and processing the information, and any new piece of information will need interfaces to be changed so that the system can support it. There are no separate layers for components serving request and response, and therefore the impact of both actions on each other is potentially a problem regarding shared resources being used, as well as, the same state of connection being shared between both components processing two entirely different requests. Although different features are implemented as components but the architecture of these components does not comply with component-based systems because there is no concept of different components talking to each other. Rather, each feature has a unique functionality trail starting from web services to component and onwards to the database. It looks more like different independent sequences of features implemented with potential duplication of the effort needed to implement a feature even when the same is partially or fully implemented already.

4.1.2 Quality of Service controlled by Architectures

As mentioned earlier that many existing architectures are focused more towards details of functional implementation and the tiers in which the system works but integration of quality of service attributes is not discussed. This type of architecture lacks the necessary information needed by a system to exhibit reliable and sustainable behavior when working in different settings of the production environment. With increasing integration of technology to make the transport systems intelligent, the information produced by them has to be reliable. The reliability parameter is also needed in operational efficiency so that the information generated from it can be trusted.

Another important and key aspect that is missing in existing architectures is their ability to address the measure of interoperability of the individual components that eventually contributes to the interoperability of the software system built on top of it. The architecture should help in controlling the ability of the system to implement and extend to new features without impacting the existing ones. Interoperability, in our context, refers to the ability of a transport system to use and share information or functionality, of another system by adhering to common standards, which is not possible if these standards are not agreed upfront. Interoperability, integration and exchange of information are confused because of sharing some commonalities in terms of issues and goals. In data integration the goal is to synthesize data from different data sources – usually independent of each other – into a unified “view” according to a “global” schema. In data exchange the goal is to take data from a given data source and transfer it to a target data source such that it “reflects” the given source data as accurately as possible. Although both data integration and data exchange problems have been largely investigated, they are still considered extremely hard tasks. Implementing data interoperability requires realising data integration and data exchange as well as enabling effective use of the data that becomes available.

The extendibility is a very important aspect of transport systems because they have dynamic scenarios to incorporate more vehicles, information processing and supporting different data sources where the they should perform efficiently. The architecture must support the extendibility of the services and their capacity if a better hardware is provided and it should also include the load balancing strategies so that regional load balancing can be achieved without impacting the overall performance of the system. This QoS perspective of the next generation of smart and intelligent transport systems is also missing in the existing architectures and will be addressed in the proposed novel architecture.

The throughput or performance of the system for increased number of processing requests plays vital role in the acceptance of the system to be able to perform when number of requesting devices and applications is virtually unknown. The expiry of the traveling information every minute demands that the response time of the services cannot be later than few seconds to allow latency and processing on client side to use this information effectively. The accuracy of information is another quality attribute that needs confirmation because it adds to the reliability of the system and is very important for the passengers to trust the correctness of the information for a better planning of time and resources.

4.1.3 System Intelligence Objectives

The intelligence of a transport system should play a role in making it better and more enabled of the context to make strategic decisions. Short-term or long-term analysis of the data should be reflected in provision of the data to be used by buses and other modes of transport integrating this information to help companies and passengers equally with reliable data. The impact of the data that is generated in a transport system is important not only for itself but also other stakeholders of the system, and that includes commuters, bus companies and other systems working in the same ecosystem of a smart environment. We have identified different aspects of an intelligent bus system that are driven by the data being produced in the operation and should enable to address the challenges attached with each dimension. Therefore, we are convinced that the system should be capable of addressing following scenarios in order to provide a reliable service to the commuters as well as help the management make strategic decisions based on real-time being collected.

4.1.3.1 Congestion detection

The term congestion is an indication of the situation when traffic in a particular area or zone is getting slower and can potentially result in a complete standstill. It raises concern in many ways when we consider an urban transport system. The vehicles that are stuck in the congestion are using bus time, fuel, wasting commuter's time and there is a cost attached with each of those factors. An efficient detection of congestion can help minimizing its impact on the performance and operation of the system. This can be achieved by designing solutions for avoiding the congestion to happen and providing alternatives to both the vehicles and the passengers. Most of the existing prediction systems focus on the prediction of arrival times on stops for the buses and have offered algorithms ranging from GPS-based prediction to consideration of data exchange between the bus and different sensor units to know the location of the bus about the next stop it is supposed to arrive. This information is not sufficient because location of the bus does not define complete context of the bus where data from other buses on the route and external data elements like weather and traffic also play role in determination of the where the bus potentially can be in next few minutes. It can be frustrating for the passengers waiting at a stop when the count down for the expected arrivals is not changing for the arriving buses, and they have no idea if the indicated information about arrivals is accurate. This undermines the reliability of the system in providing accurate information about the vehicles.

The system should be intelligent enough to adapt to the traffic situation in which the bus is currently driving, and the data generated from it should contribute to better understanding of its impact on all the connection points including stops, other vehicles, bus companies and other systems, like other transport mediums, relying on this information. Early detection of congestion can add a lot more value to this process. The congestion detection is helpful for efficient time and resource management, and it can help in planning the transport service in a passenger friendly and cost effective way; its impact on the better environment by saving fuel and gas emissions is itself very important. The arrival data being reported by the buses on stops along with their speed during the stops and waiting times at signals can project a congestion before it happens. Therefore, an intelligent transport system should use this data for on-the-fly analysis to identify happening of congestion and calculate its impact on all the vehicles using that route or zone where congestion is expected. The system should also be capable of communicating this to vehicles approaching that congestion area so that the system onboard those vehicles can suggest alternate routes and possibilities that the vehicle can take to avoid or minimize the impact of the congestion.

4.1.3.2 Incident detection

There are different incidents that can happen on the road or with the vehicle itself. The list of incidents includes but is not limited to road accidents, emergency, road works, weather restrictions and breakdown of the vehicle itself. Some of these incidents can contribute to create congestion or delayed traffic. The incidents are supposed to be unexpected happenings and things that cannot be planned ahead of time. But the system should be intelligent to calculate the risk and help in decision making. Along with a potential decision to be made, the system should be capable to communicate the impact of this incident on other vehicles in the same network and other vehicles in close by locations or roads. If a bus breaks down on the way, the passengers have no way but to wait for the next bus to arrive. There is a possibility that the next bus arriving on the stop does not have capacity to accommodate more passengers or the next bus is running on a different schedule and is not stopping on that stop. The breakdown can also be somewhere between the stops and the regular buses don’t stop there to pick up the passengers. Similarly, accidents can happen anywhere on road involving the bus itself or impacting the bus making it stop. The emergency situation on board or involving the same route bus is driving through can also cause delays for the bus to complete its journey. There is a very special kind of incidents that have a considerable impact on the journey planning of the commuters without them knowing upfront; and that is for the stops that have been closed and is known only when the bus reaches that stop or is redirected way before reaching that stop. Such planned rerouting or redirection, if it does not consider the expected delays into calculation of arrival and departure times, can undermine the satisfaction and reliance of passengers on the network and can also end in using applications or websites to plan the journey in this situation.

The system should be capable of materializing these incidents information into useful knowledge for the passengers so that they are aware of these problems before they board the bus where possible to ensure best use of their time. This knowledge will also help the bus companies to communicate accurate and updated information to the passengers who are already on board and help them save their time by suggesting suitable alternatives when such an incident happens. If the types of incidents are recorded and parameterized so that the impact analysis of incidents can be measured, then this information can be shared with all stakeholders of the transport network. Each type of incident has a separate workflow on what should be done or what can be done to minimize its impact on operation of the transport network. Having this analysis part of the intelligent infrastructure in the system can automate the sequence of steps that should be taken and help the vehicles make decision about choosing the best route to continue the traveling where possible.

4.1.3.3 Rerouting and route optimisation

Although the number of buses and planned routes network to commute passengers from one place to another is increasing to accommodate more and more passengers, the number or size of roads inside an urban area does not change to accommodate the increased capacity. The buses cannot use all the routes because of their size restrictions as the buses can travel only on those roads that have capacity to allow them drive without causing any trouble for the bus itself or other vehicles either parked or moving. Route optimization is a feature that cars use very frequently to avoid possible congestions to save their time. Route optimization for a transport network has many limitations because of the road restrictions different size and dimension of buses running on the road. Planned rerouting is easier because the directions are provided by that planned incident or activity on the road. Automatic rerouting is a challenge because the consideration is not only to provide an alternate path to avoid the congestion; comparing the time taken on that alternate route with the time it may take to ease congestion should also play role in this decision making. That is where route optimization plays its role that different alternate routes should be considered for rerouting and the most optimized route should be selected subject to its suitability. Sometimes, a route optimization may suggest the bus to continue the journey because the extra time that may take in congestion is still better than taking alternate routes.

Another aspect of route optimization can be considered after analysis of passenger count boarding the buses. If passengers boarding a bus A are going to a destination where another bus B also goes; and the bus B takes less time to reach the same destination, then such information should be recorded and used for route optimization so that it is made available to passengers through signs and mobile applications that provide specific passengers opportunity to reconsider their journey planning. It can encourage efficient use of the resources and also helps the passengers save their time. The route optimization can also be done by optimizing the amount of time that is spent on the stop waiting for passengers. There are strategies already in practice like rendering few stops where passengers have to request the bus to stop otherwise the bus is not supposed to stay on the stop. The analysis of arrival time, departure time and dwell time (the time spent on the stops) can help in deriving the information about optimizing these times as well so that the bus time can be used effectively and optimal commuting options are available for the commuter. An intelligent transport system should have capacity to evolve the collected information from the bus network and automatically suggest to the bus companies on which stops are being used efficiently and how can the change of arrival time, departure time and dwell time on any stop help in optimizing the route from both the company and the commuter perspective.

Bus connections are another aspect of the route optimization. The bus network can be visualized as a network of road intersecting each other. There are many different lines or routes that constitute a transport network with many buses serving each line during the day. These lines share many stops to enable the commuters change buses to reach their destination if one bus does not take them directly. It means if the routes are planned and the times are decided when the buses would be running, it is important to consider these intersection points so that waiting time can be minimized for the commuters who have to changes buses to complete their journey. It becomes a complicated problem when there are many intersections and it is not always possible to optimally deal with each intersection without impacting another connection on the way. The frequency of buses on one line can be different than the ones intersecting this line and making the schedules on one line compatible with other intersecting lines is challenging. This situation becomes more complex when a bus is delayed for many reasons mentioned above. Making a decision if the buses that were to connect with the delayed bus should wait or continue their journey can always be a trade-off choice because some compromise is to be made between saving time of the passengers from one line or the other and also deciding priorities for this purpose. The minimum that should be done is communicating to the commuters and keeping track of all the missed connections so that we can attach the delays with possible missed connections and this information can be used for better planning of the routes and the journey itself.

4.1.3.4 Real-time analysis

The analysis of information is a key to address all the challenges being discussed. No decision can be taken without availability of an effective analytics exposing different key performance indicators. The time to make the information retrieved from analysis available for decision making is very important for an urban transport system because the information is outdated very frequently in a transport system. The arrival and departures times are valid for up to one-minute time window because then new times are to be published for the commuters and the bus information network. This advocates the need for an on-the-fly analysis system that can process the information coming from different sources and produce decision specific data that can be published back to vehicles and applications without any delay. Therefore, an intelligent transport system should have the ability to not only incorporate the real-time data (data streams) being fed in but also process it to keep an updated snapshot of the system all the time. The response time for request of information should be processed without any delay. This apparently simple process of on-the-fly analysis becomes a challenge when the size of the network grows and the amount of data coming from the vehicles and the bus companies along with their 3rd party data providers, like Tfl or Movia that provide information about expected arrivals, weather, roadworks and traffic information, increases both in frequency and volume. Most of the existing systems provide either no implementation for real-time data capture and analysis or have very limited facility to provide this kind of analysis.

Many of the congestion, incident analysis and routing challenges discussed above can be suggested solutions only if on-the-fly analysis is possible. The lack of flexibility in existing transport systems to incorporate infrastructure for real-time data analysis limit their application. The incorporation of specific tools and techniques along with compatible resources in existing resources is not an easy task because their original design was not planned with this in consideration. An intelligent transport system should either have this feature itself or be able to integrate with architectures and infrastructures so that the intelligence can be integrated with them. The intelligence components of a system should incorporate latest techniques and technologies like Big Data, NOSQL with real-time processing capabilities. The data mining algorithms should be customized to accommodate real-time pattern recognition and then using these pieces of information to provide analytics. This introduces the concept of short-term and long term-prediction because an intelligent transport system should have both of these features to continue learning and improving. The data that has been collected from the transport network should be converted to a probability based projection and prediction database that can be incorporated to predict arrival and departure times in real-time.

4.1.4 Data Integration challenges

The way transport systems are designed work includes talking to different data standards that exist for exchange of the data related to an urban transport network. There are lot of external factors like weather, road works, planning, data coming from sensors and working with data coming from different providers. The current transport systems are working with either one of the sources but they lack the ability to process data from different sources to convert that into more useful information to help transport systems make decisions efficiently and smartly. Majority of the transport systems are working without any support for Big Data infrastructures like Hadoop or structure free databases like MongoDB and they are just two examples of many different alternatives available.

4.1.5 Environmental perspective

The concept of smart environment joins all systems in the ecosystem of future generation of applications and services and bus system is not an exception because it has great impact on the environment. Also, the bus system does not run in isolation and is closely connected with the other modes of transport like trains and planes and sharing the same road with other vehicles demands the transport systems to have connection points to import and export information. Saving time and making efficient use of the resources like roads, vehicles, fuel in the form of gas as well as moveable energy and integration of technology has transformed the transport systems into intelligent and self-adapting information systems. Therefore, it is very important that the architecture of a transport system is designed with these considerations and out of the box to discover and enable dynamic interactions with other systems.

4.2 Proposed Design/Architecture

As mentioned in previous sections, the need for a new architecture for transport system is evident that is based on not only how the system should be designed but also how will the system interact with other entities in the smart environment. The proposed architecture in this Thesis is shown below in Figure 4.2 that outlines different components of the system and how they should interact each other. It will be a generic system where data processing components will be designed to be scalable and efficient in processing. The promising features and the components implementing them of the new architecture are elaborated in the following subsections. The novelty of the architecture lies in the following areas

- Plug n Play architecture: All the components in the architecture are replaceable. That means we can replace any component with another component giving similar but better function without changing anything else in rest of the system. An example is a Analytics components. One component may be using one algorithm to process the data whereas another component may be using a better algorithm to process the data and it can easily replace the existing component and rest of the system will work with it. Same is true for transformation, system and real-time cache components. The components in the existing system are fixed for the purpose and they can’t be replaced. It will need to re-deploy the whole system for every change.
- The Real-Time Cache Manager: It enables the system to work on a novel data structure based on hierarchical regions so that the effort of performing complex spatial queries can be drastically reduced and that adds to the performance of the system. It facilitates the integration of this system with any other internal or external system without impacting the required functionality. There is no cache manager in current system and every request for information is processed through expensive spatial queries from the database.
- Automation: The whole flow of functionality is automated, which means that the process from data acquisition to data visualization is autonomous. There is no automation in the existing system and different processes of data acquisition and processing depend on each

other and one process can even deadlock and fail other components when running in parallel.

- Generic: Although the architecture is proposed for a public transport system, it is easily applicable to any other data oriented domain where we have data coming from different sources and we intend to perform predictive analytics. The Data Analytics component especially is very generic and is independent of the algorithms or the data warehouse and can virtually work with any database or data format and still produce predictive analysis on the data and visualise it for strategic decision making and operational excellence. The existing system is strictly applicable to the very system it is implemented for. It cannot be used as-is for any other transport system as well and needs lot of changes before it can support another system with different data structure and components to interact with.
- Flexible: It is easily applicable to any existing system where integration with data is needed and it empowers the system for predictive analytics and monitoring. The existing system does not have any functionality to integrate with any existing transport system and start providing any data processing or analytical capabilities. The existing system itself depends on external systems to get information about prediction of arrival times and therefore, it has no layer for exporting this information based on its own data as well.

There are major differences, as mentioned above, between the proposed architecture and the existing system where the proposed architecture can not only support the existing system but it can also help in building a new smart transport system from scratch.

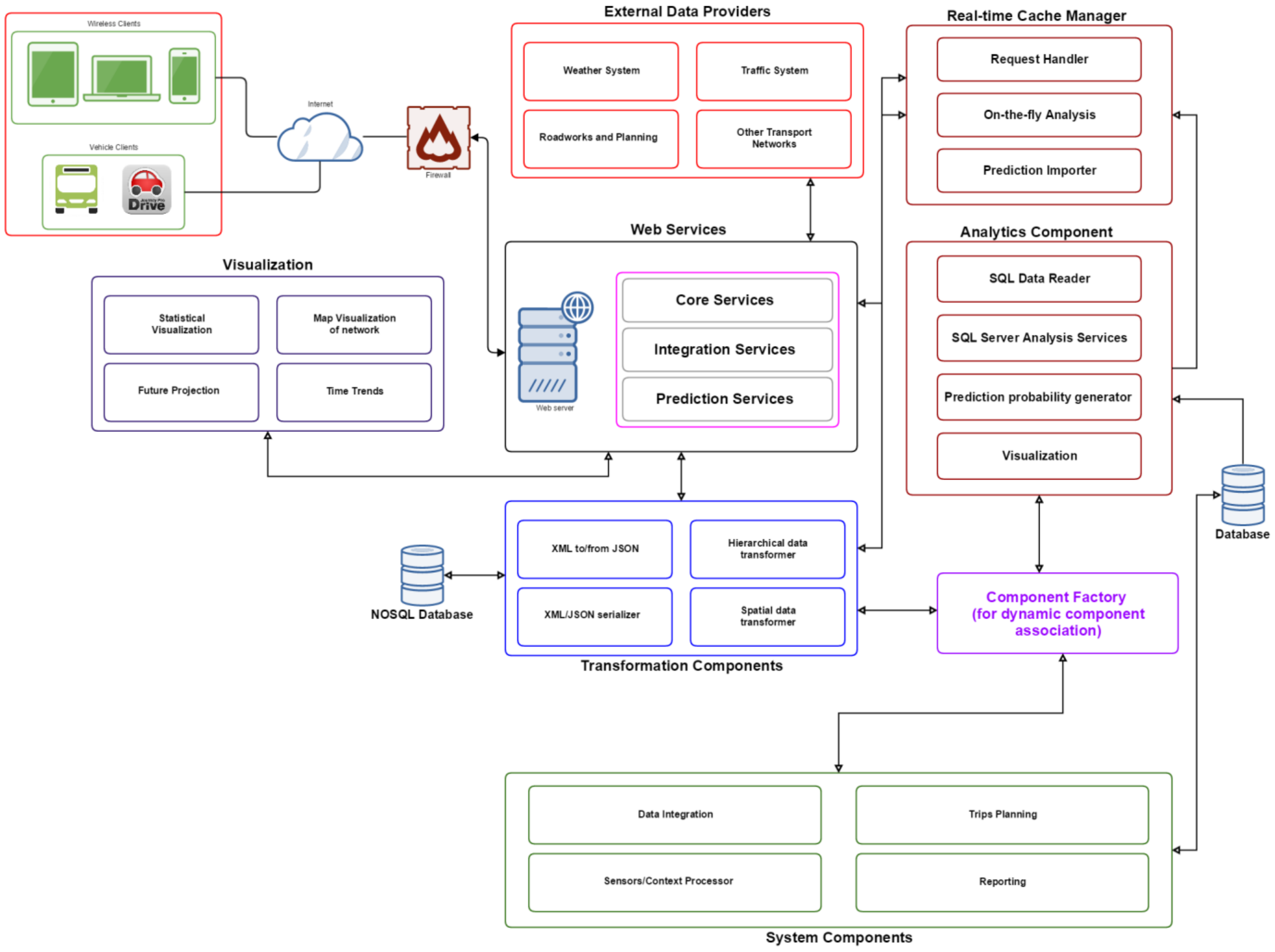


Figure 4-2: The proposed system architecture

4.2.1 Web Services

The proposed architecture presents a very dynamic WCF based web service management system that provides asynchronous access to the web services so that the requests that do not need immediate data response do not wait for the data to be stored and processed. There are many layers of web services with each layer working independently and load on one layer does not impact the performance of other web services. The Asynchronous service pool will ensure that the applications and devices that are sending data to the system do not have to wait for the process to be completed before they receive an acknowledgment. The data collected from different sensors like GPS, speed, direction and the vehicle's operational sensors is dropped at designated web services, and each of the services adds the data request to a pool for processing.

Different components of this module are shown in the Figure 4.3 below that lists the services implemented to serve those different functions of the system. The figure also indicates interactions between the services and the interactions with 3rd party systems to be able to fetch data so that it can be integrated into the prediction rules extraction where applicable. The web services make use of the transformation component to establish a mapping between different data formats.

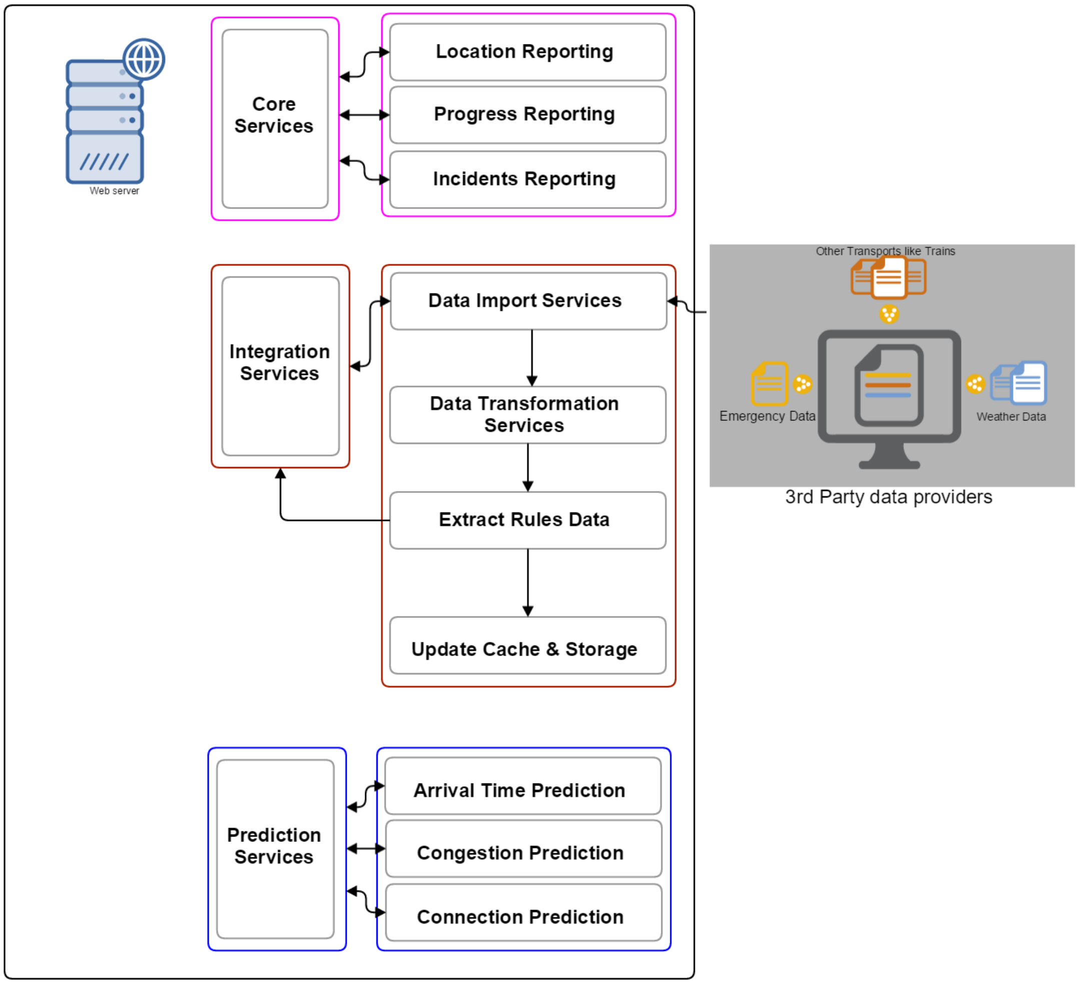


Figure 4-3: internal architecture of web services module

Different web services components will serve a different type of data requests coming into the system including the data coming from internal network as well as heterogeneous external sources like 3rd party transport and prediction data providers. The sensor data services are designed to support data collection for different sensors onboard vehicles and use the transformation components to convert this data into a format so that it can be used in both real-time prediction component and generation of long-term prediction probabilities. The Integration web services are responsible for the two-way integration of data and information. On every request that arrives on web services, it identifies the type of request and then extracts other pieces of information it should be integrated with before it is processed. One example is that data arriving at current location of the vehicle is combined with the speed of the vehicle reported around the same time so that both speed and location parameters can be attached as tags with the data and used later for analysis.

The web services component does not only work as an interface for the external world to be used by client applications and devices, but it is also used as a bridge between different components of the system itself. That is where the Export Services layer is used because it works as a data provision layer for both external and internal requesting applications. The Integration services combine the request coming from Sensor services with the required elements from Export Services based on the context identified from Sensor Services data and hand over the collaborated data for further processing to the system components. The Export Services reflect the service feature of the system to provide intelligence and prediction back to the requesting applications. These services work in collaboration with Real-time components to provide instant prediction and information about the transport network. The services also work with backend system components so that the data can be shared with another transport system to derive further contextual information from it where needed.

4.2.2 Real-time Cache Manager

This is an instant intelligence provision component of the system that is responsible for providing a short-term prediction based on the current situation of the transport network and suggests if a specific area may have congestion and delay issues. It is also responsible for providing information about real-time arrivals of the vehicles that is derived from data reported by the buses, and a collaborative analysis of the information, with existing prediction database. This component is composed of many sub-components that complete the lifecycle of real-time analysis of the information, from data acquisition, to process-ready strategic data to help in decision making.

Being a real-time component, it is very important that the response time for both incoming and outgoing data is very quick. This component maintains an optimized data structure to support storage and retrieval of the information because queries for spatial data are time-wise expensive especially when data volume is bigger, and the query wants to filter the information in close vicinity. The information is stored in such a fashion that the relationship between stops and vehicles is configured once only, and the relevant information is fetched in real-time. A special and specific data structure has been designed to facilitate this storage and retrieval of information without compromising the performance for spatial queries. It is a hierarchical data structure to split the area into zones and subzones based on the impact analysis between different stops. Each stop in the network is marked with its zone hierarchy so that the data, as well as processed information, can be accessed in either top-to-bottom or bottom-up approaches. This hierarchical data structure enables the system to run the spatial queries on filtered and context-specific data compared to running it on all the network to find out the impact in real-time.

There are two types of requests that can be processed by this component. One is for the data that is being reported and is to be stored not only in the permanent database but also in the cache database. This type of request does not need any response other than acknowledgment. The second type of request is to access the processed information that can serve different purposes like real-time arrival time prediction, delays at different levels of stop and zones, the impact of delay propagation, congestion identification in future, short-term prediction of stops and times, resource requirement and optimization of resources as well as routes. All this information is extracted and derived from the data that is stored because the links between different data elements help in combining the information to generate these different reports. This is supplemented by the data transformation components that can convert information to different formats. The transformation components make it possible to connect to any data source so that the data as a request or response can be converted to a format that is accepted by another side of the application.

The real-time prediction considers not only the live data coming in, but it is also based on the prediction probability for each stop generated through data analysis component. The Analysis component in the system is continuously updating the prediction database with new data coming in so that comprehensive scenarios for prediction are always updated. The output of the Analysis component is an equation that can be used by the Real-time cache manager to compute information about arrivals and situation of the transport network at different levels of zonal hierarchy.

4.2.3 The Analytics Component

This component is the backbone of the system because it processes data coming from different internal and external sources and can process the data so that it can be converted into usable information for strategic decision making. It works with collaboration of many different sub-components that deal with storage and retrieval of information from main database, data integration services to make the data ready for processing and that involves data cleaning and pre-processing. The other components are application of data mining techniques with customized mining models, generation of prediction probabilities for each stop and converting this prediction model into an equation that can be used by any application or process to be able to make accurate predictions.

The one-time process of fetching the history data is a very time-consuming task before this incremental data fetching is used because of the volume of the data that has been used as a test bed. This one-time process is done only once so that the information based on the history of arrivals at different stops is used to make the connection between different end points and investigate their impact on the local zone as well as the whole transport network. This process is changed to an incremental data fetching that retrieves only newly updated information from the database and ensures that already processed data is discarded. Since this is an offline process, therefore it is not real-time, and the process can take the time to conduct detailed analysis of the data. Once the data retrieval is complete, it is fed into the data clean-up component that removes any anomalies in the data, and it also transforms the structure of the data to make it compatible with the data structure required for data mining.

The data analytics component has a very comprehensive implementation to apply data mining techniques on the data to generate prediction models and probabilities that can be used to make a prediction about the stops. SQL Server analysis services have been used to apply the data mining techniques based on customized mining models to include different criteria for each mining model so that variations of the mining technique can be used to apply these mining models and derive different prediction strategies. Five different mining techniques have been used as Clustering, Linear Regression, Logistic Regression, Decision Trees and Neural Networks. The system generates different prediction models from this processing, and the models are generated specific to routes so that prediction for that route is close to accurate. These models are converted to prediction probabilities data that provides a direct prediction of arrival time for each stop based on the context of the stop. This prediction data is then exported into a new database that can be used by components like Real-time cache manager that intend to make predictions by using the dynamic context of the stop.

There is a visualization component as well, that contributes to the visual representation of the data, as well as predicted probabilities. This is a very specific visualization intended at indicating the accuracy of the prediction process more than just plotting the data. The output of this component reflects score for each prediction model so that the system can use different alternatives to make a prediction. An example is that based on data for one route, Linear regression may be more useful under specific context criteria for a stop whereas it may be another technique for the same stop that is more useful for the same stop under different criteria. The system is designed in a way that it can decide at runtime to choose which algorithm suits best with available mining models based on accuracy score and prediction probabilities produced using those techniques. The machine learning algorithms are key part of the analytics component as they are used to conduct predictive analysis on the data and enhance self-learning attribute of the system. There are many difference platforms that can be used for applying machine learning algorithms, and a comparison is made among the best suitable options. This comparison is presented in next sub section.

4.2.4 The Transformation Components

The transformation components are responsible for interoperability with other internal or external systems where data exchange is required. The data is available in many different formats, like JSON and XML, and one standard cannot be enforced in different implementation that exist in the industry. Also, there are many legacy systems that are still working on either an old format of the data or the internal structure of the data. To be able to integrate with different types and structures of the data, the transformation components implement a layer for each type so that the implementation is independent of the format of the data that is being imported into the storage or exported to other systems through web services. Therefore, the transformation components act as layer between different components of the same setup or inter-system communication without doing any specific implementation for each different format.

4.2.5 Components Factory

This module equips the system with dynamic flexibility for use and integration of different components without the requirement of a new deployment every time. It has a reference of different components available in the system to conduct different functional activities in the system like data retrieval, cleanup, and application of prediction components. Although the current system integrates data from SQL server database, the system can equally work with another component built to work with data coming from say Oracle and then using the relevant transformation components so that when the data is fed into the prediction system, it is same structure regardless of which database platform it came from. It connects with all major components of the system like Real-time cache manager and therefore adds the capacity in the system to use different real-time prediction systems if needed for different routes or systems.

4.2.6 System Components

Although the real-time cache manager component is the real interface of the system where web services are exposed to give access to the external systems for smart analytics, the system components play an important role in processing and archiving the data coming from different sources and converting them to compatible structure so that Analysis component can easily fetch it for prediction process. If the raw data from different sources is saved into the database as-is, then it becomes a challenge to make these connections in the processing stage, and it will consume a lot of time.

The integration component has end points for data coming from different sources like external data providers for planned trips, planned road work and diversions, possible events in specific criteria that have a direct impact on the transport system. It also contains information about emergency events or incidents that may have impacted the system in past so that it can be used in the criteria as well to be able to make predictions for these events in real-time. Each end point in these components will have a specialized input implementation so that a layer is added to understand the structure of the data for incorporating it into the system. An important source of data is different sensors installed on the vehicles or on the route that publishes very important data about the location and usage of the network. This data is also merged with arrival times and can be used as one of the criteria parts of the context used for prediction.

The analytics components are the base of the prediction process and enable the real-time cache manager to be able to make an on-the-fly estimation of arrival times and detection of traffic situations like congestion, delays in service, the connection between different vehicles, the impact of delays on surrounding stops and projection of these factors to zone level. The data is continuously read from the database so that training of the machine learning algorithms is improved with more scenarios being reported. There is an SQL Server Integration services component that fetches this data and performs pre-processing. This pre-processing involves cleaning up of the data and transforming it into a flat data file so that machine learning algorithms can work on it effectively. Different context parameters are extracted from the data in this stage and appended to each report of the data from vehicles. Once this flat data format is ready, it is fed into the machine learning algorithms along with different mining models where each mining model has specific context parameters defined as an input to each machine learning algorithm. Each machine learning algorithm produces an equation for the model along with probability prediction for each stop for all the scenarios included in the training data. These probability predictions are applied to non-training data, and a further comprehensive prediction database is produced from these probabilities. The model equations are also improved based on this comprehensive probability.

There are data integration system components as well that can import data from internal as well as 3rd party data providers. The data can be for weather for that specific region or planned events or the planned road works that can potentially have an impact on the situation of the transport system. Since this system architecture is not designed for any specific location or area, the integration components provide an option to integrate with such data sources where it is important to consider them part of the context.

4.2.7 Work Flows

The proposed architecture provides a solution for the problems and challenges mentioned in previous sections for what the existing architecture lack in. The section below presents work flows for some of the scenarios, which is a good example of the simple implementation and handling of requests because the proposed architecture already handles all the complication.

Congestion Detection: The concept of identifying if a location has a possible congestion now or in near future, the request is sent to the web services with the location parameter containing longitude and latitude. This location is mapped to a zone in the hierarchy, and then the data in that zone is analyzed for delays. If the combined presentation of the delays indicates congestion along with its severity, this information is prepared and made ready for sending back to the requesting device or application. The flow for it and different components involved are shown in Figure 4.4.

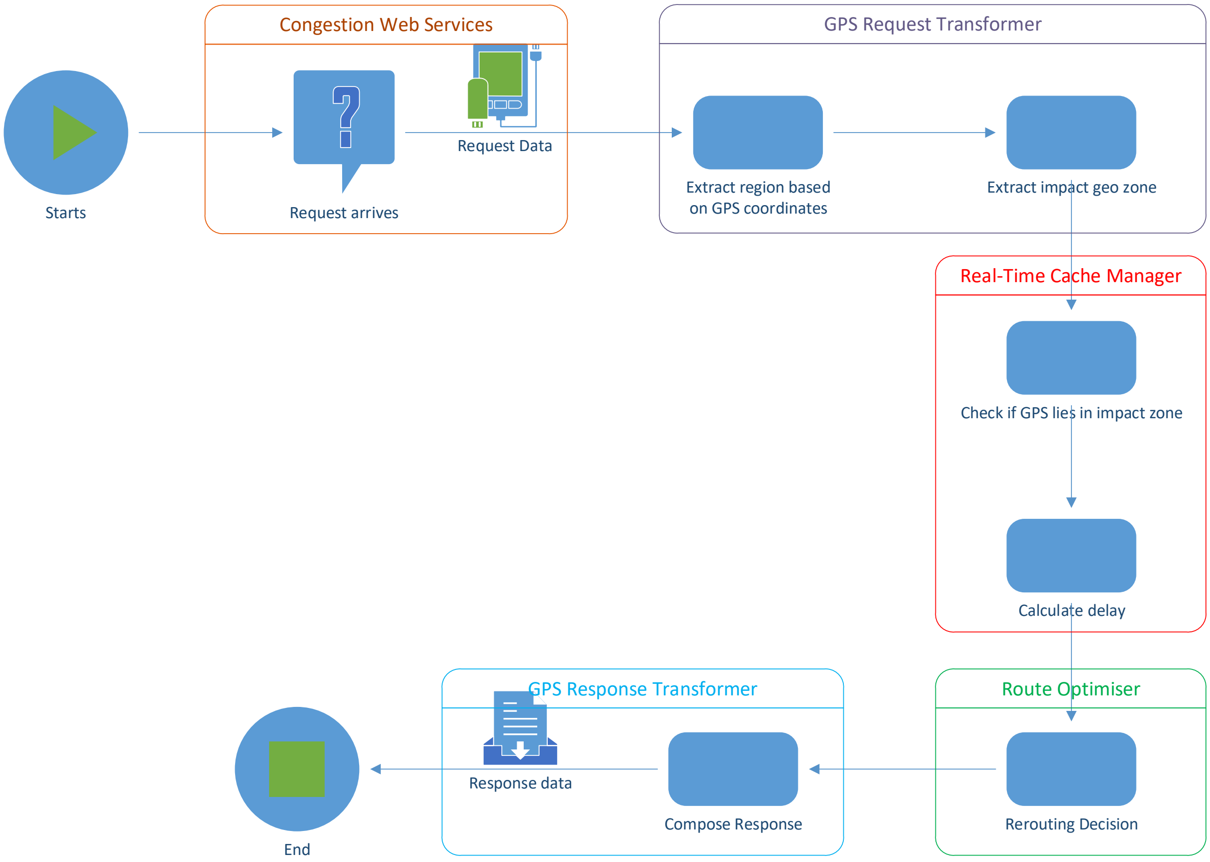


Figure 4-4: congestion detection work flow

Arrival Time Prediction: Arrival time prediction is one of the key features of the system accurate enough to be able to make short-term as well as long-term predictions. The time a bus reaches a stop is linked with many different sections of today's smart transport system because it will identify how well aligned it is with other connecting vehicles with this stop as well as other connections that can help commuters complete their journeys like trains or planes. The system proposes a very simplified process so that both internal system and any interested 3rd party system can fetch it from any location. Because the data has already been processed for delivery every time a new data report is submitted, request for the arrival time of a stop needs the application of the prediction equation to the context parameters attached with the stop request. The flow in the Figure 4.5 shows which components are involved on this process.

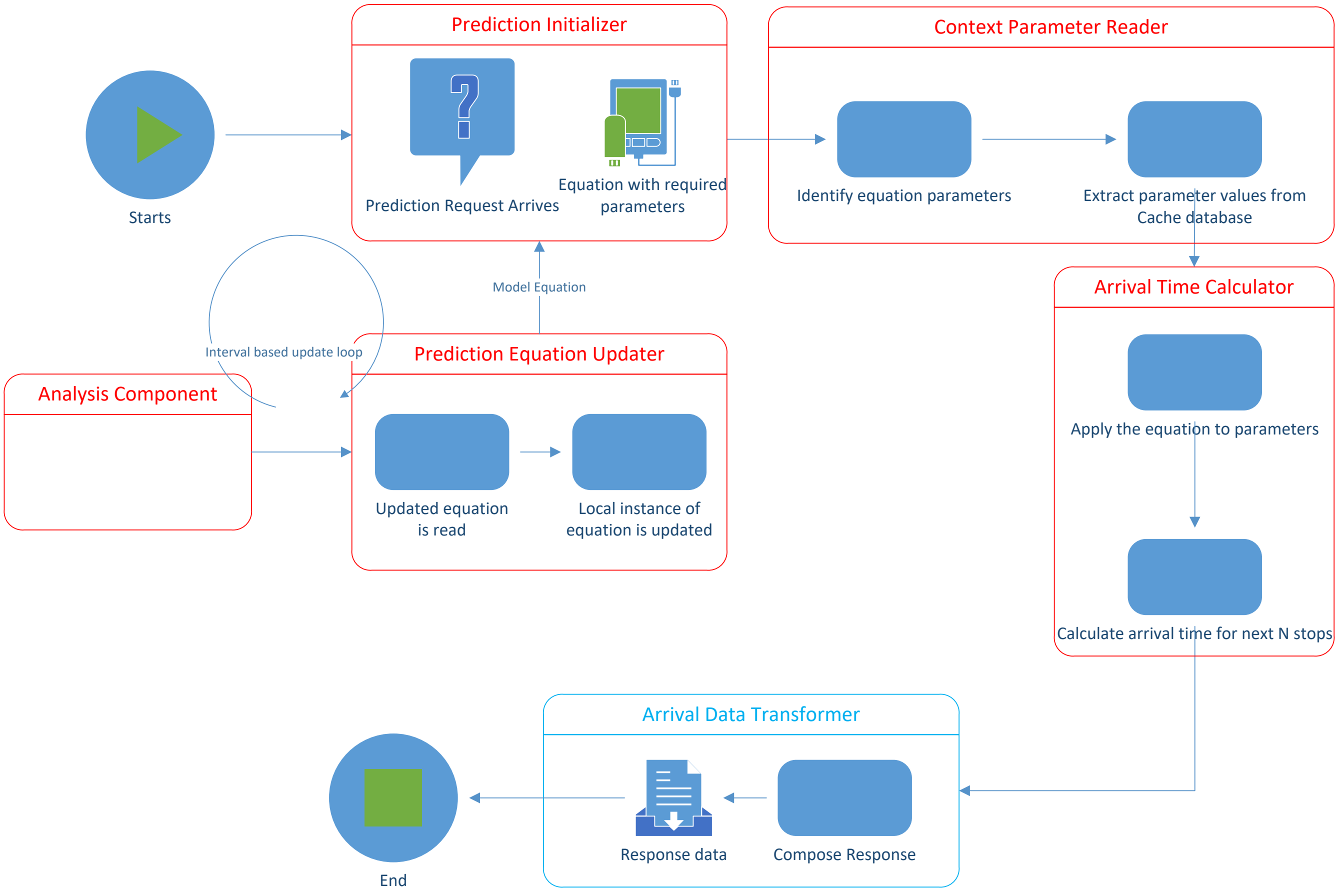

Starts
Prediction Initializer
Prediction Request Arrives
Equation with required parameters
Context Parameter Reader
Identify equation parameters
Extract parameter values from Cache database
Model Equation
Interval based update loop
Analysis Component
Prediction Equation Updater
Updated equation is read
Local instance of equation is updated
Arrival Time Calculator
Apply the equation to parameters
Calculate arrival time for next N stops
Arrival Data Transformer
Compose Response
Response data
End

Figure 4-5: arrival time prediction work flow

4.2.8 Deployment Diagram

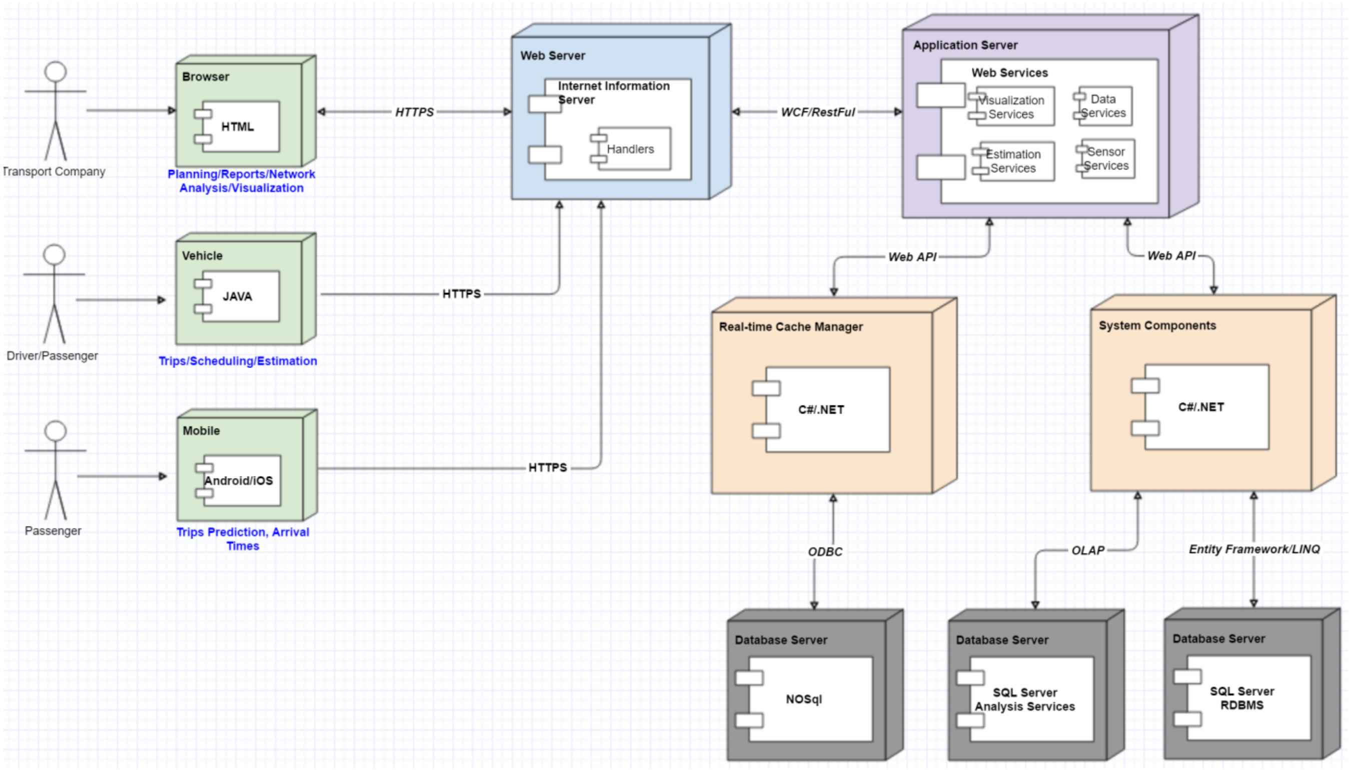


Figure 4-6: deployment diagram

The Figure 4.6 presents a potential deployment diagram for the system implemented on this architecture. The deployment environment considers Microsoft technologies in implementation of the system. It is however important that all these components can be replaced by any technology or implementation as long as their interface services maintain their input and output parameters with their respective transformation implementations available. The interface to the web services is available for any platform that support secure HTTP protocol to make calls and receive responses. This is done to ensure that reliable data can be delivered to the requesting applications and devices. The services have been implemented as Restful services on Internet Information server providing support for the web server and work as front end of the system. These services are implemented using .Net technology using C# as programming language. The same server works as the application server to provide engine for handling the requests coming before they handed over to the implementation components. The real-time cache manager component is an independent component from system core components because it can work with any system providing it the required web services interface. This cache manager component is also implemented using C# as programming platform. It uses NOSQL databases as its quick access data warehouse so that regardless of how many data nodes are, the response time of the search is fast enough to make the process real-time. The core system components are an integration of implementation and analysis services components that perform analytics on the data and produce probabilities for accurate prediction.

4.2.8.1 Data Mining platforms

The dataset maybe stored in any database system including SQL Server, Oracle and NOSQL implementations so it means that before the data is processed, it is important to transform the data into a format that is understandable by the tool used for analysis. Two main issues to handle are the data types of “DateTime” and “geography”. The reason is that different analysis tools handle the data differently and therefore need the data to be preprocessed for analysis. It was an important decision to make because the amount of data was big so we had to consider the data export and import effort. Also, it was important that the tool is compatible with the database because the application of prediction algorithms to the real-time data as it arrives was only possible if the prediction algorithm could access those incoming data stream to use the prediction data for making real-time predictions. One more thing which was a point of focus to select an analysis system that we wanted to make this prediction system integrated with an existing or improved architecture of the transport system; so integration of the proposed prediction system with ***.Net*** technology and SQL Server database were an important consideration to make.

The dataset used in this Thesis comes from SQL Server database so the selection of the database favors SQL Server when comparing with other platforms. It does not mean that other platforms have disadvantage in saving the information because each of them offers a competitive advantage but each of them have an overhead of data processing if the original dataset is stored in a different database. Copying the data to target database system needs a lot of effort because of the high volume data generated already and very high frequency data coming all the time.

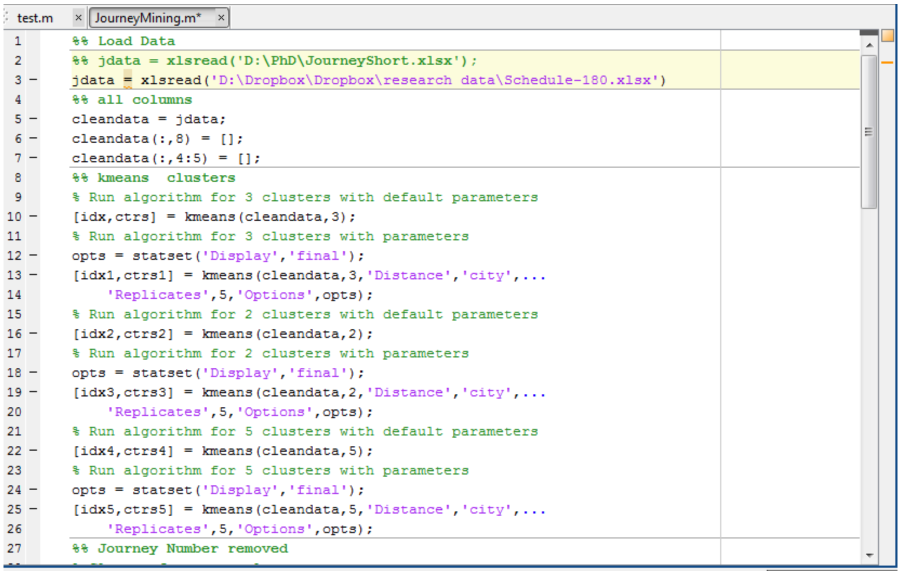
4.2.9 Comparison of mining platforms

NoSQL Databases – MongoDB

Considering this a big data problem, MongoDB was the first option tried because it provides high speed access based on its implementation of indexes on all attributes and the search on this is fast. The idea was that we use a database system from where the search is fast so that it can be used in real-time systems to make predictions. Although it produced good results when it comes to searching fast on large databases but the selection of MongoDB as storage would double the effort because the actual data is stored in SQL server. The selection of MongoDB would require that either the web services being used to PUSH data to the SQL server are changed so that they add data to MongoDB as well, and this was needed for the testing purpose. It means that we would need a change in the production system for testing the efficiency of the MongoDB storage in real-time and that is why this idea was dropped. Another way was to build a bridge component that will sync data between MongoDB and SQL server so that the prediction system is auto trained for new data as it comes and can be used for prediction system. MongoDB provides a very good platform for analytics supported by high performance access to the information but the case study has used data stored in SQL Server. Considering the size of the data and a daily effort required to transfer the data to MongoDB is an overhead in this very specific case and that is why it was not considered a viable option for the case study.

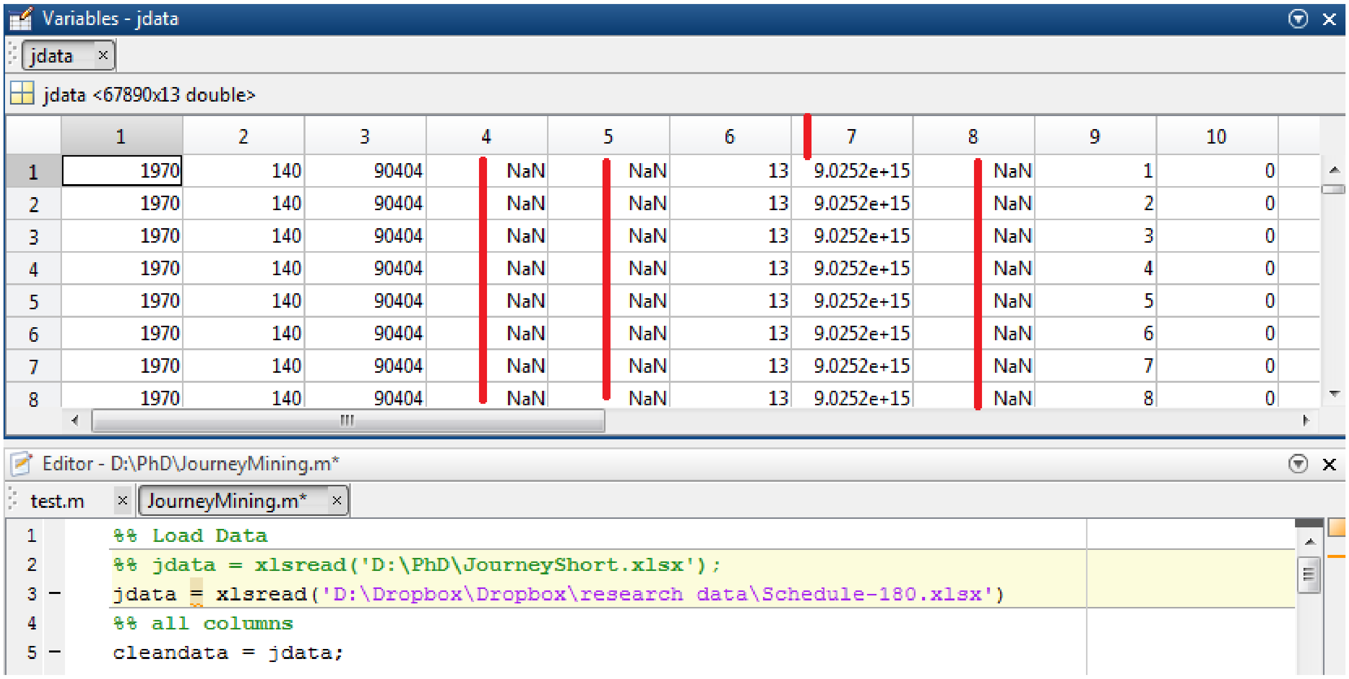
Matlab

The next tool considered was Matlab. Matlab is a well-known and popular tool used for data mining and machine learning problems and was thought to be a natural tool to build a prediction system based on the journey data. Although Matlab has a connection to the database but the handling of datetime and geography type in Matlab is not straightforward. One of the objectives of this study was to use the DateTime attribute of the data and then predict the times when the vehicles will be at a specific point. It seemed like a problem that we convert the data from SQL server types to a type Matlab understands and then transform it back to apply on data types of SQL Server when the prediction is ready would be an overhead.



```
%% Load Data
%% jdata = xlsread('D:\PhD\JourneyShort.xlsx');
jdata = xlsread('D:\Dropbox\Dropbox\research data\Schedule-180.xlsx')
%% all columns
cleandata = jdata;
cleandata(:,8) = [];
cleandata(:,4:5) = [];
%% kmeans  clusters
% Run algorithm for 3 clusters with default parameters
[idx,ctrs] = kmeans(cleandata,3);
% Run algorithm for 3 clusters with parameters
opts = statset('Display','final');
[idx1,ctrs1] = kmeans(cleandata,3,'Distance','city',...
    'Replicates',5,'Options',opts);
% Run algorithm for 2 clusters with default parameters
[idx2,ctrs2] = kmeans(cleandata,2);
% Run algorithm for 2 clusters with parameters
opts = statset('Display','final');
[idx3,ctrs3] = kmeans(cleandata,2,'Distance','city',...
    'Replicates',5,'Options',opts);
% Run algorithm for 5 clusters with default parameters
[idx4,ctrs4] = kmeans(cleandata,5);
% Run algorithm for 5 clusters with parameters
opts = statset('Display','final');
[idx5,ctrs5] = kmeans(cleandata,5,'Distance','city',...
    'Replicates',5,'Options',opts);
%% Journey Number removed
```

A typical script in Matlab looked like the figure above. The script looks very simple that it loads the data, in this example, it loads the data from an excel file exported from SQL server, and then applied clustering algorithms on this data. It is also very easy to change parameters for the clustering algorithms. Similar to clustering algorithms, we can apply other machine learning algorithms as well, so it is simple from that perspective.



jdata <67890x13 double>

	1	2	3	4	5	6	7	8	9	10
1	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	1	0
2	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	2	0
3	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	3	0
4	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	4	0
5	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	5	0
6	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	6	0
7	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	7	0
8	1970	140	90404	NaN	NaN	13	9.0252e+15	NaN	8	0

```
%% Load Data
%% jdata = xlsread('D:\PhD\JourneyShort.xlsx');
jdata = xlsread('D:\Dropbox\Dropbox\research data\Schedule-180.xlsx')
%% all columns
cleandata = jdata;
```

Figure 4-7 MATLAB data loading challenges

As can be seen in the Figure 7 above; the data it imports, it completely ignores the DateTime and geography types as it cannot understand them and need to pre-process to use them and there is no simple way of doing it. It means that this back and forth transformation of the data to use the source data from SQL server and then apply the prediction algorithm from Matlab will end up as an overhead and may become a bottleneck when applying this for a real-time prediction system. As shown in the figure below with data imported into Matlab – it is clearly visible that NaN is used for the DateTime . Playing with DateTime attribute of data is a frequent process in the data under consideration so type handling and integration problems with real system resulted in making a conclusion, and that is why Matlab was not selected for data analysis.

R

R platform was another analysis platform that was considered for data analysis. R is an emerging platform for using machine learning algorithms with an open source community making contributions to it. It has many benefits like it is the free and open source, it runs everywhere, and it supports many extensions. Also, R can connect to other languages as well as databases including SQL Server. Although R has an advantage of data clarity over Matlab on the way it handles data types but it has the problem that the components and addons being added by community; not all of them are well tested so lot of effort will be spent on making sure that the component selected for analysis will perform as expected when integrated with the real system.

SQL Server

Microsoft SQL server has a very integrated platform where storage, processing and analytics services are combined into one application. The layers have a very easy and reliable connectivity and can process the data in raw shape with provision of digging deep into the data after initial analysis is conducted. Although the data for the case study is available in SQL Server, SQL Server has some clear advantages over others because its use and customization is easier and the algorithms have been customized to do an effective analytics. It is not just data management tool, but it is a complete suite that can help on data storage as well as data integration and data analysis. It comes with three main components; SQL Server Analysis Services, SQL Server Integration Services, and SQL Server Reporting and this makes it a good candidate as all the overhead of data transformation is removed, and everything is very local to the actual database and the .Net technology is working with it. SQL Server was selected as the tool to conduct analysis considering it a natural selection based on the original database and the target integration system. Some advantages are mentioned below

- Multiple data sources: You do not have to create a data warehouse or an OLAP cube to do data mining. You can use tabular data from external providers, spreadsheets, and even text files. You can also easily mine OLAP cubes created in Analysis Services. However, you cannot use data from an in-memory database.

- Integrated data cleansing, data management, and ETL: Data Quality Services provides advanced tools for profiling and cleansing data. Integration Services can be used to build ETL processes for cleaning data, and also for building, processing, training, and updating models.
- Multiple customizable algorithms: In addition to providing algorithms such as clustering, neural networks, and decisions trees, the platform supports the development of your own custom plug-in algorithms.
- Model testing infrastructure: Test your models and data sets using important statistical tools as cross-validation, classification matrices, lift charts, and scatter plots. Easily create and manage testing and training sets.
- Querying and drill through: Create prediction queries, retrieve model patterns and statistics and drill through to case data.
- Client tools: In addition to the development and design studios provided by SQL Server, you can use the Data Mining Add-ins for Excel to create, query, and browse models. Or, create custom clients, including Web services.
- Scripting language support and managed API: All data mining objects are fully programmable. Scripting is possible through MDX, XMLA, or the PowerShell extensions for Analysis Services. Use the Data Mining Extensions (DMX) language for fast querying and scripting.
- Security and Deployment: Provides role-based security through Analysis Services, including separate permissions for drill through to model and structure data. Easy deployment of models to other servers, so that users can access the patterns or perform predictions

- The data to be analysed is in SQL server already. That can save us lot of pre-processing task to filter and transform data because we can use comprehensive capabilities of SQL server to process the data before it is ready for mining and analysis.
- SQL Server Data Mining tools are also equipped with enhanced Business Intelligence tools so we can have the option to further explore dimensions of the data without writing any scripts.
- The improvements that may be suggested for data acquisition and processing and also the short term and long term predictions will be tested in the same environment where the system is deployed. So SQL Server will provide a closer match for deployment environment as well

Considering the objectives of the analysis and saving effort on conversion of the data to make it ready for processing along with data warehouse of the current dataset indicates that Microsoft SQL Server is a better choice for applying machine learning algorithms. It also allows an easy customization of the techniques to try different mining models to exploit the data from different perspectives.

CHAPER 5: IMPLEMENTATION & EVALUATION

5.1 Introduction

This chapter provides details of the implementation of the proposed system architecture presented in the previous chapter. The novel architecture solution is applied to an existing and operational transport system, run by the partner organization on this project, Mermaid Technology, using a subset of their transport data for Buses. The subset is taken from the data because this phase will implement and evaluate algorithms and then the whole dataset will be used in next chapter where a comprehensive predictive monitoring system is implemented and evaluated. Also, the dataset chosen for this phase of implementation and evaluation picks the routes where complex scenarios exist and the functionality of the system is tested to the maximum level. It aims to improve the reliability and performance of the transport system, in real-time, by applying predictive analysis methods. The implemented system is evaluated for its performance and compared with the existing system for its ability to address challenges pointed out in previous chapter. This chapter starts with an insight into the existing data structure of the Mermaid Technology system. The data, in its current form, is visualized to enable drawing comparisons between the performance of the existing system and the proposed one. Various data mining algorithms have been applied to experiment with the system in order to obtain an optimal reliability and performance. In this chapter, the journey through the selected scenarios that led to the final choice of methods and techniques to achieve the aim is presented and discussed. It is followed by the presentation of the process through which selected machine learning algorithms are applied to the data to identify patterns of behaviours and make predictions based on the available information, in real-time. A comparison between different technologies and software environments pertinent to the problem of this project has been drawn and an example application created using historical data to demonstrate the advantages and disadvantages of the 'old' and the proposed system performance. It also facilitates juxtaposing, in visual terms, the existing data representation methods with the patterns identified through the application of different data mining models.

This chapter also demonstrates that the implemented methods constitute a self-learning system whereby the process of data capturing through to a transformation to the format ready for processing is automated. The data being captured in real-time is not only used for immediate predictions but also is fed into the process of predictive analysis methods so that its impact is also added to the prediction component for future results. A snapshot of the system's performance and status in different parts of the transport network is maintained and always updated with the new data coming in. It helps in providing short-term prediction results without querying large volumes of data in permanent storage. The prediction component makes use of a novel caching strategy that stores the processed data in a purpose-designed hierarchical data structure, which facilitates real-time storage and retrieval of analyzed data tagged with the spatial coordinates. The novel data structure is designed to provide access to the data based on a customized zone level strategy without comparing location of the requesting device with all the locations, which can potentially make the comparison slow. This novel hierarchical data structure is explained and its use is described towards end of this chapter.

The implementation of the predictive analysis component is presented in detail making use of the prediction probabilities generated by machine learning algorithms. It provides real-time prediction of the bus arrival times, which are then used for short-term forecasting of congestion and delays. The predicted arrival times are compared with the real data coming from the buses to evaluate the accuracy of the prediction and the consistency and reliability of the information. The implementation technology used to implement different components of the proposed architecture such as Web Services, Novel Caching Strategy and Hierarchical Data Structure, and the Prediction Module - is also discussed in this chapter. This chapter presents the architecture implementation and application for a subset of the entire data set of journeys considering a limited number of routes detailed in the experimental settings in next section. The detailed implementation of the proposed architecture, using the full dataset, is presented and discussed in the Chapter 6, as a Case Study, where the full automation of the process of data acquisition and data analytics is explained and demonstrated.

5.2 Experimental Setting

The experiment is aimed at evaluating the proposed architecture using a subset of data obtained from the industrial partner on this project, Mermaid Technology. They run an urban transport system with 1200 buses equipped with many on-board sensors transmitting the data to the central hub in real-time. The sensors on the buses are connected to the hub for sharing information through the server to optimize the journeys and to manage transport effectively.

This information is also made available to an external data provider (e.g. Movia), to be combined with similar information obtained from other modes of transport such as trains, to plan an optimal use of time and resources in facilitating the commuters' journeys. The information is converted into routes estimations and distributed to different display networks that manage display of this information on different important places like train stations, airports, hotels and tourist places around the country.

Out of 1200 buses operational on more than 1298 routes, 188 routes were selected for this subset, with 128 buses providing commuting to thousands of passengers every day. Each bus is equipped with sensors for GPS, Speed, Direction, Bus Operations, Signs, Traffic Lights and Cameras for CCTV recording of the bus. Each bus generates a data volume of at least 1 KB every second representing values for these different sensors and the information presented to passengers for staying updated with the progress of their journey. The buses send this information on configurable intervals to the server hub where this information is stored and processed so that it can be shared with other buses to keep them aware of each other when they are on the same route. The information is collected using different components installed on the bus and these components merge the information so that a proper structured information can be sent to the server to facilitate the analysis. This rich context information collected from each bus enables the server system to support provision of near to accurate prediction mechanisms.

5.3 Predictive Analysis

This implementation goes through the whole sequence of processing the data and making it ready for prediction using different steps of data acquisition, clean-up, basic visualization for pattern identification and data understanding and eventually applying machine learning techniques to convert the analysis into prediction probabilities, so that the prediction of arrival times for the selected route can be calculated. The selected techniques will be applied to a larger dataset in the next chapter 6 of Case Study to apply predictive analysis to the whole system and present a novel prediction system using the rich context of the system itself. The dataset is available in a data warehouse based on SQL Server Database. There are two strategies that have been followed to get the data for this analysis. One is to take history of the data from the warehouse and copy data for selected route into a local data store server used for the analysis.

The real-time data is captured using the web services and also applying triggers to the database so that new data is exported for analysis as soon as it arrives.

5.3.1 Technical Limitations of the Data

The data review is done to make sure that outliers are identified and any corruption in the data is discovered before analysis is started as it will have impact on the accuracy of the results. When looking at the limited dataset, it was discovered that we have scenarios to consider so that we can check what we can provide to make the predictions work better. Also, because it is typically a big data problem, it is important to consider if there is any data cycle that we can consider. The data cycle may reflect some parameters which we should consider in using the data for predictions. Data inspection is a very important stage as this decides the input of the prediction algorithms and if we don't have valid training data; it will reflect in our predictions. We find anomalies in the data and associate incidents with those anomalies and then also associate the impact factor with those incidents. The following are some known scenarios.

5.3.1.1Journeys starting late

There are many journeys that start late. It means there is a good chance that the prediction for the first stop will be frequently wrong as we don't know and we cannot find if and why a bus can be late on the first stop. The first stop is also the starting station so generally we have no idea about these delays. However, we can find a pattern and associate the delay with those journeys or some specific stops so that it can be considered in the prediction calculation.

5.3.1.2 Cancelled journeys

There are some journeys that are cancelled once started for many reasons. It means that the journeys for which it was cancelled at some specific stop, the remaining stops will have unknown state about arrival and departure. We should not consider any such stops because absence of arrival and departure data on these stops will have impact on the calculation if we use different formulas like mean or average or median. This should be cleaned in a pre-processing stage so that such stops are not considered. This is another state of the journey and we need to associate the incident with these journeys so that they can be pruned completely or just consider only those stops of the journey where its stops were actually driven through.

5.3.1.3 Accidents/buses broken down/traffic

There are some journeys that are genuinely delayed but they still continue. The reasons can be a road accident involving some other vehicle on the road or a traffic jam or any emergency or roadwork that has caused the bus to wait. If such journeys actually continue then it means such incidents can happen again and they should be used for prediction. There is no data available for the such incidents in the dataset.

5.3.1.4 Midnight problems

One thing which is clearly established from the data that the journeys which start around midnight or more specifically that start before midnight and complete after midnight have invalid data. So we can consider this an incident. We can also decide to remove any such data or event fix this data so that we can use it for prediction models.

5.3.1.5 Last stop problems

There are quite a few journeys where the last stop has reported delay that is quite higher. There is no known incident for this reason but we need to find a pattern for this as well so that prediction for last stop can also be accurate.

5.3.2 Application of Machine Learning Algorithms

There are four algorithms that have been employed to analyze the data and convert the raw information into understandable predictions about behavior of the transport network. The list includes Linear Regression, Neural Networks, Clustering and Decision Tree. The details for these techniques are presented in Chapter 3. The algorithms did not perform well with the raw data and also the mining models produced based on the raw data are not useful. This happens because the raw data structure is not friendly for predictive analysis and needs to be converted to a data structure which has different dimensions of the data available for exploitation. The algorithms were trialed on different stages as the data was prepared for acceptance by the algorithms. This resulted in transformation of the data into a shape, which can produce reasonably good mining models that can be input to the algorithms for effective analysis. The behavior of all these four algorithms was customized with different parameters like sampling parameters, splitting of information like timestamp into individual hours and minutes to help the algorithms perform accurately. The output of each algorithm was converted to probability index and then accuracy index was calculated for each probability index. This work applied a novel incremental yet recursive approach to conduct the analysis of data where the probability output of one algorithm is combined with output of another algorithm to find the best match for making prediction. Although the score indicator for each algorithm indicates how fit it is to make the arrival predictions, the fitness of this score was matched with each individual stop and different approaches were considered like taking the average of probability from all four algorithms or selecting the best probability only. This gave us the opportunity to find the best selection algorithm to find better predictions.

5.3.2.1 Important Concepts in the Existing System

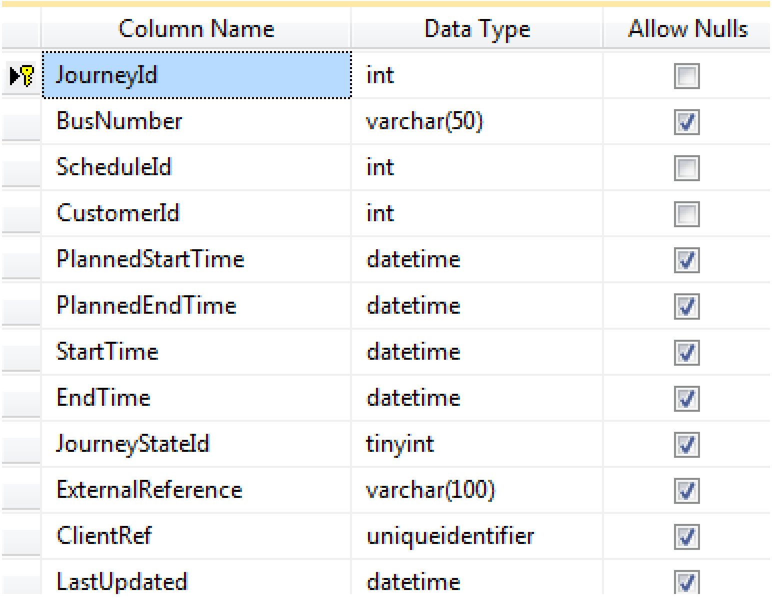
This section presents information about some important concepts in the existing system that reflect understanding of how it works and the way different information elements are connected to each other. There is a great deal of parameters that influence the performance of an urban transport system, and the list of parameters includes time of traveling, the day of the week, peak hours, the location of the stops, flow of traffic and availability of special lines for vehicles. The urban transport companies around the globe plan times for the vehicles for when they should be where so that the consumers can use this information to plan their journeys. The Figures 8 and 9 show structure of relevant entities from the existing transport system. It has some entities, and a brief description of each entity is given below.

- **Buses**: This entity represents the vehicles that are part of the urban transport system. It will store information about all the vehicles such as their current location, last time they contacted the system, the customer they belong to and information about if they should currently be driving.
- **Stops**: These are physical locations where buses are supposed to stop during the journeys.
- **Zones**: Every city in a developed country is divided into zones based on how close that is to the city center to make management of traffic and transport easier. The stops information includes the zone in which they are located.
- **Schedules**: This entity saves a list of different routes that are driven by the buses. Each route has a source point, a destination point and identification information to be displayed on buses to let consumers know about the destination of the vehicle. This

identification information includes Line, which shows the number of the route the bus is supposed to be driving when it is driving a specific schedule.

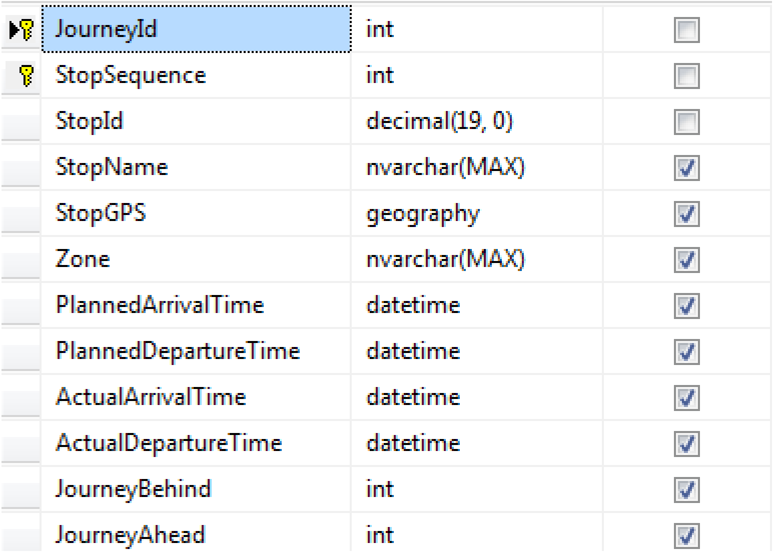
- **Journeys**: When a bus starts driving a route, it means it is attaching a specific time with a route and selecting a bus to drive at that time. Therefore, a journey has time information attached to a schedule. The concept of journeys is explained in further detail later in this chapter.

A trip made by the bus from a source point to a destination point at any specific time is referred as Journey and is the point of focus for all evaluation and implementation. As described in the Figure 5.1 and 5.2, for a potential transport system, we have selected two important tables for an explanation so that the reader has an understanding of the starting point for this research and how will the implementation transform the system architecture. The tables are called "**Journeys**" and "**JourneyStops**."



Column Name	Data Type	Allow Nulls
JourneyId	int	☐
BusNumber	varchar(50)	☑
ScheduleId	int	☐
CustomerId	int	☐
PlannedStartTime	datetime	☑
PlannedEndTime	datetime	☑
StartTime	datetime	☑
EndTime	datetime	☑
JourneyStateId	tinyint	☑
ExternalReference	varchar(100)	☑
ClientRef	uniqueidentifier	☑
LastUpdated	datetime	☑

Figure 5-1 Journeys table



▶🔑	JourneyId	int	☐
🔑	StopSequence	int	☐
	StopId	decimal(19, 0)	☐
	StopName	nvarchar(MAX)	☑
	StopGPS	geography	☑
	Zone	nvarchar(MAX)	☑
	PlannedArrivalTime	datetime	☑
	PlannedDepartureTime	datetime	☑
	ActualArrivalTime	datetime	☑
	ActualDepartureTime	datetime	☑
	JourneyBehind	int	☑
	JourneyAhead	int	☑

Figure 5-2 journey stops table

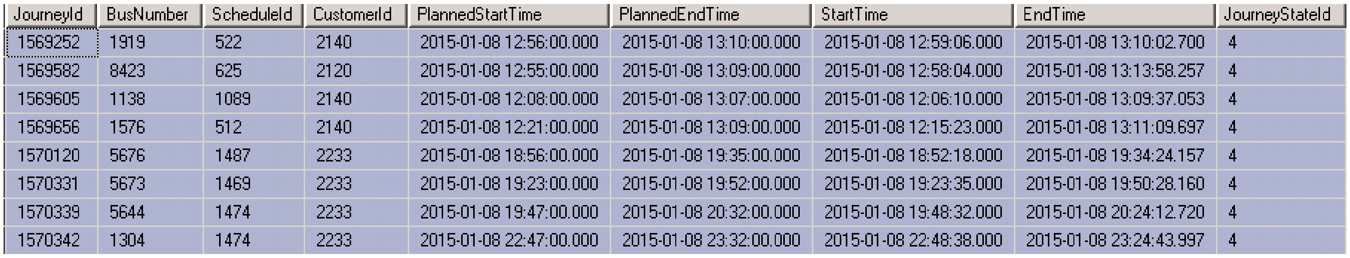
The detailed understanding of the two entities of Journeys and JourneyStops is presented below. The important elements of the data are Planned times, Actual times and Expected Times. These come from planning data of a transport company.

- PlannedStartTime – when the journey is supposed to start as per planned journeys
- PlannedEndTime – when the journey is supposed to complete
- PlannedArrivalTime – This is linked with a stop of that journey and shows the time when the bus should arrive at that stop as per planning
- PlannedDepartureTime – This is linked with a stop of that journey and shows when the bus should leave the stop as per planning
- ExpectedArrivalTime – This is linked with a stop of that journey and shows the time when the bus is expected to arrive at that stop when it is moving.
- ExpectedDepartureTime – This is linked with a stop of that journey and shows when the bus is expected to depart from the stop when bus is moving
- ActualArrivalTime – This is linked with a stop of that journey and shows the time when the bus actually arrived at that stop when it is moving.
- ActualDepartureTime – This is linked with a stop of that journey and shows the time when the bus actually departed from a stop when it is moving.
- Zone – This is the zone the stop is part of

- Schedule – This is the route being driven that has Line, From, Destination and Via information attached with it.

Journeys

This table stores and represents the journeys being driven or that have been driven by the buses at different times of the day. In the planning process for a transport system, the planning company will decide a schedule for the journeys that will decide how many routes exist in the system and how many visits each bus will make for each route during the day. The schedule contains expected information about what time the bus will depart from first stop and then with expected times on all the stops of that route; it will mention what time will the bus reach the destination. The Journey is a representation of the planned schedule in real-time, which means that it will show the actual times when the bus departed and reached all stops while driving before reaching the final destination. The Journeys table represents overall information about the journey and that includes the number of the bus that drove that this journey, the planned start time of the journey extracted from schedule, actual start time of the journey when bus started the journey, planned end time of the journey extracted from the schedule, actual end time when the bus finished the journey and state of the journey that indicates if the journey has been driven or it is still in progress. Other information in the table is for linking and referencing the journey for management and administration purposes. It is important to note that arrival time on the first stop becomes start time of the journey and arrival time at last stop becomes end time of the journey. A typical representation of journeys is shown in Table 5.1.



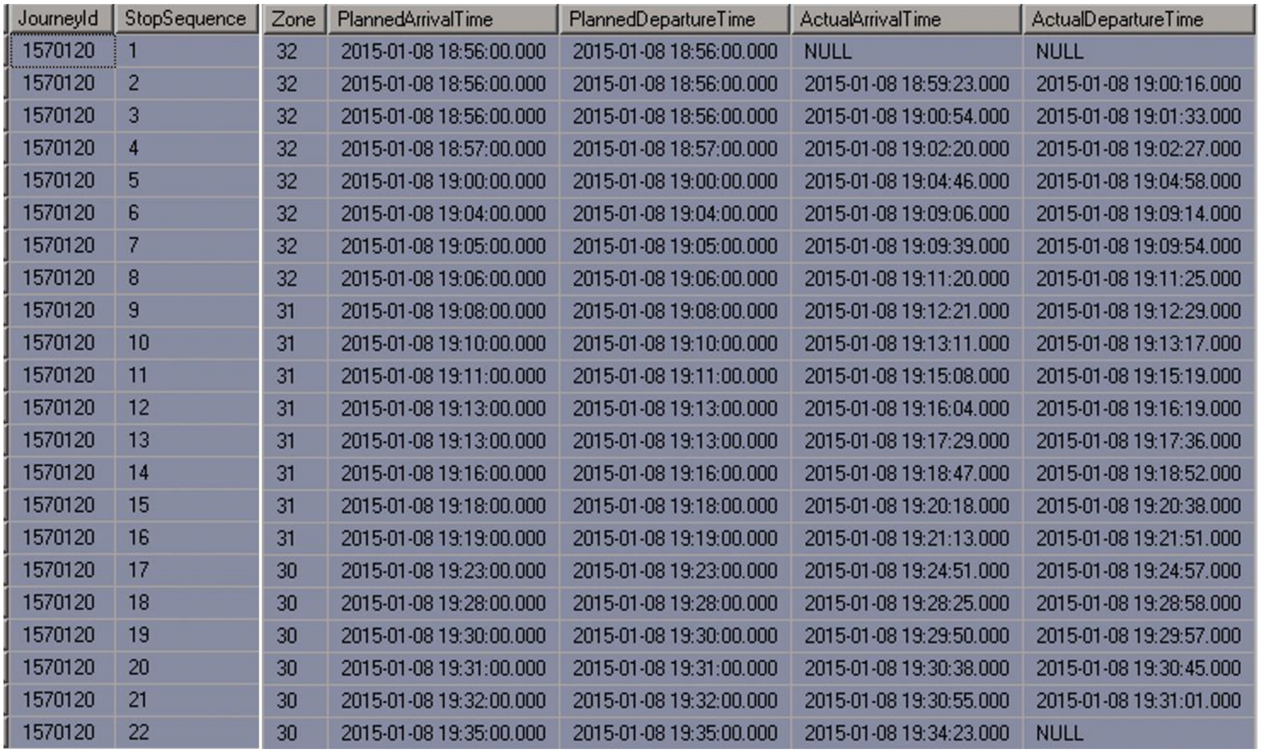
JourneyId	BusNumber	ScheduleId	CustomerId	PlannedStartTime	PlannedEndTime	StartTime	EndTime	JourneyStateId
1569252	1919	522	2140	2015-01-08 12:56:00.000	2015-01-08 13:10:00.000	2015-01-08 12:59:06.000	2015-01-08 13:10:02.700	4
1569582	8423	625	2120	2015-01-08 12:55:00.000	2015-01-08 13:09:00.000	2015-01-08 12:58:04.000	2015-01-08 13:13:58.257	4
1569605	1138	1089	2140	2015-01-08 12:08:00.000	2015-01-08 13:07:00.000	2015-01-08 12:06:10.000	2015-01-08 13:09:37.053	4
1569656	1576	512	2140	2015-01-08 12:21:00.000	2015-01-08 13:09:00.000	2015-01-08 12:15:23.000	2015-01-08 13:11:09.697	4
1570120	5676	1487	2233	2015-01-08 18:56:00.000	2015-01-08 19:35:00.000	2015-01-08 18:52:18.000	2015-01-08 19:34:24.157	4
1570331	5673	1469	2233	2015-01-08 19:23:00.000	2015-01-08 19:52:00.000	2015-01-08 19:23:35.000	2015-01-08 19:50:28.160	4
1570339	5644	1474	2233	2015-01-08 19:47:00.000	2015-01-08 20:32:00.000	2015-01-08 19:48:32.000	2015-01-08 20:24:12.720	4
1570342	1304	1474	2233	2015-01-08 22:47:00.000	2015-01-08 23:32:00.000	2015-01-08 22:48:38.000	2015-01-08 23:24:43.997	4

TABLE 5-1 Journeys from real database

Here JourneyId represents unique id of the journey in the system. BusNumber is a unique number given to the bus driving this journey, ScheduleId refers to the planned schedule that this bus selected to drive for this journey and remaining are the timestamps already explained.

Journey Stops

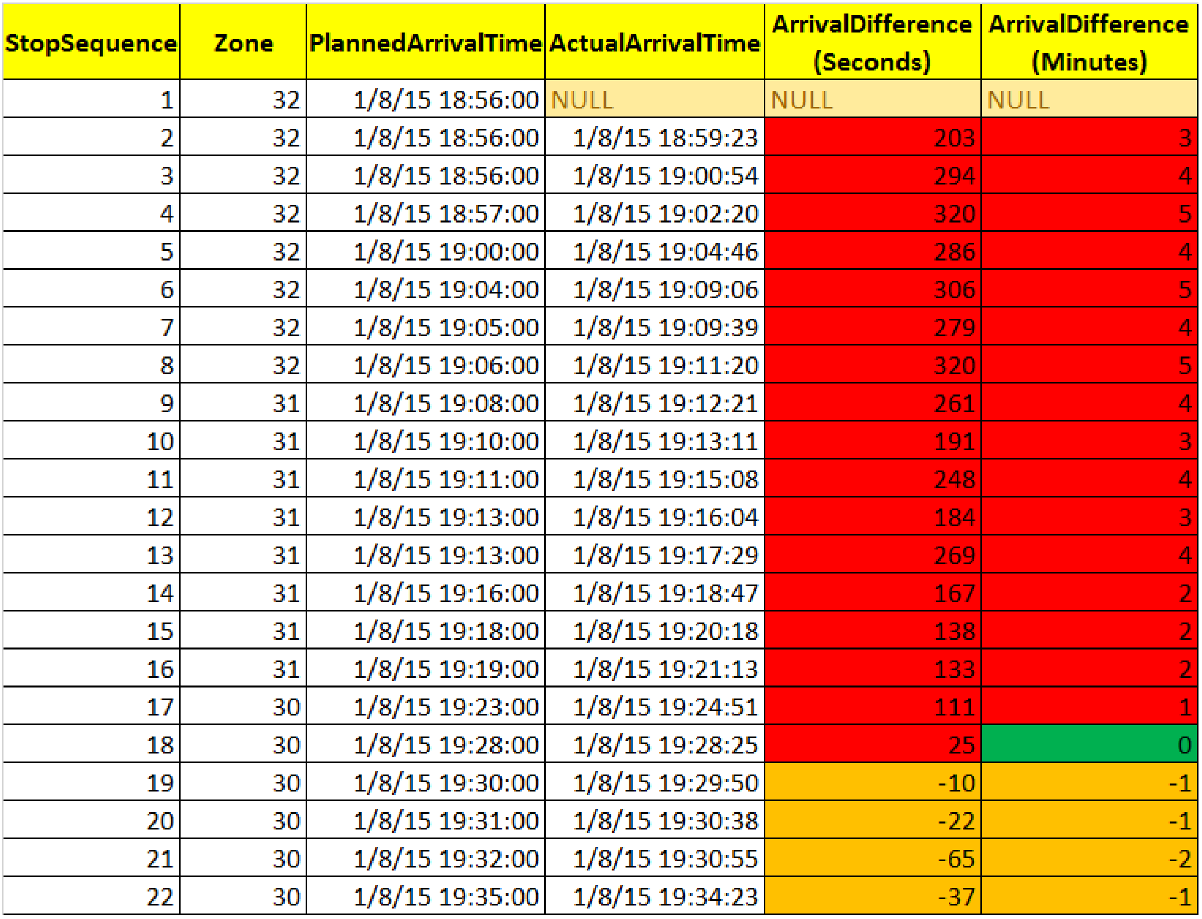
This table stores information about individual stops of the route. It joins planned time information from the schedule and actual time from the real-time driving to know the status of the specific journey at any point of time during the journey. Although this table stores a lot of information about each stop, important parameters are StopID as a unique identification of the stop, StopName as the name of the stop; PlannedArrivalTime extracted from the schedule and ActualArrivalTime from the actual arrival of the bus in real-time. It also has PlanneddepartureTime extracted from the schedule, ActualDepartureTime from the actual departure of the bus in real-time, Longitude and Latitude represented as StopGPS, StopSequence representing a sequence of the stop in the list and Zone representing the physical zone of the city where the stop is located. Stops detail for one of the journeys listed in Table 1 is shown in Table 5.2.



JourneyId	StopSequence	Zone	PlannedArrivalTime	PlannedDepartureTime	ActualArrivalTime	ActualDepartureTime
1570120	1	32	2015-01-08 18:56:00.000	2015-01-08 18:56:00.000	NULL	NULL
1570120	2	32	2015-01-08 18:56:00.000	2015-01-08 18:56:00.000	2015-01-08 18:59:23.000	2015-01-08 19:00:16.000
1570120	3	32	2015-01-08 18:56:00.000	2015-01-08 18:56:00.000	2015-01-08 19:00:54.000	2015-01-08 19:01:33.000
1570120	4	32	2015-01-08 18:57:00.000	2015-01-08 18:57:00.000	2015-01-08 19:02:20.000	2015-01-08 19:02:27.000
1570120	5	32	2015-01-08 19:00:00.000	2015-01-08 19:00:00.000	2015-01-08 19:04:46.000	2015-01-08 19:04:58.000
1570120	6	32	2015-01-08 19:04:00.000	2015-01-08 19:04:00.000	2015-01-08 19:09:06.000	2015-01-08 19:09:14.000
1570120	7	32	2015-01-08 19:05:00.000	2015-01-08 19:05:00.000	2015-01-08 19:09:39.000	2015-01-08 19:09:54.000
1570120	8	32	2015-01-08 19:06:00.000	2015-01-08 19:06:00.000	2015-01-08 19:11:20.000	2015-01-08 19:11:25.000
1570120	9	31	2015-01-08 19:08:00.000	2015-01-08 19:08:00.000	2015-01-08 19:12:21.000	2015-01-08 19:12:29.000
1570120	10	31	2015-01-08 19:10:00.000	2015-01-08 19:10:00.000	2015-01-08 19:13:11.000	2015-01-08 19:13:17.000
1570120	11	31	2015-01-08 19:11:00.000	2015-01-08 19:11:00.000	2015-01-08 19:15:08.000	2015-01-08 19:15:19.000
1570120	12	31	2015-01-08 19:13:00.000	2015-01-08 19:13:00.000	2015-01-08 19:16:04.000	2015-01-08 19:16:19.000
1570120	13	31	2015-01-08 19:13:00.000	2015-01-08 19:13:00.000	2015-01-08 19:17:29.000	2015-01-08 19:17:36.000
1570120	14	31	2015-01-08 19:16:00.000	2015-01-08 19:16:00.000	2015-01-08 19:18:47.000	2015-01-08 19:18:52.000
1570120	15	31	2015-01-08 19:18:00.000	2015-01-08 19:18:00.000	2015-01-08 19:20:18.000	2015-01-08 19:20:38.000
1570120	16	31	2015-01-08 19:19:00.000	2015-01-08 19:19:00.000	2015-01-08 19:21:13.000	2015-01-08 19:21:51.000
1570120	17	30	2015-01-08 19:23:00.000	2015-01-08 19:23:00.000	2015-01-08 19:24:51.000	2015-01-08 19:24:57.000
1570120	18	30	2015-01-08 19:28:00.000	2015-01-08 19:28:00.000	2015-01-08 19:28:25.000	2015-01-08 19:28:58.000
1570120	19	30	2015-01-08 19:30:00.000	2015-01-08 19:30:00.000	2015-01-08 19:29:50.000	2015-01-08 19:29:57.000
1570120	20	30	2015-01-08 19:31:00.000	2015-01-08 19:31:00.000	2015-01-08 19:30:38.000	2015-01-08 19:30:45.000
1570120	21	30	2015-01-08 19:32:00.000	2015-01-08 19:32:00.000	2015-01-08 19:30:55.000	2015-01-08 19:31:01.000
1570120	22	30	2015-01-08 19:35:00.000	2015-01-08 19:35:00.000	2015-01-08 19:34:23.000	NULL

TABLE 5-2 Details of stops information for a journey

The Table 5.2 shows stop information for a specific journey with JourneyId 1570120. It shows the planned and actual times for the stops with actual arrival time missing for first stop and actual departure time missing for last stop. Although there are many different scenarios that exist for the data to be empty or populated but in the ideal case, all the stops should have their planned and actual times populated. To be able to understand this table better, the Table 5.3 represents the some extra information extracted from these planned and actual times and that is status about early, on-time and late arrivals. The information is presented in both minutes and seconds as consideration of delay in minutes or seconds can change the way we look at the system. The cells marked red in Table 5.3 indicate that the bus was delayed on the stops against the planned arrival time, the green indicates on-time arrival and the orange cells indicate that the bus reached the stops earlier. We can also see one stop where it shows as on-time when considered as minutes and late when considered as seconds. It depends on the how the data is processed and it can influence the evaluation as well as the prediction of arrival and departure times.



StopSequence	Zone	PlannedArrivalTime	ActualArrivalTime	ArrivalDifference (Seconds)	ArrivalDifference (Minutes)
1	32	1/8/15 18:56:00	NULL	NULL	NULL
2	32	1/8/15 18:56:00	1/8/15 18:59:23	203	3
3	32	1/8/15 18:56:00	1/8/15 19:00:54	294	4
4	32	1/8/15 18:57:00	1/8/15 19:02:20	320	5
5	32	1/8/15 19:00:00	1/8/15 19:04:46	286	4
6	32	1/8/15 19:04:00	1/8/15 19:09:06	306	5
7	32	1/8/15 19:05:00	1/8/15 19:09:39	279	4
8	32	1/8/15 19:06:00	1/8/15 19:11:20	320	5
9	31	1/8/15 19:08:00	1/8/15 19:12:21	261	4
10	31	1/8/15 19:10:00	1/8/15 19:13:11	191	3
11	31	1/8/15 19:11:00	1/8/15 19:15:08	248	4
12	31	1/8/15 19:13:00	1/8/15 19:16:04	184	3
13	31	1/8/15 19:13:00	1/8/15 19:17:29	269	4
14	31	1/8/15 19:16:00	1/8/15 19:18:47	167	2
15	31	1/8/15 19:18:00	1/8/15 19:20:18	138	2
16	31	1/8/15 19:19:00	1/8/15 19:21:13	133	2
17	30	1/8/15 19:23:00	1/8/15 19:24:51	111	1
18	30	1/8/15 19:28:00	1/8/15 19:28:25	25	0
19	30	1/8/15 19:30:00	1/8/15 19:29:50	-10	-1
20	30	1/8/15 19:31:00	1/8/15 19:30:38	-22	-1
21	30	1/8/15 19:32:00	1/8/15 19:30:55	-65	-2
22	30	1/8/15 19:35:00	1/8/15 19:34:23	-37	-1

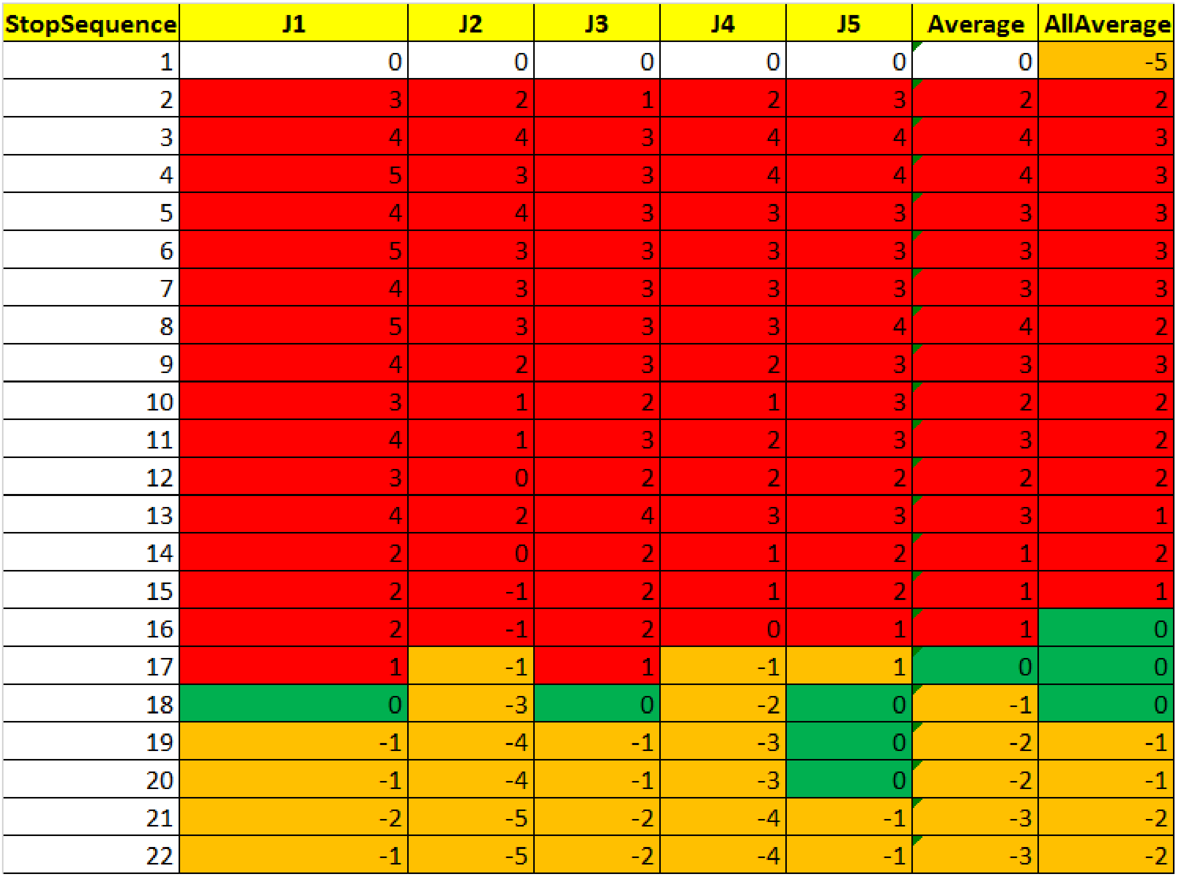
TABLE 5-3 Arrival difference data for a journey

The trend chart for the above data is shown in Figure 5.3 below where a trend line indicates how the journey performed when driven. It shows two different perspectives and the impact each perspective has on system performance. If we look at trend line of both minutes and seconds together, it shows that probably the seconds' line is indicating clearly the delay trend whereas the minute's line is not. In reality, the minute's line is hidden just because of the value it has compared to the seconds. On the right side in the same Figure 5.3, the same minute trend line is represented separately and it shows a similar trend like seconds line and therefore both charts are representing the journey with the same trend that it starts with delays and then catches up

Figure 5-3 trend line

as the vehicle reaches the last stop and it reports little early arrivals on last stops.

Although the visualization shown above gives information about the pattern of journey delay for this one journey, information about journeys at the same time will give insight into the behavior of the journeys on that route and the time.



StopSequence	J1	J2	J3	J4	J5	Average	AllAverage
1	0	0	0	0	0	0	-5
2	3	2	1	2	3	2	2
3	4	4	3	4	4	4	3
4	5	3	3	4	4	4	3
5	4	4	3	3	3	3	3
6	5	3	3	3	3	3	3
7	4	3	3	3	3	3	3
8	5	3	3	3	4	4	2
9	4	2	3	2	3	3	3
10	3	1	2	1	3	2	2
11	4	1	3	2	3	3	2
12	3	0	2	2	2	2	2
13	4	2	4	3	3	3	1
14	2	0	2	1	2	1	2
15	2	-1	2	1	2	1	1
16	2	-1	2	0	1	1	0
17	1	-1	1	-1	1	0	0
18	0	-3	0	-2	0	-1	0
19	-1	-4	-1	-3	0	-2	-1
20	-1	-4	-1	-3	0	-2	-1
21	-2	-5	-2	-4	-1	-3	-2
22	-1	-5	-2	-4	-1	-3	-2

Table 5-4 Journeys delay data for 5 journeys

Information about five journeys and average of delays for all those five journeys is shown in Table 5.4. The data indicates that that journeys on this route are late at initial stops, but they catch up towards the last stops. The comparison is shown as a chart with all five journeys, and their average delay information is shown in Figure 5.5 below.

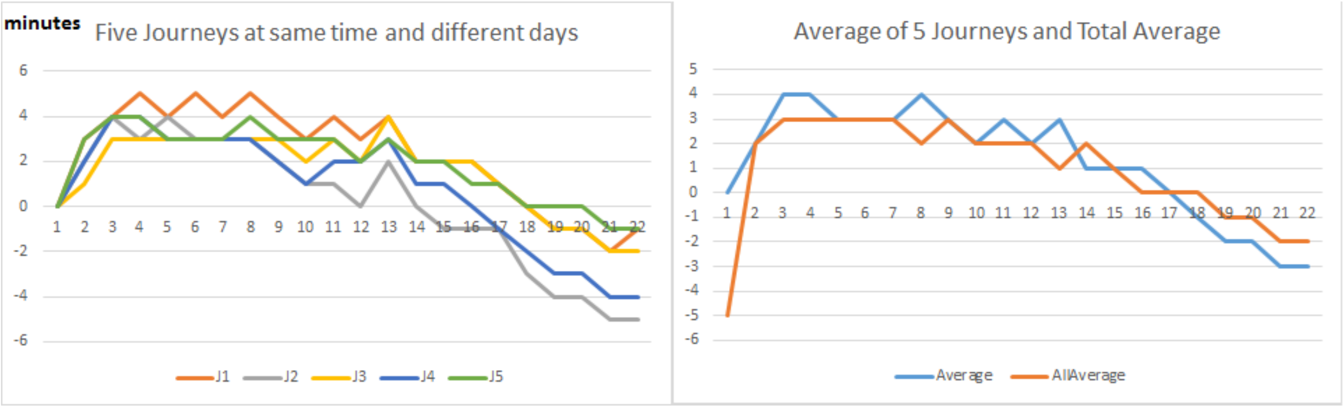
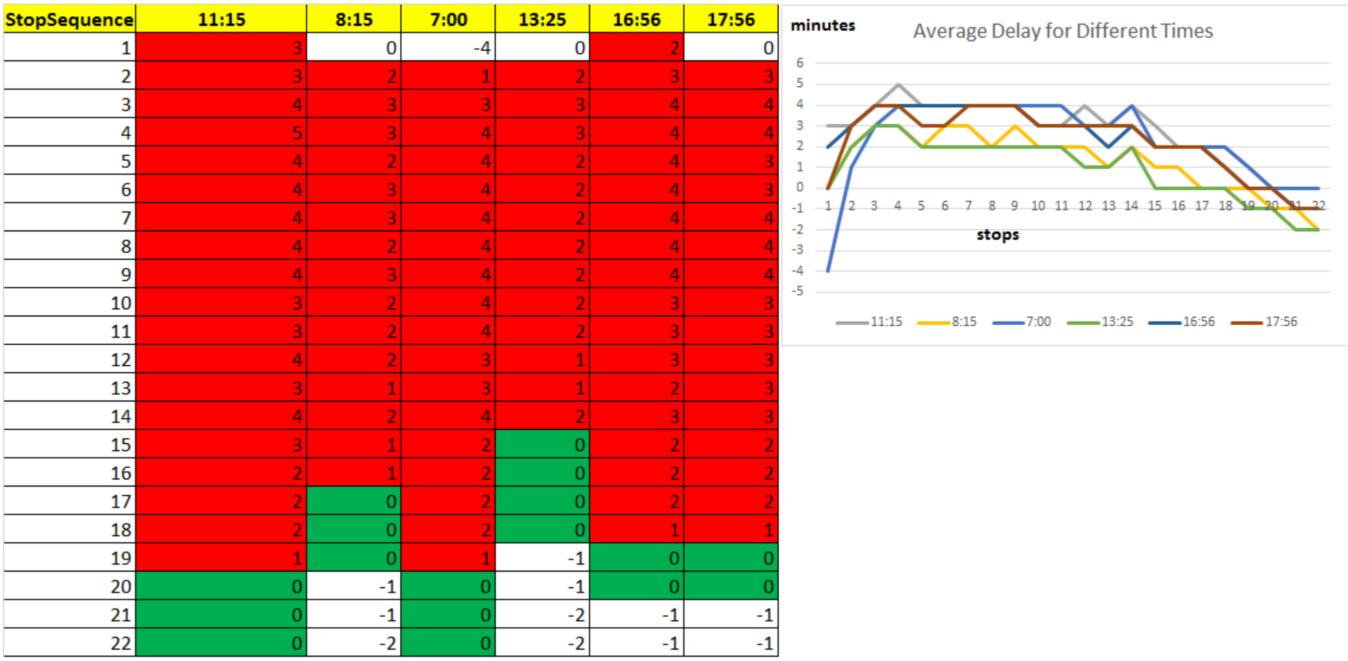


Figure 5-5: Line chart showing delay for 5 journeys

The trend of delay is very close to the average delay, and that indicates that at this time “18:56” on this route, the journeys are late in the start but catch up towards the end of the journey. The diagram below Figure 5.6 represents the average delays for others times of the day on the same route.



StopSequence	11:15	8:15	7:00	13:25	16:56	17:56
1	3	0	-4	0	2	0
2	3	2	1	2	3	3
3	4	3	3	3	4	4
4	5	3	4	3	4	4
5	4	2	4	2	4	3
6	4	3	4	2	4	3
7	4	3	4	2	4	4
8	4	2	4	2	4	4
9	4	3	4	2	4	4
10	3	2	4	2	3	3
11	3	2	4	2	3	3
12	4	2	3	1	3	3
13	3	1	3	1	2	3
14	4	2	4	2	3	3
15	3	1	2	0	2	2
16	2	1	2	0	2	2
17	2	0	2	0	2	2
18	2	0	2	0	1	1
19	1	0	1	-1	0	0
20	0	-1	0	-1	0	0
21	0	-1	0	-2	-1	-1
22	0	-2	0	-2	-1	-1

Figure 5-6 Delay analysis of journey at diFferent times

There are some key advantages of this dataset available from the industrial partner – Mermaid Technology compared to other datasets available like the one available from Traffic for London. It is raw data collected from the network giving liberty to the implementation to interpret and structure it for best use in the algorithms. It is real data from an operational system and its relevance is further strengthened with a consistent influx of real-time data stream. Both the planning as well as operational data is available in one place and therefore, integration and comparison of planned versus operational data is realistic. One very important benefit of this dataset is that there is a completely operational system available to trial the algorithms and results and the extent of access to the system for both data access and application in the field is priceless.

5.3.2.2 Delay Analysis at different time of the day

The Figures 5.7 to 5.11 present a basic analysis of the delay situation in different trips made by the buses. The delay is noted at different days and different times of the day for some routes to get an idea of how much delays are being reported and then this information is used to establish rules for prediction of delays.

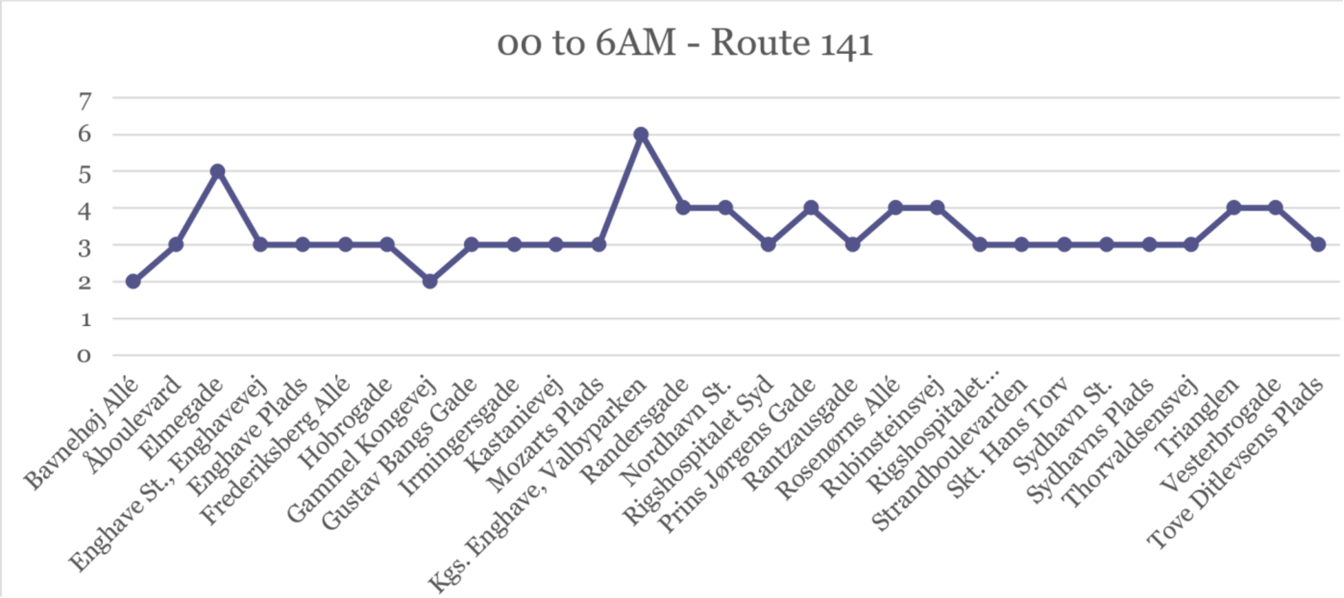


Figure 5-7 12am to 6am route 141, delay in minutes

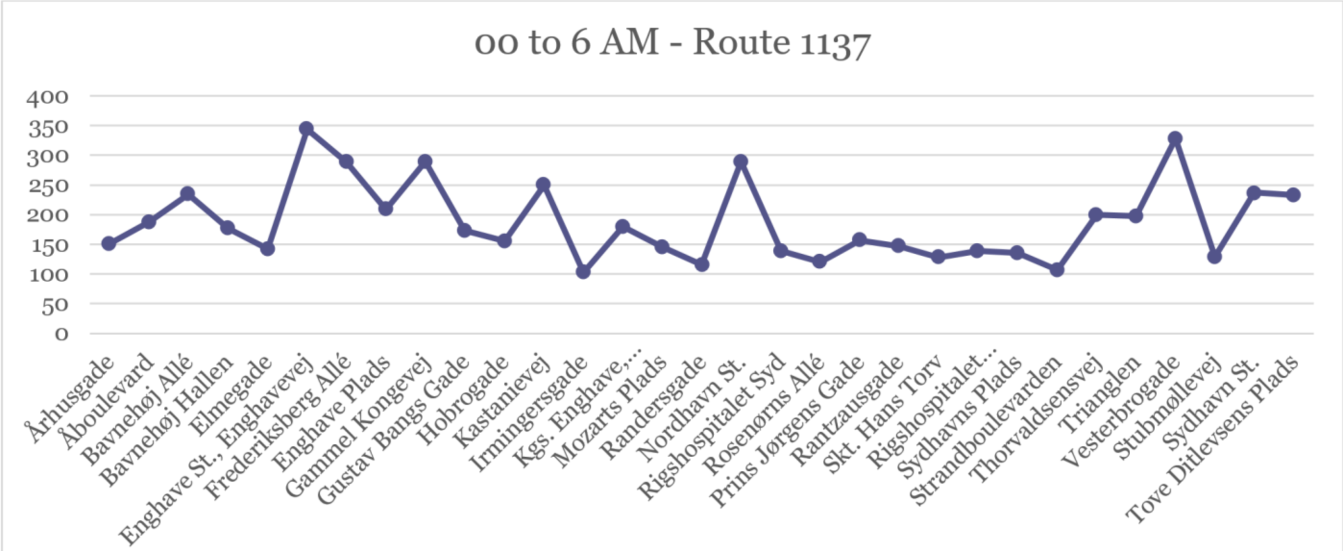


Figure 5-8 12 am to 6 am route 1137, delay in seconds

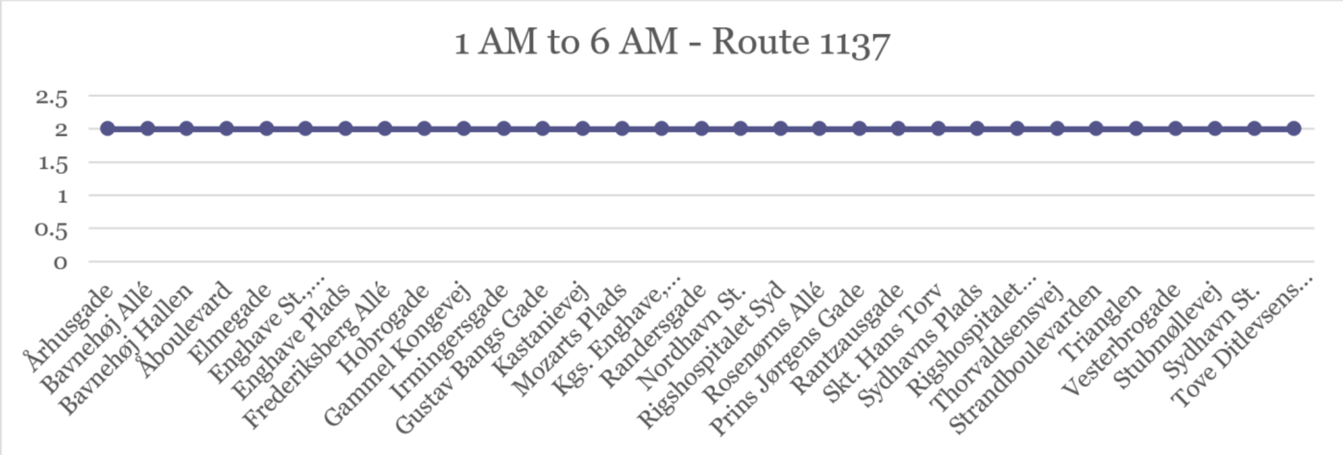


Figure 5-9 1am to 6 AM route 1137, delay in minutes

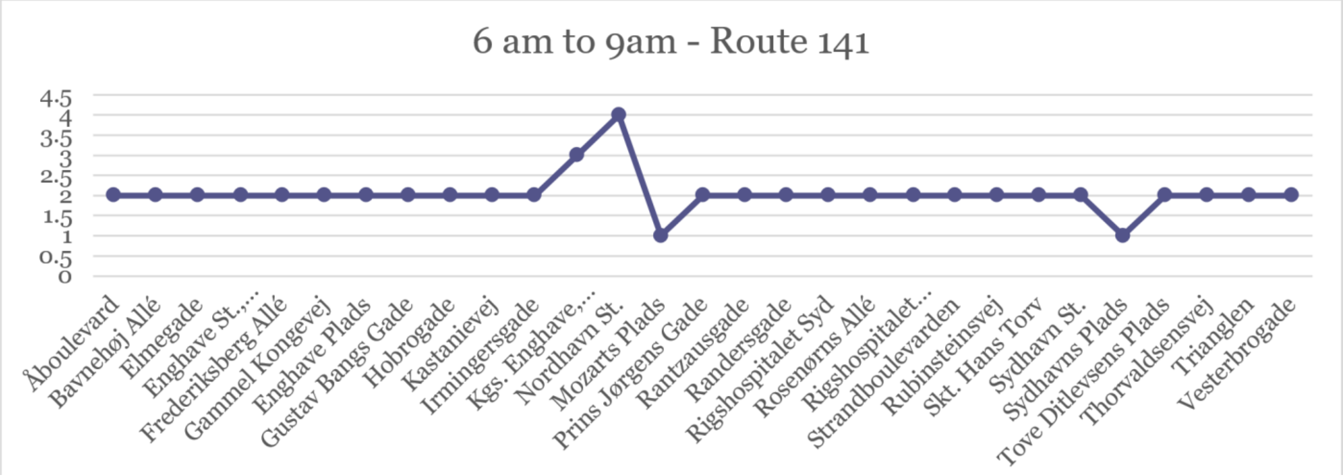


Figure 5-10 6 am to 9 am route 141, delay in minutes

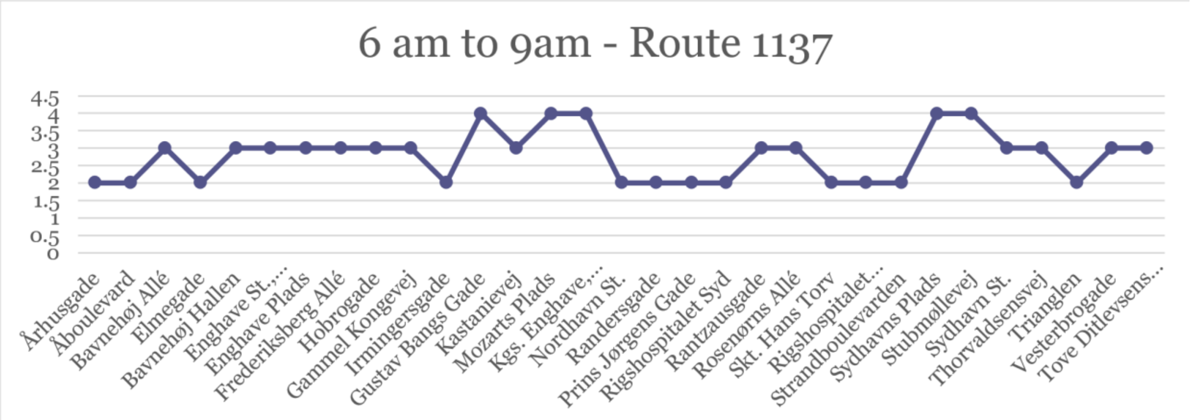


Figure 5-11 6am to 9am route 1137, delay in minutes

These plots indicate a trend of delays consistent across different times of the day and the routes. It means that plotting these delays accurately is very important so that the people waiting on the stops can be informed with reliable delay information to avoid their discomfort with the transport service. The analysis indicates that the delay is different on different routes and also the delay situation is different in peak and off-peak times. Because the arrival times are linked with departure times for the buses, it is important to plot departure time as well along with arrival time so that the impact of delayed or early arrival times can be seen on the departure time of the buses. Although it is up to the driver to make a decision if he should wait for the same dwell time on the stops when the buses are late or early on stops but having them plotted together will give an insight on not only the relationship of these two times but also it can indicate about behavior of the bus drivers for their decision making. The Figures 5.12 to 5.17 in next section plot Arrival and Departure time together for this purpose. This plotting is not done for a specific route and multiple routes are picked and the plot is made for the days in general. The plan is to identify the maximum and minimum delays reported on stops and see how the departure is plotted against it. It is also the purpose to identify any patterns that may exist on days so that these variations can be considered in machine learning algorithms.

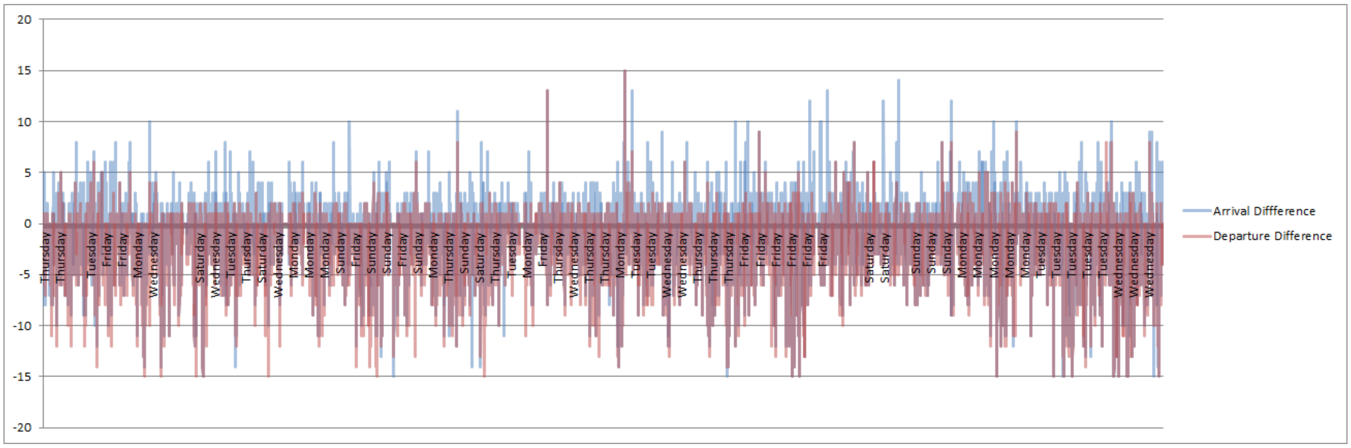


Figure 5-12 early and delay plotted together, delay in minutes

This shows day wise data for all the journeys in the dataset. It was filtered for 15 minutes plus/minus value of ArrivalDifference and DepartureDifference. Although DepartureDifference is being shown at some places but we are considering only ArrivalDifference to establish some rules that can be used to include more parameters like DepartureDifference as that will have impact on arrival time of next stops. We can see from the picture above that we have many instances when the bus arrived late or early. Usually arrivals were late and Departures were early. The pictures below will show the data for different time slots so that we can see journey punctuality snapshot for different timeslots of the day compared to above picture that shows it for the whole data.

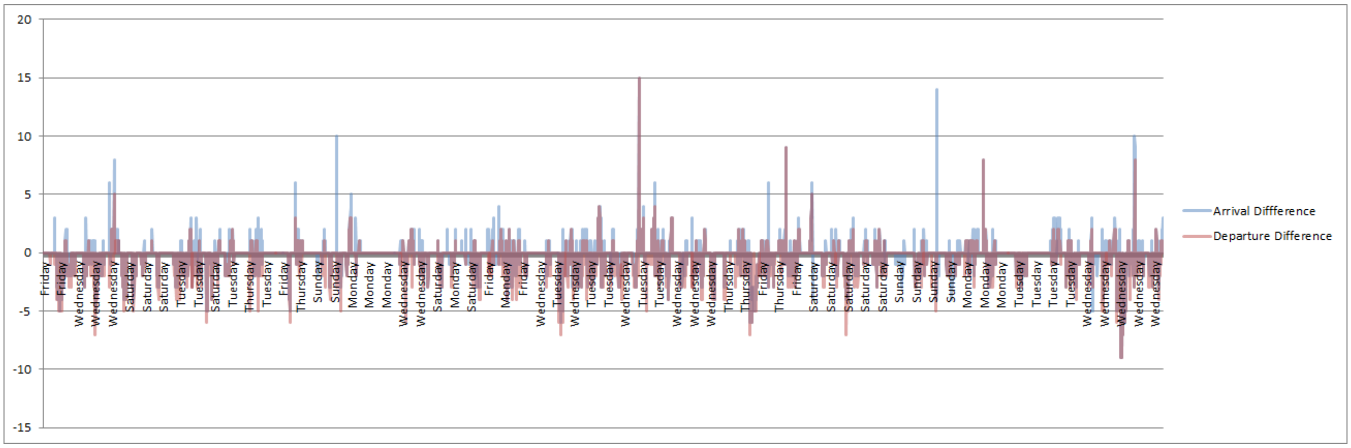


Figure 5-13 plotting of both early and late arrival 12 AM TO 5 AM, delay in minutes

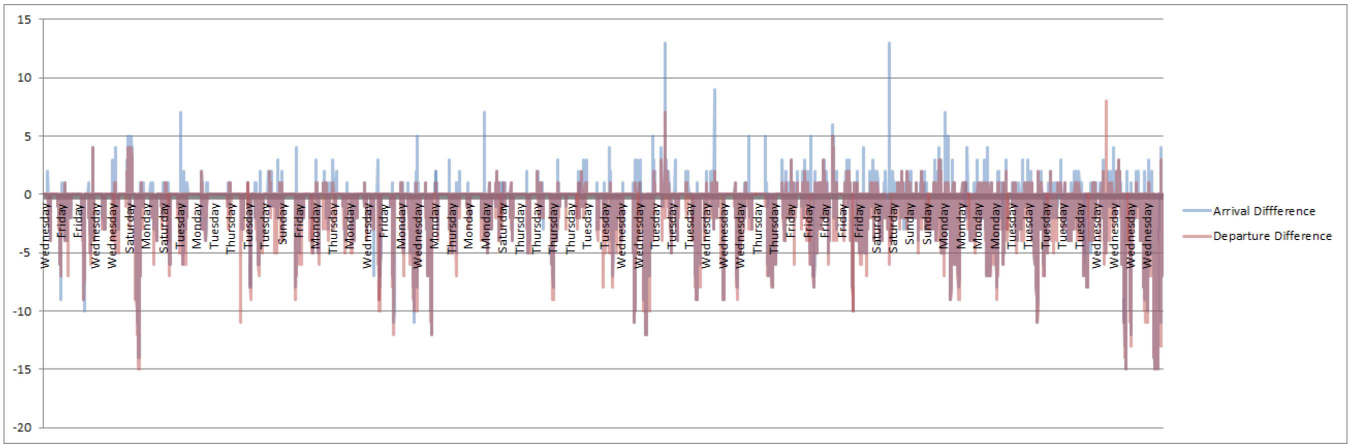


Figure 5-14 Plotting of both early and late arrival 6 AM TO 9 AM, delay in minutes

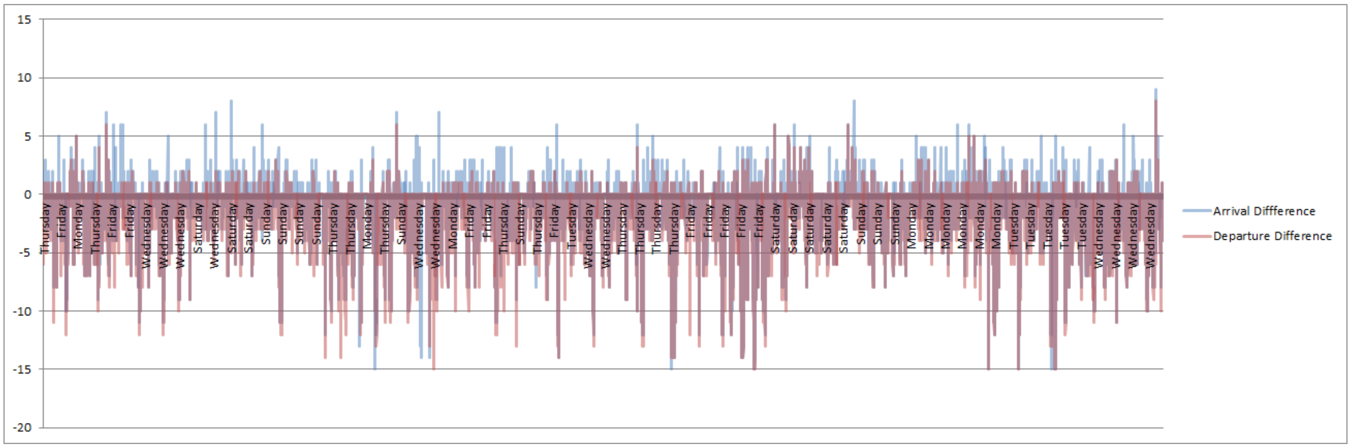


Figure 5-15 Plotting of both early and late arrival 10 AM TO 3 PM, delay in minutes

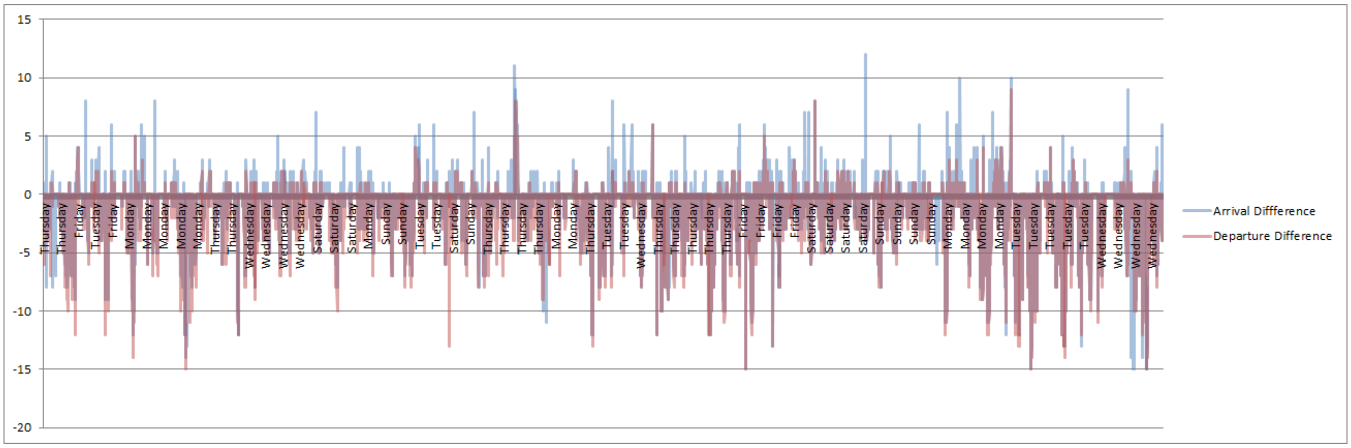


Figure 5-16 Plotting of both early and late arrival 4 PM TO 7 PM, delay in minutes

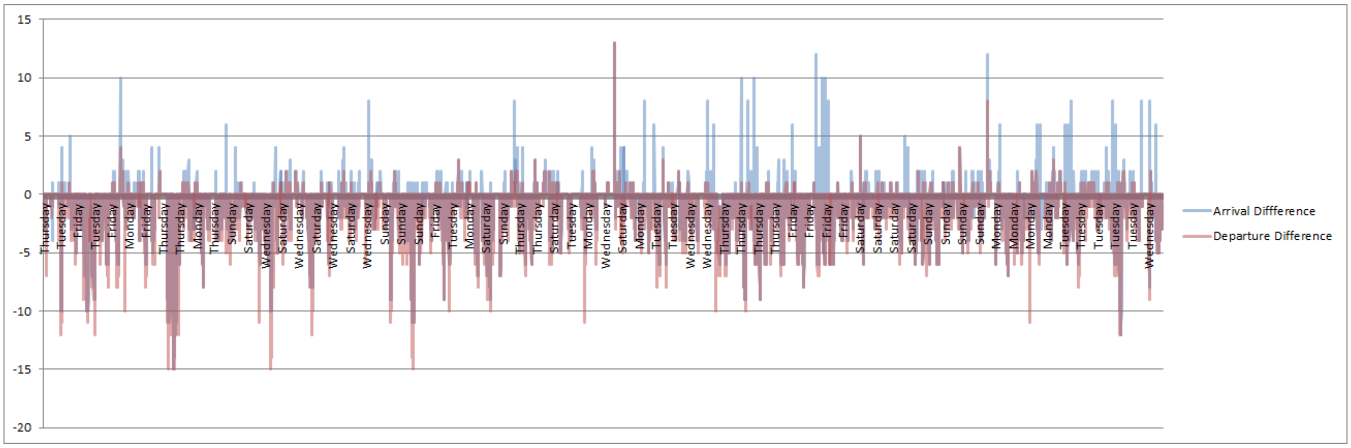
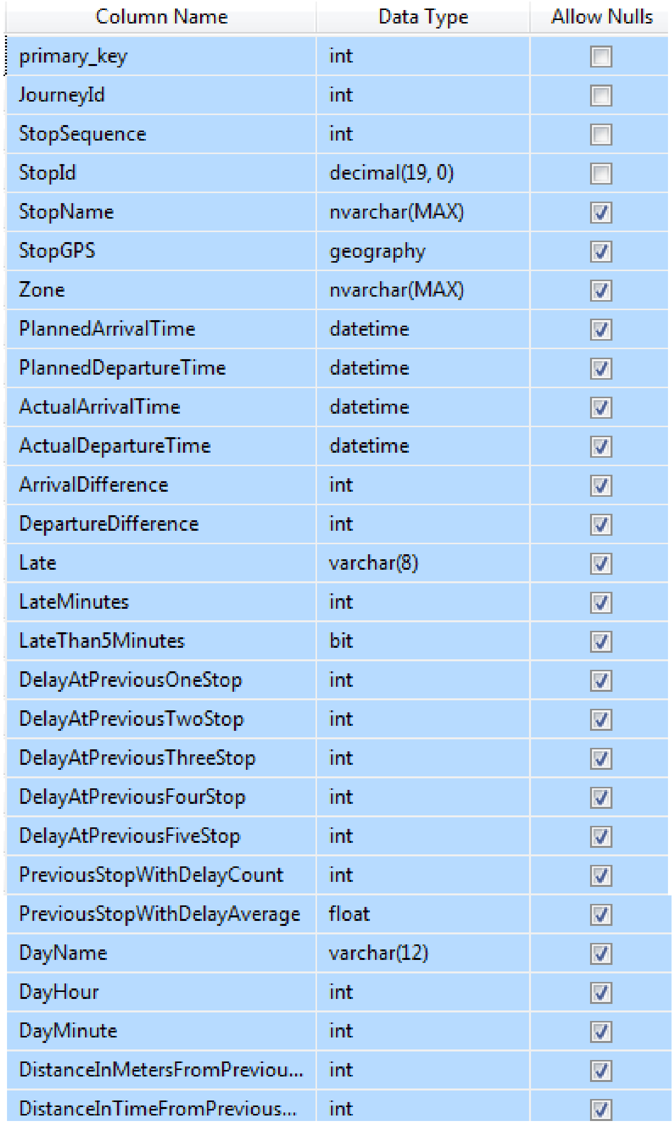


Figure 5-17 Plotting of both early and late arrival 7 PM TO 12 AM, delay in minutes

We can clearly see the difference of behavior for different timeslots. Although we cannot establish a pattern based on this because almost all time slots are reporting early as well as late arrivals with some indications on when it is more frequent than others where it is less frequent but this information is not sufficient to use that as a rule. This indicated that there is need to look into this data deeply so that new patterns can be identified and rules can be established and then these rules can facilitate prediction of arrival and departure times. The findings are:

- Although the rules can be established delay is quantified on a day and hour of the day based on actual arrival time reported. But because the vehicles move in different regions and even cities therefore application of a rule which is made on data in one region may not be true in another region. So we need some extra parameters to consider.
- Also the delay calculated is based on one independent incident. That means stop wise. But the data does not indicate what impact a delay on a stop had on next coming 5 stops for instance. Or the delay is calculated for a region covering more than one stops and then its impact is studies on next stops to come. If this is quantified, then we can establish a rule based on this that what is probability of the delay propagated to the next stops.

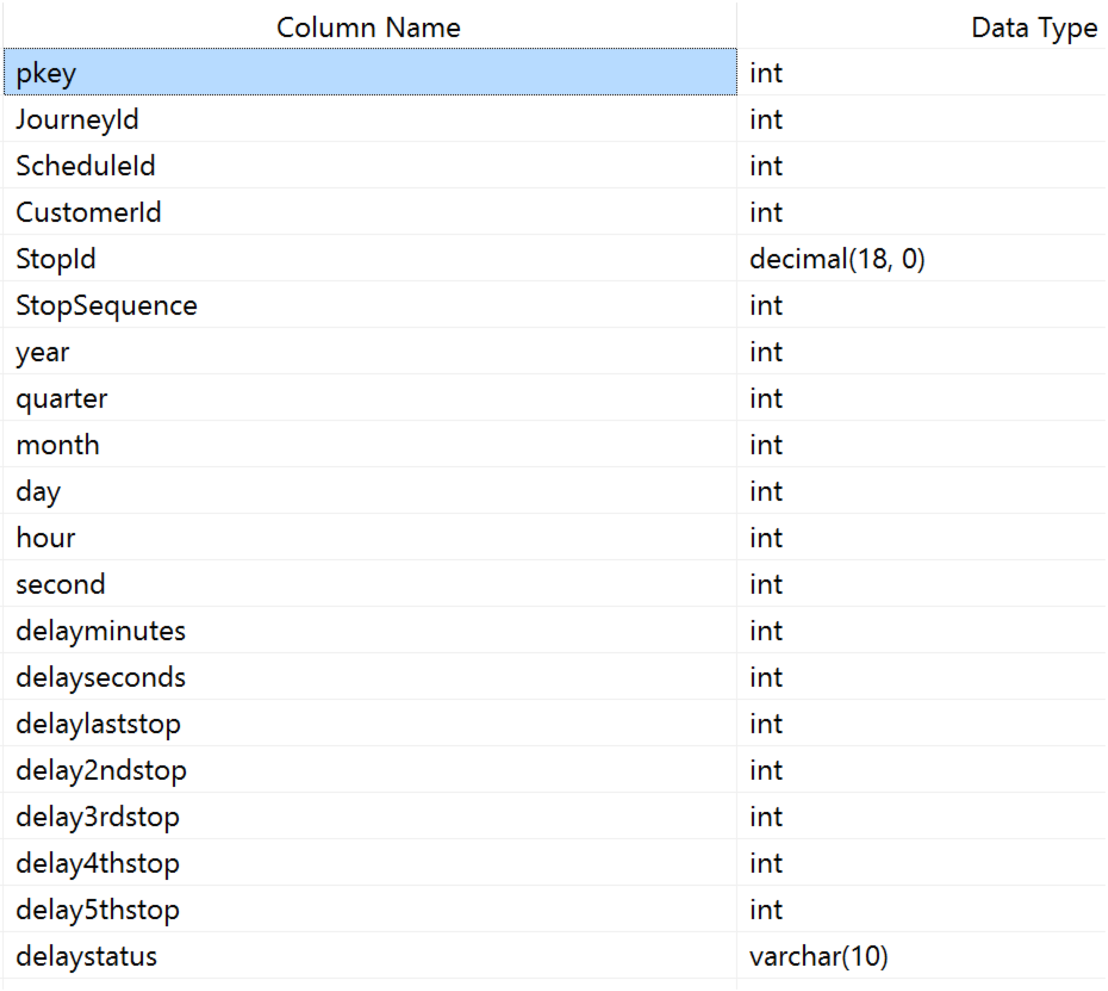
The prediction of the arrival times is a key part of the contribution because it helps the system use this information to address the challenges mentioned in the design chapter that existing systems face today to support a reliable urban transport system. The information that has been used so far in the visualization and analysis is based on the day and time of the arrivals made in different bus trips. Every route has more than one stop and it means that a delay at one stop can contribute to delay at the next stop. Therefore, information about delays on each stop should be evaluated for impact on the next stop in sequence where the bus is expected to arrive as it progresses with the journey. Before machine learning techniques can be applied, the data in raw format was converted to specific format to include all possible dependent and independent variables in the data. The datetime is also split to days, hours and minutes to differentiate the role of these entities together as well as separately. The delay on previous stops were checked and for the sake of this study, five previous stops data was collected to be included in the analysis. In order to identify the relationship between delay at a stop with its distance from previous stops is also important to establish. In order to include all these identified parameters, the structure of the data was changed as shown in the Figure 5.18 below:



Column Name	Data Type	Allow Nulls
primary_key	int	□
JourneyId	int	□
StopSequence	int	□
StopId	decimal(19, 0)	□
StopName	nvarchar(MAX)	☑
StopGPS	geography	☑
Zone	nvarchar(MAX)	☑
PlannedArrivalTime	datetime	☑
PlannedDepartureTime	datetime	☑
ActualArrivalTime	datetime	☑
ActualDepartureTime	datetime	☑
ArrivalDifference	int	☑
DepartureDifference	int	☑
Late	varchar(8)	☑
LateMinutes	int	☑
LateThan5Minutes	bit	☑
DelayAtPreviousOneStop	int	☑
DelayAtPreviousTwoStop	int	☑
DelayAtPreviousThreeStop	int	☑
DelayAtPreviousFourStop	int	☑
DelayAtPreviousFiveStop	int	☑
PreviousStopWithDelayCount	int	☑
PreviousStopWithDelayAverage	float	☑
DayName	varchar(12)	☑
DayHour	int	☑
DayMinute	int	☑
DistanceInMetersFromPreviou...	int	☑
DistanceInTimeFromPrevious...	int	☑

Figure 5-18 The new data structure to store data for prediction

After careful comparison of different mining platforms, presented in the previous chapter, and their suitability to the system being studied, SQL Server Analysis services with support from Integration Services to do pre-processing proved to be the best choice to conduct an in-depth analysis of the data. The tabular representation of the existing dataset is converted to the flat format shown in Figures 5.19 and 5.20 so that SQL Server Analysis Services can apply machine learning algorithms to produce prediction for arrival times to be used as the basis of short-term as well as long-term predictions. There are two prediction elements in this, one is ‘delay minutes’ and other one is ‘delay status’. The ‘delay minutes’ identifies how much delay is expected and ‘delay status’ indicates as yes or no for the delay to happen.



Column Name	Data Type
pkey	int
JourneyId	int
ScheduleId	int
CustomerId	int
StopId	decimal(18, 0)
StopSequence	int
year	int
quarter	int
month	int
day	int
hour	int
second	int
delayminutes	int
delayseconds	int
delaylaststop	int
delay2ndstop	int
delay3rdstop	int
delay4thstop	int
delay5thstop	int
delaystatus	varchar(10)

Figure 5-19: Data structure for mining models

5.3.2.3 Customized Mining Models

Before a machine learning algorithm can start working on a dataset, a mining model is defined based on the dataset so that input and output expectations are clearly defined in the process. The objective of this work includes identification of delay status and then quantification of the delay so, following are the basic mining models generated from this data and then four machine learning techniques Clustering, Regression, Decision Tree and Neural Networks) are run on these models to find the patterns of delay identification.

5.3.2.3.1 Late/Early Mining Model

The purpose of this model, presented in Figure 5.21, was to be able to predict if the bus will be late or not based on the training data we used from the real data. As shown in the Figure 5.19, the field to be predicted is *delaystatus* and this is the first step to identify patterns in the data and know which kind of data points support the result to be late. The training data is used to train two types of clustering techniques called Scalable K-means and Scalable Expectation Maximisation (EM). Non-scalable versions of K-means and EM are not being used because they load all the data in memory and use more storage and computing power with little difference of improvement. Number of clusters (K) is kept 10 for both algorithms to find out arrangement of data points, clustering seed is random to generate random clusters as a starting point and modelling cardinality is set to 10 for improving performance of each algorithm. The input parameters are hour, minute, delayseconds and delaylaststop only. The number of attributes will be increased to include delay further back in time for previous stops to see the impact of going into history of delays for stops. The cluster groups for both K-means and EM are shown in Figure 5.20. The data belongs to one route and contains 218,027 data points covering different trips made on that route in different times of the day. The cluster profiles for K-means and EM are shown in Figure 5.21.

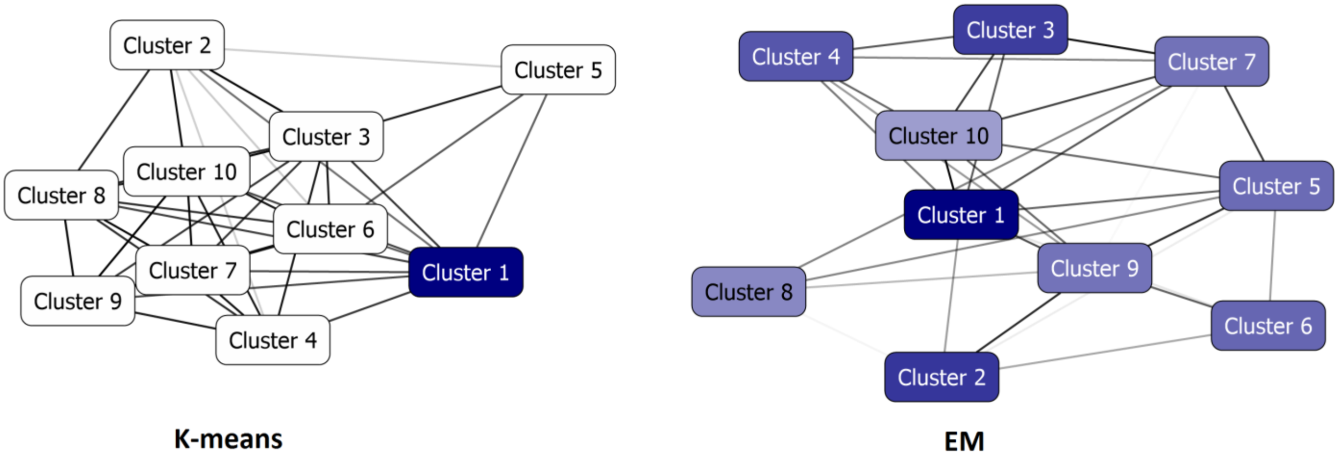
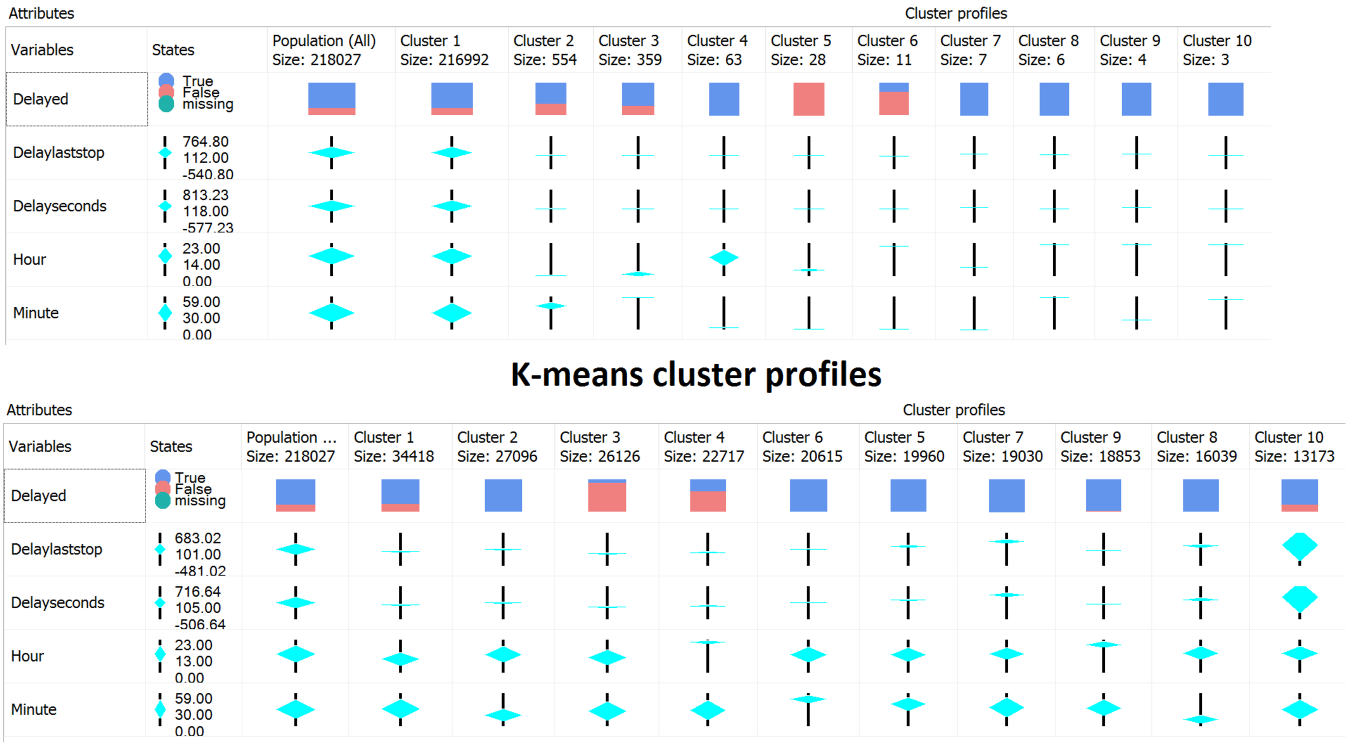


Figure 5-20: Cluster groups produced from K-means and EM

The clusters produced from K-means and EM as shown in the Figure 5.20 indicate that K-means could not identify the pattern because the mean value calculated from attributes does not have variation and it put most of the data points in one cluster because it did not differentiate the difference of values with their relation to hours and minutes. The repetition of different values on the same route on different days yet same hour and minute values makes them redundant and they are considered similar values and put into same cluster. Also, K-means is a hard clustering method and puts one data point strictly in one one cluster. However, EM grouped them better than K-means because it takes one data point at a time and establishes its relationship with the cluster and probability is calculated for the data point and each cluster group it belongs to. The cluster profiles presented in Figure 5.21 reflect this behaviour of the algorithms in grouping of data points and their role in decision of the prediction attribute *delaystatus*. K-means cluster profiles clearly show that only hour attribute differentiated compared to other attributes but all other attributes favoured in making one main cluster because the algorithm considered each of them as a separate attribute and did not identify relationship between hour and minute so that the group represents a different slot for grouping. The hour and minute attributes are more evenly distributed in EM clusters and support delay at current and last stop as well in a different cluster. The score of EM is 0.97 with 91.73% prediction probability compared to K-means with score 0.77 and 77.36% prediction probability as shown in Figure 5.22 that also presents model line for both algorithms drawn with an ideal line. EM clearly performs better. The score indicates fitting of the model and therefore higher score means better performance as indicated by EM over K-means.



Attributes

Cluster profiles

Variables	States	Population (All) Size: 218027	Cluster 1 Size: 216992	Cluster 2 Size: 554	Cluster 3 Size: 359	Cluster 4 Size: 63	Cluster 5 Size: 28	Cluster 6 Size: 11	Cluster 7 Size: 7	Cluster 8 Size: 6	Cluster 9 Size: 4	Cluster 10 Size: 3
Delayed	● True ● False ● missing											
Delaylaststop	764.80 112.00 -540.80											
Delayseconds	813.23 118.00 -577.23											
Hour	23.00 14.00 0.00											
Minute	59.00 30.00 0.00											

K-means cluster profiles

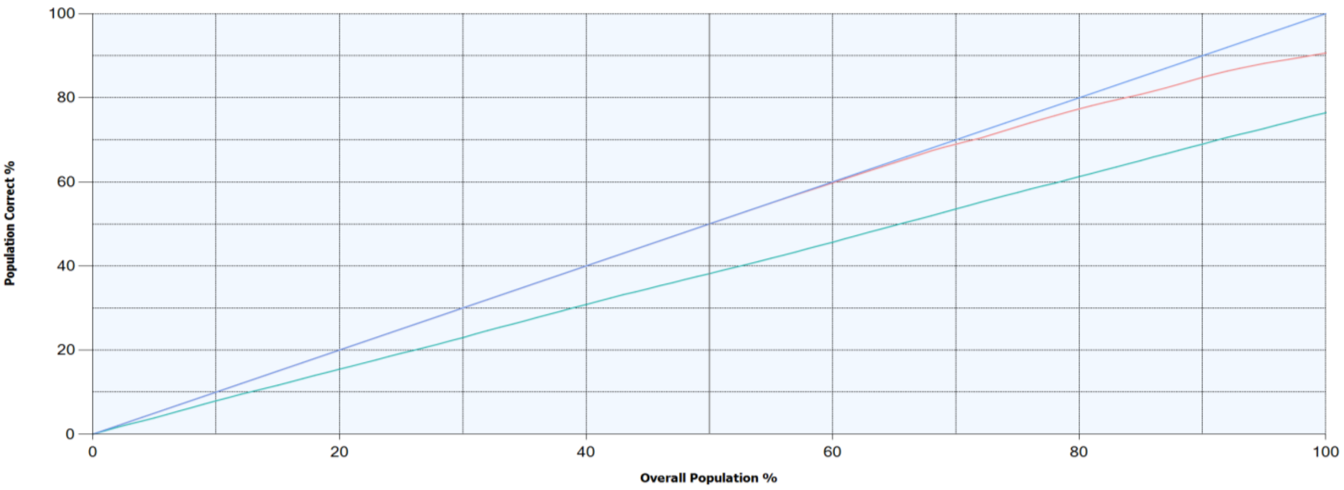
Attributes

Cluster profiles

Variables	States	Population ... Size: 218027	Cluster 1 Size: 34418	Cluster 2 Size: 27096	Cluster 3 Size: 26126	Cluster 4 Size: 22717	Cluster 6 Size: 20615	Cluster 5 Size: 19960	Cluster 7 Size: 19030	Cluster 9 Size: 18853	Cluster 8 Size: 16039	Cluster 10 Size: 13173
Delayed	● True ● False ● missing											
Delaylaststop	683.02 101.00 -481.02											
Delayseconds	716.64 105.00 -506.64											
Hour	23.00 13.00 0.00											
Minute	59.00 30.00 0.00											

EM cluster profiles

Figure 5-21: cluster profiles for K-means and EM



Legends

Series, Model	Score	Population correct	Predict probability
🟥 Clustering Scalable EM	0.97	49.98%	91.73%
🟩 Clustering Scalable KMeans	0.77	38.16%	77.36%
🟦 Ideal Model		50.00%	

Score: Best fitting for prediction
Population Correct: Percentage of data points predicted acurately
Predict probability: Probability for accurate prediction

Figure 5-22: Prediction line, score and prediction probability for K-means and EM

We add more input attributes to the model to identify the impact of delay2ndstop, delay3rdstop, distancetolaststop (derived from GPS location of adjacent stops) and day of the week. The cluster groups, cluster profiles and prediction line (lift chart) are shown respectively in Figures 5.23, 5.24 and 5.25 respectively.

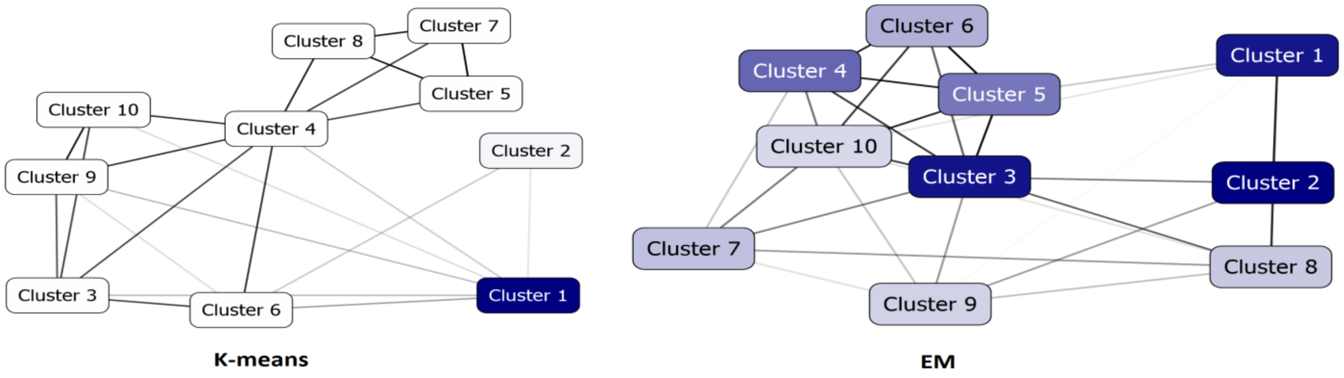


Figure 5-23: Cluster groups for K-means and EM using more input attributes

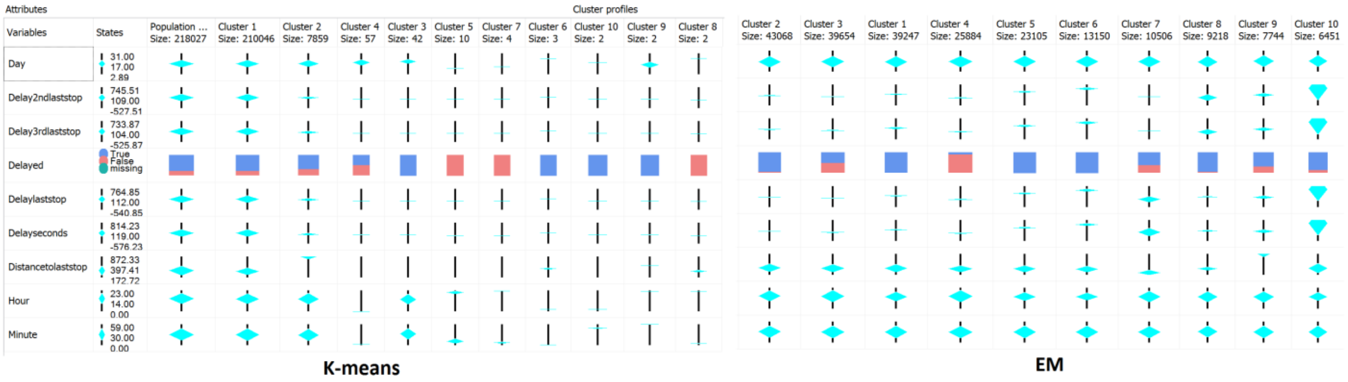
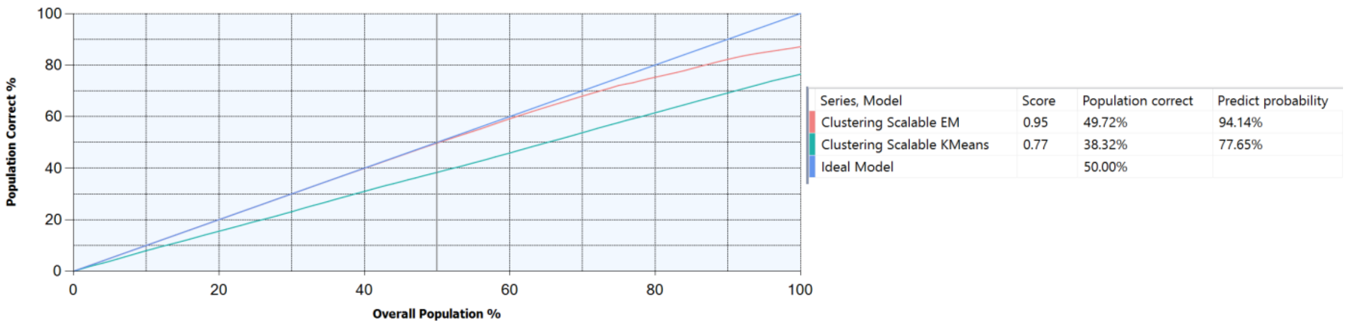


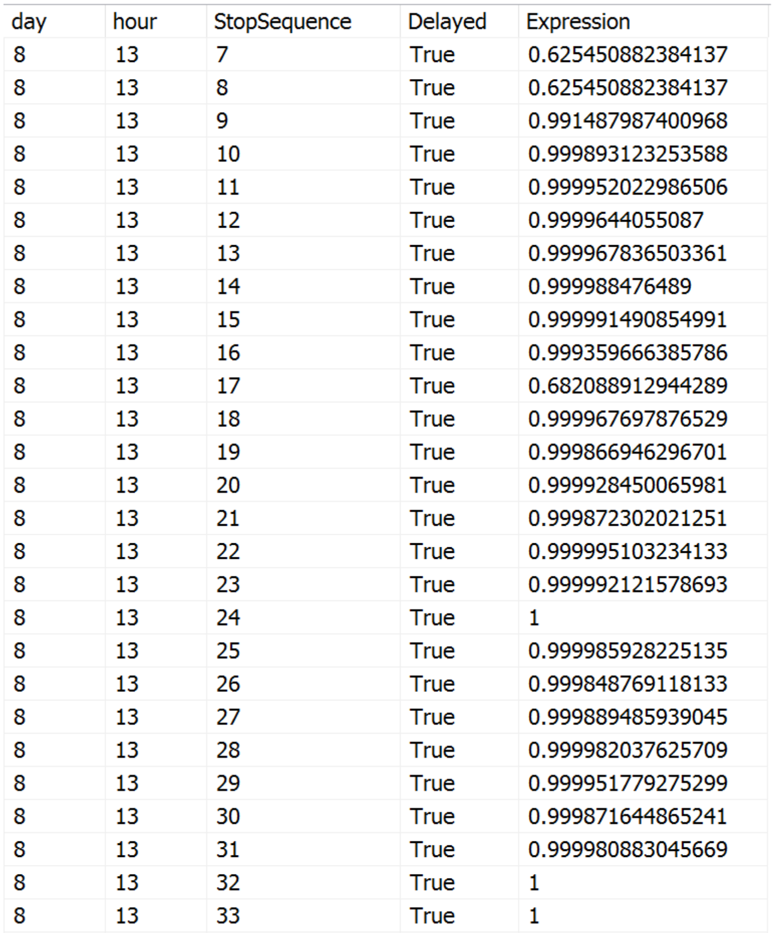
Figure 5-24: Cluster profiles for K-means and EM using more input attributes



Series, Model	Score	Population correct	Predict probability
Clustering Scalable EM	0.95	49.72%	94.14%
Clustering Scalable KMeans	0.77	38.32%	77.65%
Ideal Model		50.00%	

Figure 5-25: Prediction line, score and prediction probability for K-means and EM (more input attributes)

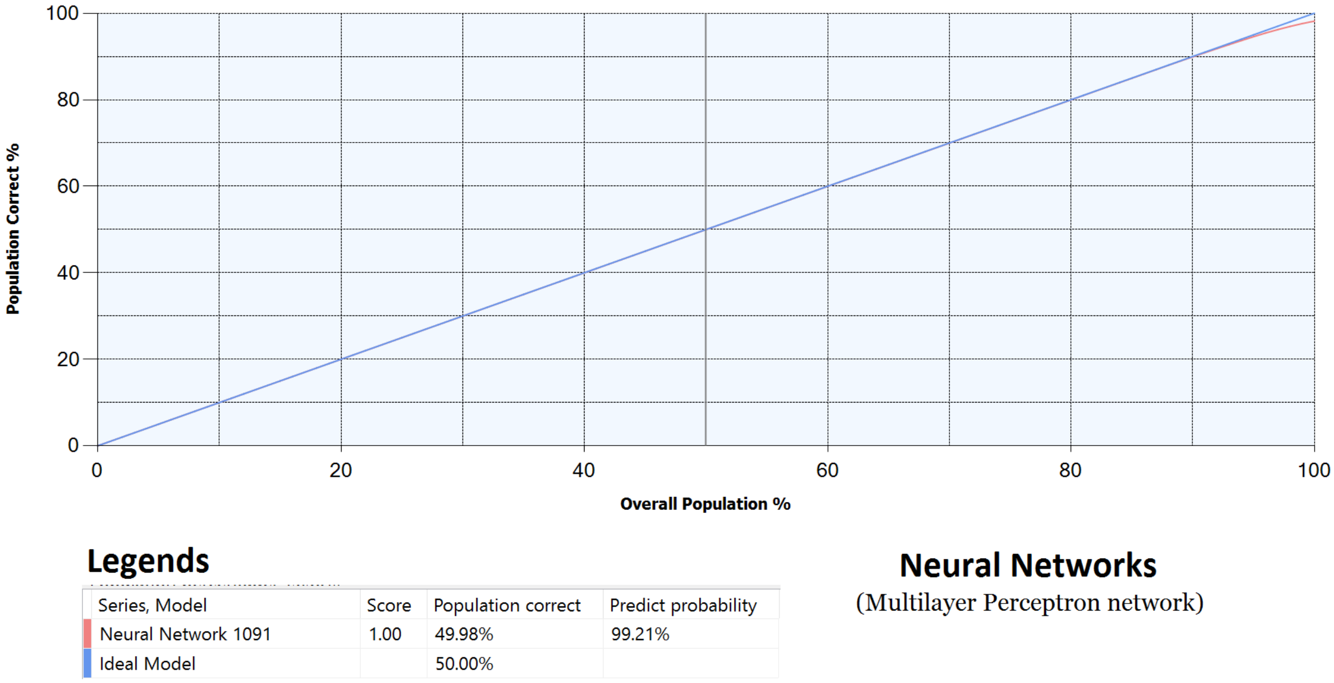
As shown in Figures 5.23, 5.24 and 5.25 – addition of more attributes has not helped in improvement of the results and EM has still performed better than K-means because of its data point association with multiple clusters to get insight on relationship between different attributes of each data point. A snapshot of the application of the trained EM model on data is shown in Table 5.1 that indicates the prediction probability of more than 99% for most of the cases.



day	hour	StopSequence	Delayed	Expression
8	13	7	True	0.625450882384137
8	13	8	True	0.625450882384137
8	13	9	True	0.991487987400968
8	13	10	True	0.999893123253588
8	13	11	True	0.999952022986506
8	13	12	True	0.9999644055087
8	13	13	True	0.999967836503361
8	13	14	True	0.999988476489
8	13	15	True	0.999991490854991
8	13	16	True	0.999359666385786
8	13	17	True	0.682088912944289
8	13	18	True	0.999967697876529
8	13	19	True	0.999866946296701
8	13	20	True	0.999928450065981
8	13	21	True	0.999872302021251
8	13	22	True	0.999995103234133
8	13	23	True	0.999992121578693
8	13	24	True	1
8	13	25	True	0.999985928225135
8	13	26	True	0.999848769118133
8	13	27	True	0.999889485939045
8	13	28	True	0.999982037625709
8	13	29	True	0.999951779275299
8	13	30	True	0.999871644865241
8	13	31	True	0.999980883045669
8	13	32	True	1
8	13	33	True	1

Table 5-1: Fitting EM model on prediction of delaystatus

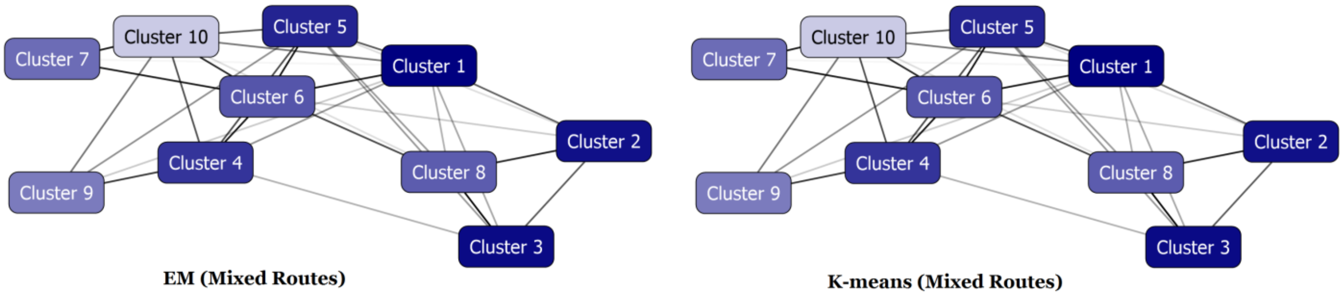
Application of Neural Networks (NN) algorithm to the same dataset produced better results than EM using Multilayer Perceptron network, also called a Back-Propagated Delta Rule network using 4 input variables for the input layer, two hidden layers and one output layer to predict for *delaystatus*. These input variables are same as tried in first experiment of K-means and EM elaborated earlier. It is run with node ratio of 4, 30% holdout percentage and zero seed so that the algorithm uses same approach to generate content for the model during the processing. The prediction line for Neural Network is shown in Figure 5.26.



Series, Model	Score	Population correct	Predict probability
🟥 Neural Network 1091	1.00	49.98%	99.21%
🟦 Ideal Model		50.00%	

Figure 5-26:Prediction line (lift chart) for Neural Networks

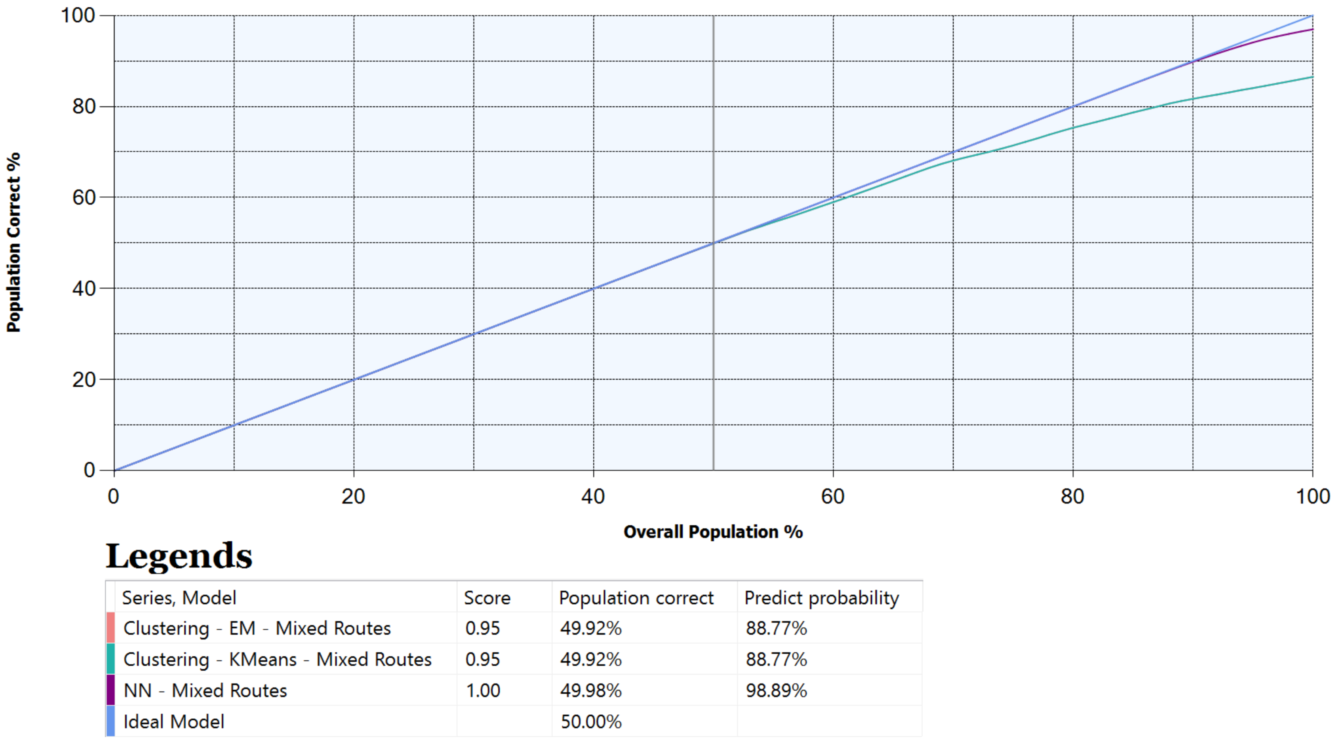
It has a score of 1 and 99.21% prediction probability to accurately predict if the bus will be late or not on stop having information about hour, minute, history of delays reported for the stop for which prediction is being made and the delay at last stop of current journey. It was interesting to observe that K-means did not perform at all when the data was used for multiple journeys of the same route. It makes sense because the means calculation for input attributes does not vary much because the pattern of delays on the same route supports one category of means and that is why most of the data points were categorized in the same cluster. Next, we mixed the dataset so that it contains journeys from different routes but the number and size of the sample was kept the same. There are 227 routes with each route having at least one data point and the maximum number of data points are 7866 in one of the routes. This gives lot of variety to the data and it has a minimum of 227 patterns but some of the patterns would have similarities because many routes go to the same destination so they share some common stops and therefore, the impact of these stops can reduce the unique patterns that may exist in the dataset. We applied all three algorithms (EM, K-means, NN) on the dataset with four input attributes and delaystatus as the output attribute. The performance of K-means improved tremendously and its score is same as that of EM, which validates the prior assumption we made that lack of variety in the patterns was the reason for K-means for not performing. NN still performed better than both EM and K-means. The cluster groups for EM and K-means, and prediction line for all three algorithms (EM, K-means, NN) with their score and prediction probabilities are shown in Figures 5.27 and 5.28 respectively.



EM (Mixed Routes)

K-means (Mixed Routes)

Figure 5-27: Cluster groups for Mixed Routes (EM & K-means)



Legends

	Series, Model	Score	Population correct	Predict probability
🟥	Clustering - EM - Mixed Routes	0.95	49.92%	88.77%
🟩	Clustering - KMeans - Mixed Routes	0.95	49.92%	88.77%
🟪	NN - Mixed Routes	1.00	49.98%	98.89%
🟦	Ideal Model		50.00%	

Figure 5-28: Prediction line for EM, K-means and NN (Mixed Routes)

These results support dynamic selection of a data mining algorithm depending on the scope of the prediction being performed. If the purpose is to know about the bus to be late or not and the number of routes is limited, EM can be used in comparison to K-means whereas both EM and K-means can be used when there is diverse type and number of routes. NN performs better than both EM and K-means for limited as well as extensive list of routes. The results from NN were verified with cross-validation so that results for mining model are validated through multiple cross-sections and testing of structure with associated mining models. Table 5.2 shows cross-validation results for NN.

5.3.2.3.2 Measuring predicted delay

After measuring the chances of a bus being late or not, the next step is to identify how much delay is expected. Linear Regression has been applied to the data for route 1091, which is the same dataset for which EM, K-means and NN have been applied to know status of the delay earlier. Linear Regression applied is special version of a decision tree algorithm that has been optimised for modelling pairs of continuous attributes. This specialisation is controlled through parameters for restricting the growth of the tree and keep all the data in one node and feature selection is done by using Interestingness as a method of analysis. Interestingness identifies how interesting and therefore important a feature can be in prediction of delay. It is done based on entropy so that the attributes with random distribution have higher entropy and therefore having not enough information for a concrete decision. That means attributes with high entropy will be less interesting. The reason for selection of Interestingness as a feature selection method is because all other methods supported by decision trees algorithm apply to discrete variables only whereas variables in the dataset we are using are continuous. There is no enforced regressor in these models and the algorithms choose regressor based on the data automatically. The prediction line for route 1091 with inout attributes of hour, minute and delay at last stop is shown in Figure 5.29.

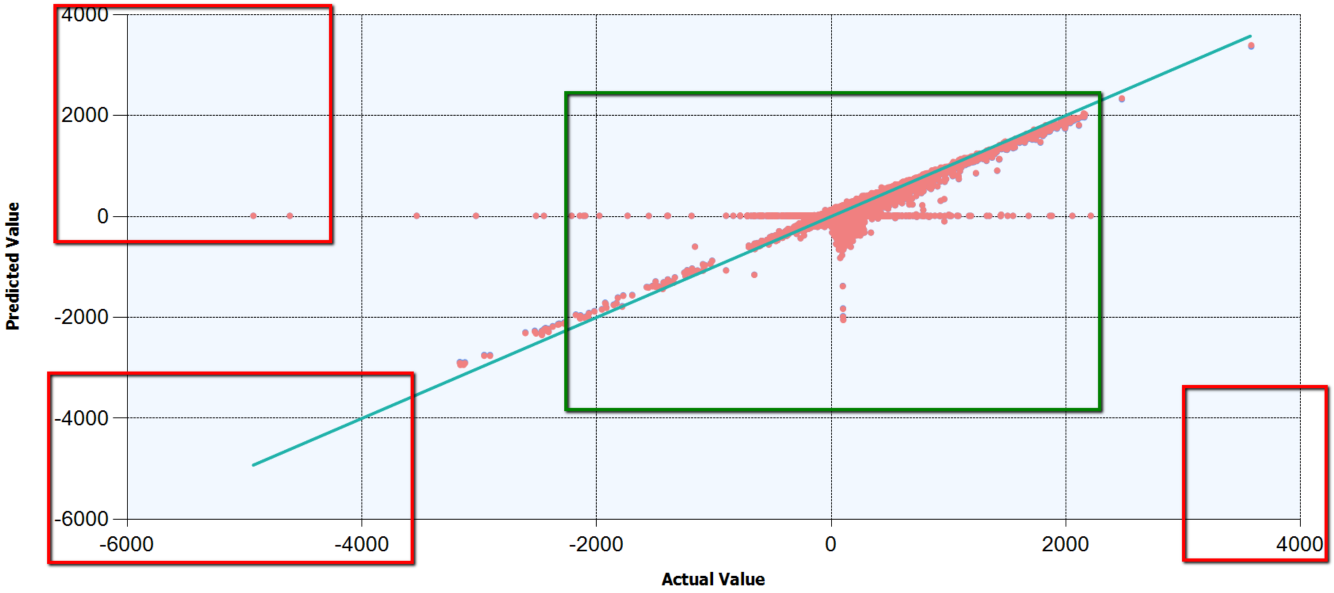
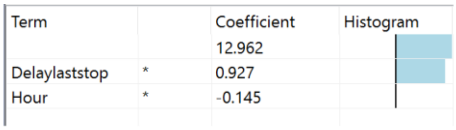


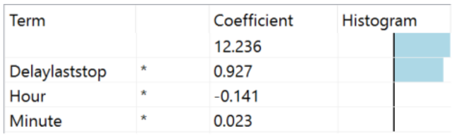
Figure 5-29: Prediction line for Linear Regression (Route 1091)

As marked by red rectangles in Figure 5.29, there are clearly some outliers in the data reflecting a delay in the range from 33 to 100 minutes (shown in seconds in the Figure 5.29). When cross-checked in the dataset, there are two journeys where the reported delay is abnormal and further investigation shows that the time reported was wrong for those journeys and regression has pointed them out as outside the normal range of +/- ten minutes. As presented in Figure 5.30 that shows new prediction line for route 1091 after forcing minutes to be included in applying regression. When applied with default settings, the regression algorithm ignored minutes input, but application of enforced inclusion of minutes improved the score and fitting of the algorithm for route 1091. The predicted data points are more aligned with the ideal prediction line as indicated by the area encircled green.



Term		Coefficient	Histogram
		12.962	
Delaylaststop	*	0.927	
Hour	*	-0.145	

Without minutes regression coefficients



Term		Coefficient	Histogram
		12.236	
Delaylaststop	*	0.927	
Hour	*	-0.141	
Minute	*	0.023	

Minutes inclusion regression coefficients

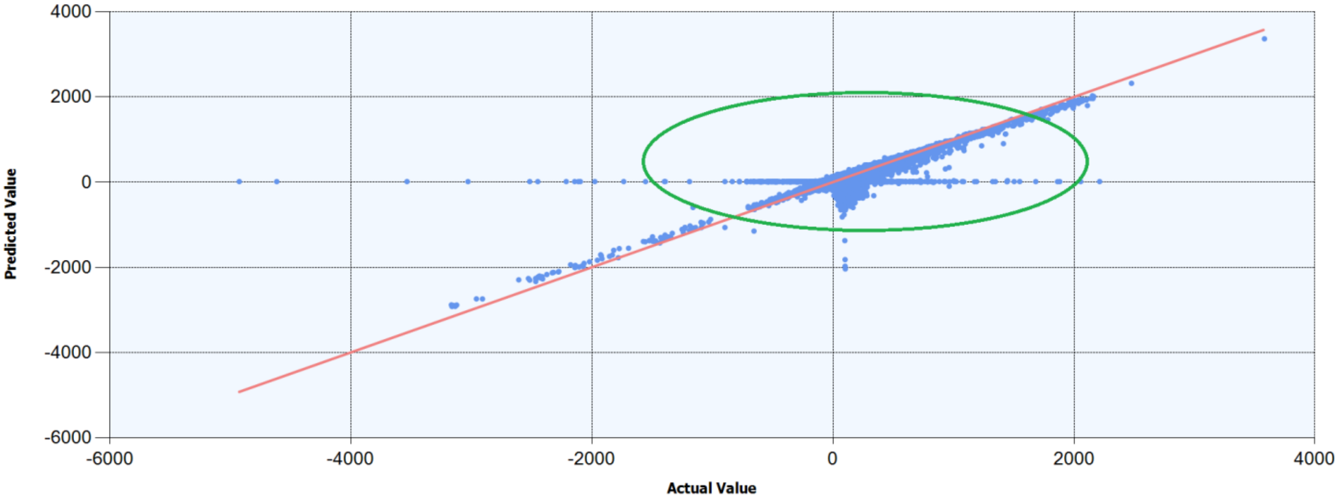


Figure 5-30: Prediction line of Linear Regression (Route 1091) with minutes parameter included

The application of Linear Regression to the data of 227 routes is shown in Figure 5.31, which maintains the pattern of prediction as tried with route 1091 but it has better performance and more data points are predicted close to the center point (delay of +/- 10 minutes) because the variety of data helps in ignoring the outliers when they happen very rare and that is why have minimal impact on the prediction. The impact of outliers can be further well defined by introducing new attributes derived from values of outlier data points and using them as factors, which is not in the scope of this work. The regression will be further tested with inclusion of more attributes like delay to more stops in history and the distance between previous stop and the stop for which the delay is being predicted.

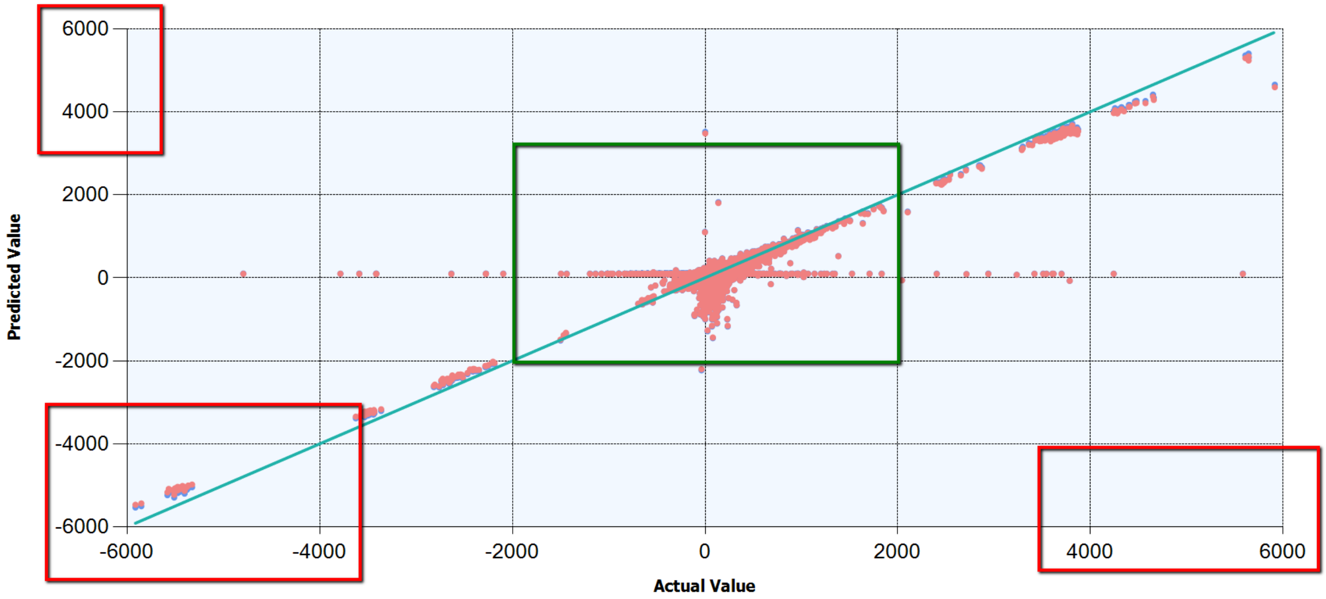
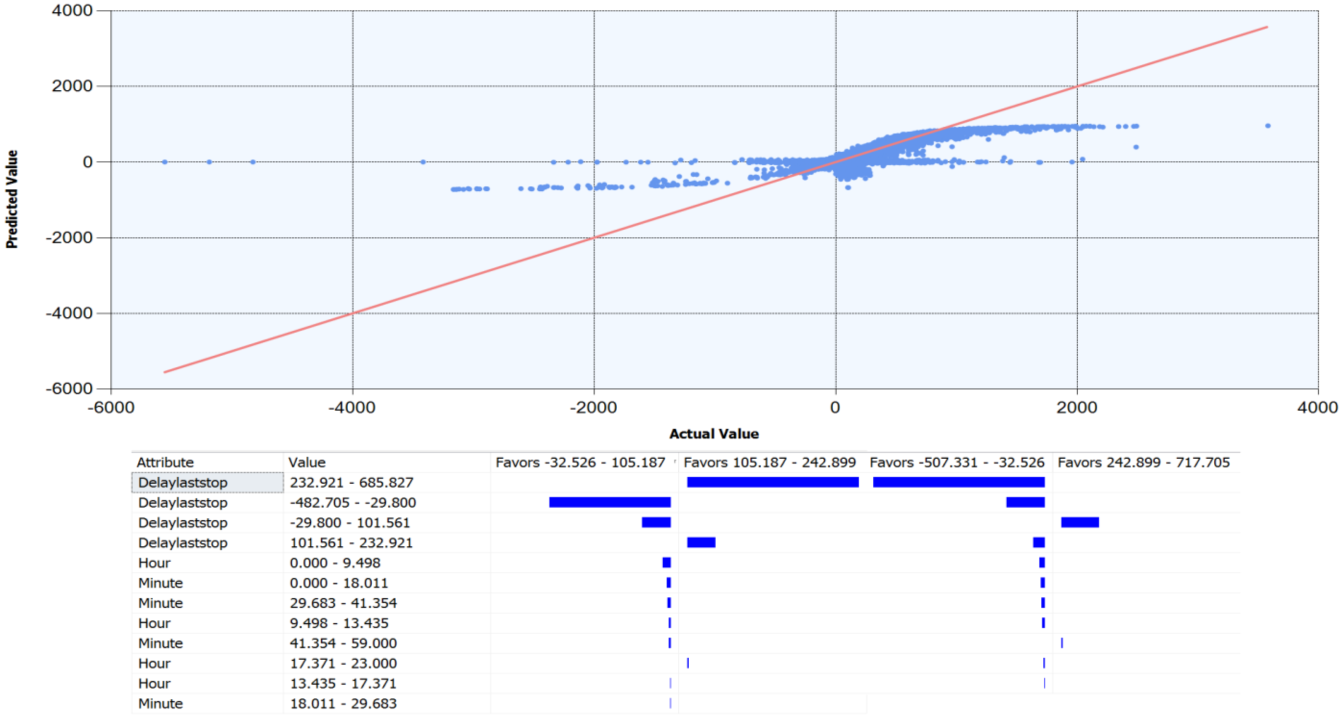


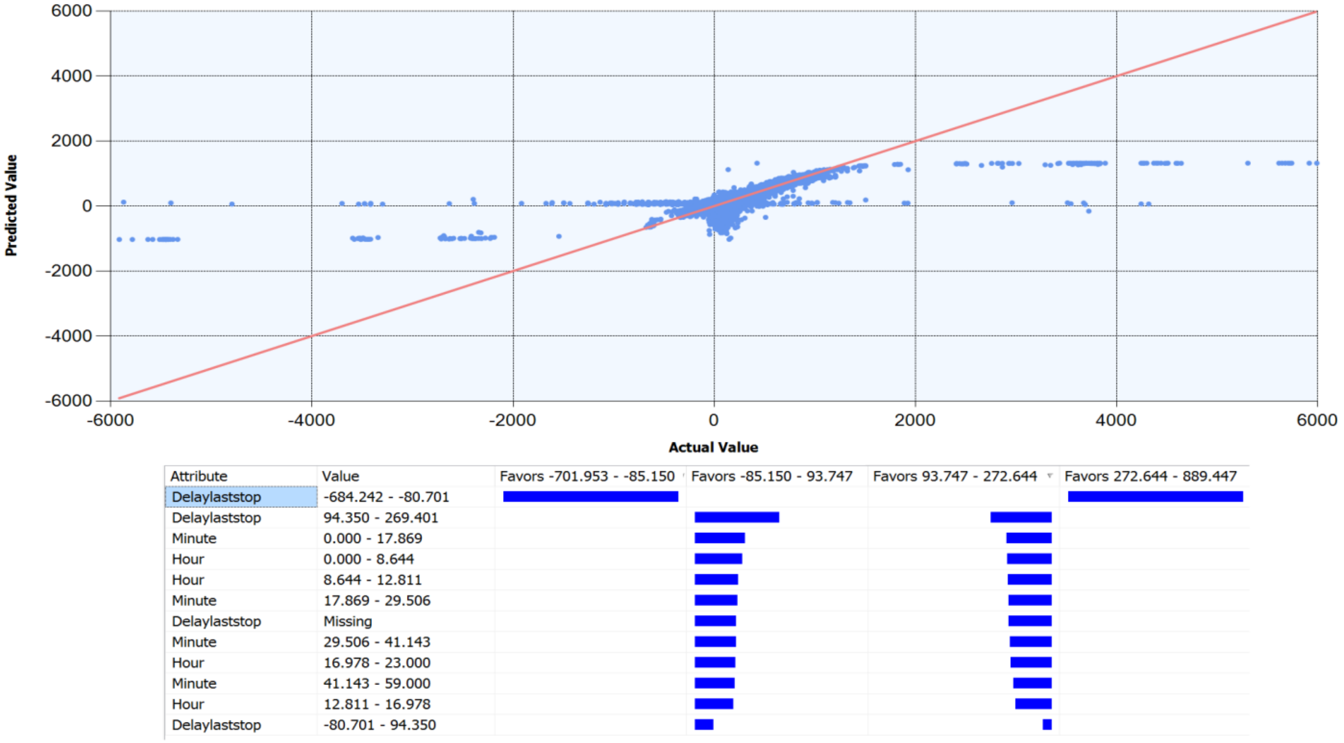
Figure 5-31: Prediction line for Linear Regression (Mixed Routes)

Considering better performance of NN in initial prediction of *delaystatus*, it was applied to prediction of amount of delay as well on the same data use by Linear Regression. NN made the predicted data points more close to the ideal line and the prediction probability is better than Linear Regression. NN also indicated support vector of different value ranges of the attributes to find their support for a specific prediction. As presented in Figures 5.32 and 5.33, NN used Delayatlaststop as a major input neuron that impacted the predicted value when applied on data for the same route but the impact of hour and minutes increased when we applied it to data for 227 routes because the relevance of hour and minute increased with journeys processed in more variety of times during the day. To be able to identify how far the projection of delay at a stop reflects in its arrivals in future stops, it is important to validate the impact of delays from previous delays more than just the last stop delay. Also, the dataset contains data points for different months and information about the time spent by buses on different stops along with distance to previous stops as a matter of time. Inclusion of these attributes as input to the algorithms will expose further data relationships and their support for each other.



Attribute	Value	Favors -32.526 - 105.187	Favors 105.187 - 242.899	Favors -507.331 - -32.526	Favors 242.899 - 717.705
Delaylaststop	232.921 - 685.827		■	■	
Delaylaststop	-482.705 - -29.800	■		■	
Delaylaststop	-29.800 - 101.561	■			■
Delaylaststop	101.561 - 232.921		■	■	
Hour	0.000 - 9.498	■		■	
Minute	0.000 - 18.011	■		■	
Minute	29.683 - 41.354	■		■	
Hour	9.498 - 13.435	■		■	
Minute	41.354 - 59.000	■			■
Hour	17.371 - 23.000		■	■	
Hour	13.435 - 17.371	■		■	
Minute	18.011 - 29.683	■			

Figure 5-32: Prediction line of NN (Route 1091) with attributes support vector



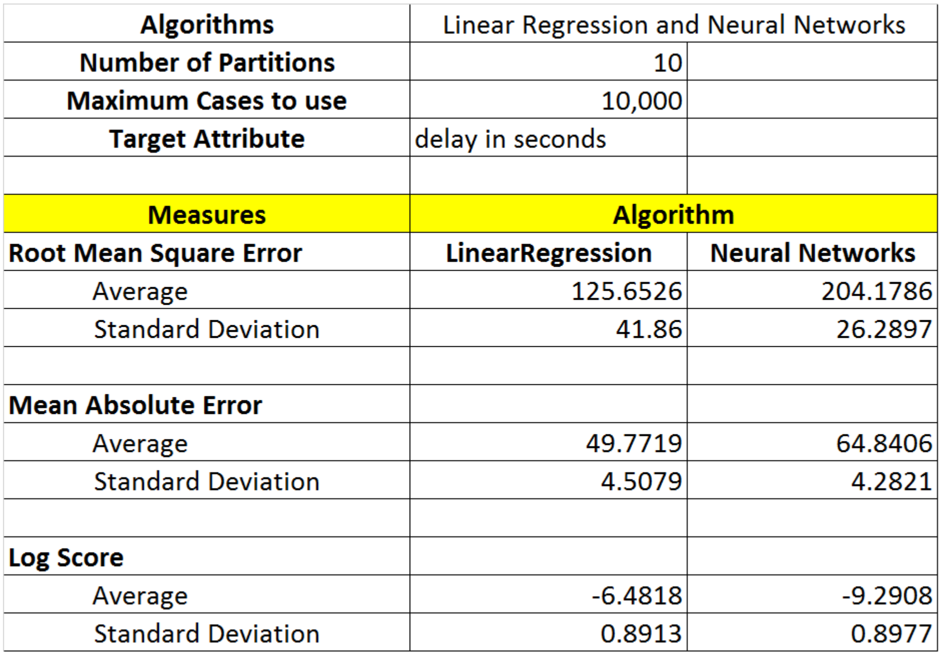
Attribute	Value	Favors -701.953 - -85.150	Favors -85.150 - 93.747	Favors 93.747 - 272.644	Favors 272.644 - 889.447
Delaylaststop	-684.242 - -80.701	■			■
Delaylaststop	94.350 - 269.401		■	■	
Minute	0.000 - 17.869		■	■	
Hour	0.000 - 8.644		■	■	
Hour	8.644 - 12.811		■	■	
Minute	17.869 - 29.506		■	■	
Delaylaststop	Missing		■	■	
Minute	29.506 - 41.143		■	■	
Hour	16.978 - 23.000		■	■	
Minute	41.143 - 59.000		■	■	
Hour	12.811 - 16.978		■	■	
Delaylaststop	-80.701 - 94.350		■	■	

Figure 5-33: Prediction line of NN (Mixed routes) with attributes support vector

All these algorithms have one thing in common, they all support impact of delay at last stop and the history of delay at last stop as one of the major contributing factors that can be segregated for specific timeslots and patterns with help of extra attributes of time and space. Regression and NN have performed well and primarily regression will be applied on the whole dataset in Chapter 6.

5.3.3 Cross Validation of Models (Regression & NN)

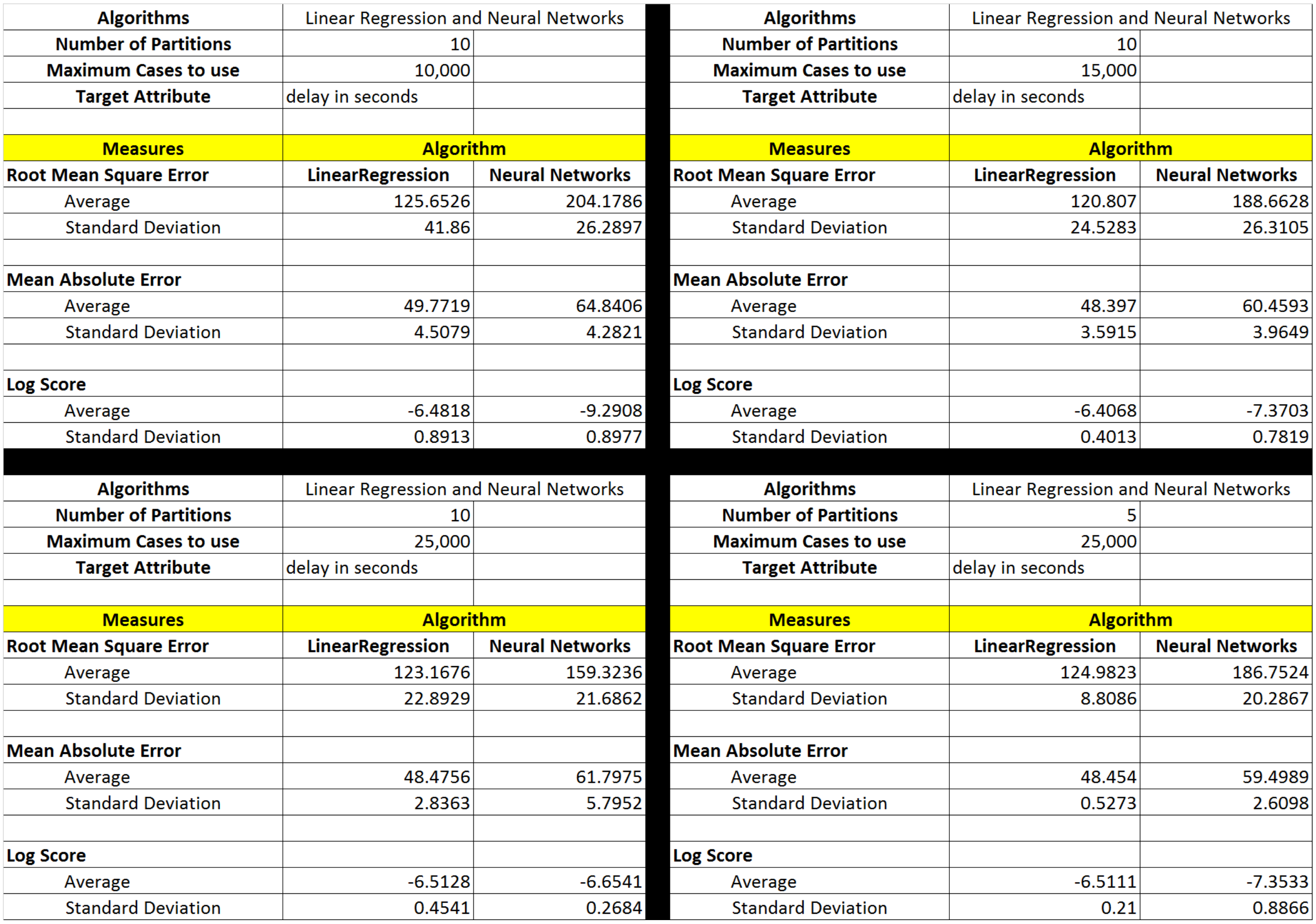
The models generated by regression and NN are put through cross-validation where different partitions were created from the training set to validate the model and measure it using *Root Mean Square Error (RMSE), Mean Absolute Error (MAE)* and *Log Score (LS)*. First cross-validation is done with 10 partitions, 10,000 cases and target attribute of delay in seconds as shown in Table 5.2.



Algorithms	Linear Regression and Neural Networks	
Number of Partitions	10	
Maximum Cases to use	10,000	
Target Attribute	delay in seconds	
Measures	**Algorithm**	
Root Mean Square Error	**LinearRegression**	**Neural Networks**
Average	125.6526	204.1786
Standard Deviation	41.86	26.2897
Mean Absolute Error		
Average	49.7719	64.8406
Standard Deviation	4.5079	4.2821
Log Score		
Average	-6.4818	-9.2908
Standard Deviation	0.8913	0.8977

Table 5-2: Cross-Validation of Regression and NN with 10 partitions

As shown in Table 5.2, the regression has an average RMSE of 125.65 seconds with confidence of 41.86 seconds compared to neural networks that has the average 204.18 seconds with a confidence level of 26 seconds, which is better than regression but if we add/subtract the confidence value to average value, regression prediction line gives better confidence. The combined value of average with deviation gives regression achievement better than neural networks by 64 seconds. MAE has better confidence for both algorithms and the average is also less compared to RMSE. Log Score gives the best estimation for both algorithms with best average and confidence value compared with both RMSE and MAE. Comparing all three estimations support Linear Regression to be better performing than NN. Different partition sizes and number were tried as shown in Table 5.3 where increase in number of cases had less positive impact than number of partitions on the confidence in all measures.



Algorithms	Linear Regression and Neural Networks	
Number of Partitions	10	
Maximum Cases to use	10,000	
Target Attribute	delay in seconds	
Measures	**Algorithm**	
Root Mean Square Error	**LinearRegression**	**Neural Networks**
Average	125.6526	204.1786
Standard Deviation	41.86	26.2897
Mean Absolute Error		
Average	49.7719	64.8406
Standard Deviation	4.5079	4.2821
Log Score		
Average	-6.4818	-9.2908
Standard Deviation	0.8913	0.8977

Algorithms	Linear Regression and Neural Networks	
Number of Partitions	10	
Maximum Cases to use	15,000	
Target Attribute	delay in seconds	
Measures	**Algorithm**	
Root Mean Square Error	**LinearRegression**	**Neural Networks**
Average	120.807	188.6628
Standard Deviation	24.5283	26.3105
Mean Absolute Error		
Average	48.397	60.4593
Standard Deviation	3.5915	3.9649
Log Score		
Average	-6.4068	-7.3703
Standard Deviation	0.4013	0.7819

Algorithms	Linear Regression and Neural Networks	
Number of Partitions	10	
Maximum Cases to use	25,000	
Target Attribute	delay in seconds	
Measures	**Algorithm**	
Root Mean Square Error	**LinearRegression**	**Neural Networks**
Average	123.1676	159.3236
Standard Deviation	22.8929	21.6862
Mean Absolute Error		
Average	48.4756	61.7975
Standard Deviation	2.8363	5.7952
Log Score		
Average	-6.5128	-6.6541
Standard Deviation	0.4541	0.2684

Algorithms	Linear Regression and Neural Networks	
Number of Partitions	5	
Maximum Cases to use	25,000	
Target Attribute	delay in seconds	
Measures	**Algorithm**	
Root Mean Square Error	**LinearRegression**	**Neural Networks**
Average	124.9823	186.7524
Standard Deviation	8.8086	20.2867
Mean Absolute Error		
Average	48.454	59.4989
Standard Deviation	0.5273	2.6098
Log Score		
Average	-6.5111	-7.3533
Standard Deviation	0.21	0.8866

Table 5-3: Cross-Validation of Regression and NN with 10 partitions (Variation of Cases and Partitions)

Linear Regression performs better than NN and its confidence increases with increasing the number of input cases and reducing the number of partitions.

5.4 Web Services Implementation

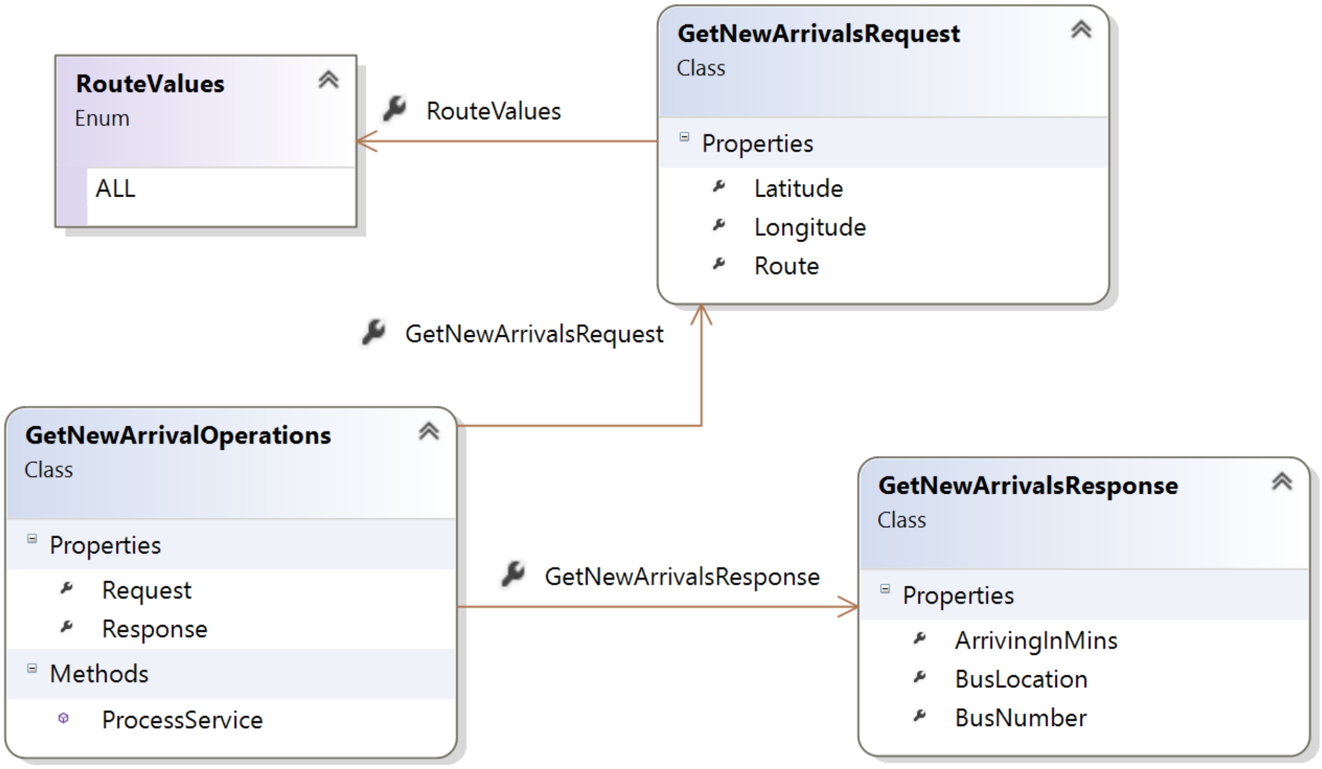


Figure 5-34 a model for web services implementation

The web services implementation is very simple and straightforward, and a model for that implementation is presented in the Figure 5.34. The idea is to build asynchronous web services for accepting the data acquisition requests so that the reporting devices and services don't have to wait for any processing of the data. The requests for data, however, are not asynchronous and wait for the data to be returned. Each web service is implemented as a pair of request/response architecture that supports JSON format for request and response. Each request is assigned to a specific request handler so that the relevant transformation can be used. Once the request is transformed into a hierarchical structure, the information is handed over to response handler. The response handler connects to the relevant system component for fetching the required data. When the data is handed over to the response handler, it uses transformation components to convert the format back to JSON structure and send back to the requesting device or service.

5.4.1 Web Services Evaluation

The web services are an important part of the proposed system to provide the performance and scalability required for a real-time data processing. They have been implemented with asynchronous architecture so that a support for higher number of users with graceful response time can be maintained. The evaluation of the web services component was designed to test the performance, availability, reliability, correctness and scalability of the web services. Different client platforms were tried to test the web services and eventually Apache JMeter was chosen as it provides an easy to use interface along with excellent representation of the information. A throughput graph for 10,000 concurrent calls looks like the Table 5.4. It shows data for four different measures as Average, Media, Deviation and total Throughput. We can assess that the response time for all the calls out of 10,000 needs less than 100 milliseconds to respond back and this data is not a static data, it is dynamic data that changes and needs to be read from caching based database. It simulated 500 unique users sending calls every few milliseconds to test the throughput and availability of the services. The measures for response time are presented in the table below and the charts showing the trend of increase in response time and its impact on throughput are presented in Figure 5.35 to 5.38.

Unique Users	Iterations	Throughput	Average (milliseconds)	Median (milliseconds)	Deviation (milliseconds)	Total calls made
500	20 each	49,147 calls/min	69	54	109	10,000
500	50 each	70,501 calls/min	137	84	357	25,000
1000	25 each	46,143 calls/min	383	105	1242	25,000
500	No limit for 3 mins	73,715 calls/min	395	108	1483	238,345
25,000	5	29,400 calls/mins	895	523	2243	125,000

Table 5-4: web service performance measures (Response time & throughput)

The results indicate an encouraging trend that the number of calls that can be processed per minute by the web services components are representation of the scalability of the system. The worst case tried was with 25,000 unique users and the system was able to handle 29,400 calls/min, which shows the ability of the system to be able more than 25,000 unique clients per minute. It is important to mention that the server used for this trial has no special very high end specifications. It is an Intel Xerox 2 core CPU with only 2 GB RAM installed. If a good specs hardware machine is used with at least 12 GB RAM and latest available multi core machines, this response time as well throughput can be increased. It is worth noting that none of the calls failed in this process and each call was served with proper data it requested for. That shows the uptime and availability of the service under stress time and the request is served within the threshold time of data availability – which is a minute time. The changes in the throughout, response time as average and median along with deviation are shown below.

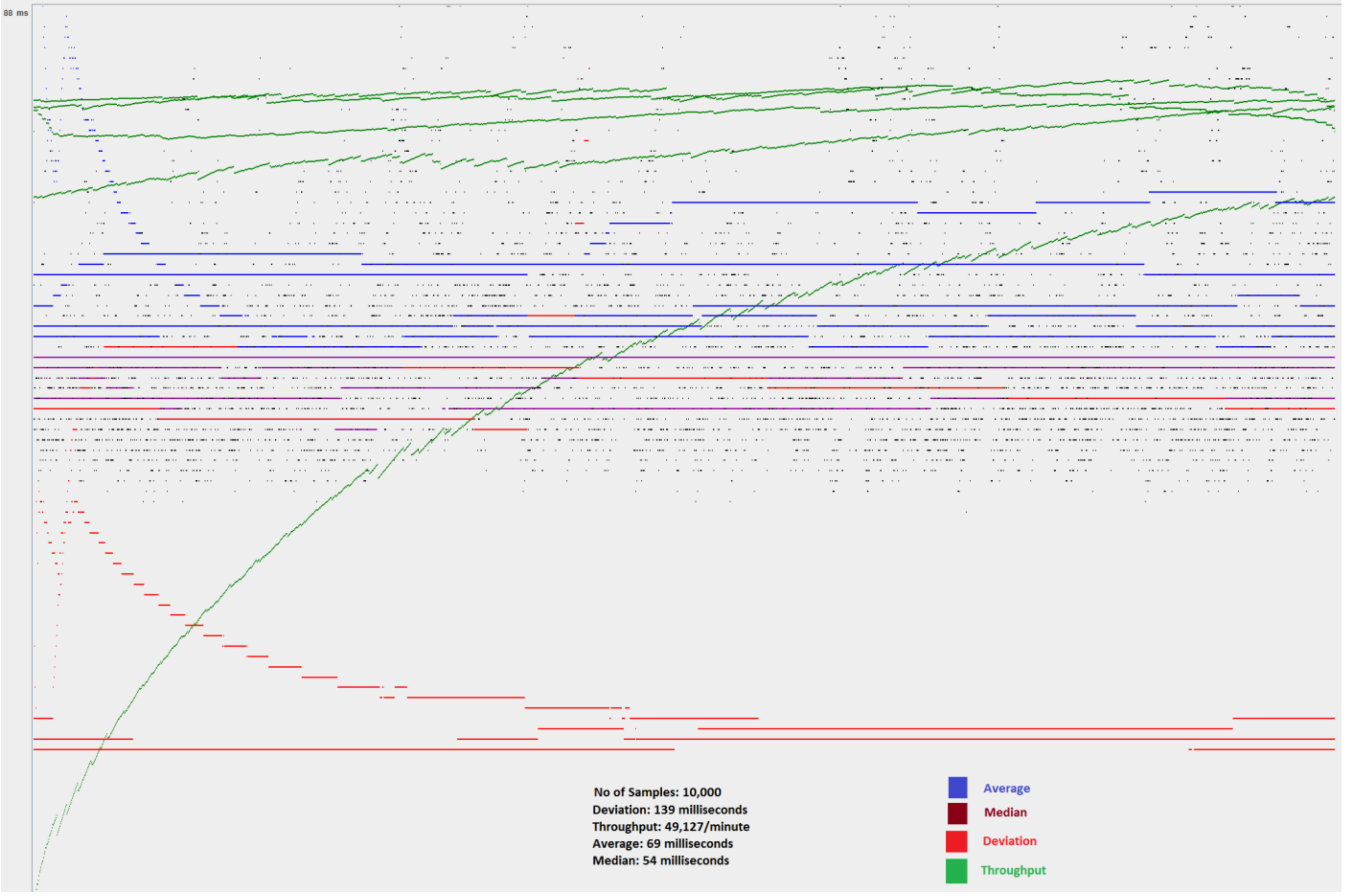


Figure 5-35 sample of 10,000 calls made to web service for data request

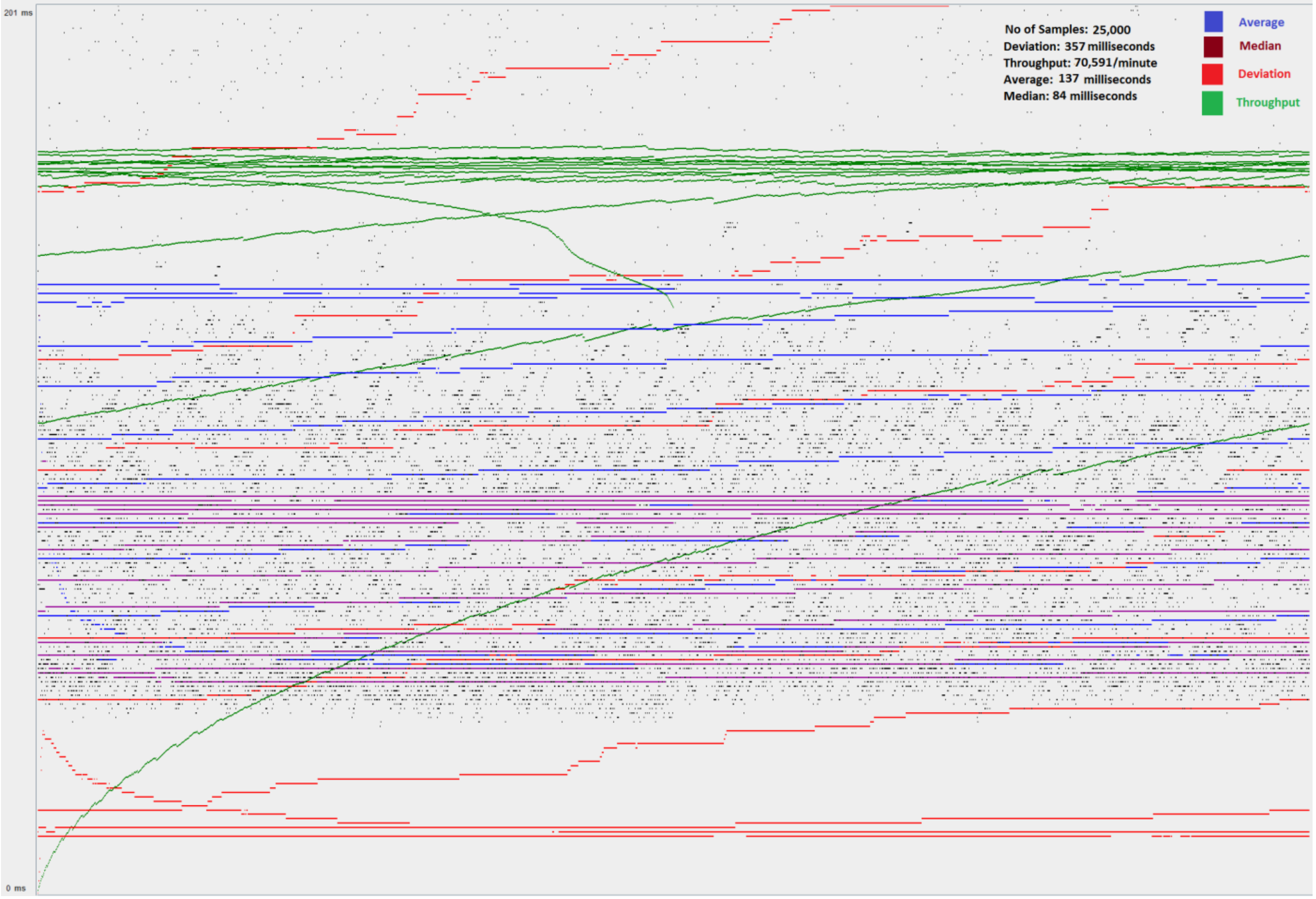


Figure 5-36: 25,000 web service calls from 500 unique users

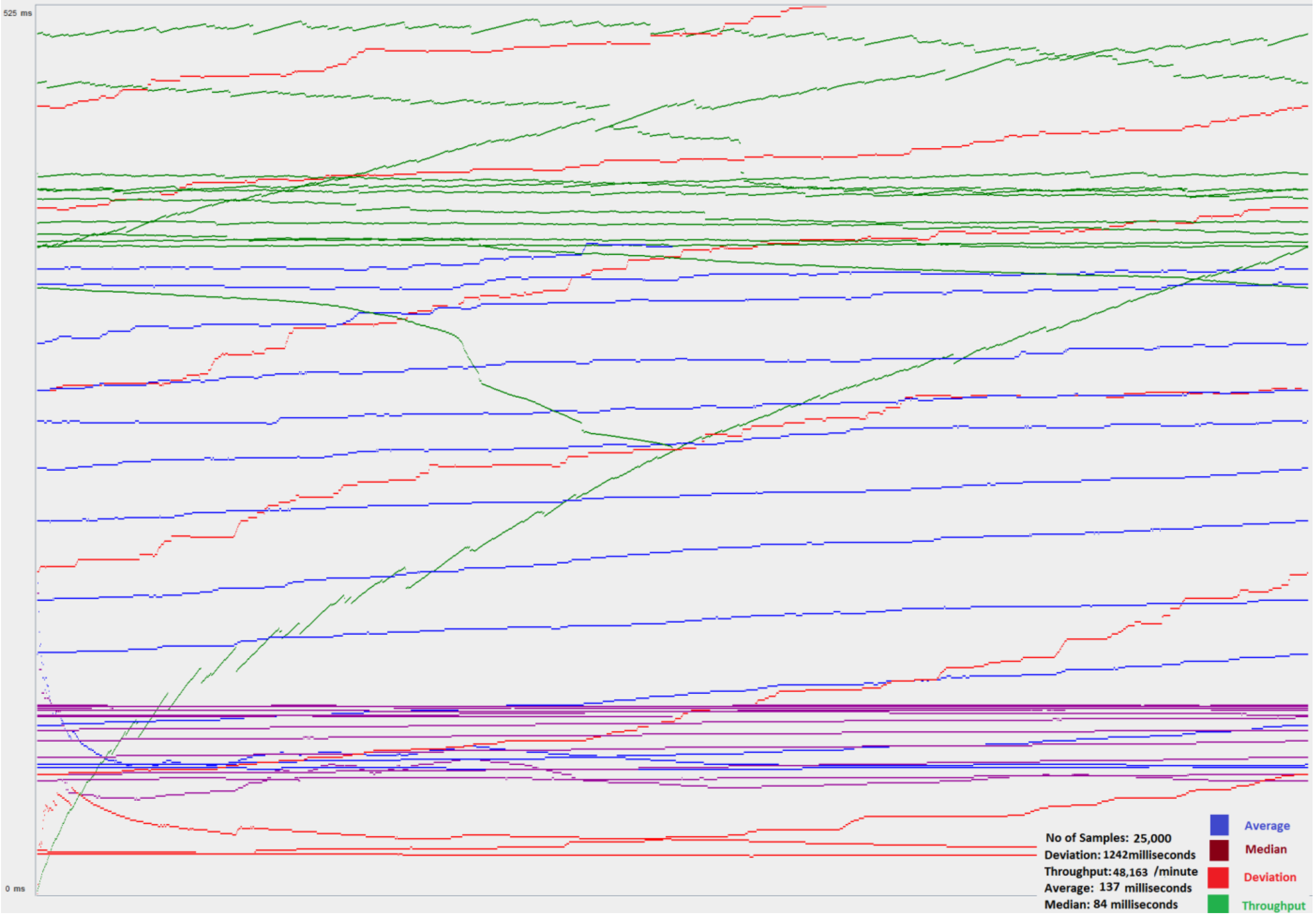


Figure 5-37: 1000 unique users making 25,000 calls

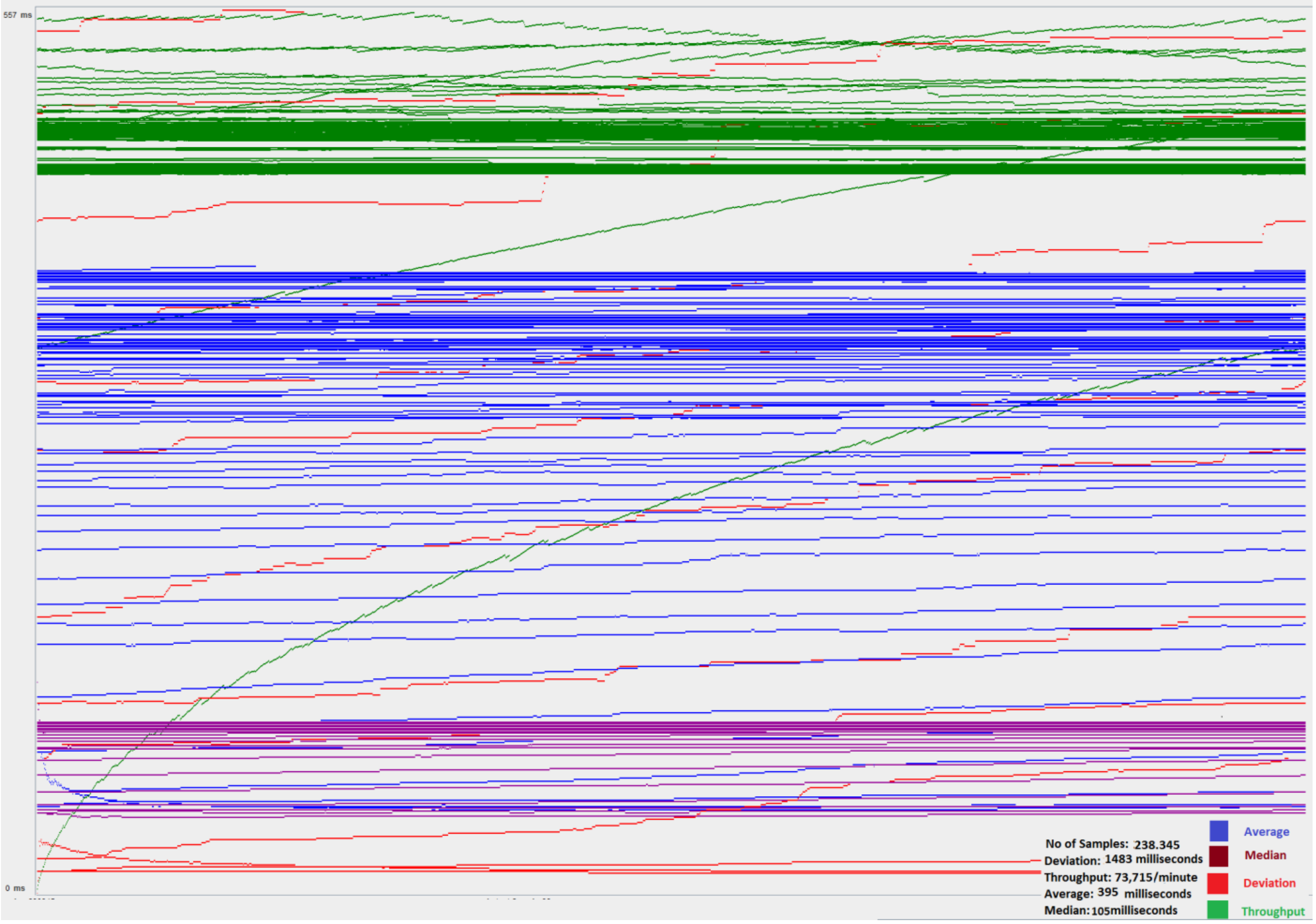


Figure 5-38: 500 unique users storming the services in a loop

5.5 Conclusion

This chapter has presented analysis of the data using Machine Learning algorithms applied through SQL Server Analysis Services. It has also given glimpse into the implementation of web services component and a detailed evaluation of the web services is presented as well. The predictive analysis was performed on a subset of the data to exploit the efficiency of these machine learning algorithms. Application of machine learning techniques to dig deep into the data discovered interesting relationships between different attributes of the data and provides a guideline on different perspectives of the data. The implementation and evaluation of both the web services component and the prediction component were discussed to make a basis for detailed case study in next chapter where a fully automated implementation will be discussed. The next chapter will talk in depth about different machine learning techniques that have been employed to convert the history data into a real-time prediction database. The integration of techniques with different custom mining models, visualization techniques and automation strategies for data integration from different sources will be presented along with the application of those models to processed data so that probabilities for prediction of arrival times and possibilities of delay can be generated using the simple inclusion of the context parameters attached with each report of the arrival time on a stop.

CHAPER 6: CASE STUDY: MERMAID TECHNOLOGY TRANSPORT SYSTEM

6.1Introduction

The realisation of an intelligent transportation system depends, among other things, on the structure in which the data is stored to allow for an easier data manipulation, including facilitating predictability in order to maintain a smooth operation of the system. The work conducted in this thesis is centred around examining one current urban transport system in order to propose substantial innovations, chiefly at the system architecture level, so that the data acquisition, manipulation, and storage can be optimized for introducing an effective prediction engine. The prediction engine is in place to make informed decisions, in real-time, on the operations of the system. This complex system-level change should aid planning and dynamic allocation of the resources required for an optimized operation of the transport network and management of the service.

This chapter starts with a brief introduction of the system used by a partner company on this project, Mermaid Technology, Copenhagen. The company provides an access to the full set of the sensory and context data, in real-time. It is followed by a presentation of the currently used system architecture and a brief analysis of the problems and challenges that a rather non-connected structure and an off-line approach to data handling brings to the management of the operations.

The proposed architecture, as presented in Chapter 4, is then applied to the Mermaid Technology system and implemented according to the evaluation results presented in Chapter 5. A comprehensive evaluation of the proposed architecture versus the existing, Mermaid Technology transport system – is presented at the end of this. The parameters taken into consideration to draw comparisons between the two systems - 'before' and 'after' - include the impact of predictability on the adherence to the timetable , the impact on the overall smoothness of the operation, optimisation of the use of resources, extendibility of the system – to name just some of the key factors by which the performance of the new system is evaluated.

The elaborations on the Case Study and the comprehensiveness with which the proposed architecture has been implemented and tested on real data, in real-time is showcasing the research performed on this project through allowing an 'in vivo' evaluation and testing of the new architecture, as well as the quality and fitness for purpose of the tailor made data mining algorithms to optimise the use of resources and to make the running of an urban transport network smooth and with minimum delays.

6.2 Case Study –Bus Management System in Copenhagen

The public transport system is, by its very nature, complex - and Denmark is no exception, especially the congested city of Copenhagen where hundreds of thousands of bus trips are made to more than 20 thousand bus stops every day. There are two types of main public transport mediums used in Copenhagen: i) bus network, and ii) over- and under-ground trains network. The buses use mobile network based internet connection that is shared between communication with server and provision of WIFI services inside the bus. This can potentially result in offline scenarios where bus has no connection or poor connection and it has difficulty in fetching information about arrivals and departures, especially real-time information. The map is presented in the Fig. 6.1: Copenhagen Bus Map, which shows bus lines servicing various destinations around the city. This Case Study deals with only a bus network management system with an emphasis on the subset of buses serving in the urban cycle of the city of Copenhagen.

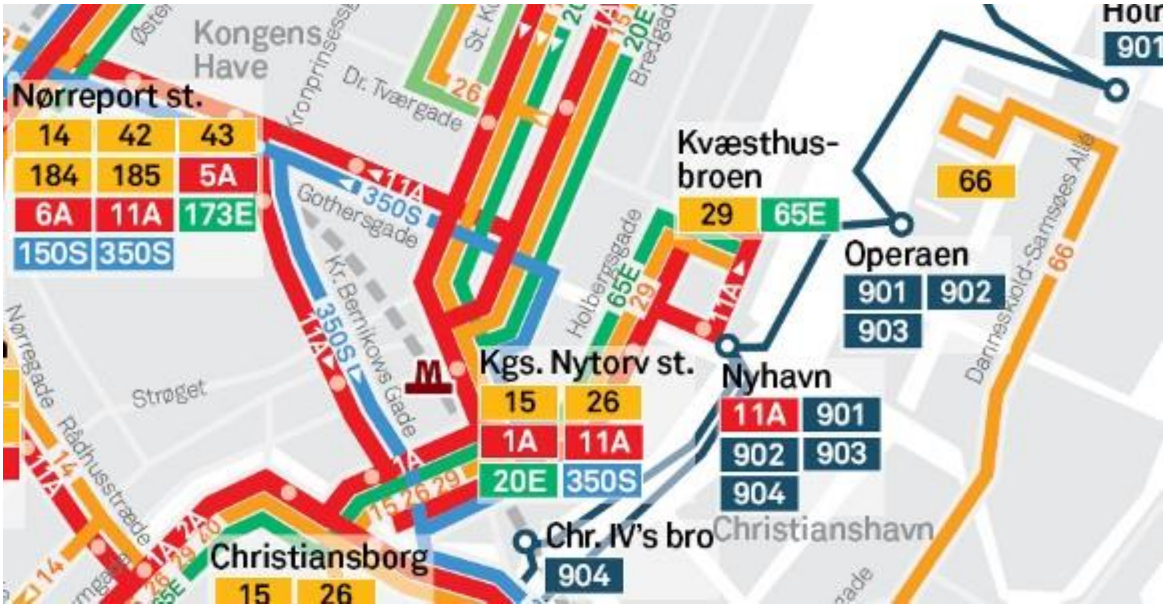


Figure 6-1: Copenhagen bus map

The zonal map of the bus stops is presented in Figure. 6.2 where all the zones in Denmark are presented to demonstrate the scale of the system and the potential amount of data being collected from buses from all these zones. Most of the routes operate in multiple zones.

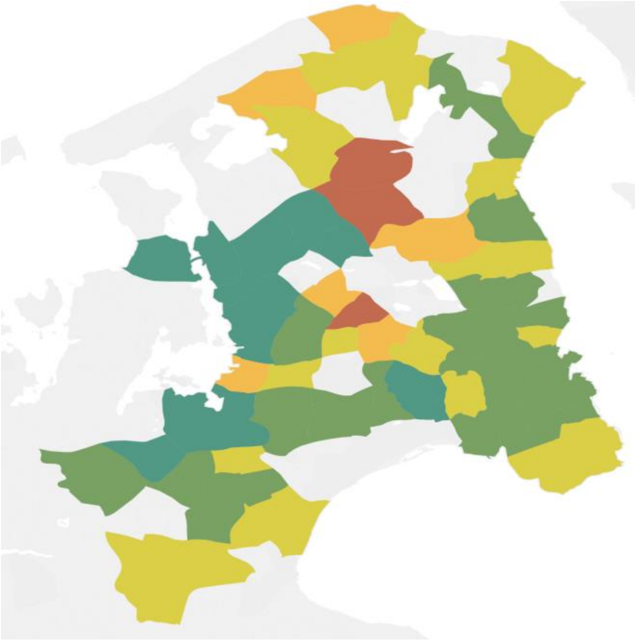


Figure 6-2: Zonal map of Denmark with buses operating in

The city of Copenhagen is the only major city of Denmark with direct and quick connection to other European countries such as Sweden, via both the train and bus services. It is a typical EU city with urban environment, modern establishment, heavy footfall and streets and roads insufficient to handle the traffic requirements of the city. The use of cycles is very frequent but public transport is the preferred option to save time in commuting. Although there is a reasonable metro system with trains connecting different destinations in the city the buses are still the main means of transport. The government policies also encourage its citizens to use the public transport since the environment related objectives are high on the government's agenda. Figure 6.3 and 6.4 indicate the operational stops on only 30 routes and then Live buses mapped on the stops indicating how many connections exist between different routes. The connections add lot of complication to an information system that can provide information about arrivals.

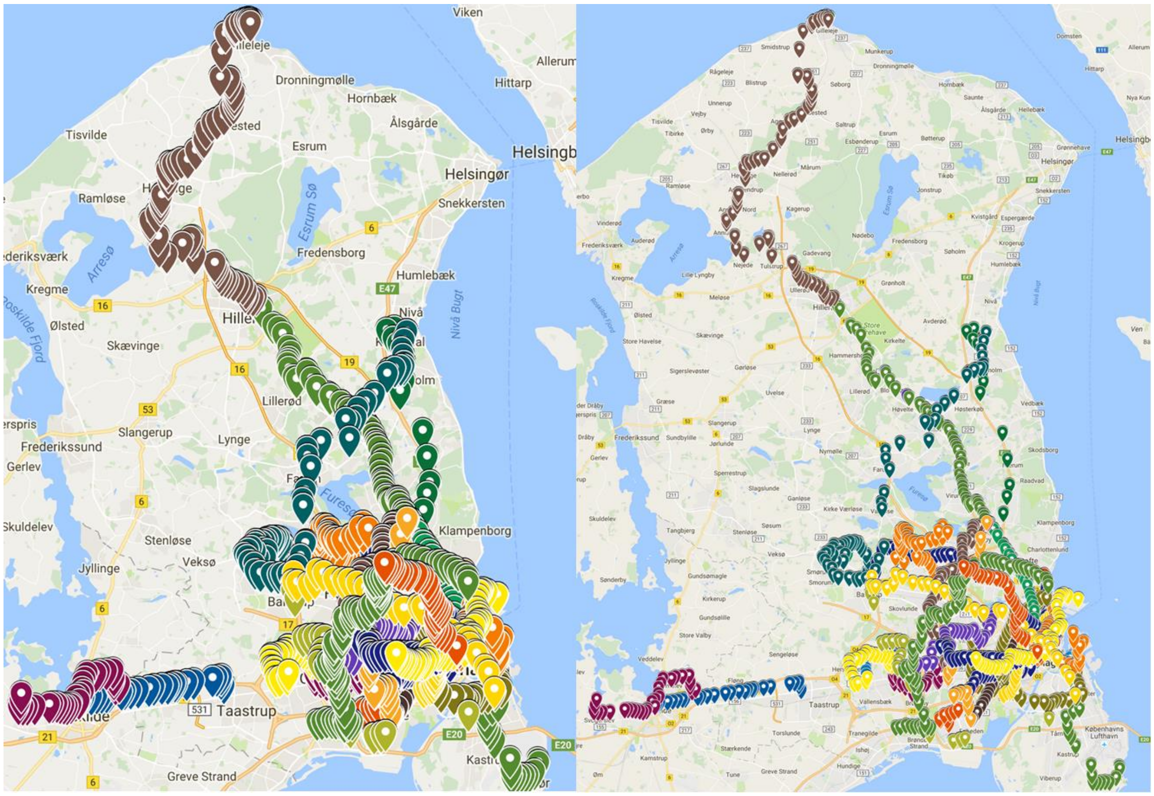


Figure 6-3: Bus stops shown in 30 routes

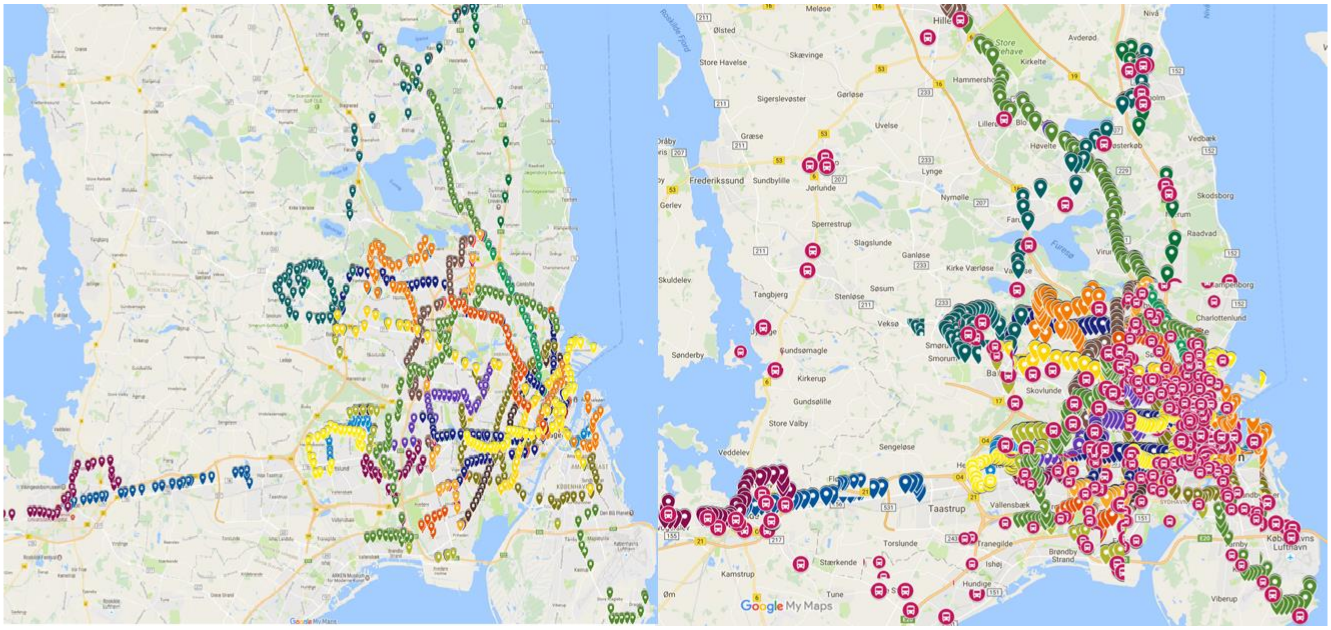


Figure 6-4: Live buses mapped on stops during the day - off peak time

6.3 Existing Infrastructure

Mermaid Technology's existing system is based on service oriented architecture, a diagram for which is presented in Error! Reference source not found. Although the system implements web services for importing the data and exporting selected data to sub-systems such as the client applications on the devices on-board the buses - the design of the overall system does not support predictability without a major change in its core layers. The system has organically grown into a large entity based on the requirements as they evolved in time, and, therefore, the interaction between different components needs continuous maintenance to ensure they can work together. The information is acquired through many different interfaces and saved into the database without any linkage to other parts of the system that may need this information. A good example of this is that the current location of a bus is reported, for each bus in the network, every few seconds and stored in a separate database, whereas the information about the stops the buses are at, and correspondent times of arrival are stored in a different database. It means that if there is a need for the data of all reported locations of a bus between any two stops - for deriving the speed at which the vehicle was traveling - this information will need to be retrieved from two different databases, the link then established at run-time in order to perform any queries or analytics. This type of a scenario makes the usage of data very complicated and the processing operation lengthy and unnecessarily repetitive, because there is a pre-processing required every time there is a need to perform the same type of analysis on the data. The current system faces some of the challenges identified in the existing transport management system architectures, particularly the Quality of Service (QoS), the lack of prediction, the dependence on external data providers, the unreliability of the bus operation data etc.

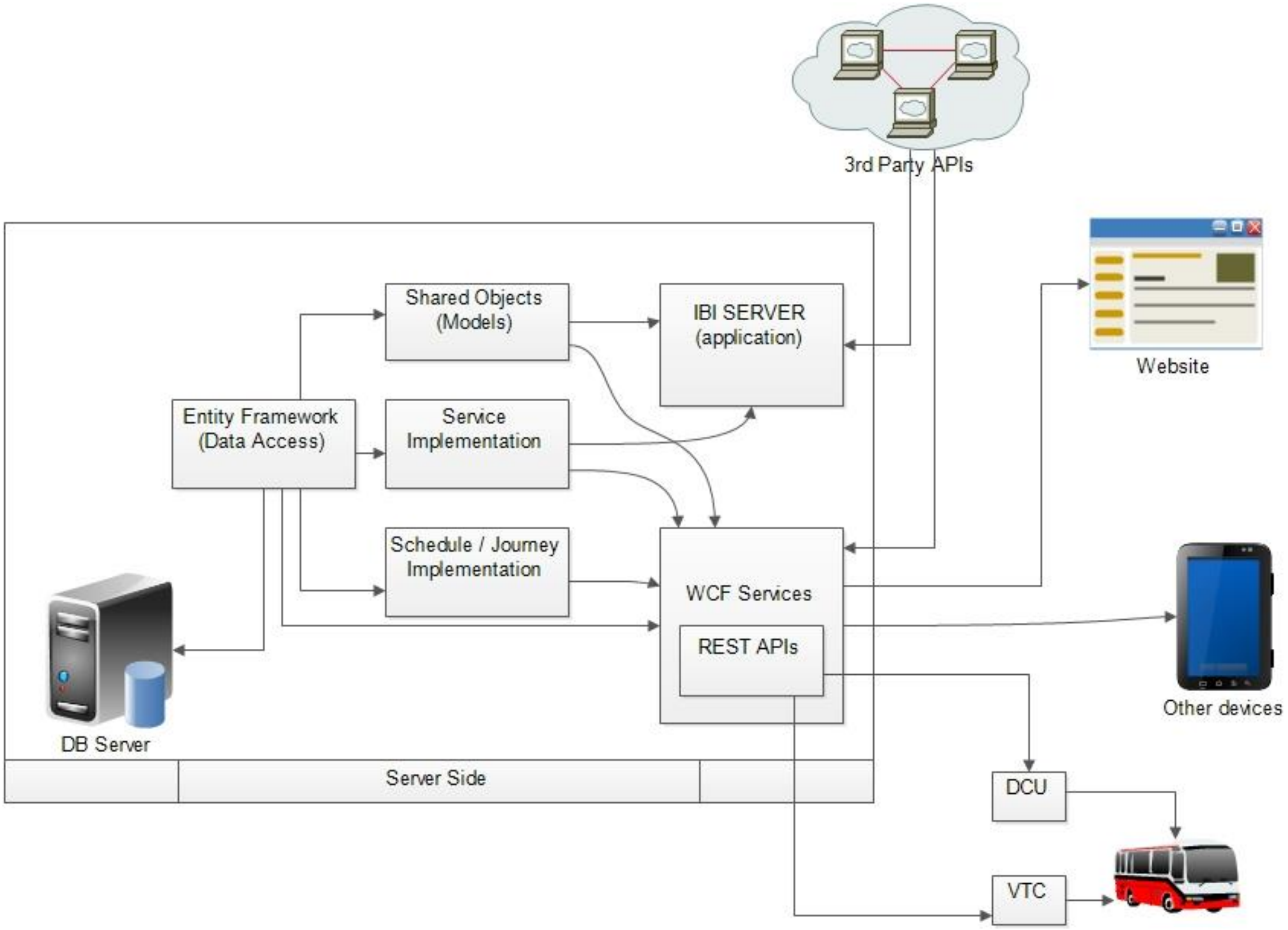


Figure 6-5: Existing system – Mermaid Technology

The system has grown and is still increasing the number of sensors being installed on the buses. Every new sensor needs a new service to be implemented on the system so that the data acquired from that sensor can be imported into the system. This generates yet another schema where data is saved and the integration effort increases – in time and resources - with every new implementation. All the services designed as part of the 'service oriented architecture' are, in fact, working as data acquisition services. The exporting data operations are serving the data that is already in the database without any effective analysis in real-time ('on the fly'). In an attempt to make the system, to an extent, predictable, Mermaid Technology recently started to export the location data to external vendors for processing in order to develop their prediction engines, but the exported data misses the basic context information of each vehicle. There is hardly any analysis performed on the data and no prediction algorithms, or system, is in place to facilitate it - even when the data sets being collected is fit for performing sufficiently accurate predictions for running a smart, smooth urban transport operation.

6.3.1 Technical Details of the Existing Architecture

As mentioned earlier, the data is strictly stored in a normalised relational database and, since it is rapidly increasing in volume, the efficiency of the system services is affected resulting in significantly delayed response-times on the data being exported. This is critical for a dynamically managed transport system where delays of the order of a second render the information out-dated in a matter of minutes. This constitutes the rationale for introducing a novel architecture that will cater for predictive analyses and the management of an urban transport system in real-time.

The existing system architecture has primarily three layers of implementation. On the server side, there are 2 layers: one for web services and one data layers. The services layer is primarily responsible for retrieving data about the planned, future journeys for the buses including, for example, information on bus line allocations to particular journeys, the starting point of each journey and the stops and destinations involved in each journey. The services have embedded implementation mechanisms for various different sources of data, including from 3rd party data operators, such as MOVIA [174] – storing all the data, eventually, into the local database. The data comes in different formats, such as JavaScript Object Notation (JSON) and XML, and it has to be transformed into the format suitable for the local database.

It means that, if there is JSON format data coming from two different sources, then the parsing logic is potentially duplicated in each route, with no central component in place to share the unique implementation. This makes the maintenance of the operation complicated and the inclusion of more sources of data becomes very cumbersome.

The data layer is used to process all the requests from the web services layer and it is used also to import and export the data. The data layer is based on an SQL Server database and provides access to other databases created for various entities impacting on the system and on the operation – such as journeys, location logs, operational logs etc. The server machines for hosting databases are of an appropriate specification, but they are mostly under extreme load and processing all the time. This often results in time-outs because the same database and its tables are being accessed simultaneously from different services, which creates deadlocks leading to various failures involving a data loss. This is due to an insufficiently structured architecture of the existing system – the problem that is being addressed, in part, in this thesis.

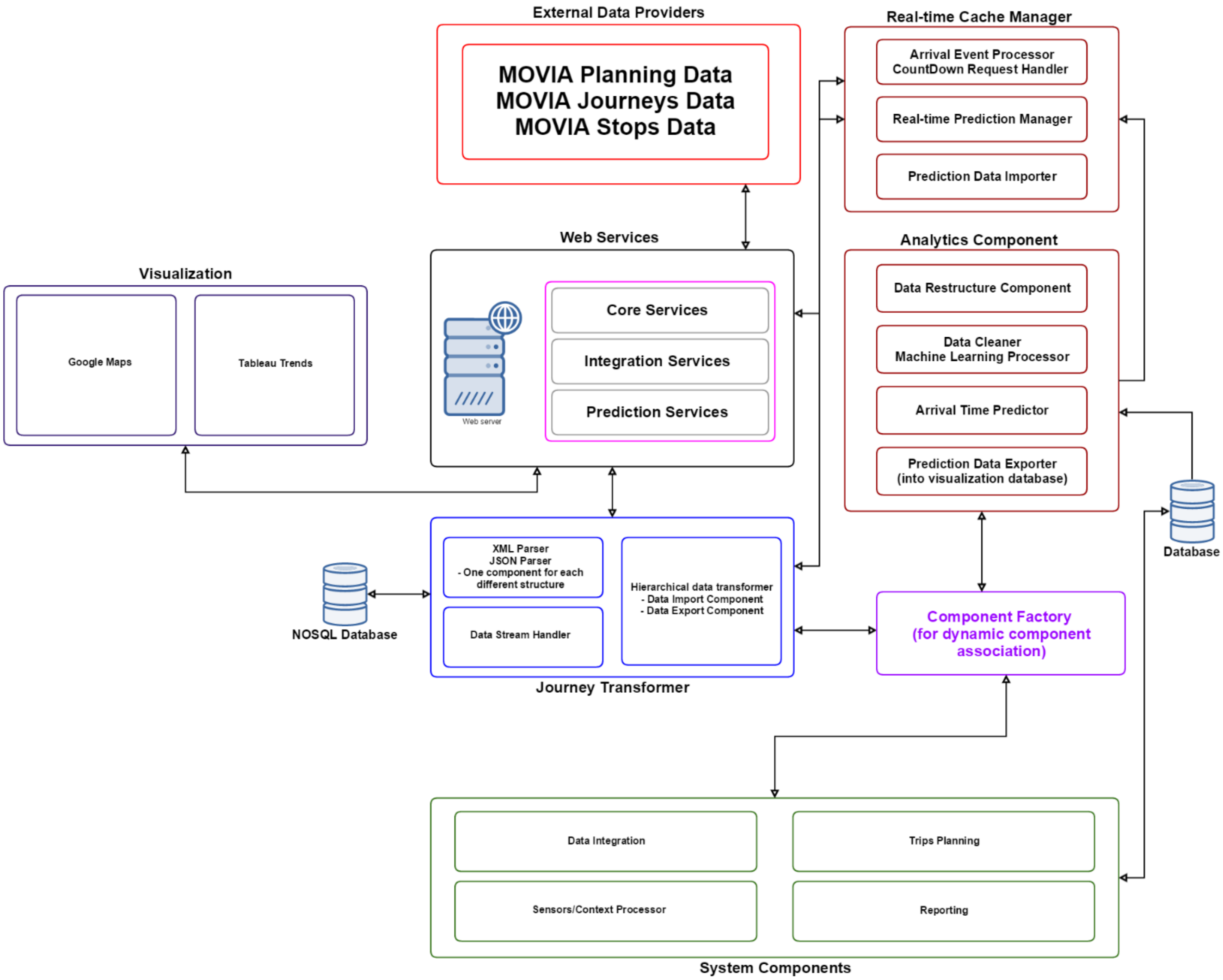


Figure 6-6: Adapted architecture - bus system

These practical challenges are a barrier to using the existing system architecture for a modern, continuously changing transport system where new sensors and interfaces are being added to make the system smarter and more environment-friendly. These challenges also hamper the efforts to make the data reliable for the commuters as end-users, as well as for the transport staff, as the decision-making on both sides is based on this data.

These problems may be addressed by, for example, putting more hardware resources into the system, but that would be a temporary solution and would not address the core of the problem. It is for this reason that the requirement of a new architecture emerged – as it was felt that it can not only provide a basis for building a reliable information system, but also address the many challenges of the existing system by providing a tailor made prediction engine based on the data that is already being acquired in this vast system. The proposed new architecture, presented in Chapter 4 and adapted for the Mermaid Technology transport system, as shown in Figure 6.6, addresses these challenges using the strategies discussed in Chapter 4. The implementation of it, within the Mermaid Technology remit is discussed here, as a Case Study. The architecture intends to make the data handling - from acquisition to storage and processing of it – seamless and fit for real-time predictive analytics. The details of how it handles the data are described in the next section of this Case Study where migration to the new architecture is discussed.

6.4 Migration to the new system

The diagram for the proposed architecture presented above, in Figure 6.6, for convenience and to aid understanding of the migration process. The details of the new architecture are presented in Chapter 4. The key features of the proposed architecture include: i) an automated, efficient data acquisition system, ii) a flexible data transformation facility, with the ability to add new formats, iii) a prediction engine based on a novel caching approach, iv) an integrated predictive analytics system, v) diverse data storage mechanisms to support predictive analytics and vi) an ability to supplement the architecture of the existing system with vital components that will integrate predictability and intelligence in data processing.

6.4.1 Automated Data Acquisition

The lifecycle of the data generation, processing and storage changes when a new architecture is implemented. For example, data filtering and transformation become an integral part of the processing to make it ready for applying machine learning algorithms and the Real-Time Cache manager component to integrate the data with both short-term and long-term predictions. The automation of the data acquisition process into the system enables processing of real-time data thus allowing for an impact analysis on existing situation of the transport network, in real-time. A novel data acquisition approach is applied in this project so that a filtered, standardized format data is added into the system, ready for processing and further analysis.

The automated data acquisition process has been tailor made for this application and implemented as an *Integration Services* application. The significant excerpts from the code for this application are presented in the Appendix A, for reference. This automation script is run in regular intervals and it follows an incremental approach to acquire the latest data and append it to the dataset being used for updating the prediction engine, so that it is always ready to make predictions based on the latest data collected from the transport system. As shown in Figure 6.7, this acquisition and transformation package creates an integrated, unique source of data and a unique destination end-point, from which the data is processed. This makes the prediction engine capable of working with virtually any database because the only intervention that would have to be made is to change the source properties in this package according to the nature of the data source and it will be automatically transformed into the required format.

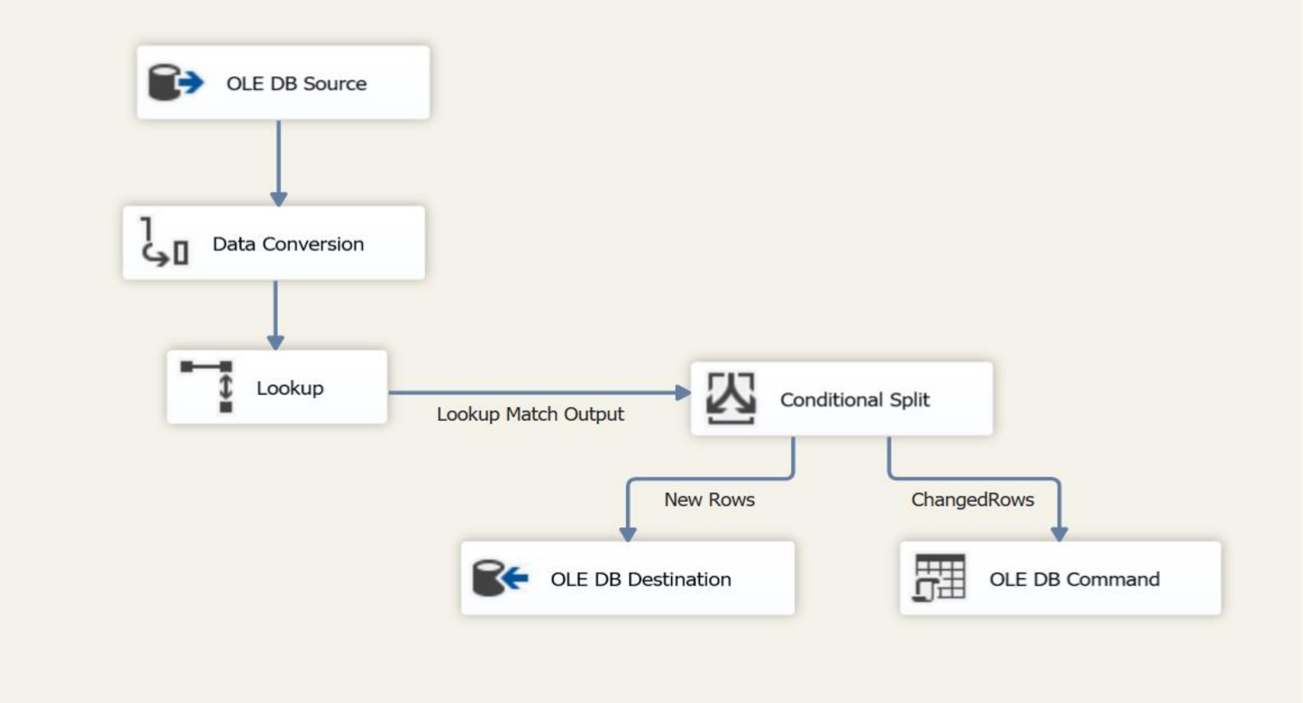


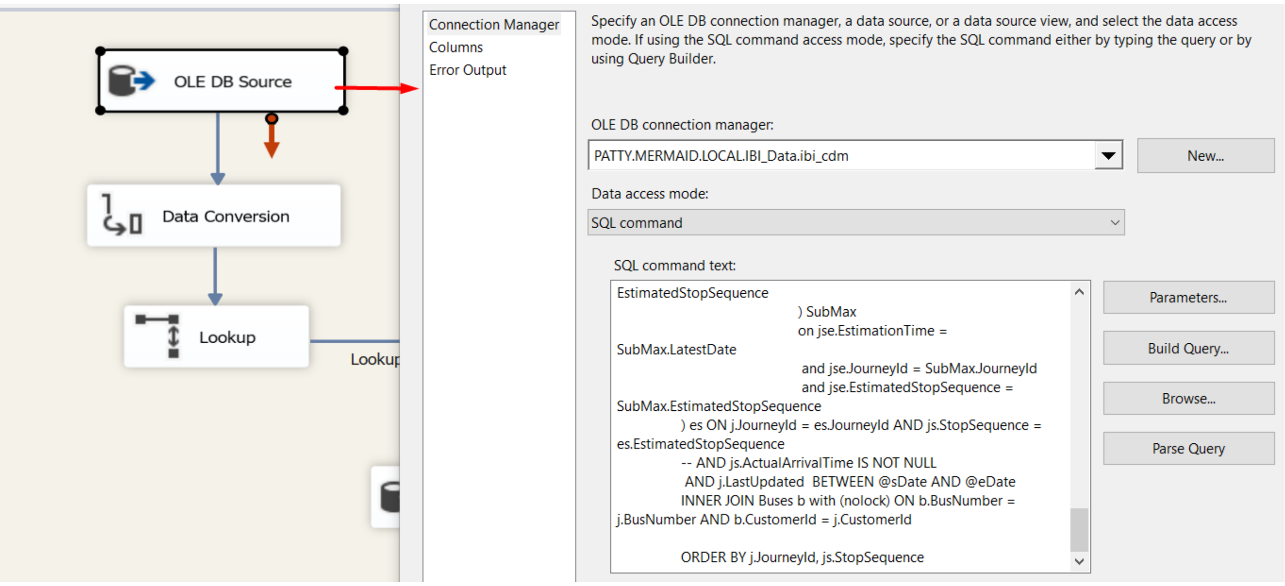
Figure 6-7: Architecture for real-time data acquisition

Figure 6.7 also shows different stages of the automated data pre-processing, where, in the first step, the data is acquired/retrieved from the source, in this case from an SQL server database. The live data-stream is being monitored and this package identifies instantly when the new data set becomes available. At this point, the raw data from the first batch is fed into the Data Conversion component that has a separate facility for each data format of the raw data based on the source acquired from. This same component serves the purpose of integrating data from heterogeneous sources and converting it into one format, standardized for the prediction engine implemented.

Once the data is converted, it is passed onto the *data validation* component, which validates the data for possible duplications or redundancy. All required modifications of data are automatically applied. All the new entries are simply appended to the destination dataset to update the new information required to be analysed. Each step is explained in the following sub-sections.

6.4.1.1 Data Acquisition – configuration of incoming data sources

The dataset used for this case study exists in the SQL Server database and therefore, this data acquisition component uses OLE DB connection to connect to the source database. Figure 6.8 presents configuration interface for setting up incoming data sources that can be both real-time and event driven. This connection is used to fetch the new or updated data, when available, on the regular intervals.



Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

OLE DB connection manager: PATTY.MERMAID.LOCAL.IBI_Data.ibi_cdm

Data access mode: SQL command

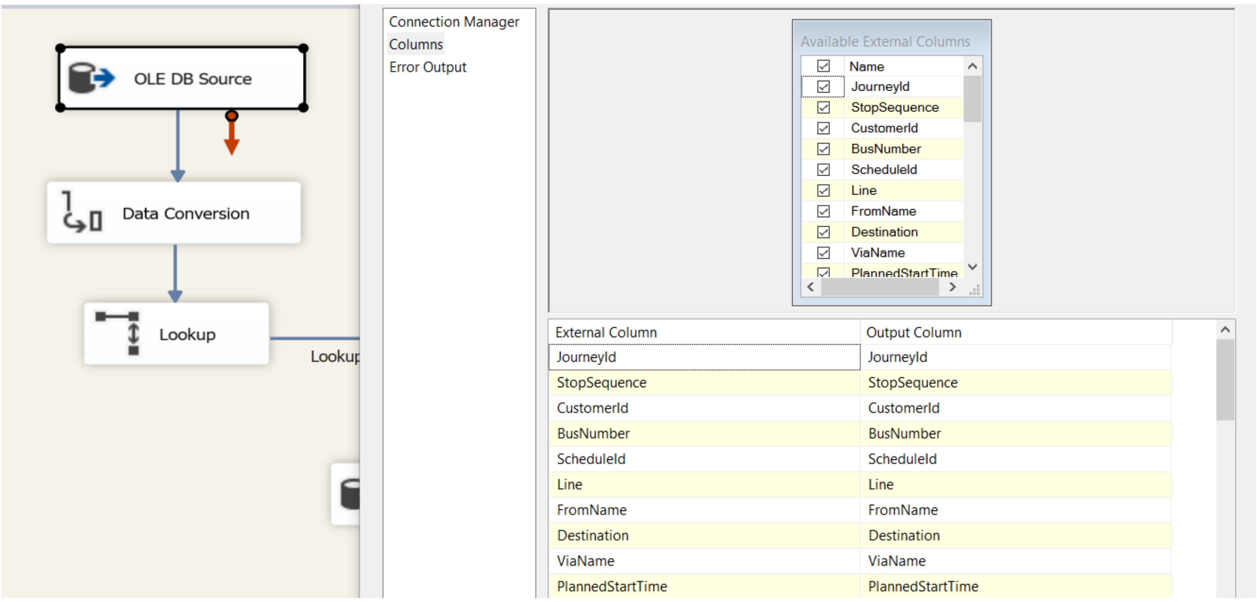
SQL command text:

```
EstimatedStopSequence
                                    ) SubMax
                                    on jse.EstimationTime =
SubMax.LatestDate
                                     and jse.JourneyId = SubMax.JourneyId
                                     and jse.EstimatedStopSequence =
SubMax.EstimatedStopSequence
             ) es ON j.JourneyId = es.JourneyId AND js.StopSequence =
es.EstimatedStopSequence
             -- AND js.ActualArrivalTime IS NOT NULL
              AND j.LastUpdated  BETWEEN @sDate AND @eDate
             INNER JOIN Buses b with (nolock) ON b.BusNumber =
j.BusNumber AND b.CustomerId = j.CustomerId

             ORDER BY j.JourneyId, js.StopSequence
```

Figure 6-8: Configuration of input data sources

It needs a SQL Query to read the data in raw structure and output it into one flat structure that can be used for conversion to the format compatible with the system. The script contains information about which attributes of the data we are interested in like arrival and departure time, existing estimations for those stops and information about the stops so that their identity can be attached with different features defined for processing. Figure 6.9 shows the extracted attributes based on the query and they define each instance uniquely for its route and timeline.



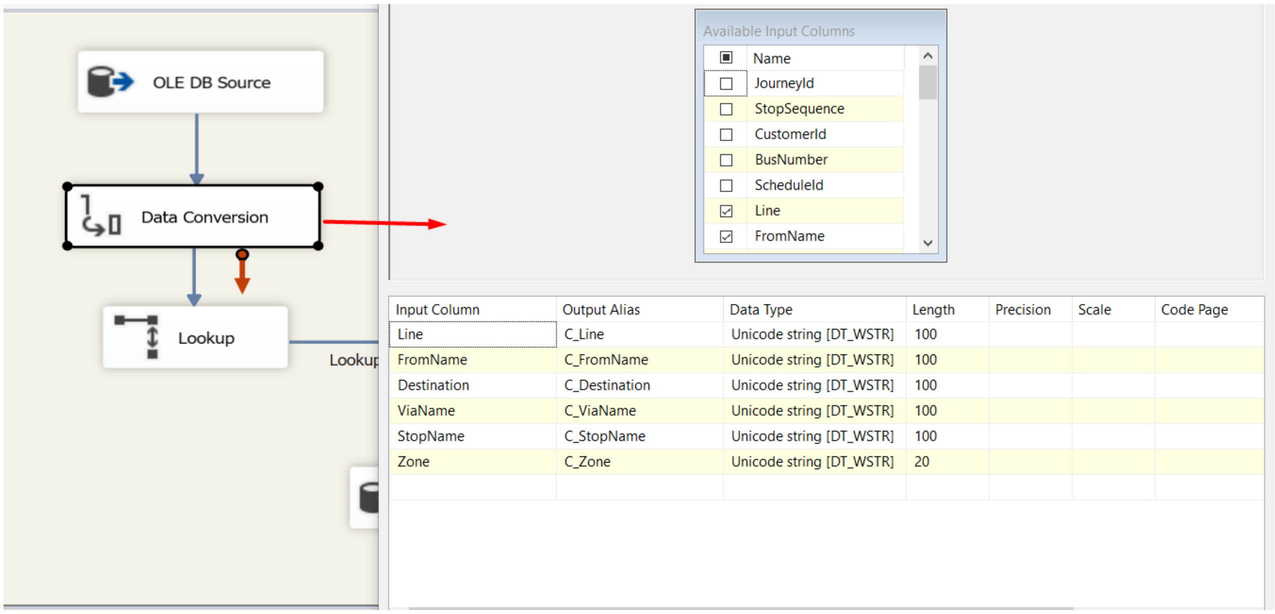
☑	Name
☑	JourneyId
☑	StopSequence
☑	CustomerId
☑	BusNumber
☑	ScheduleId
☑	Line
☑	FromName
☑	Destination
☑	ViaName
☑	PlannedStartTime

External Column	Output Column
JourneyId	JourneyId
StopSequence	StopSequence
CustomerId	CustomerId
BusNumber	BusNumber
ScheduleId	ScheduleId
Line	Line
FromName	FromName
Destination	Destination
ViaName	ViaName
PlannedStartTime	PlannedStartTime

Figure 6-9: Mapping of attributes in data sources to destination structure

6.4.1.2 Transformation of heterogenous data sources

Once the data is fetched from the source or a set of sources, it is filtered so that the attributes important for predictive analysis like time and route information are extracted from the fetched data. Also, the transformation of raw data being collected into dependent and independent variables needs to intelligently identify the type and content of the data and then present alternatives for its transformation. Although some level of the filtering is possible at the query stage too in acquisition stage, but the attributes fetched at acquisition stage are kept in the system so that the extension of the features to be considered does not need any changes in the source dataset and they can simply be extracted from the existing fetched structure of the data. The Figure 6.10 shows the attributes filtered from the selected set and information about their type and the destination type is synced to make sure data is transformed as expected. The naming of the attributes also changes because the raw data has different names for features and it is important that regardless of the type of data source, the destination attribute names are always same so that no change is needed in the algorithms to process the filtered dataset. This stage is used not only for filtering of the attributes to be used in current scope of the study, but also it is used to derive new attributes based on existing attributes through an automated transformation process. An example mapping configuration is shown in Figure 6.9 that defines the output attributes, both primary and derived, generated from the data being collected. The timestamp attribute is one such clear example that is converted to year, month, day, hour and minute attributes to make the dataset compatible with most of the mining platforms that do not support the timestamp type directly.



Available Input Columns

■	Name
□	JourneyId
□	StopSequence
□	CustomerId
□	BusNumber
□	ScheduleId
☑	Line
☑	FromName

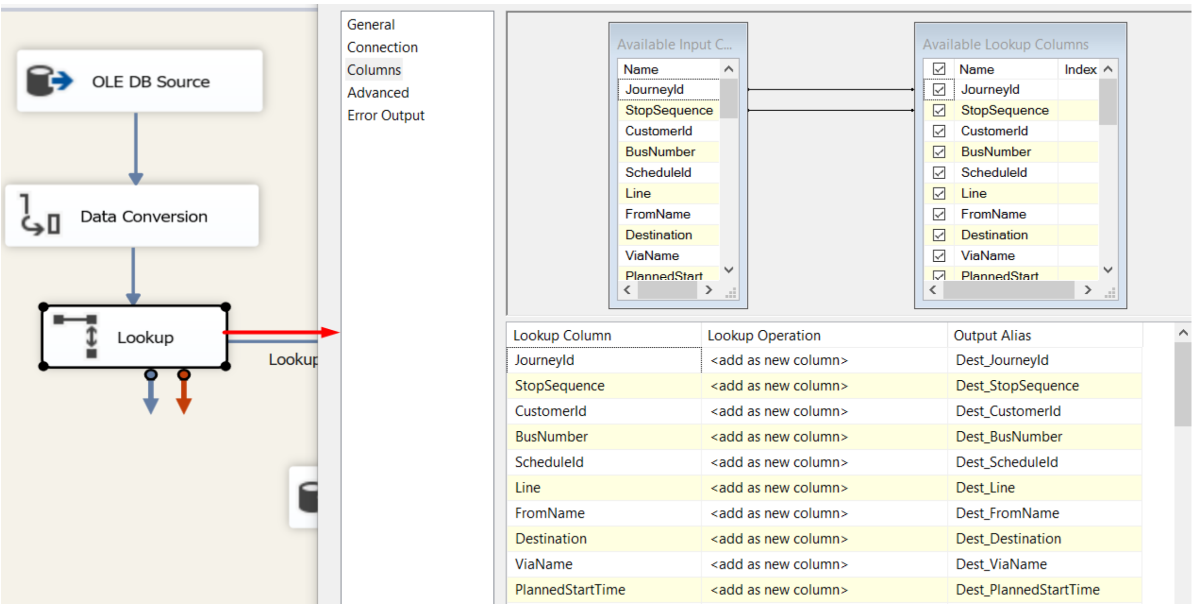
Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
Line	C_Line	Unicode string [DT_WSTR]	100			
FromName	C_FromName	Unicode string [DT_WSTR]	100			
Destination	C_Destination	Unicode string [DT_WSTR]	100			
ViaName	C_ViaName	Unicode string [DT_WSTR]	100			
StopName	C_StopName	Unicode string [DT_WSTR]	100			
Zone	C_Zone	Unicode string [DT_WSTR]	20			

Figure 6-10: Transformation and multi-language mappings

The conversion stage also helps in filtering out the instances that are not needed or exist as anomalies and can impact the results because of a corrupt instance occurred in the data. An example is that if a stop arrival event was not recorded then that instance must be discarded because its value of NULL or ZERO will have its part considered in the predictive analysis and the algorithm will consider it as one of the valid values to consider in processing. To support data in different languages, Unicode encoding scheme is implemented so that any data that is coming in non-English language and represents special characters is transformed without any data loss. This configuration mapping is shown in Figure 6.10.

6.4.1.3 Data validation and filtering

When the data is coming from difference sources, we can have lot of duplicate data in it that needs to be sorted out before it is fed into the prediction engine. For example, data coming through location of a bus and arrivals at a stop contains location reported from different sources and can potentially be different. We have implemented a lookup process that checks for conflicting references of the data when it is duplicate and redundant and extracts the correct information while discarding the redundant or out-dated data. The lookup component establishes a link between the new data being fetched in this iteration and the destination data that already exists. As shown in the Figure 6.11, each column in the available input columns is mapped to available lookup column in the expected destination dataset. This mapping ensures that each attribute in the source dataset has an equivalent attribute in the destination dataset and this information is used later to identify if this instance is a new entry or an update to the existing instance fetched in previous iterations. Each lookup mapping is controlled by the Lookup operation that defines if it should be compared before it is copied or the data can be copied as it is. In addition, the derived attributes use these operations to output the value that eventually is copied to the destination attribute.

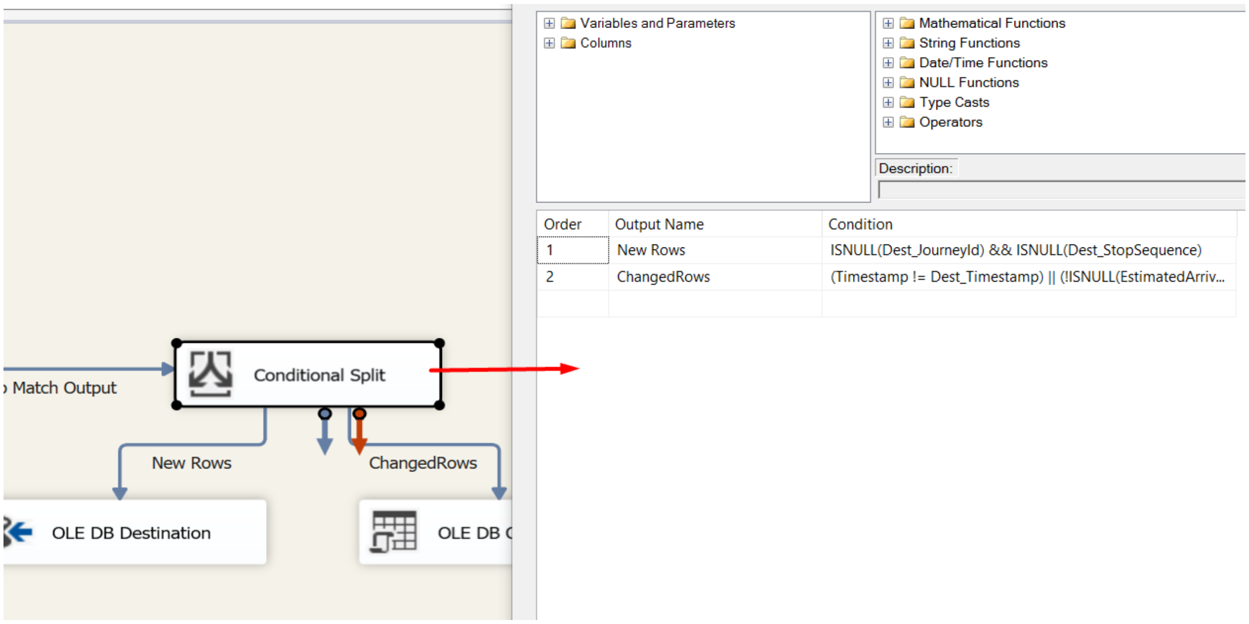


Lookup Column	Lookup Operation	Output Alias
JourneyId	<add as new column>	Dest_JourneyId
StopSequence	<add as new column>	Dest_StopSequence
CustomerId	<add as new column>	Dest_CustomerId
BusNumber	<add as new column>	Dest_BusNumber
ScheduleId	<add as new column>	Dest_ScheduleId
Line	<add as new column>	Dest_Line
FromName	<add as new column>	Dest_FromName
Destination	<add as new column>	Dest_Destination
ViaName	<add as new column>	Dest_ViaName
PlannedStartTime	<add as new column>	Dest_PlannedStartTime

Figure 6-11: Lookup configuration for discarding unnecessary data and relationship of different data attributes

6.4.1.4 Data exporting configuration

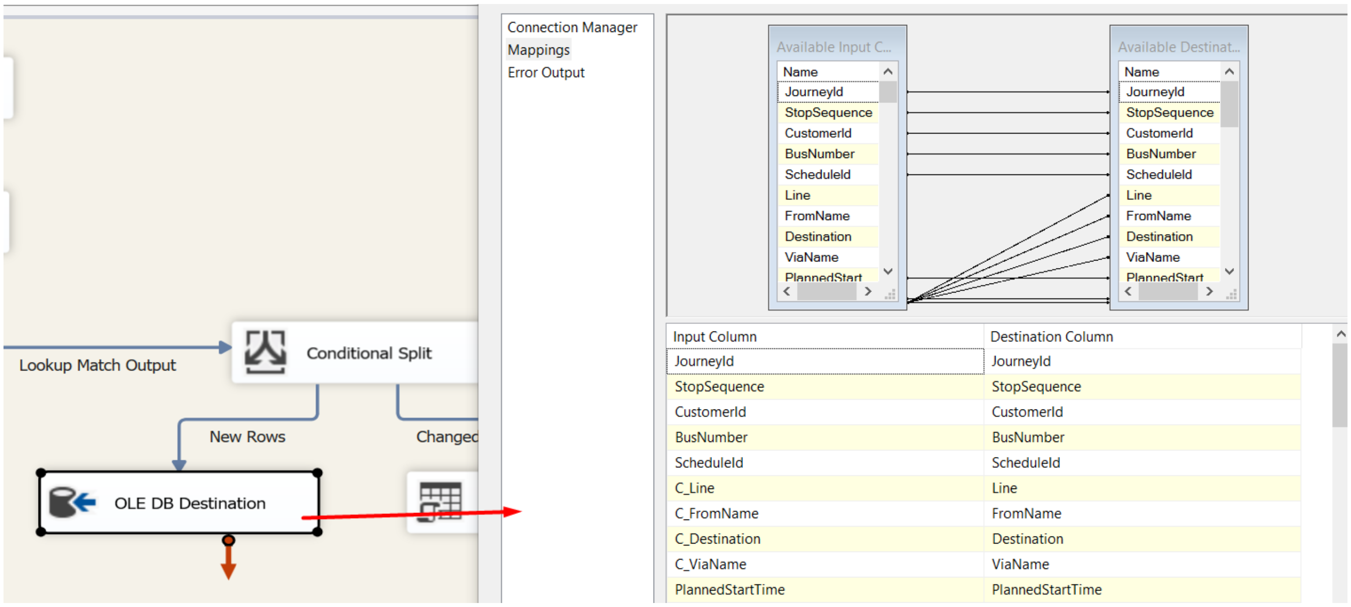
After the data goes through the data validation and filtering stage, it is ready for copying into the final destination tables or instances depending on the type of the destination data source used. It is important to mention that the process is completely independent from the type of data store and can virtually export data to any destination and therefore, extends the use of this novel acquisition and processing process for virtually any data oriented system. Like the conversion and validation components, this data export configuration component also needs a separate implementation instance for each type of destination data source to be supported. It helps to get the data into a format and storage that is compatible with the tools being used to conduct the predictive analysis on the data. In this study, the destination database is also SQL Server and therefore two OLE DB output connections are used to facilitate the induction of new rows or updated rows into the destination data source. The Figure 6.12 highlights the condition for the split to identify if the instance already exists in the destination dataset or a new instance is inserted. As shown in Figure 6.12, different criteria can be applied to each type of export and each export can be going to different type of data sources altogether.



Order	Output Name	Condition
1	New Rows	ISNULL(Dest_JourneyId) && ISNULL(Dest_StopSequence)
2	ChangedRows	(Timestamp != Dest_Timestamp) || (!ISNULL(EstimatedArriv...

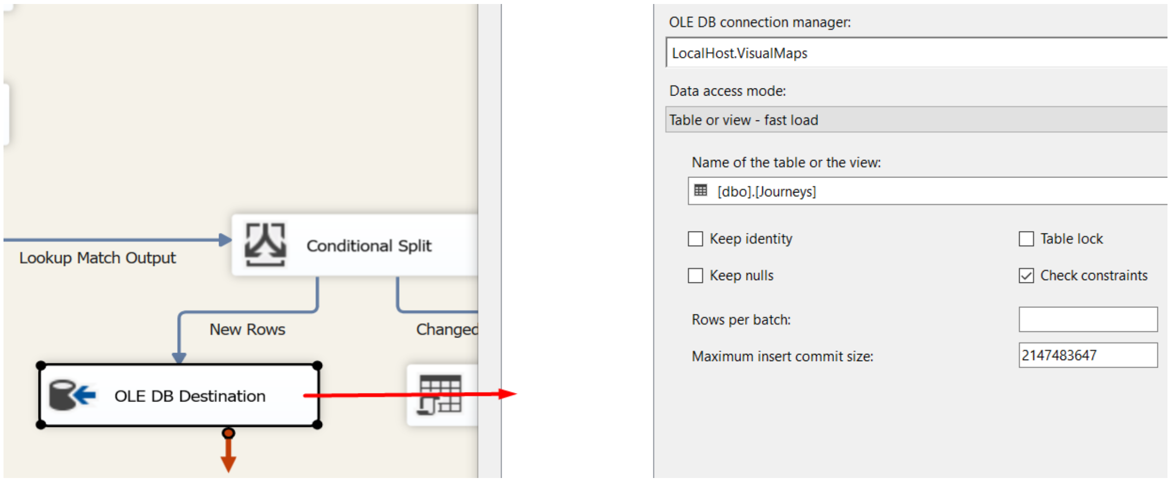
Figure 6-12: Criteria configuration for removing anomalies and redundant data

Figures 6.13, 6.14 and 6.15 indicate the workflow of association of source and destination attributes both for new and updating instanced with information needed for the destination storage. The connection parameters for the destination storage along with the name of the entities that will be used to store the information is also provided. A mapping is established here as well so that the system knows how to apply conditions and on which attributes before they are copied or upgraded. This completes the cycle of data pre-processing so that it is ready for predictive analysis through application of machine learning techniques selected in chapter 5 where they were applied to a limited dataset for the same source.



Input Column	Destination Column
JourneyId	JourneyId
StopSequence	StopSequence
CustomerId	CustomerId
BusNumber	BusNumber
ScheduleId	ScheduleId
C_Line	Line
C_FromName	FromName
C_Destination	Destination
C_ViaName	ViaName
PlannedStartTime	PlannedStartTime

Figure 6-13: Exporting the data into prediction engine - mappings and transformation



OLE DB connection manager:
LocalHost.VisualMaps

Data access mode:
Table or view - fast load

Name of the table or the view:
[dbo].[Journeys]

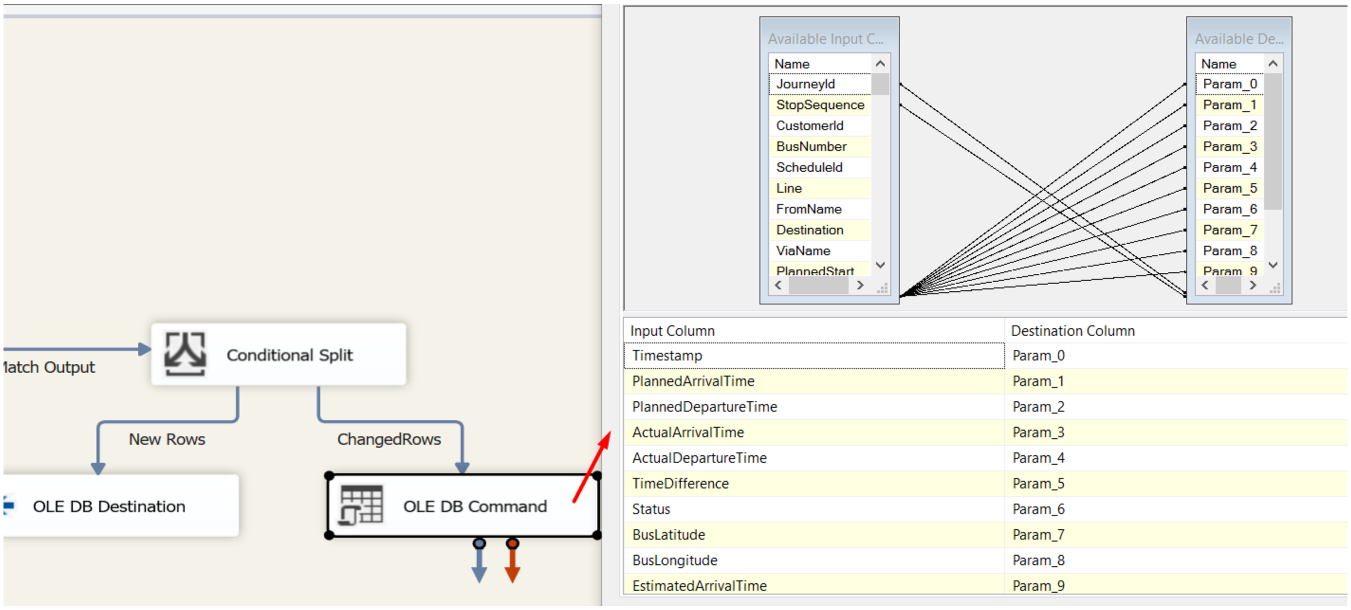
□ Keep identity □ Table lock

□ Keep nulls ☑ Check constraints

Rows per batch:

Maximum insert commit size: 2147483647

Figure 6-14: Exporting the data into prediction engine - control outflow configuration



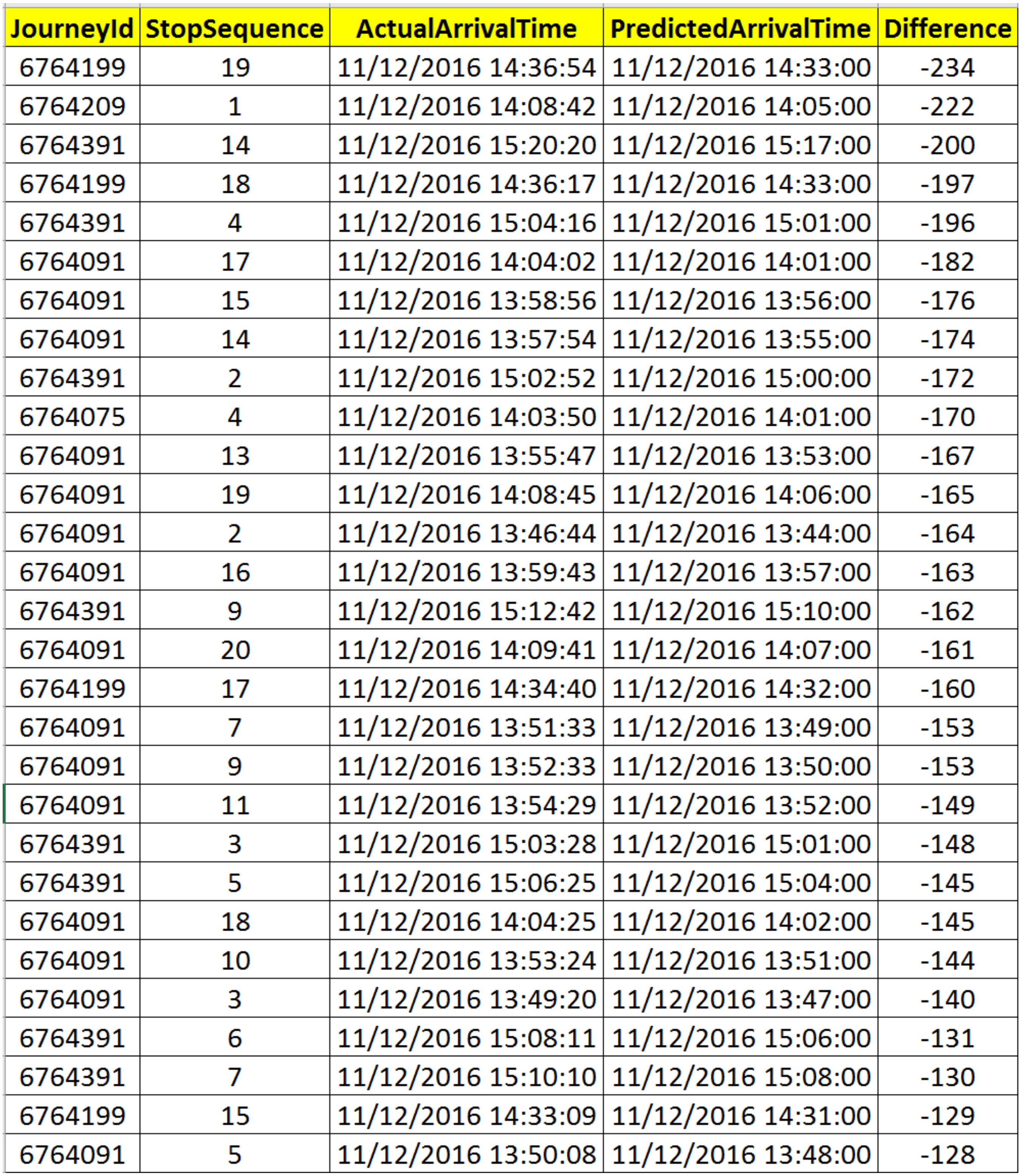
Input Column	Destination Column
Timestamp	Param_0
PlannedArrivalTime	Param_1
PlannedDepartureTime	Param_2
ActualArrivalTime	Param_3
ActualDepartureTime	Param_4
TimeDifference	Param_5
Status	Param_6
BusLatitude	Param_7
BusLongitude	Param_8
EstimatedArrivalTime	Param_9

Figure 6-15: Mapping of data attributes to dependent and independent variables in prediction engine

6.5 Comparative Analysis of the Prediction Engines

6.5.1 Third Party Prediction Engine

As indicated earlier, the existing system currently uses a 3rd party prediction engine for some routes to send departure information to the buses and the same information is shared with the devices showing incoming buses at the bus-stop. Table 6.1 shows a subset of the predictions of the arrival time, made by the 3rd party system, together with a comparison with the actual data. Although the internal details of the 3rd party system are not known and there is no information about the algorithm being used, it is still possible to review and evaluate the accuracy of the predicted arrival times and compare it with the actual arrival times. The system takes location of the buses as an input and then provides information about estimated arrival times on the next stops. The data in the table below shows the difference between the predicted time and the actual time for various stops and various trips made. The negative value indicates a late arrival of the bus compared to the predicted time, whereas a positive value shows an early arrival of the bus compared to the predicted time.



JourneyId	StopSequence	ActualArrivalTime	PredictedArrivalTime	Difference
6764199	19	11/12/2016 14:36:54	11/12/2016 14:33:00	-234
6764209	1	11/12/2016 14:08:42	11/12/2016 14:05:00	-222
6764391	14	11/12/2016 15:20:20	11/12/2016 15:17:00	-200
6764199	18	11/12/2016 14:36:17	11/12/2016 14:33:00	-197
6764391	4	11/12/2016 15:04:16	11/12/2016 15:01:00	-196
6764091	17	11/12/2016 14:04:02	11/12/2016 14:01:00	-182
6764091	15	11/12/2016 13:58:56	11/12/2016 13:56:00	-176
6764091	14	11/12/2016 13:57:54	11/12/2016 13:55:00	-174
6764391	2	11/12/2016 15:02:52	11/12/2016 15:00:00	-172
6764075	4	11/12/2016 14:03:50	11/12/2016 14:01:00	-170
6764091	13	11/12/2016 13:55:47	11/12/2016 13:53:00	-167
6764091	19	11/12/2016 14:08:45	11/12/2016 14:06:00	-165
6764091	2	11/12/2016 13:46:44	11/12/2016 13:44:00	-164
6764091	16	11/12/2016 13:59:43	11/12/2016 13:57:00	-163
6764391	9	11/12/2016 15:12:42	11/12/2016 15:10:00	-162
6764091	20	11/12/2016 14:09:41	11/12/2016 14:07:00	-161
6764199	17	11/12/2016 14:34:40	11/12/2016 14:32:00	-160
6764091	7	11/12/2016 13:51:33	11/12/2016 13:49:00	-153
6764091	9	11/12/2016 13:52:33	11/12/2016 13:50:00	-153
6764091	11	11/12/2016 13:54:29	11/12/2016 13:52:00	-149
6764391	3	11/12/2016 15:03:28	11/12/2016 15:01:00	-148
6764391	5	11/12/2016 15:06:25	11/12/2016 15:04:00	-145
6764091	18	11/12/2016 14:04:25	11/12/2016 14:02:00	-145
6764091	10	11/12/2016 13:53:24	11/12/2016 13:51:00	-144
6764091	3	11/12/2016 13:49:20	11/12/2016 13:47:00	-140
6764391	6	11/12/2016 15:08:11	11/12/2016 15:06:00	-131
6764391	7	11/12/2016 15:10:10	11/12/2016 15:08:00	-130
6764199	15	11/12/2016 14:33:09	11/12/2016 14:31:00	-129
6764091	5	11/12/2016 13:50:08	11/12/2016 13:48:00	-128

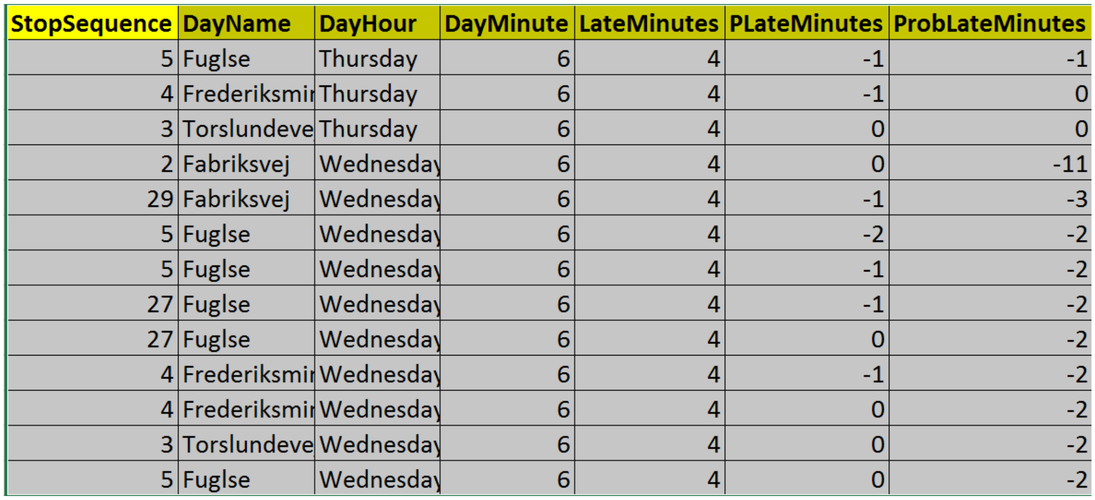
Table 6-1: accuracy of the existing prediction engine

There are many issues with the accuracy of this prediction engine. The main issue is the accuracy of the prediction. Namely, the differences between the real and the predicted values of arrival time are measured crudely, in *minutes*, which gives too crude an indication of the accuracy of the bus service is indicative of some shortcomings of the prediction engine. Also, the prediction data is not always available to integrate because it is dependent on the location of the bus, sent through sometimes unreliable communication channels.

6.5.2 Proposed Prediction Engine Characteristics and Performance

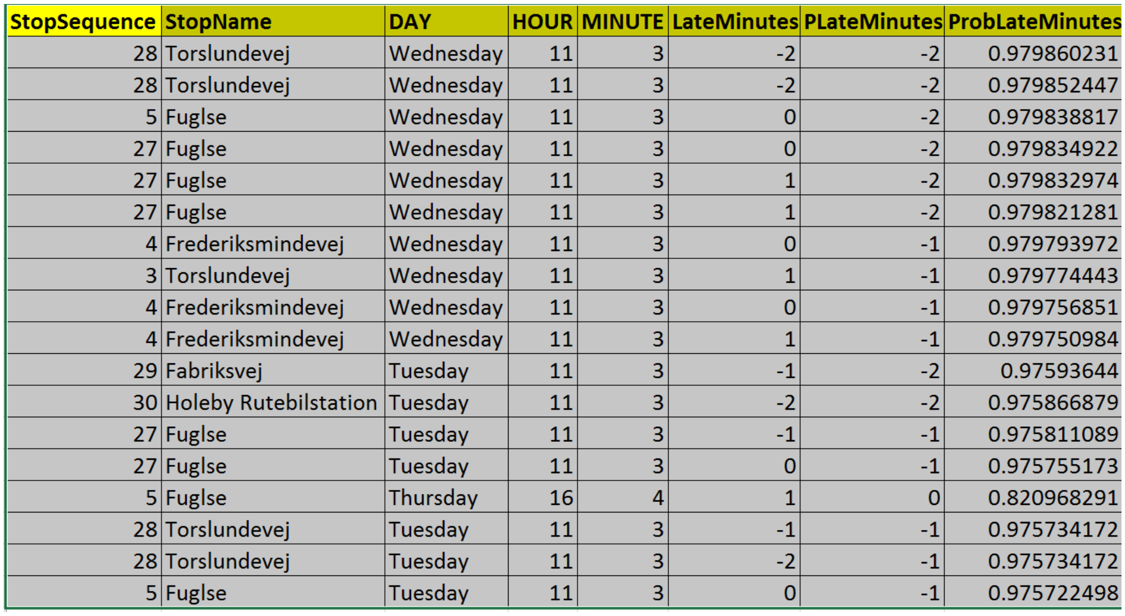
The proposed prediction engine is based on the following four standard machine learning algorithms: Clustering, Decision Trees, Neural Networks and Linear Regression. It is combined with customised data mining models and applied to the large datasets, as detailed in Chapter 5. The calculated prediction probabilities indicate that clustering and linear regression perform better, but only when the traditional algorithms are customized with pre-processing the data. While clustering is not considered an ideal technique for making predictions, the obtained results proved better than expected due to an innovative pre-processing procedure (Automated Data Acquisition) prior to feeding the data into the algorithm. On the other hand, neural networks and decision tree algorithms are considered effective in some cases, and they have been applied to this project, but linear regression still performed better than the other three, in most of the cases tested (as explained in Chapter 5, section 5.3).

Various data mining models were designed to train the algorithms to identify the best models for prediction. The data about prediction probabilities was collected for all four algorithms and stored in a separate database so that the prediction component can use this data and make real-time predictions. Tables 6.2, 6.3 and 6.4 indicate the snapshots from the generated prediction probabilities.



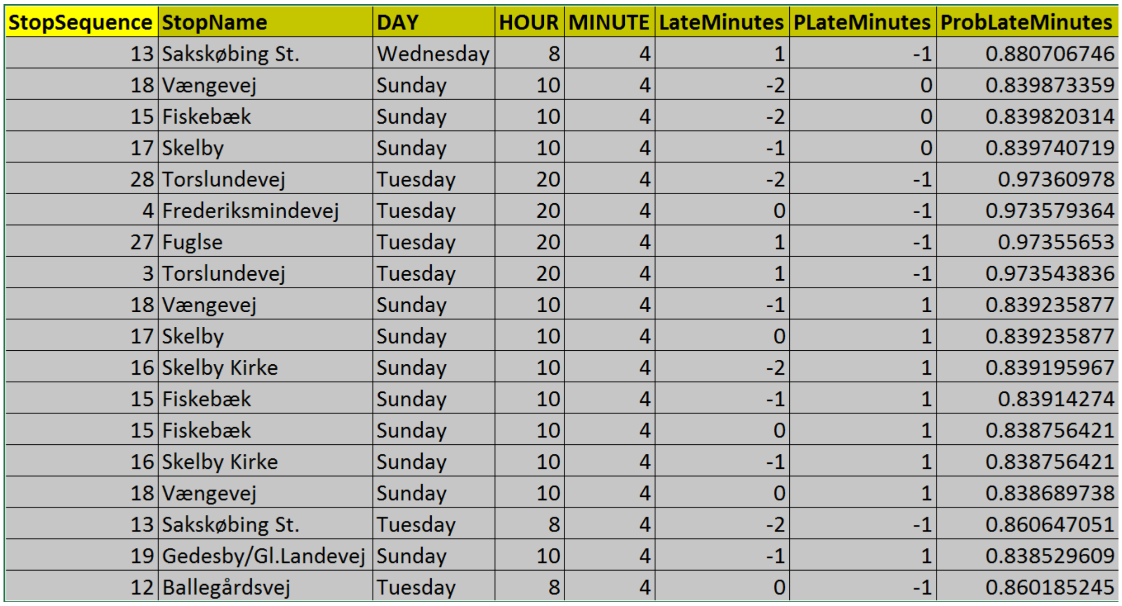
StopSequence	DayName	DayHour	DayMinute	LateMinutes	PLateMinutes	ProbLateMinutes
5	Fuglse	Thursday	6	4	-1	-1
4	Frederiksmi	Thursday	6	4	-1	0
3	Torslundeve	Thursday	6	4	0	0
2	Fabriksvej	Wednesday	6	4	0	-11
29	Fabriksvej	Wednesday	6	4	-1	-3
5	Fuglse	Wednesday	6	4	-2	-2
5	Fuglse	Wednesday	6	4	-1	-2
27	Fuglse	Wednesday	6	4	-1	-2
27	Fuglse	Wednesday	6	4	0	-2
4	Frederiksmi	Wednesday	6	4	-1	-2
4	Frederiksmi	Wednesday	6	4	0	-2
3	Torslundeve	Wednesday	6	4	0	-2
5	Fuglse	Wednesday	6	4	0	-2

Table 6-2: probabilities for latency using linear regression



StopSequence	StopName	DAY	HOUR	MINUTE	LateMinutes	PLateMinutes	ProbLateMinutes
28	Torslundevej	Wednesday	11	3	-2	-2	0.979860231
28	Torslundevej	Wednesday	11	3	-2	-2	0.979852447
5	Fuglse	Wednesday	11	3	0	-2	0.979838817
27	Fuglse	Wednesday	11	3	0	-2	0.979834922
27	Fuglse	Wednesday	11	3	1	-2	0.979832974
27	Fuglse	Wednesday	11	3	1	-2	0.979821281
4	Frederiksmindevej	Wednesday	11	3	0	-1	0.979793972
3	Torslundevej	Wednesday	11	3	1	-1	0.979774443
4	Frederiksmindevej	Wednesday	11	3	0	-1	0.979756851
4	Frederiksmindevej	Wednesday	11	3	1	-1	0.979750984
29	Fabriksvej	Tuesday	11	3	-1	-2	0.97593644
30	Holeby Rutebilstation	Tuesday	11	3	-2	-2	0.975866879
27	Fuglse	Tuesday	11	3	-1	-1	0.975811089
27	Fuglse	Tuesday	11	3	0	-1	0.975755173
5	Fuglse	Thursday	16	4	1	0	0.820968291
28	Torslundevej	Tuesday	11	3	-1	-1	0.975734172
28	Torslundevej	Tuesday	11	3	-2	-1	0.975734172
5	Fuglse	Tuesday	11	3	0	-1	0.975722498

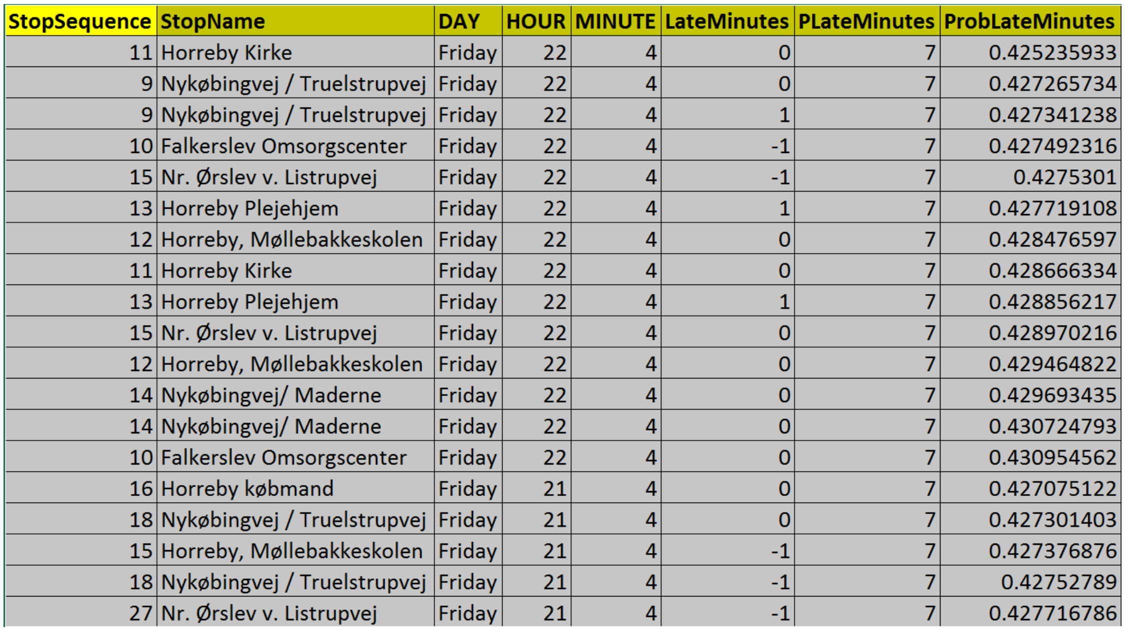
Table 6-3: probabilities for latency using linear regression



StopSequence	StopName	DAY	HOUR	MINUTE	LateMinutes	PLateMinutes	ProbLateMinutes
13	Sakskøbing St.	Wednesday	8	4	1	-1	0.880706746
18	Vængevej	Sunday	10	4	-2	0	0.839873359
15	Fiskebæk	Sunday	10	4	-2	0	0.839820314
17	Skelby	Sunday	10	4	-1	0	0.839740719
28	Torslundevej	Tuesday	20	4	-2	-1	0.97360978
4	Frederiksmindevej	Tuesday	20	4	0	-1	0.973579364
27	Fuglse	Tuesday	20	4	1	-1	0.97355653
3	Torslundevej	Tuesday	20	4	1	-1	0.973543836
18	Vængevej	Sunday	10	4	-1	1	0.839235877
17	Skelby	Sunday	10	4	0	1	0.839235877
16	Skelby Kirke	Sunday	10	4	-2	1	0.839195967
15	Fiskebæk	Sunday	10	4	-1	1	0.83914274
15	Fiskebæk	Sunday	10	4	0	1	0.838756421
16	Skelby Kirke	Sunday	10	4	-1	1	0.838756421
18	Vængevej	Sunday	10	4	0	1	0.838689738
13	Sakskøbing St.	Tuesday	8	4	-2	-1	0.860647051
19	Gedesby/Gl.Landevej	Sunday	10	4	-1	1	0.838529609
12	Ballegårdsvej	Tuesday	8	4	0	-1	0.860185245

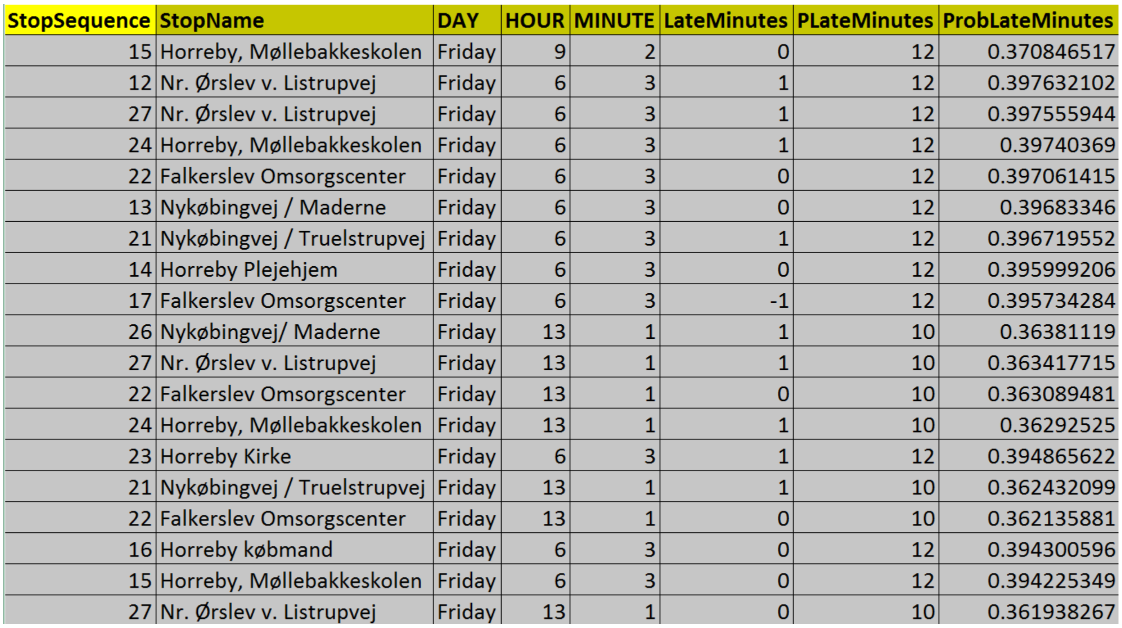
Table 6-4: probabilities for latency using linear regression

The above Tables indicate two probabilities: i) for the lateness of a bus at each stop, presented in minutes, and ii) the arrival time difference between the predicted and the actual value, presented in seconds. These two probabilities demonstrate clearly an advantage of using seconds as a unit, as they add an extra level of precision to the prediction results, as the information on the times is closer to the actual values. The prediction probabilities shown for different times of the day, as well as comparing weekdays and weekends, are very encouraging showing 97% congruence with the actual times, in most of the cases. The instances where the discrepancy is higher can be attributed to insufficient data available to train the prediction algorithm, for the given scenario.



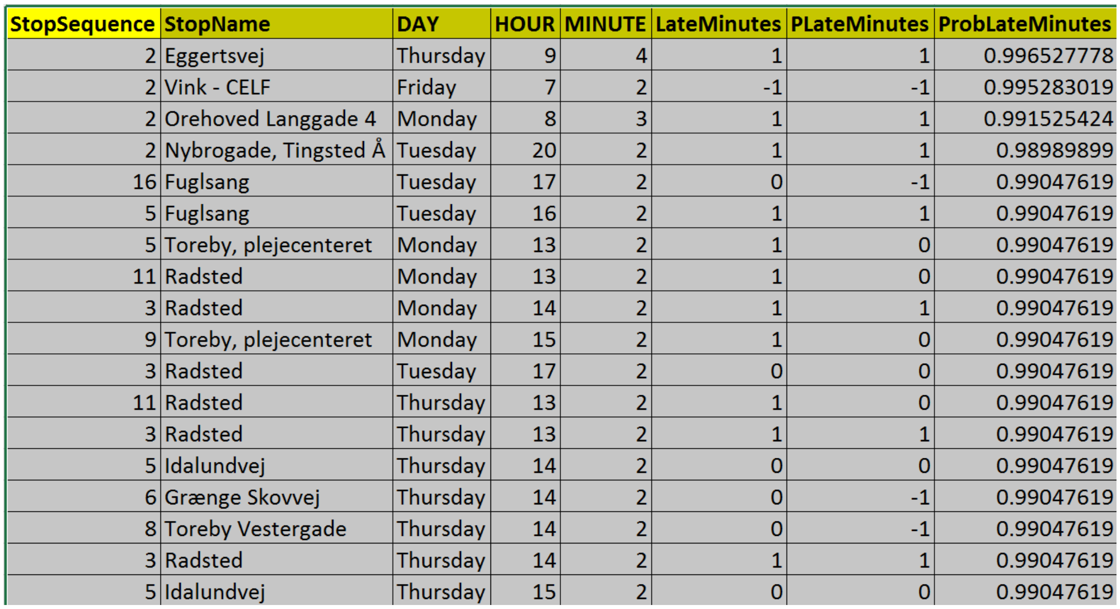
StopSequence	StopName	DAY	HOUR	MINUTE	LateMinutes	PLateMinutes	ProbLateMinutes
11	Horreby Kirke	Friday	22	4	0	7	0.425235933
9	Nykøbingvej / Truelstrupvej	Friday	22	4	0	7	0.427265734
9	Nykøbingvej / Truelstrupvej	Friday	22	4	1	7	0.427341238
10	Falkerslev Omsorgscenter	Friday	22	4	-1	7	0.427492316
15	Nr. Ørslev v. Listrupvej	Friday	22	4	-1	7	0.4275301
13	Horreby Plejehjem	Friday	22	4	1	7	0.427719108
12	Horreby, Møllebakkeskolen	Friday	22	4	0	7	0.428476597
11	Horreby Kirke	Friday	22	4	0	7	0.428666334
13	Horreby Plejehjem	Friday	22	4	1	7	0.428856217
15	Nr. Ørslev v. Listrupvej	Friday	22	4	0	7	0.428970216
12	Horreby, Møllebakkeskolen	Friday	22	4	0	7	0.429464822
14	Nykøbingvej/ Maderne	Friday	22	4	0	7	0.429693435
14	Nykøbingvej/ Maderne	Friday	22	4	0	7	0.430724793
10	Falkerslev Omsorgscenter	Friday	22	4	0	7	0.430954562
16	Horreby købmand	Friday	21	4	0	7	0.427075122
18	Nykøbingvej / Truelstrupvej	Friday	21	4	0	7	0.427301403
15	Horreby, Møllebakkeskolen	Friday	21	4	-1	7	0.427376876
18	Nykøbingvej / Truelstrupvej	Friday	21	4	-1	7	0.42752789
27	Nr. Ørslev v. Listrupvej	Friday	21	4	-1	7	0.427716786

Table 6-5: probabilities for 11 AM showing probabilities for latency using neural network



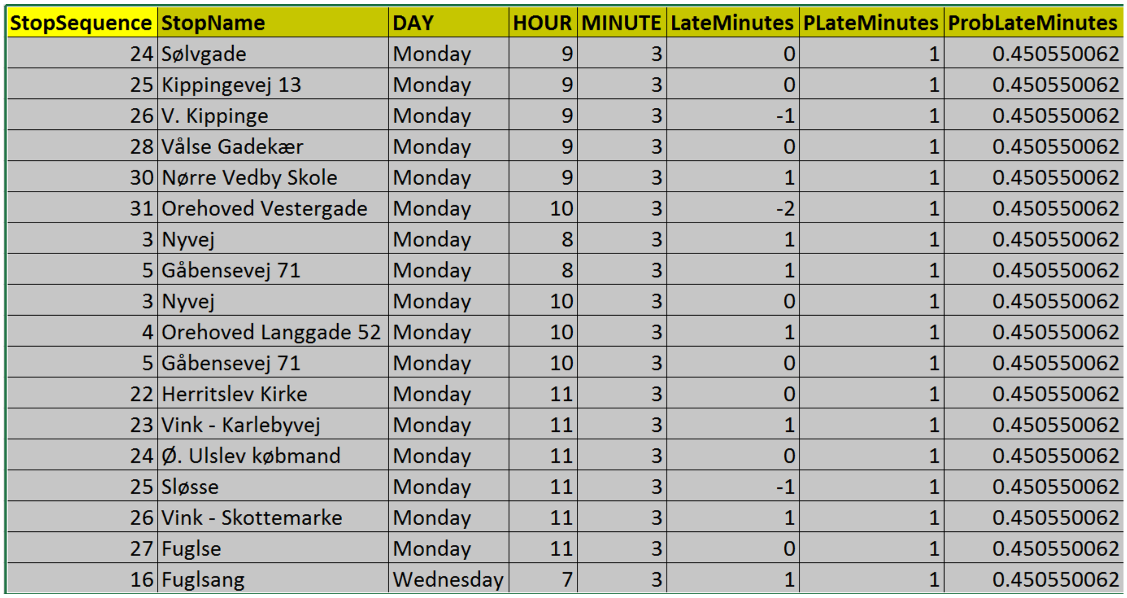
StopSequence	StopName	DAY	HOUR	MINUTE	LateMinutes	PLateMinutes	ProbLateMinutes
15	Horreby, Møllebakkeskolen	Friday	9	2	0	12	0.370846517
12	Nr. Ørslev v. Listrupvej	Friday	6	3	1	12	0.397632102
27	Nr. Ørslev v. Listrupvej	Friday	6	3	1	12	0.397555944
24	Horreby, Møllebakkeskolen	Friday	6	3	1	12	0.39740369
22	Falkerslev Omsorgscenter	Friday	6	3	0	12	0.397061415
13	Nykøbingvej / Maderne	Friday	6	3	0	12	0.39683346
21	Nykøbingvej / Truelstrupvej	Friday	6	3	1	12	0.396719552
14	Horreby Plejehjem	Friday	6	3	0	12	0.395999206
17	Falkerslev Omsorgscenter	Friday	6	3	-1	12	0.395734284
26	Nykøbingvej/ Maderne	Friday	13	1	1	10	0.36381119
27	Nr. Ørslev v. Listrupvej	Friday	13	1	1	10	0.363417715
22	Falkerslev Omsorgscenter	Friday	13	1	0	10	0.363089481
24	Horreby, Møllebakkeskolen	Friday	13	1	1	10	0.36292525
23	Horreby Kirke	Friday	6	3	1	12	0.394865622
21	Nykøbingvej / Truelstrupvej	Friday	13	1	1	10	0.362432099
22	Falkerslev Omsorgscenter	Friday	13	1	0	10	0.362135881
16	Horreby købmand	Friday	6	3	0	12	0.394300596
15	Horreby, Møllebakkeskolen	Friday	6	3	0	12	0.394225349
27	Nr. Ørslev v. Listrupvej	Friday	13	1	0	10	0.361938267

Table 6-6: probabilities for latency using neural network



StopSequence	StopName	DAY	HOUR	MINUTE	LateMinutes	PLateMinutes	ProbLateMinutes
2	Eggertsvej	Thursday	9	4	1	1	0.996527778
2	Vink - CELF	Friday	7	2	-1	-1	0.995283019
2	Orehoved Langgade 4	Monday	8	3	1	1	0.991525424
2	Nybrogade, Tingsted Å	Tuesday	20	2	1	1	0.98989899
16	Fuglsang	Tuesday	17	2	0	-1	0.99047619
5	Fuglsang	Tuesday	16	2	1	1	0.99047619
5	Toreby, plejecenteret	Monday	13	2	1	0	0.99047619
11	Radsted	Monday	13	2	1	0	0.99047619
3	Radsted	Monday	14	2	1	1	0.99047619
9	Toreby, plejecenteret	Monday	15	2	1	0	0.99047619
3	Radsted	Tuesday	17	2	0	0	0.99047619
11	Radsted	Thursday	13	2	1	0	0.99047619
3	Radsted	Thursday	13	2	1	1	0.99047619
5	Idalundvej	Thursday	14	2	0	0	0.99047619
6	Grænge Skovvej	Thursday	14	2	0	-1	0.99047619
8	Toreby Vestergade	Thursday	14	2	0	-1	0.99047619
3	Radsted	Thursday	14	2	1	1	0.99047619
5	Idalundvej	Thursday	15	2	0	0	0.99047619

Table 6-7: probabilities for latency using decision trees



StopSequence	StopName	DAY	HOUR	MINUTE	LateMinutes	PLateMinutes	ProbLateMinutes
24	Sølvgade	Monday	9	3	0	1	0.450550062
25	Kippingevej 13	Monday	9	3	0	1	0.450550062
26	V. Kippinge	Monday	9	3	-1	1	0.450550062
28	Vålse Gadekær	Monday	9	3	0	1	0.450550062
30	Nørre Vedby Skole	Monday	9	3	1	1	0.450550062
31	Orehoved Vestergade	Monday	10	3	-2	1	0.450550062
3	Nyvej	Monday	8	3	1	1	0.450550062
5	Gåbensevej 71	Monday	8	3	1	1	0.450550062
3	Nyvej	Monday	10	3	0	1	0.450550062
4	Orehoved Langgade 52	Monday	10	3	1	1	0.450550062
5	Gåbensevej 71	Monday	10	3	0	1	0.450550062
22	Herritslev Kirke	Monday	11	3	0	1	0.450550062
23	Vink - Karlebyvej	Monday	11	3	1	1	0.450550062
24	Ø. Ulslev købmand	Monday	11	3	0	1	0.450550062
25	Sløsse	Monday	11	3	-1	1	0.450550062
26	Vink - Skottemarke	Monday	11	3	1	1	0.450550062
27	Fuglse	Monday	11	3	0	1	0.450550062
16	Fuglsang	Wednesday	7	3	1	1	0.450550062

Table 6-8: probabilities for latency using clustering technique

6.5.3 Summary

As shown in Tables 6.6, 6.7 and 6.8, the clustering technique, while promising due to grouping the data points – gives poor predictability in most of the cases. Table 6.8 shows around 55% discrepancy, or less than 50% accuracy. This inaccuracy is due to the nature of the algorithm, which is not designed for predicting continuous values such as bus delays, or congestions. These results provide evidence that the regression algorithm with customized mining models and varied number of parameter lists can produce a high level of accuracy for arrival times and the delay information as shown in Figures 6.3 and 6.4

Prediction probabilities obtained using Neural Networks, Decision Trees and Clustering are shown in Tables 6.5 to 6.8., which demonstrate what was already concluded – that the accuracy of the Linear Regression method is higher than both the Neural Networks and the Decision Tree algorithms.

6.6 Real-Time Monitoring Using Prediction Engine

After the real-time data acquisition process and the application of comprehensive data mining algorithms, this information was employed to monitor and manage the transport network in real-time. It constitutes one of the key contributions of this thesis to knowledge and application of established methods to a new, complex environment.

Many of the challenges addressed by the proposed intelligent transport system architecture, such as the real-time system monitoring, an informed congestion anticipation, evaluation and propagation of delays, and dynamic re-routing of buses will be presented in this section.

The development of data visualisation techniques and strategies suited to the real-time monitoring of a constantly changing complex system constitutes a significant portion of this phase of the project. The results presented in this chapter are based on an integrated process of data acquisition, data analysis in real-time - and refreshing of the interface according to the correlation between the prediction and the real data. In addition, the aim of this phase was to facilitate real-time transport management such as operators, traffic managers, bus companies, traffic authorities and planning department through appropriate visualisation techniques ensuring clarity of information through innovative ways of presenting complex data in real time. These end users would use this presentation of information for strategic decision making and operational performance of the transport system. The output from the prediction engine is directly inputted into suitable visualization frameworks to facilitate strategic decision-making.

Thus, the following section will present different aspects of system monitoring based on real-time visualisation - with each subsection addressing the problems encountered and solved by a particular visualisation strategy. Some standard means of data visualization, such as diagrams and graphs, to identify and analyse patterns in the existing data have already been presented in Chapter 5. In all the visualization scenarios three colours (red, green and blue) have been used to indicate the status of the bus operation and the routes they are taking at any one time. The red colour indicates a delay, the green colour indicates on-time arrival and blue colour indicates an early arrival. The pie chart is used to indicate the ratio of the trips made on the route and arrival events reported by buses on the route. With each arrival event, a point on the map is drawn so that a particular area, or zone, can be highlighted for each point in time and space.

6.6.1 System Visualisation

As shown earlier in Figures 6.3 and 6.4, presentation of the data for only 30 routes creates a complex visualisation map, which does not help, in clear understanding of the situation of buses and the routes they are. Therefore, the implemented monitoring system creates visualisations for different levels such as system, zone, route so that information can be looked at in detail. Figure 6.16 presents an updated status of all the routes and is continuously updating in real-time. The strength of the colour indicates the amount of delay being reported with light colour showing less or no delay and dark colour showing more than normal delay being reported. It also indicates relationship between different routes in terms of delays being reported.

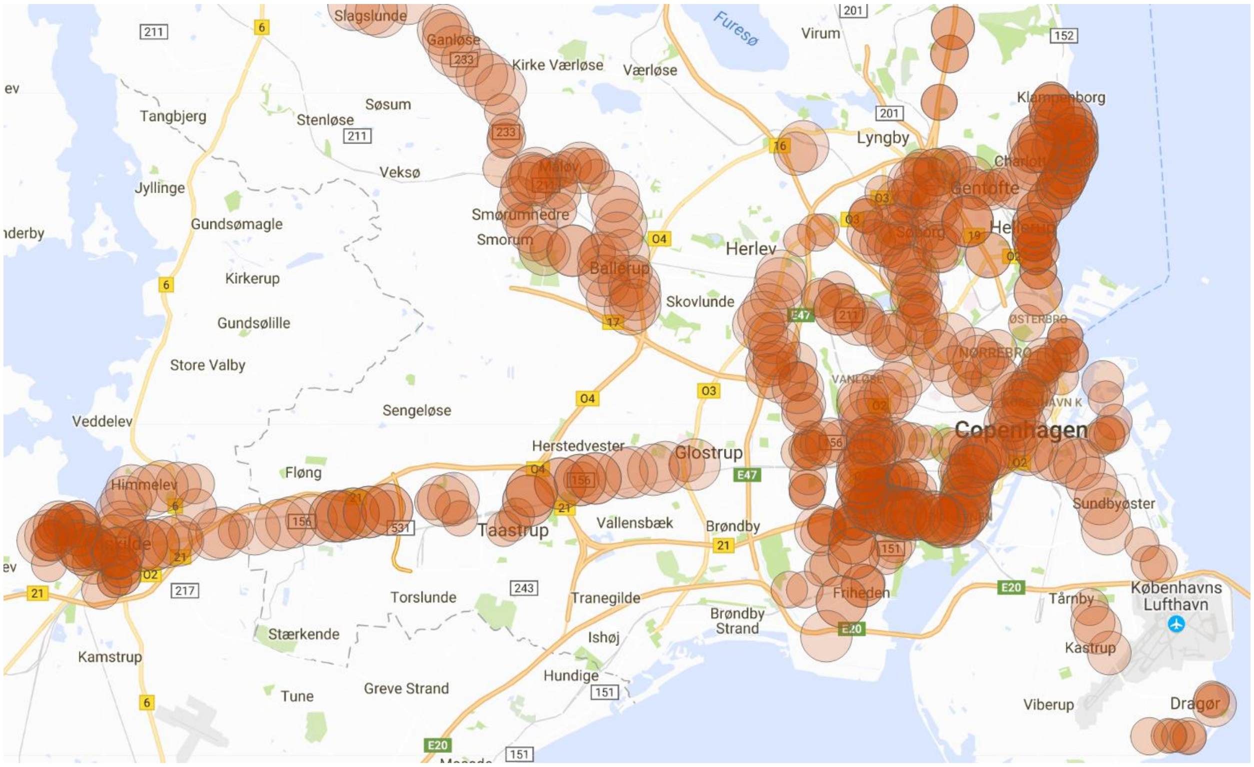


Figure 6-16: Current operational situation of all routes

Once the overall perspective of the routes is accessed, the visualisation shown in Figure 6.17 presents an option to watch a specific area closely. The data about three routes going to the same destination is shown with start and end of the route shown in detail as marked by red and green boxes. Although there are many stops where some delays are reported but the red colour of buses is an indication that the current situation on those stops and the roads covering those stops are causing delays for the buses driving those routes. The buses that are in the middle of these routes are shown in green colour but the buses to the end of the route specifically are red, which means that buses coming from all three routes with the same destination are getting late and more information should be looked to clarify if there is any congestion building up. This can be an indication of a starting congestion. We have different filters to apply to this data so that growth of current status can be seen as it developed with the help of buses as well as time selection. The time selection offers a range of options from past 10 minutes to 7 weeks so that both the remote and recent history can be investigated to see if there is a pattern that causes this situation.

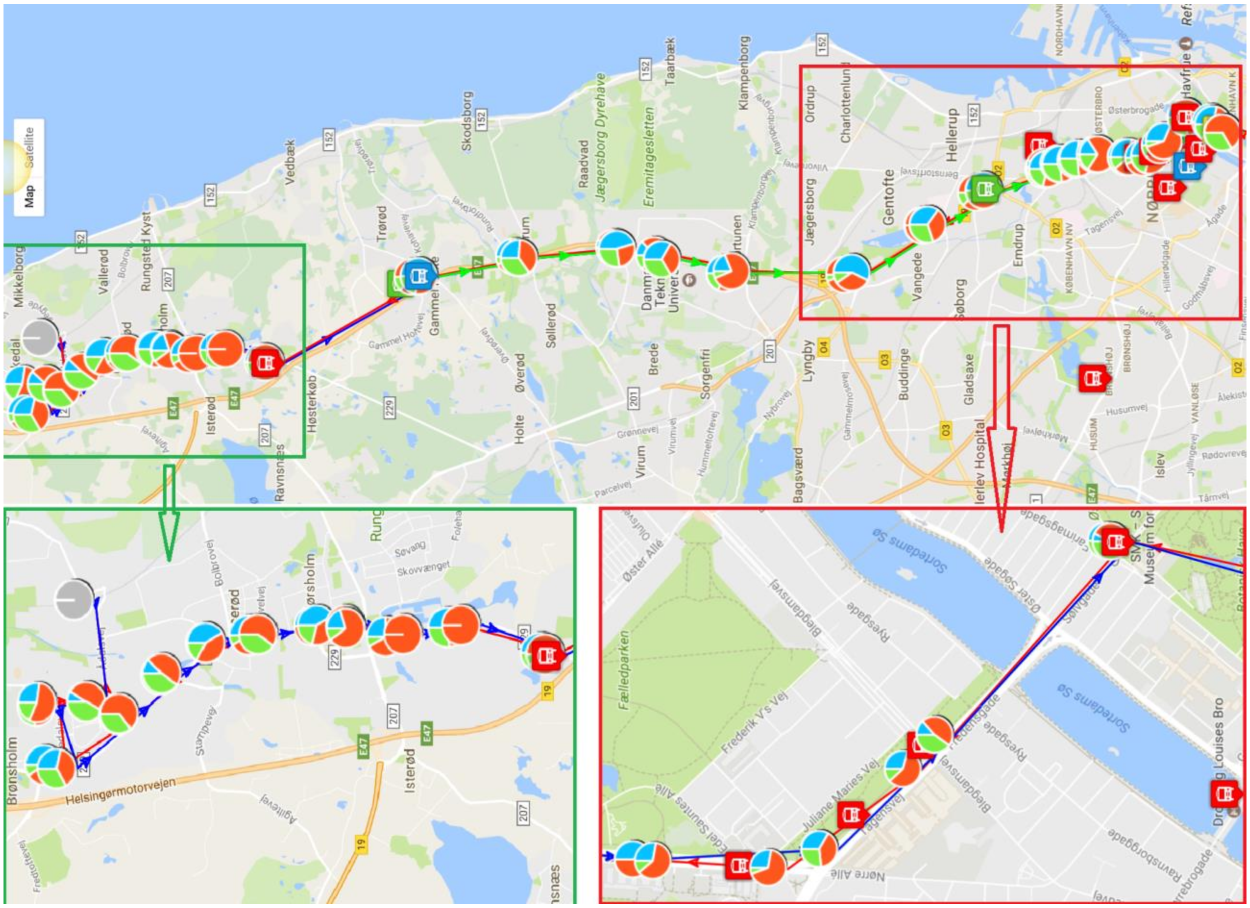


Figure 6-17: Real-time stops and buses for 3 routes going to same destination

The Figure 6.18 presents the data for six routes that are not close to each other to a level that they can impact or transfer the delay from one route to another. The colors in three main areas of the visualization indicate a completely different aspect of the system. The area marked green had most of the stops indicated with major percentage of red but the color of the bus shows that the traffic has started getting better and the traffic will improve. Whereas, the areas marked red are showing that most of the buses in that area are still delaying and only one of the areas has traffic situation getting better on one side of the route. This helps to drill down the monitoring effort on the areas marked red so that decision can be made about more buses going to the same route or area. Figures 6.19 and 6.20 present route and time wise situation for all the routes where number of routes are indicated by the number of circles, size of the circle represents view for average delay for each route and hour.

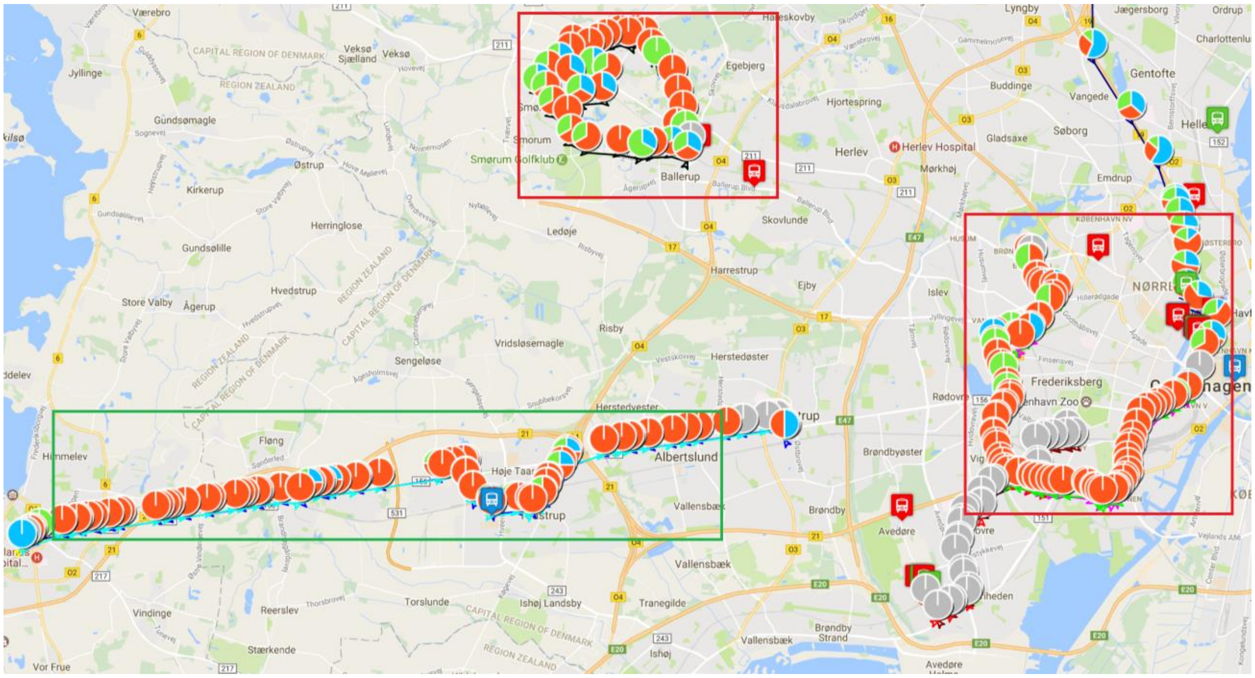


Figure 6-18: 6 routes from different zones

Figure 6.19 presents an overall perspective of the system displaying average delay plotted for each route with respect to the hour of the day it was driven in. It shows that from 7 to 10 AM in the morning and 3 to 7 PM in the evening, the routes are showing higher delays and that is indicated by the size of circle. The interesting part of this visualization is that it shows information about all the routes in one place, which is not possible to look at on the map because each of these circles contains on the average 35 stops and GPS coordinates of the stops make them widely spread on the map. We can look at this information from the hour and routes, zone and routes as presented in Figure 6.20, destination and routes or any other dimension of the data and we can learn about operational situation of the buses and routes depending on the priority of the transport service being provided in that area. This information is available as APIs as well so that it can be shared with internal and external systems like other transport companies operating in the same area and the traffic management authorities so that it helps them plan the routes and required buses on each route to serve the commuters with minimum delays and higher customer satisfaction.

9
8
19
1
1
19
15

Figure 6-19: All 472 routes with size of circle indicating average delay. Arrangement of route circles is random

Figure 6.20 splits the whole network of routes and stops into zones around the city. It clearly indicates which zone is projecting more delays and which routes in a specific zone are facing higher delays. There is virtually no limit on the number of dimensions we can use to look at the data because all different attributes have been converted to dimensions and each dimension can be combined with data plotted for other attributes for relationship based visualization. The selected routes from this display can then be monitored in real-time by visuals such as Figure 6.17 and 6.18.

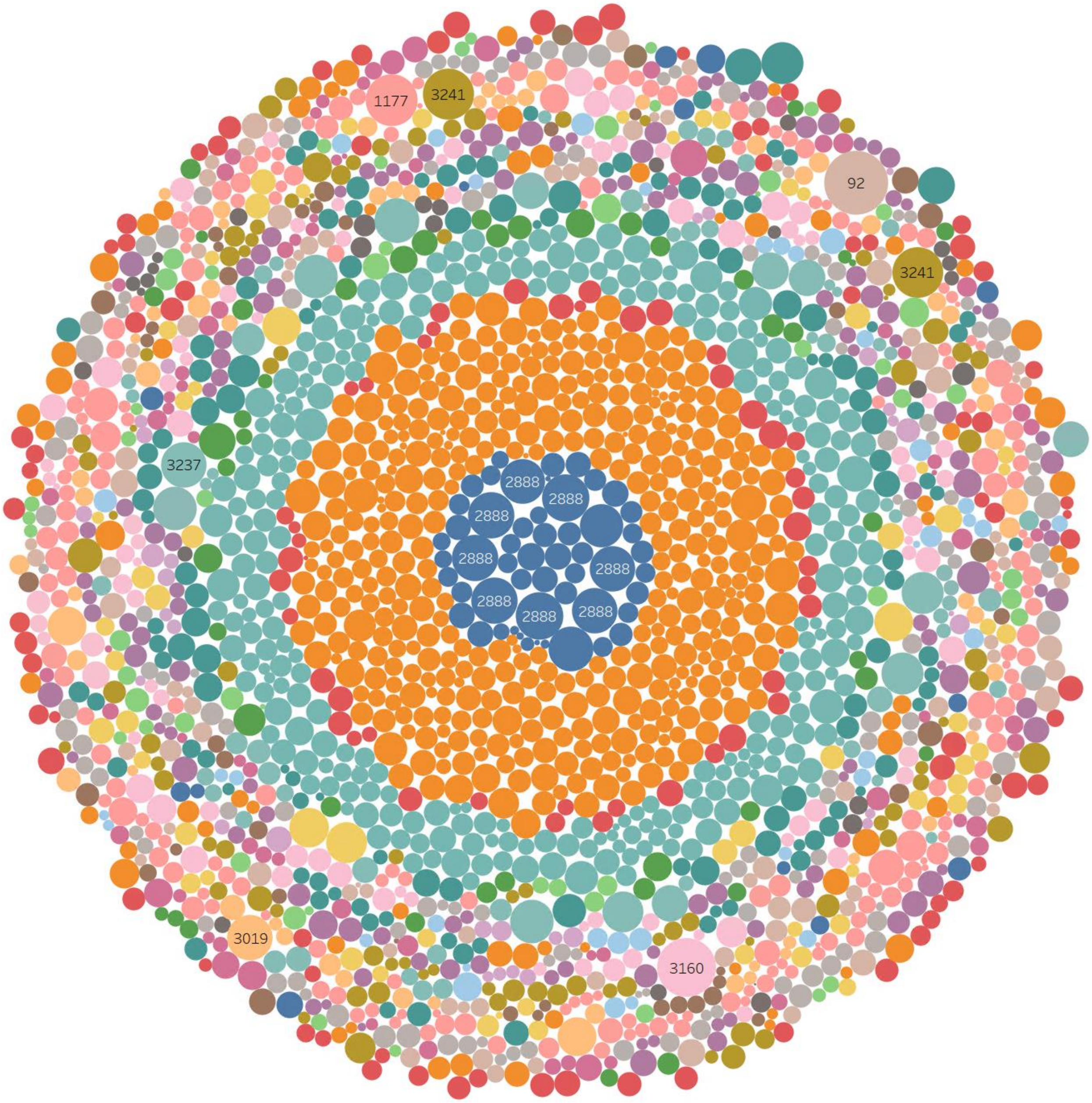


Figure 6-20: indication of zonewise routes causing the delays

6.6.2 Congestion Prediction and Delay Propagation

The visualization in Figure 6.21 shows potential situation of a route in particular area in future time (10 minutes selected for this route). The considerable delay is set to two minutes, which can be changed and makes the system configurable so that tolerance threshold of delay can be added and false alarms of a bus being delay can be avoided. Each node shows a bus number with a probability of delay and the amount of delay expected on the next arrival. The red icon shows the bus number with delay in minutes.

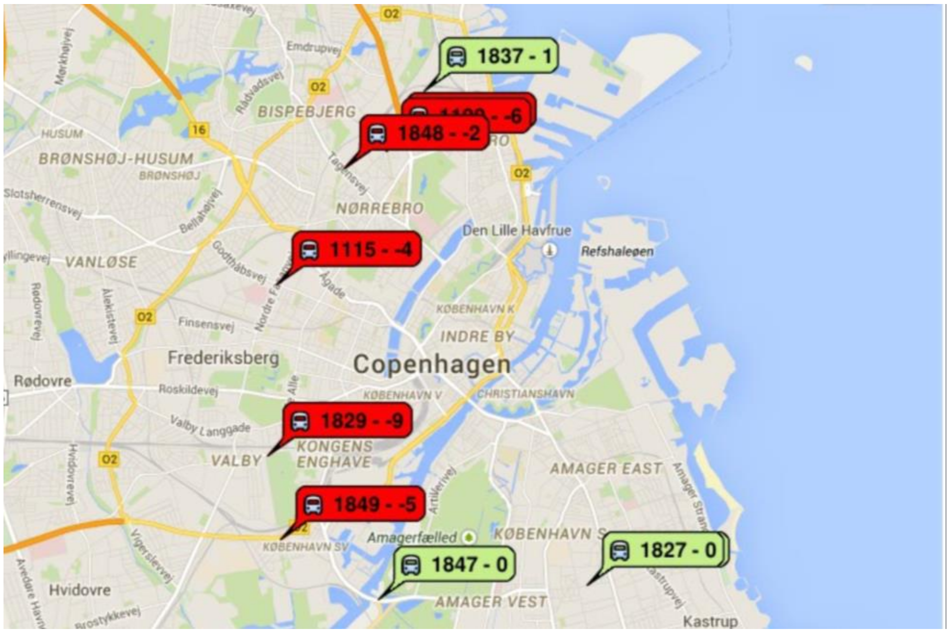


Figure 6-21 Congestion indication in future

There is a threshold value of 1 minute in this case that delay is considered only when it is more than one minute. This threshold value can be used to control the impact factor of different incidents. For example, if it is known that on a specific day or time, the buses will start 3 minutes late from the first stop, the threshold value can be set to three minutes. This can help save lot of administration hours because the system will indicate delay only if it is more than threshold time. The delay propagated across different routes is shown in the Figure 6.22. This presents an extended visualisation of the predicted delay in next 10 minutes based on history and current progress being made on 20 routes in a zone. It indicates specific areas where congestion is expected in next ten minutes and how many routes will be impacted with that congestion. In addition, it shows that after how many stops the congestion is expected to ease out to make alternate route selections. Figure 6.23 presents a long-term projection of delay on two routes.

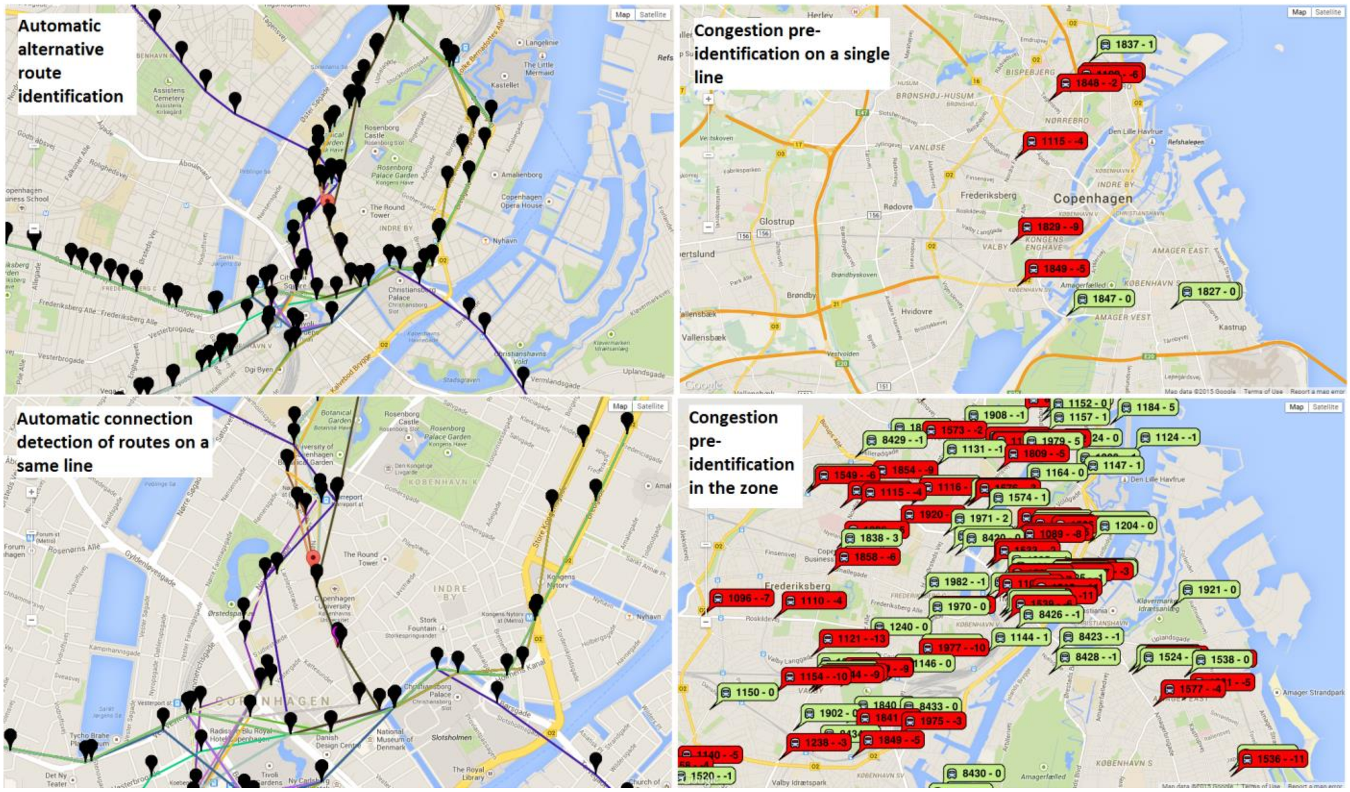

Automatic alternative route identification
Congestion pre-identification on a single line
Automatic connection detection of routes on a same line
Congestion pre-identification in the zone

Figure 6-22 delay propagation at zone level

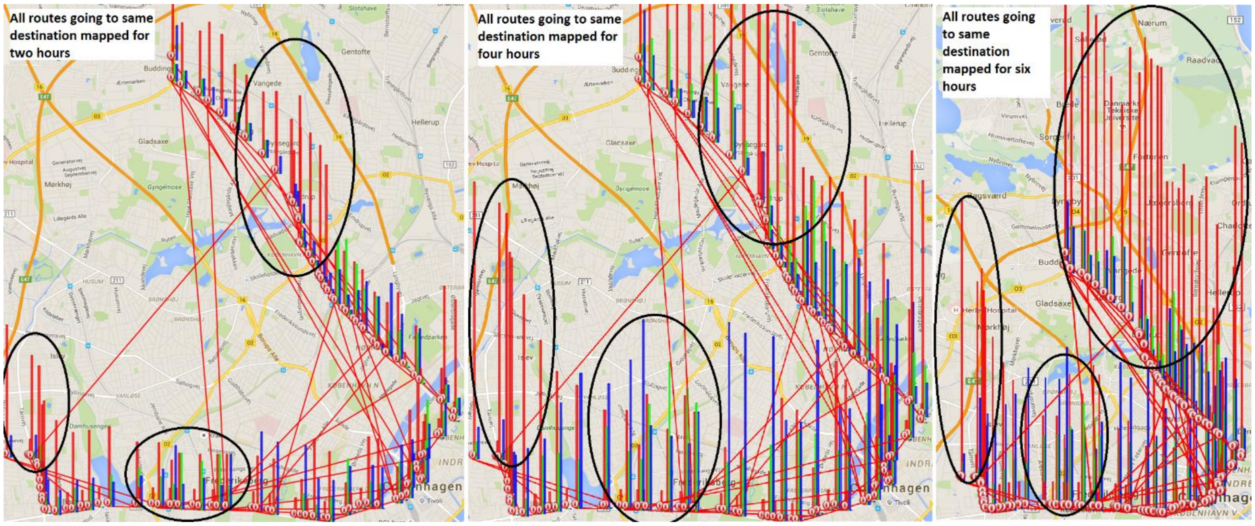

All routes going to same destination mapped for two hours
All routes going to same destination mapped for four hours
All routes going to same destination mapped for six hours

Figure 6-23: History based long term future projection of two routes

6.6.3 Rerouting for Buses

As mentioned in Chapter 4 for challenges faced by ITS, the buses having specific routes to travel have limited options to choose an alternate route when there is congestion or an accident on the route. The main reason is that buses cannot use all the road like small or private vehicles because of their size and the road infrastructure requirement to ensure safety of the bus and the commuters on-board. We have solved this problem by automatically identifying congestion that can delay a bus and propose one or more alternate routes to the driver so that either the driver can decide or the operator at the bus company can advise him to take a different route to avoid being stuck in the traffic. As shown in Figure 6.24, an area is marked where traffic is building up and the percentage of delays being reported indicate that the traffic is slowing down or going into a traffic jam. The system proposes in real-time the alternate routes that are used by buses already and are currently not facing any heavy traffic. This gives a chance to the driver or the bus company to make a decision on how many stops can be or should be skipped to connect back to the same route so that the commuters on-board can be notified and the same information can be shared with different screens and services dispensing traffic information on the stops. This is achieved through continuous monitoring of the route a bus is driving and any linked routes that are connected with this route and can potentially be an alternate. The common areas of the routes are identified and connection points are created so that as soon as a bus is facing congestion and has the option to take a different route from a common stop, the system makes this information available in the system that can be displayed as needed. One such example of rerouting visualisation is implemented in the system as presented in Figure 6.24. Each point in the shown visualisation has information about how many trips have been made to that point and the colour of buses indicates how is the situation on road impacting their current journey. The connected routes with the current route being monitored can be selected to be shown in the system to see impact of a route on the routes connected. Like in this Figure 6.24, the connected routes indicate that there are no delays being reported and the routes can be used as alternate.

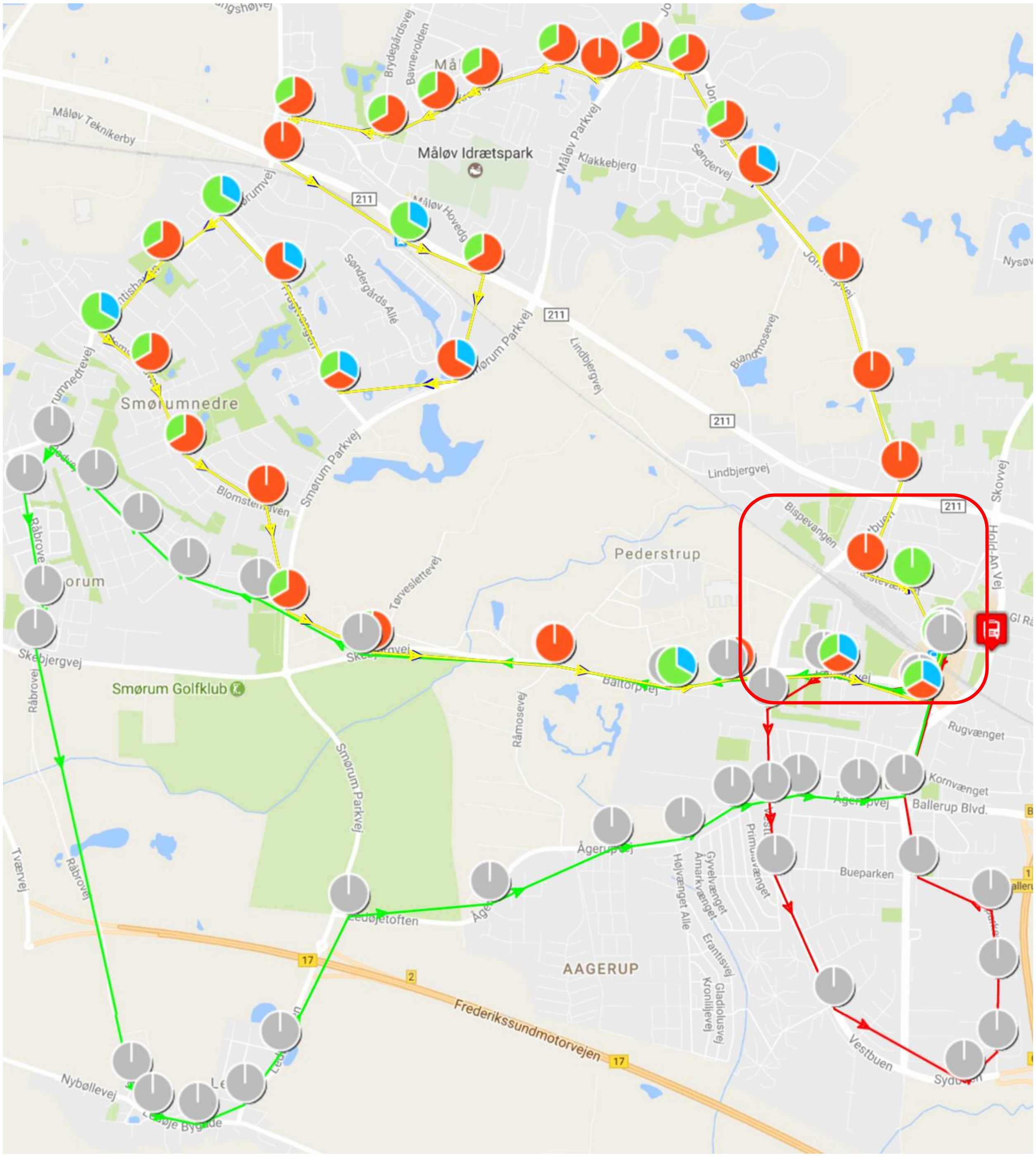


Figure 6-24: Buses rerouting

6.7 Offline Prediction engine

As elaborated in Chapter 3 in review of existing prediction engines and then in Chapter 5 where a new framework for predictive analytics evaluation was presented, the majority of the prediction engines need multi-tier infrastructure where server and client tiers are involved in the process of building and using prediction based models. The implementation and automation of a real-time predictive analytics architecture in this thesis has extended the use of implemented prediction engine to scenarios where there is no server side available and the bus does not have any connection to server side infrastructure. For example, a bus loses the connection to the server while driving on a journey and has no access to the information it needs to inform the bus passengers about when the bus is expected to reach on the stop to come. There is another scenario where the location of bus is being reported wrong because of a faulty GPS and that means the prediction engine, even when it is working and available to the bus from server side, is not of much use because it will rely on location of the bus to make predictions for the next stops. It is not only about the buses, but it is also about the devices at bus stops displaying arrivals and departures in future. These devices will not be able to give this information about arrivals and departures if they have no connection to the server from where the fetch the information to be displayed.

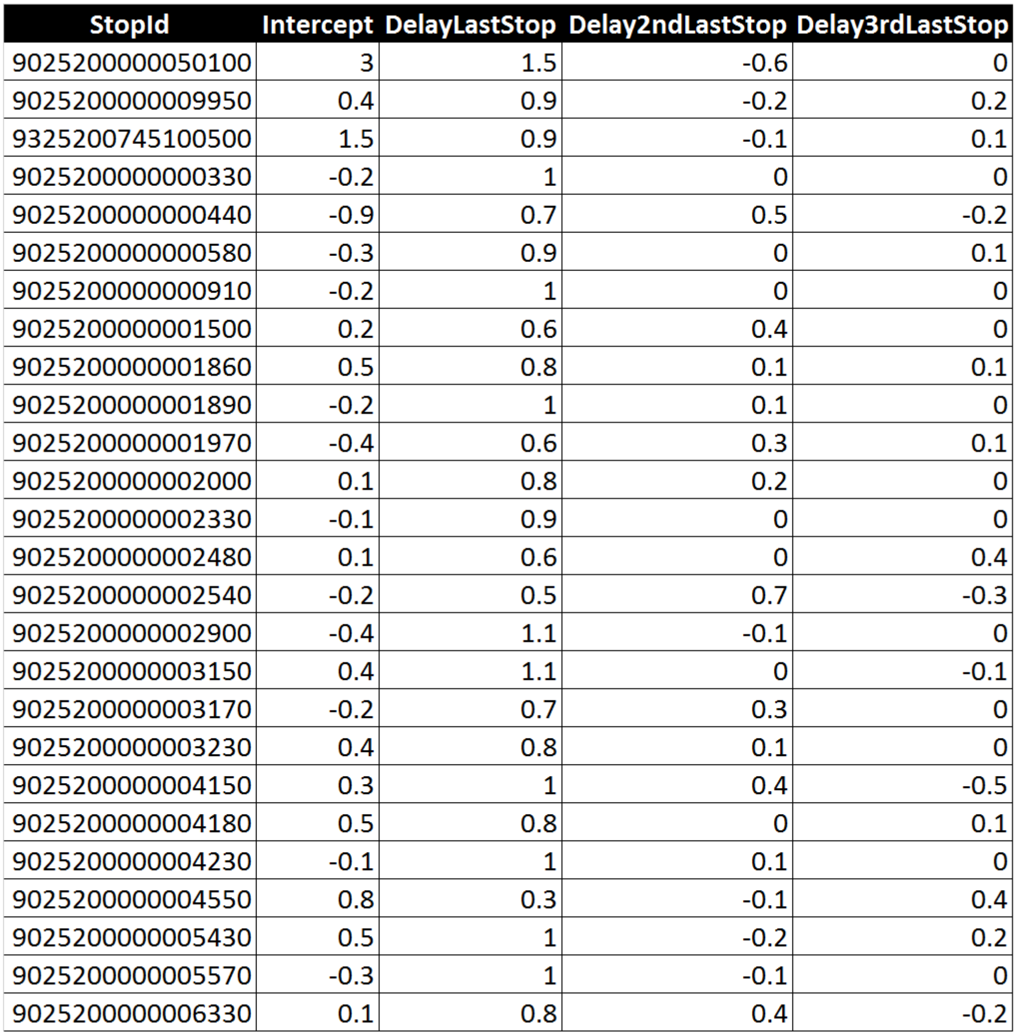
6.7.1 Prediction Coefficients

One of the key information produced by regression based data mining techniques is that the output contains information about coefficients for each input attribute. The coefficients are like weights assigned to each input attribute so that these weights can be used to build an equation that can be used for making prediction about a specific instance of the journey. Table 6.9 presents a list for some of the stops and their coefficients produced from automated application of data mining techniques that make prediction of delays based on delay reported on three previous stops. The example equation that can be derived from these coefficients is:

Delay-at-Stop = Intercept + DelayLastStop * DelaySecondsLastStop + Delay2ndLastStop*DelaySeconds2ndLastStop + Delay3rdLastStop*DelaySeconds3rdLastStop

$$Delay(9025200000050100) = 3 + 1.5 * 200 + (-0.6)*180 + 0*50$$

Similarly, delay for any stop can be calculated at any time with these coefficients that are upgraded after every iteration of data mining techniques.



StopId	Intercept	DelayLastStop	Delay2ndLastStop	Delay3rdLastStop
9025200000050100	3	1.5	-0.6	0
9025200000009950	0.4	0.9	-0.2	0.2
9325200745100500	1.5	0.9	-0.1	0.1
9025200000000330	-0.2	1	0	0
9025200000000440	-0.9	0.7	0.5	-0.2
9025200000000580	-0.3	0.9	0	0.1
9025200000000910	-0.2	1	0	0
9025200000001500	0.2	0.6	0.4	0
9025200000001860	0.5	0.8	0.1	0.1
9025200000001890	-0.2	1	0.1	0
9025200000001970	-0.4	0.6	0.3	0.1
9025200000002000	0.1	0.8	0.2	0
9025200000002330	-0.1	0.9	0	0
9025200000002480	0.1	0.6	0	0.4
9025200000002540	-0.2	0.5	0.7	-0.3
9025200000002900	-0.4	1.1	-0.1	0
9025200000003150	0.4	1.1	0	-0.1
9025200000003170	-0.2	0.7	0.3	0
9025200000003230	0.4	0.8	0.1	0
9025200000004150	0.3	1	0.4	-0.5
9025200000004180	0.5	0.8	0	0.1
9025200000004230	-0.1	1	0.1	0
9025200000004550	0.8	0.3	-0.1	0.4
9025200000005430	0.5	1	-0.2	0.2
9025200000005570	-0.3	1	-0.1	0
9025200000006330	0.1	0.8	0.4	-0.2

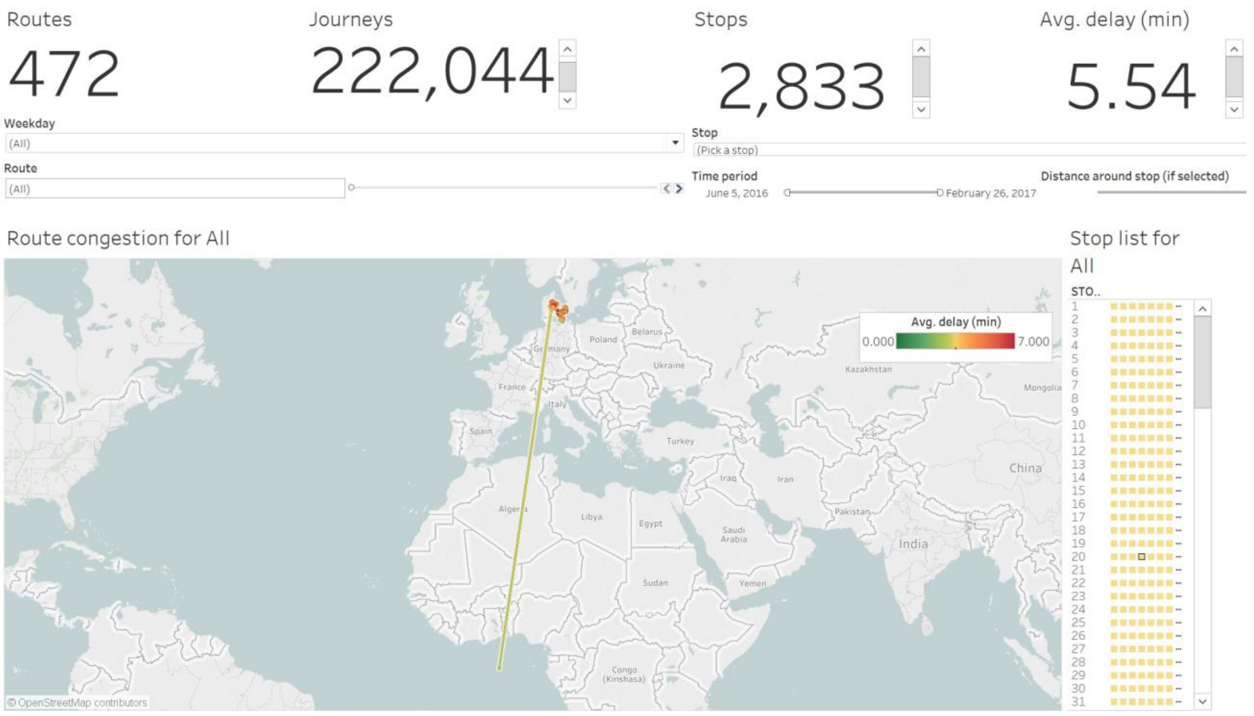
Table 6-9: Prediction coefficients for stops and delays at previous stops

6.7.2 Prediction in buses on-board

As the buses are connected to the server all the time, the prediction component installed on the bus is interacting with predictive analytics system on the server side. In addition to the data downloaded for arrivals and departures from the server, the buses also have access to the prediction coefficients for the stops of the route and the stops of linked routes. There can be a configurable way to sync with updates of these coefficients so that the buses or any devices and services fetching this information always have latest values. This helps the buses and devices to generate localised prediction even when there is no server side available and the predictions will still be close to accurate except urgent and emergency situations where bus is not aware of the incident.

6.8 Predictive Monitoring – Management Perspective

Figure 6.25 presents statistics about the dataset being used in the case study where 472 routes were selected and we have 222, 044 journeys completed over 2,833 bus stops recording 34.3 million bus events on the road. The average delay gives average of all the journeys completed, which is not very useful from overall system perspective but it does give indication about the potential of delays being faced by the buses.



Routes 472

Journeys 222,044

Stops 2,833

Avg. delay (min) 5.54

Weekday (All)

Route (All)

Stop (Pick a stop)

Time period June 5, 2016 – February 26, 2017

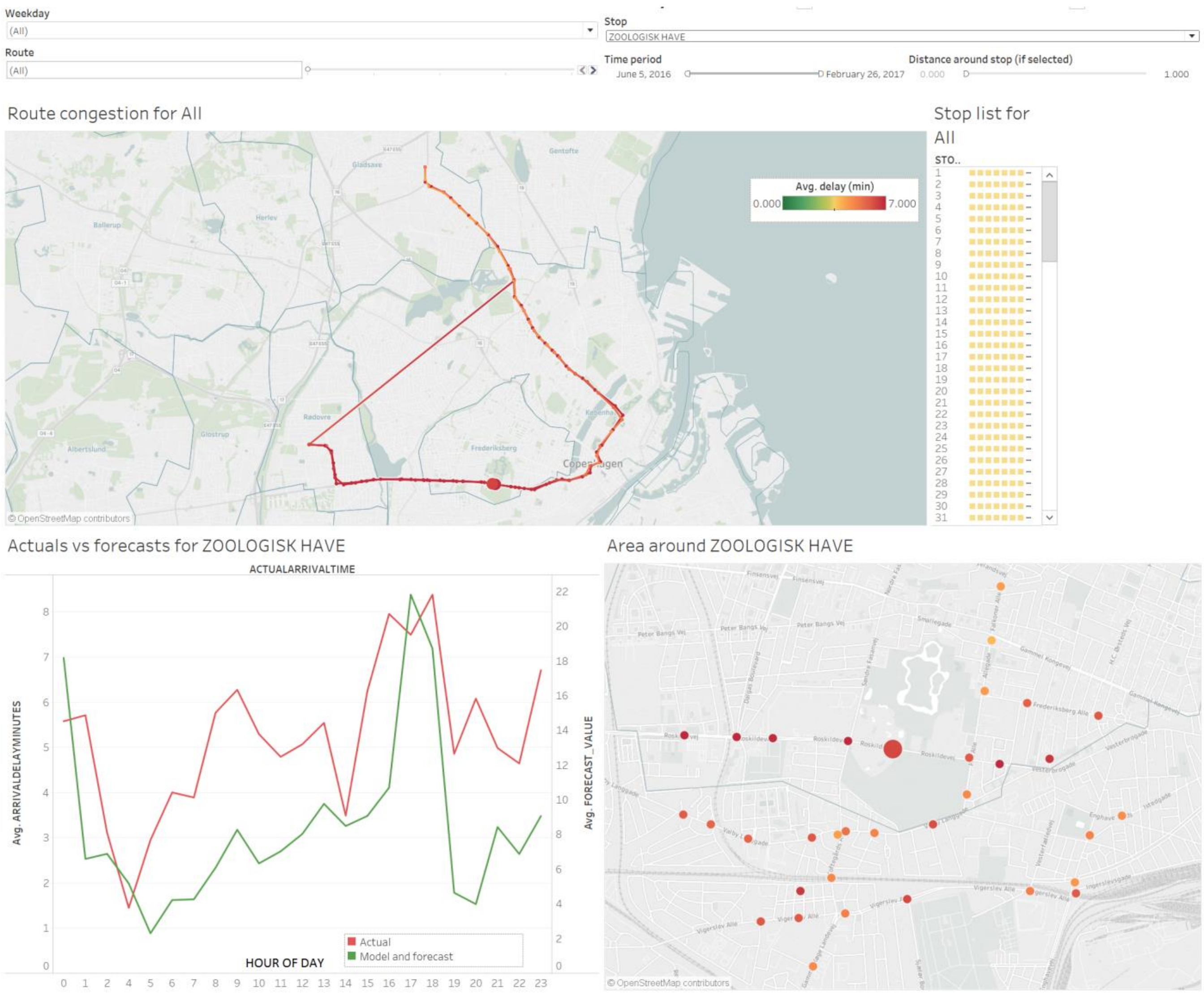
Distance around stop (if selected)

Route congestion for All

Stop list for All

Figure 6-25: Case Study data statistics

This application gives lot of options to explore the operational situation of the bus system and drill down to the level of routes and stops. The area within 1 mile of the targeted stop is marked with all the stops in that area and the delay value encountered on those stops to see projection of delay in nearby stops. There is seven days forecast as well that considers day and hour for segregating the forecast developed based on the predictive analytics. Considering the amount and volume of data, hundreds of different angle and aspects of the system can be monitored using this application but we will list only few here for elaboration. We can select any stop and the system will update the visualization for that stop as shown in Figure 6.26.



Weekday (All)
Route (All)
Stop ZOOLOGISK HAVE
Time period June 5, 2016 February 26, 2017
Distance around stop (if selected) 0.000 1.000

Figure 6-26: Forecast for a stop with impact on the whole route and nearby region

A more complex example is shown in Figure 6.27 for a stop that is part of multiple routes and therefore, delay on this stop can be either coming from other routes or it may be causing delay to stops of other routes. Figure 6.28 presents delay situation for 6 routes that contain this stop.

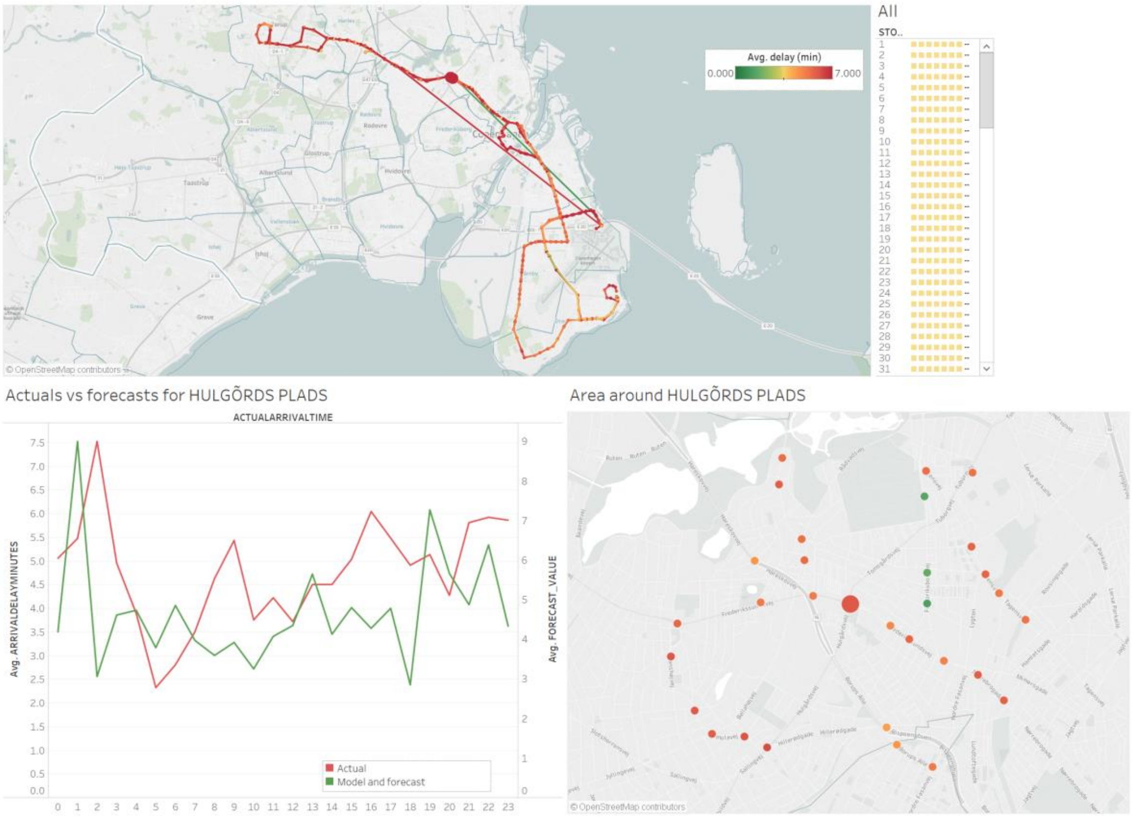


Figure 6-27: complete route with forecast and surrounding area for a stop existing in multiple routes

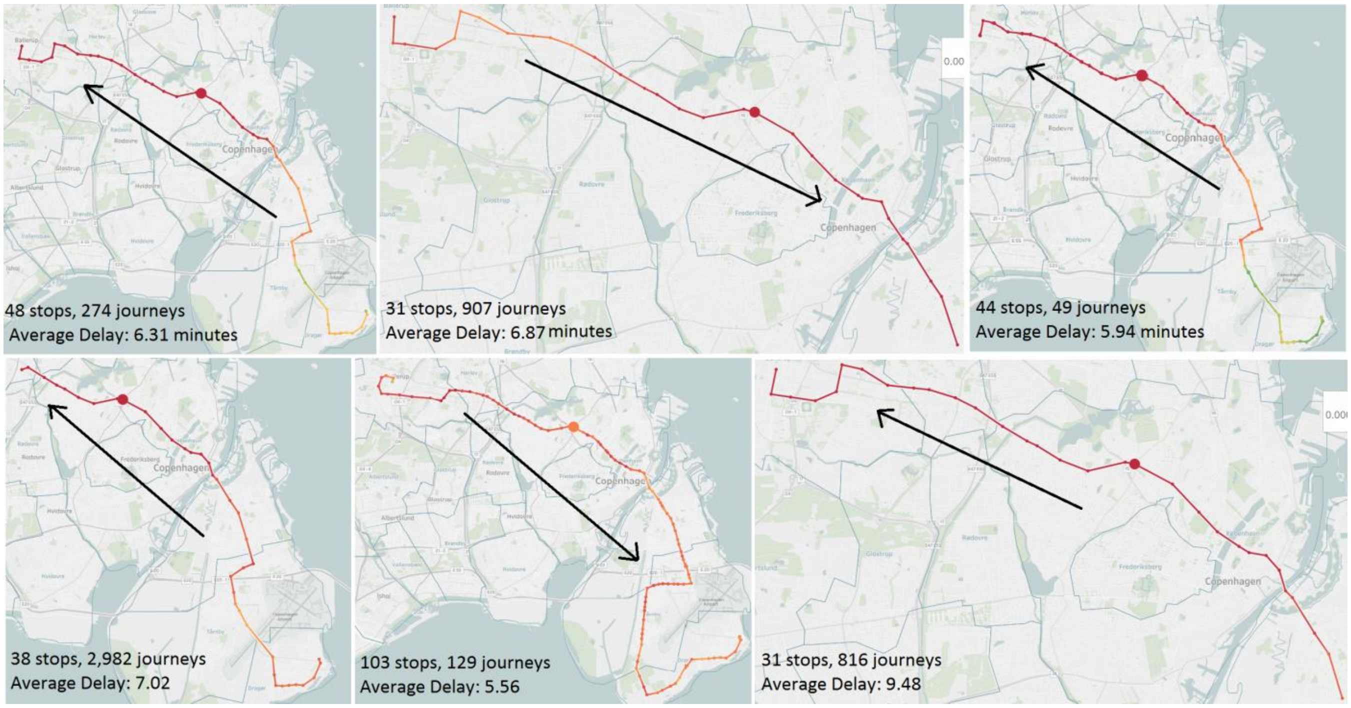


Figure 6-28: Delay status in 6 routes that contain stop presented in Figure 6.27

We can easily measure from these elaborations on which route may be causing delay in other routes having some common stops. Day wise forecast for Monday, Wednesday, Saturday and Sunday is shown in Figure 6.29based on the history data and real-time data being reported.

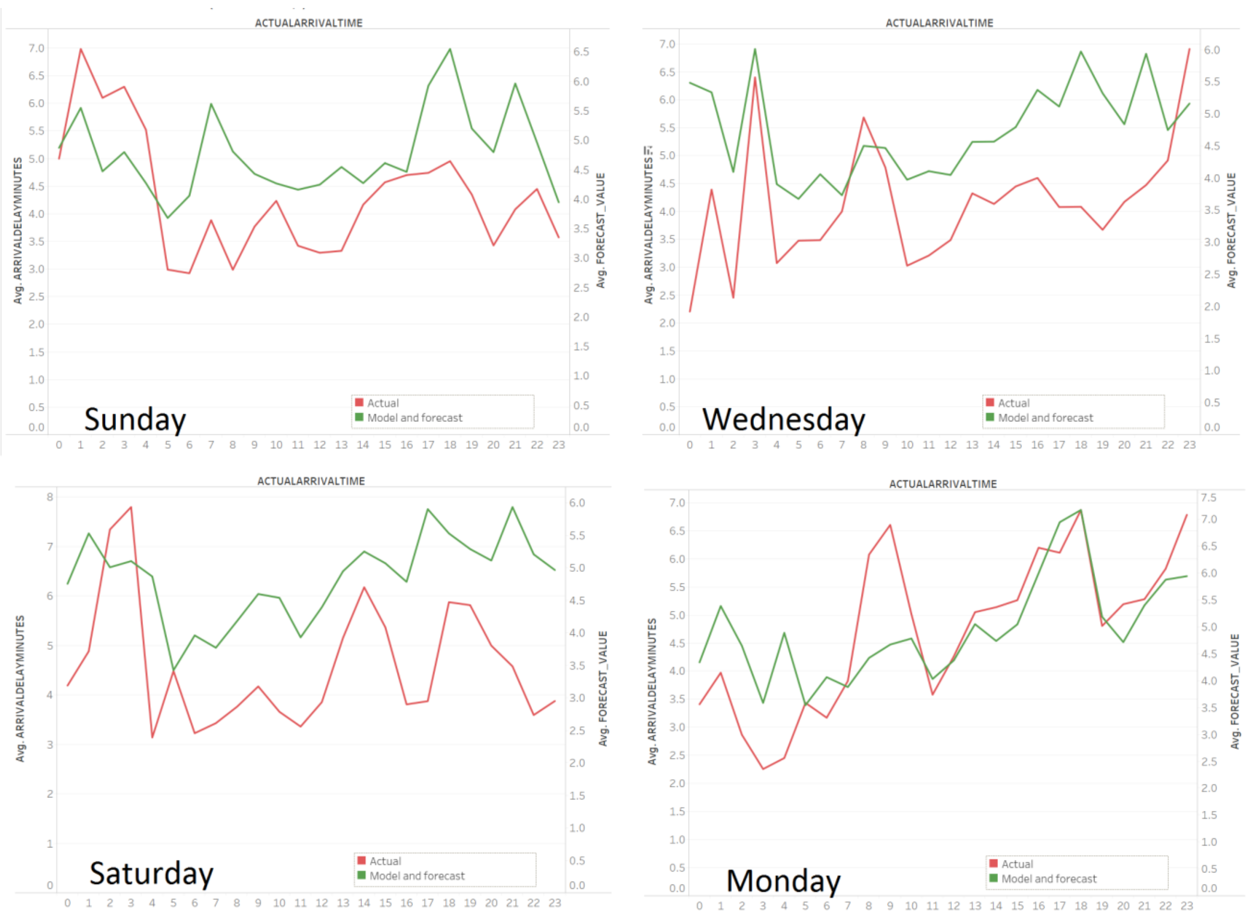


Figure 6-29: Forecast for 4 days of week

As mentioned at the start of this section, there are hundreds of screenshots that can be taken from the system but we cannot put all of them here because although they represent different patterns, they may look like repetition of the same type of diagrams for different stops, areas, week days, hour of the day.

6.9 Summary

This chapter presented a comprehensive case study for the application of the real-time predictive monitoring system based on the data provided by our industrial partner – Mermaid Technology. The problems in existing system about lack of predictability were highlighted and then a migration into new architecture implementation was elaborated. The implementation of new architecture involves no changes in the existing system and rather provides an implementation that can be used with any existing system as well as a new system can be built based on this architecture. That means a completely new architecture is implemented during this Thesis. There was a consistent feedback process involved where technical as well as operational staff from the company were kept in loop and all the findings and improvements were discussed. The problems in the existing system were highlighted through interviews on Skype, email and face-to-face meetings in the company's office in Denmark. The results were also shared with the management and technical team and there was great interest in applying the solution for evaluation and integration. This elaboration contained information about automatic real-time data acquisition and processing of the data that makes short-term and long-term predictions. The presented data acquisition process is independent of the data source or destination storage where the raw and processed data should be stored because it can integrate with virtually any data source. A detailed comparison of prediction implemented in this thesis was made with existing predictions and validity of the prediction was measured in terms of accuracy. This was followed by comprehensive presentation of the visualization component that enables the predictive monitoring system to facilitate in real-time monitoring of the buses and routes.

The implementation of the proposed architecture on the existing system has offered many benefits such as real-time monitoring of buses and routes, a novel way of rerouting the buses when they are in a traffic jam or stuck in a road incident/accident and forecast of prediction based on the data that belongs to the system itself. This case study has offered evidence for a software driven solution to problems for congestion and buses rerouting without using any expensive hardware infrastructure. Detailed analysis of the transport system with an option to drill down to the level of individual routes and stops can help in planning of the routes more effectively. The ability of the proposed architecture to integrate with any existing or new system has extended the concept of ITS architectures. Efficient handling of large volumes of data for this time-critical transport system using real-time integration of data mining techniques has opened new opportunities for designing specific algorithms that can handle data for futuristic transport systems.

The next chapter presents conclusions on the work achieved in this thesis and and some directions for future work.

CHAPER 7: CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

The primary aims and contributions of this thesis are the design, implementation, and evaluation of a new architecture for a real-time predictive monitoring system for urban transport and adaptive framework for intelligent transport systems with emphasis on buses. The work which was carried out to achieve these aims, as well as the results of the work, was reported in the previous chapters of this thesis. In this section, the achievements and conclusions which have been previously drawn will be summarised.

The research began by a thorough investigation into existing intelligent transport systems in order to examine their support for specialised urban transport and provision of an automated monitoring system. The investigation started by reviewing the growth of smart sensors has impacted different sectors of life, particularly transport. This was followed by a discussion on different intelligent transport system architectures that already exist, which can take advantage of their context derived from sensors installed on the vehicles and transport infrastructure. Then, different data mining techniques, that can be used to process the data for knowledge discovery with high volumes of data being generated, were presented.

Similarly, the thesis has investigated existing arrival time prediction systems using data mining techniques in order to determine the effectiveness of data mining approaches and examine how they are applied in making short-term and long-term arrival time predictions. The investigation started by discussing and comparing the approaches of data mining, application on arrival time prediction, evaluation of prediction models, and their advantages as well as deficiencies with respect to dynamically updated complex public transport scenarios. Moreover, it investigated a suitable integration mechanism of data mining techniques and data processing strategies in order to specify how they can be used in different stages of data from acquisition to prediction. In addition, the thesis highlighted a suitable paradigm for the implementation of the real-time data processing as well as a suitable data-driven decision-making mechanism for urban transport.

The identified deficiencies of the existing intelligent transport system architectures have been addressed in this thesis by developing a new architecture, which is able to handle short-term and long-term predictions in real time. The proposed architecture is based on the intelligent integration of heterogeneous sources of data coming from sensors installed on buses about location, journey progress, and the relationship between different trips being performed based on the route and time of the day. The architecture presented an integration process that combines novel caching strategy demonstrated in the implementation and evaluation of efficient data acquisition with data processing using data mining techniques. Different data acquisition and transformation strategies have been employed for cleaning the data before it is fed into the data prediction engine. Different mining models are derived from the data attributes such as time of the day, delays reported at the stops that have passed and the coefficients being continuously updated from the data mining process. Four data mining techniques were applied to the data using a combination of mining models addressing correlation of different attributes with each other and the prediction of arrival time. This process generates prediction coefficients for each mining model, and these coefficients are applied to the dataset for which we intend to make a prediction. The score for each combination of data mining technique and mining model is calculated and results prediction coefficients from the best-fit model are updated in the prediction database.

Moreover, the novel aspects of the system monitoring have been demonstrated in the implementation of comprehensive visualisation engine that integrates with the live feed coming from the data prediction engine and current context information acquired from bus data. The visualisation engine monitors the progress of thousands of buses completing their scheduled trips through tens of thousands of bus stops, and multiple data visualisation components are implemented addressing different aspects of monitoring like congestion, system performance, routes and buses causing the delay. The impact of each arrival event is captured in real-time and integrated with other buses travelling in the zone to reflect the combined projection of delay on the overall situation of the network. Different aspects of real-time monitoring covered by the visualisation engine are route based monitoring, multiple routes monitoring originating from same source or going to same destination, routes with intersecting stops, routes covering multiple zones, current position and delay situation of each bus. The delay on each stop can be correlated with other stops in a specific radius to see the impact of delay on the stop to other stops in the region as well as a reflection of delay at other stops to the current stop being monitored. A causal analysis to know more about the potential cause of delay can also be performed by monitoring different segments of each route and correlating with other route segments in the same area. A real-time monitoring of delay projection is implemented that expands to all the routes and all the stops with the inclusion of buses and rendering of the live location of buses and contribution of delay for each bus.

The design, implementation, and evaluation of an intelligent transport system based on proposed architecture are presented that selects a partial dataset from the data of the industrial partner, Mermaid Technology. A detailed introduction to the data, how it defines the context of the bus and relationship of this context data with the bus is elaborated. The process of transforming the structured data into a flat format is explained where this transformation process excludes all the data considered unnecessary for analysis and retains information about previous delays reported, delays from previous stops and dependency between currently reported delay and previously reported delays. The selection of Microsoft SQL Analysis engine is justified for applying data mining techniques, which is followed by implementation of an analysis package that contains mining models created for each of the four selected data mining techniques. The data mining techniques are applied to prediction models, and the results of prediction are analysed for their accuracy. The web services component of the system is also evaluated and benchmarked for performance for different number of concurrent users asking for the information through web services or pushing information into the system through web services.

Finally, a case study involving the proposed architecture and complete implementation and evaluation with data, from an industrial partner, for last eight months was experimentally evaluated. The case study involved data acquisition from thousands of buses in real-time through the web services, parallel processes of storing the data in persistent storage and recording of the instant snapshot for real-time analysis. An automated data acquisition from the database is integrated with a real-time data feed, and it is transformed into prediction-ready data. The transformed data is automatically exported into data mining engine that processes the data using pre-evaluated prediction models and generates prediction probabilities with prediction coefficients that represent the combination of existing history of data and the real-time data just arrived. The prediction coefficients are mapped to the whole dataset, and updated set of prediction results are produced. The prediction results are fed into a continuous integration loop of the visualisation engine that updates the reports and dashboards designed on the prediction results combined with existing visualisations.

The implementation of the system has achieved the aims that were set for the case study namely, design and implement a novel system architecture for intelligent transport systems in order to allow existing and new public transport systems to implement context-specific arrival time prediction system. The previous task has been performed by implementing an automated data acquisition process, diverse transformation to convert the data into a prediction-ready format and continuous integration of this information into the prediction system. The results from the prediction system are made available through web services for easy and direct access to the relevant prediction data. Moreover, the experimental evaluation has demonstrated that the data mining techniques used in the thesis proved to have significant effectiveness by conducting in-depth pattern recognition of the travelling behaviour in the context data collected from buses. The analysis of the results has shown that the accuracy of the predicted data is higher when compared to another prediction system being used by the industrial partner. Implementation of a real-time predictive monitoring system in the form of comprehensive visualisation was another objective that was achieved in this thesis.

Although the system has addressed challenges identified for this Thesis, there are few things that needs to be evaluated when integrated with this architecture. The impact of new sources of data such as data from traffic authorities for planning and road work, incidents and accidents data from emergency department and weather data integration is not known as that was not accomplished in the scope of this work. More influx of data will have an impact on the whole life cycle of data and will have an impact on performance of data acquisition to visualization. The system is independent of the data format but it does not comply with industry standards like HAFAS format for storing and exchanging transport data. This was discovered when I was interviewing one of the potential company that implements different data formats for being used by transport systems around the world. The lack of implementation for these formats stops many different data providers from being integrated unless a custom transformation component is built. Another important aspect where the system can be improved, and is considered a future work, that its prediction is associated with GPS points of the stops. That means that it can predict when a bus will arrive on a particular stop but it does not reflect the progress on bus location between two stops. Availability of this information can improve connectivity between different transport systems because the decisions can be made when the buses are in transit rather than making these decisions only when they are on a stop.

7.2 Future Work

In this section, we give suggestions about how the work presented in this thesis can be carried out further. The future work which can presently be seen in this field may be classified into a number of categories.

The first category for future research is related to the fact that the currently proposed architecture should be customised while applying the same structure and methods for other modes of transport like trains, trams and other private services like taxi and ride-hailing and ride-sharing services. This can be achieved by using building a connector component that can integrate with a data source from those services and transform that into prediction-ready data. This can be easily done because the bus is also a vehicle and this concept can be replaced with any other type of vehicle, and the architecture as well as the implementation does not provide any bus specific components and is kept open for any type of vehicle. However, the operation of taxis, ride-hailing and ride-sharing services is very different compared to the buses, trains and trams where there are specific starting and end points along with known stopping points. Also, public transport vehicles travel on already specified routes most of the time. That means the architecture cannot be applied as-is to taxis, ride-hailing and ride-sharing services unless the data is transformed to match the structure the system can process. The transformation will consider pickup and destination point and distance between them as input parameter and then frequency of trips between those points will reflect amount of time it takes and time of the day a trip was made. Additional attributes can be added for type of car used, number of passengers and other traffic parameters. Once this processing and integration is done for the data sources, the predictive engine can predict arrival time for these services as well and visualization will present location of vehicles with the progress of the trip being made.

The second category is related to the extension of proposed ITS architecture to be implemented in multiple heterogonous bus systems through integration with data collected from these systems. It was seen in the evaluation of the architecture that it could handle thousands of buses going through tens of thousands of bus stops to complete their scheduled journeys, this analysis should be evaluated for data from multiple systems and therefore increasing the amount of data and connected devices to produce prediction results. The accuracy of prediction should be evaluated across systems to enhance sharing of information between the systems through the prediction interface.

The third category is related to the optimisation strategies. The experimental evaluation of the case study has indicated the possibility of developing decision support system for urban transport. The urban transport systems and their components installed on the devices inside vehicles should extend this function to enable the vehicles to make smart decisions based on the localised data available. The rules for these decisions can be updated on a regular basis from the server, but the vehicles should have the ability to make decisions on their own if the system goes offline.

Finally, the research area that can be contributing to future work is the customisation and application of Data Mining techniques. Domain-specific data mining techniques should be designed that use context of the transport system as a criterion for decision-making as their built-in feature. This customisation of data mining techniques will give researchers an opportunity to use and build on those data mining techniques and extend the features to include advanced context parameters from growing vehicle infrastructures. Development of Big Data initiatives to integrate with ITSs to expand the scope of these public transport systems should be encouraged so that ITSs can be integrated with external systems and establish a reliable information sharing.

REFERENCES

[1] Hu, Bo, Zhixue Wang, and Qingchao Dong. "A modeling and reasoning approach using description logic for context-aware pervasive computing." Emerging Research in Artificial Intelligence and Computational Intelligence. Springer Berlin Heidelberg, 2012. 155-165.

[2] Jeon, Paul Barom. "Context Aware Intelligent Mobile Platform for Local Service Utilization." Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on. Vol. 1. IEEE, 2012.

[3] Delalandre, G., McCarthy, J., Mechant, P., Muller, R., & Ysebaert, G. "Supporting and enhancing a sense of community in smart cities through big data." European Institute of Innovation and Technology (EIT) Foundation's Annual Innovation Forum. EITF, 2012.

[4] E. Costa-Montenegro, F. Quinoy-Garcia, F. J. Gonzalez-castano and F. Gil-Castineira, "Vehicular Entertainment Systems: Mobile Application Enhancement in Networked Infrastructures," in IEEE Vehicular Technology Magazine, vol. 7, no. 3, pp. 73-79, Sept. 2012.

[5] Artikis, Alexander, Marek Sergot, and Georgios Paliouras. "Run-time composite event recognition." Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems. ACM, 2012.

[6] Cugola, Gianpaolo, and Alessandro Margara. "Processing flows of information: From data stream to complex event processing." ACM Computing Surveys (CSUR) 44.3 (2012): 15.

[7] Etzion, O., and P. Niblett. "Event Processing in Action, Manning Publications." (2011).

[8] Gal, Avigdor, Segev Wasserkrug, and Opher Etzion. "Event processing over uncertain data." Reasoning in Event-Based Distributed Systems. Springer Berlin Heidelberg, 2011. 279-304.

[9] Jeffery, Shawn R., Minos Garofalakis, and Michael J. Franklin. "Adaptive cleaning for RFID data streams." Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006.

[10] Kowalski, Robert, and Marek Sergot. "A logic-based calculus of events." Foundations of knowledge base management. Springer Berlin Heidelberg, 1989. 23-55.

[11] Kim, Mi Jeong. "A framework for context immersion in mobile augmented reality." Automation in construction 33 (2013): 79-85.

[12] Escobedo, L., Nguyen, D. H., Boyd, L., Hirano, S., Rangel, A., Garcia-Rosas, D., Hayes, G. "MOSOCO: a mobile assistive tool to support children with autism practicing social skills in real-life situations." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2012.

[13] Amft, Oliver, and Paul Lukowicz. "From backpacks to smartphones: Past, present, and future of wearable computers." IEEE Pervasive Computing 8.3 (2009).

[14] Roussos, George, Andy J. Marsh, and Stavroula Maglavera. "Enabling pervasive computing with smart phones." IEEE Pervasive Computing 4.2 (2005): 20-27.

[15] R. Achkar, G. A. Haidar, R. Pharaoun, A. Rokaya and M. A. Ahmar, "i-Display Using Smart Phone Application," 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, Valetta, 2012, pp. 379-384.

[16] Mitchener, Jonathan. "What we'll wear-[comms futures]." Engineering & Technology 3.18 (2008): 74-74.

[17] Nageba, Ebrahim, Paul Rubel, and Jocelyne Fayn. "Context-aware mobile services adaptation to dynamic resources. Application to mHealth." Mobile and Wireless Networking (iCOST), 2012 International Conference on Selected Topics in. IEEE, 2012.

[18] Dunn, B. K., Galletta, D. F., Hypolite, D., Puri, A., & Raghuwanshi, S. "Development of smart phone usability benchmarking tasks." System Sciences (HICSS), 2013 46th Hawaii International Conference on. IEEE, 2013.

[19] Mogg, Trevor. "Smartphone sales exceed those of PCs for first time, Apple smashes record." Retrieved August 6 (2012): 2013.

[20] Singh, Sameer. "Impact of iOS & Android on the PC Replacement Cycle‖." (2012).

[21] Fagrell, Henrik, Kerstin Forsberg, and Johan Sanneblad. "FieldWise: a mobile knowledge management architecture." Proceedings of the 2000 ACM conference on Computer supported cooperative work. ACM, 2000.

[22] Poushter, Jacob. "Smartphone ownership and Internet usage continues to climb in emerging economies." Pew Research Center (2016).

[23] Dey, Anind K. "Understanding and using context." Personal and ubiquitous computing 5.1 (2001): 4-7.

[24] Dorn, Lisa, ed. Driver behaviour and training. Vol. 3. Ashgate Publishing, Ltd., 2008.

[25] Engstrom, J., and T. Victor. "Real-time recognition of large-scale driving patterns." Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE. IEEE, 2001.

[26] Fuller, Ray. "A conceptualization of driving behaviour as threat avoidance." Ergonomics 27.11 (1984): 1139-1155.

[27] Laurence, H., & Nick, M. "Review of fatigue detection and prediction technologies." National Road Transport Commission (2000).

[28] Henricksen, Karen, Jadwiga Indulska, and Andry Rakotonirainy. "Modeling context information in pervasive computing systems." International Conference on Pervasive Computing. Springer Berlin Heidelberg, 2002.

[29] Yasuo, Kumagai T. Sakaguchi, Okuwa Masayuki, and Akamatsu Motoyuki. "Prediction of Driving behaviour through Probabilistic Inference." Proceedings of the 8th Conf on Engineering Applications of neural Networks (EANN'03) Malaga. 2003.

[30] Näätänen, Risto, and Heikki Summala. "Road-user behaviour and traffic accidents." Publication of: North-Holland Publishing Company (1976).

[31] Oliver, Nuria, and Alex P. Pentland. "Graphical models for driver behavior recognition in a smartcar." Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE. IEEE, 2000.

[32] Olsson, Carl Magnus. "Taking the next step in context-aware applications." Proceedings of IRIS26, Helsinki, Finland (2003).

[33] Rakotonirainy, A., Cutmore, T., Steele, T., & James, D. A. "An investigation into peripheral physiological markers that predict monotony." Proceedings. 2004.

[34] Artikis, A., Etzion, O., Feldman, Z., & Fournier, F. "Event processing under uncertainty." Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems. ACM, 2012.

[35] Artikis, Alexander, Marek Sergot, and Georgios Paliouras. "Run-time composite event recognition." Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems. ACM, 2012.

[36] Cugola, Gianpaolo, and Alessandro Margara. "Processing flows of information: From data stream to complex event processing." ACM Computing Surveys (CSUR) 44.3 (2012): 15.

[37] Buthpitiya, S., Luqman, F., Griss, M., Xing, B., & Dey, A. K. "Hermes--A Context-Aware Application Development Framework and Toolkit for the Mobile Environment." Advanced Information Networking

and Applications Workshops (WAINA), 2012 26th International Conference on. IEEE, 2012.

[38] Radio, N., Zhang, Y., Tatipamula, M., & Madisetti, V. K. "Next-generation applications on cellular networks: trends, challenges, and solutions." Proceedings of the IEEE 100.4 (2012): 841-854.

[39] Min, C., Han, W., Hwang, I., Lee, S. J., Lee, Y., Shin, I., & Song, J. "Poster: towards mobile GPU-accelerated context processing for continuous sensing applications on smartphones." MobiSys. 2012.

[40] Sörös, Gábor, and Christian Flörkemeier. "Towards next generation barcode scanning." Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia. ACM, 2012.

[41] H. H. Hsu, K. C. Tsai, Z. Cheng and T. Huang, "Posture Recognition with G-Sensors on Smart Phones," 2012 15th International Conference on Network-Based Information Systems, Melbourne, VIC, 2012, pp. 588-591. [42] http://www.internetworldstats.com/stats.htm, December 2016

[43] http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats, June 2016

[44] Hahn, Jim. "Mobile augmented reality applications for library services." New library world 113.9/10 (2012): 429-438.

[45] Huang, Gary. "Powerful smartphone solutions unleashing new technology innovations." VLSI Design, Automation, and Test (VLSI-DAT), 2012 International Symposium on. IEEE, 2012.

[46] Lim, Tek Yong. "Designing the next generation of mobile tourism application based on situation awareness." Network of Ergonomics Societies Conference (SEANES), 2012 Southeast Asian. Ieee, 2012.

[47] Chen, Yiqiang, and Yuan Miao. "Next Generation Mobile E-commerce based on Opportunistic Context Sensing." International Journal of Information Technology 18.2 (2012).

[48] Figueiredo, L., Jesus, I., Machado, J. T., Ferreira, J. R., & De Carvalho, J. M. "Towards the development of intelligent transportation systems." Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE. IEEE, 2001.

[49] Cruz, I. R., Huiyong Xiao, and Feihong Hsu. "An ontology-based framework for XML semantic integration." Database Engineering and Applications Symposium, 2004. IDEAS'04. Proceedings. International. IEEE, 2004.

[50] Liu, W. E. I. N. I. N. G., Sun, D., Song, W., & Fu, L. I. P. I. N. G. "A virtual common information platform for intelligent transportation systems." Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on. IEEE, 2004.

[51] Shi, Qixin, and Weizhong Zheng. "Architecture Analysis of Common Information Platform for Intelligent Transportation Systems (ITS) and Its Construction Means." Journal of Transportation Engineering and Information 1.1 (2003): 41-47.

[52] Li, R. M., Lu, H. P., Qian, Z., & Shi, Q. X. "Research of in the integrated transportation information platform based on XML." Intelligent Transportation Systems. 2005.

[53] R, Qiu. "ITS development of Singapore", ITS communication. Volume 5, Number 1, pp.5–11, (2003).

[54] Sheth, Amit P. "Changing focus on interoperability in information systems: from system, syntax, structure to semantics." Interoperating geographic information systems. Springer US, 1999. 5-29.

[55] Bellavista, P., Corradi, A., Fanelli, M., & Foschini, L. "A survey of context data distribution for mobile ubiquitous systems." ACM Computing Surveys (CSUR) 44.4 (2012): 24.

[56] Barba, Evan, Blair MacIntyre, and Elizabeth D. Mynatt. "Here we are! where are we? locating mixed reality in the age of the smartphone." Proceedings of the IEEE 100.4 (2012): 929-936.

[57] Tsai, Mavis. "The trends and adoption behaviors of smart phones in Taiwan: A comparison between persons over 45 years of age and youth under 25." Technology Management for Emerging Technologies (PICMET), 2012 Proceedings of PICMET'12:. IEEE, 2012.

[58] Praveen, Dr Kumar, Reddy Dhanunjaya, and Singh Varun. "intelligent transport system using GIS." Map India conference. 2003.

[59] Figueiredo, L., Jesus, I., Machado, J. T., Ferreira, J. R., & De Carvalho, J. M. "Towards the development of intelligent transportation systems." Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE. IEEE, 2001.

[60] Mirchandani, Pitu, and Fei-Yue Wang. "RHODES to intelligent transportation systems." IEEE Intelligent Systems 20.1 (2005): 10-15.

[61] Wang, Dasheng, Lei Ren, and Jing Li. "Modeling intelligent transportation systems with multi-agent on SOA." Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on. IEEE, 2010.

[62] Kamran, Shoaib, and Olivier CL Haas. "Semantic agent-based controls for Service Oriented Architecture (SOA) enabled Intelligent Transportation Systems (ITS)." Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. IEEE, 2011.

[63] Li-Der Chou, Bo-Teng Deng, D. C. Li and Kai-Wei Kuo, "A passenger-based adaptive traffic signal control mechanism in Intelligent Transportation Systems," 2012 12th International Conference on ITS Telecommunications, Taipei, 2012, pp. 408-411.

[64] Barth, Matthew, and Michael Todd. "Intelligent transportation system architecture for a multi-station shared vehicle system." Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE. IEEE, 2000. [65] Yang, Zhang. "Pedestrian detection for intelligent vehicle based on bilayer difference features algorithm." Transportation Information and Safety (ICTIS), 2015 International Conference on. IEEE, 2015.

[66] Y. m. L. Roux and P. Lassudrie-Duchesne, "Palmyre : a MIMO reconfigurable transmission platform for ITS applications," 2006 6th International Conference on ITS Telecommunications, Chengdu, 2006, pp. 907-912.

[67] Zhu, H., Chang, S., Lu, L., & Zhang, W. "RUPS: Fixing Relative Distances among Urban Vehicles with Context-Aware Trajectories." Parallel and Distributed Processing Symposium, 2016 IEEE International. IEEE, 2016.

[68] Waite, J., Benke, M., Nguyen, N., Phillips, M., Melton, S., Oman, P., Johnson, B. K. "A combined approach to ITS vulnerability and survivability analyses." Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on. IEEE, 2004.

[69] Bosc, P., Sentieys, O., Peyret, F., Ray, C., & Bonnin, J. M. "Gis its bretagne: status and perspectives." ITS Telecommunications Proceedings, 2006 6th International Conference on. IEEE, 2006.

[70] Chen, S., Ray, C., Tan, J., & Claramunt, C. "Integrated Transportation GIS for the City of Guangzhou, China." ITS Telecommunications Proceedings, 2006 6th International Conference on. IEEE, 2006.

[71] Lye, S. C. K., Tan, S. E., Siew, Z. W., Chin, Y. K., & Teo, K. T. K. "Adaptive modulation in public transport network system with network coding." Global High Tech Congress on Electronics (GHTCE), 2012 IEEE. IEEE, 2012.

[72] B. Kolosz and S. Grant-Muller, "Sustainability assessment approaches for intelligent transport systems: the state of the art," in IET Intelligent Transport Systems, vol. 10, no. 5, pp. 287-297, 6 2016.

[73] Zhang, Mingchen, Jingyan Song, and Yi Zhang. "Three-tiered sensor networks architecture for traffic information monitoring and processing." Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on. IEEE, 2005.

[74] Hamza-Lup, G. L., Hua, K. A., Lee, M., & Peng, R. "Enhancing intelligent transportation systems to improve and support homeland security." Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on. IEEE, 2004.

[75] Chih-Ju, C., Sheng-Hao, S., Kuo-Hsiung, T., & To-Cheng, L. "A novel SCADA system design and application for intelligent traffic control." Control and Decision Conference (CCDC), 2015 27th Chinese. IEEE, 2015.

[76] Jeong, Ranhee, and R. Rilett. "Bus arrival time prediction using artificial neural network model." Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on. IEEE, 2004.

[77] Maiti, S., Pal, A., Pal, A., Chattopadhyay, T., & Mukherjee, A. "Historical data based real time prediction of vehicle arrival time." Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. IEEE, 2014.

[78] Liu, J., Gong, C., Li, J., Li, J., & Cui, X. "Social Sensing Enhanced Time Ruler for Real-Time Bus Service." Semantics, Knowledge and Grids (SKG), 2013 Ninth International Conference on. IEEE, 2013.

[79] Liu, T., Ma, J., Guan, W., Song, Y., & Niu, H. "Bus arrival time prediction based on the k-nearest neighbor method." Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference on. IEEE, 2012.

[80] Pongnumkul, S., Pechprasarn, T., Kunaseth, N., & Chaipah, K. "Improving arrival time prediction of Thailand's passenger trains using historical travel times." Computer Science and Software Engineering (JCSSE), 2014 11th International Joint Conference on. IEEE, 2014.

[81] Moreira-Matias, L., Mendes-Moreira, J., de Sousa, J. F., & Gama, J. "Improving mass transit operations by using AVL-based systems: a survey." IEEE Transactions on Intelligent Transportation Systems 16.4 (2015): 1636-1653.

[82] Agafonov, Anton, and Vladislav Myasnikov. "An Adaptive Algorithm for Public Transport Arrival Time Prediction Based on Hierarhical Regression." Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on. IEEE, 2015.

[83] Maiti, S., Pal, A., Pal, A., Chattopadhyay, T., & Mukherjee, A. "Historical data based real time prediction of vehicle arrival time." Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. IEEE, 2014.

[84] Liu, J., Gong, C., Li, J., Li, J., & Cui, X. "Social Sensing Enhanced Time Ruler for Real-Time Bus Service." Semantics, Knowledge and Grids (SKG), 2013 Ninth International Conference on. IEEE, 2013.

[85] Papadimitratos, P., De La Fortelle, A., Evenssen, K., Brignolo, R., & Cosenza, S. "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation." IEEE Communications Magazine 47.11 (2009).

[86] Liu, W. E. I. N. I. N. G., Sun, D., Song, W., & Fu, L. I. P. I. N. G. "A virtual common information platform for intelligent transportation systems." Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on. IEEE, 2004.

[87] Shi, Qixin, and Weizhong Zheng. "Architecture Analysis of Common Information Platform for Intelligent Transportation Systems (ITS) and Its Construction Means." Journal of Transportation Engineering and Information 1.1 (2003): 41-47.

[88] Li, R. M., Lu, H. P., Qian, Z., & Shi, Q. X. "Research of in the integrated transportation information platform based on XML." Intelligent Transportation Systems. 2005.

[89] R, Qiu. “ITS development of Singapore”, ITS communication. Volume 5, Number 1, pp.5–11, (2003).

[90] Sheth, Amit P. "Changing focus on interoperability in information systems: from system, syntax, structure to semantics." Interoperating geographic information systems. Springer US, 1999. 5-29.

[91] Cruz, I. R., Huiyong Xiao, and Feihong Hsu. "An ontology-based framework for XML semantic integration." Database Engineering and Applications Symposium, 2004. IDEAS'04. Proceedings. International. IEEE, 2004.

[92] Cui, Zhan, and Paul O'Brien. "Domain ontology management environment." System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000.

[93] Li, Yang, Jun Zhai, and Yan Chen. "Using ontology to achieve the semantic integration of the intelligent transport system." Information Technology 6 (2005): 10-13.

[94] Samper, J. J., Tomás, V. R., Martinez, J. J., & van den Berg, L. "An ontological infrastructure for traveller information systems." Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE. IEEE, 2006.

[95] Gruber, Thomas R. "A translation approach to portable ontology specifications." Knowledge acquisition 5.2 (1993): 199-220.

[96] Klein, Michel. "Interpreting XML documents via an RDF schema ontology." Database and expert systems applications, 2002. proceedings. 13th international workshop on. IEEE, 2002. [97] Noy, Natalya Fridman, and Mark A. Musen. "Algorithm and tool for automated ontology merging and alignment." Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). Available as SMI technical report SMI-2000-0831. 2000.

[98] J. Crowcroft ; I. Wakeman ; Z. Wang ; D. Sirovica. "Is layering, harmful? (remote procedure call)." IEEE Network 6.1 (1992): 20-24.

[99] Hutchinson, Norman C., and Larry L. Peterson. "The x-kernel: An architecture for implementing network protocols." IEEE Transactions on Software engineering 17.1 (1991): 64-76.

[100] Haas, Zygmunt. "A protocol structure for high-speed communication over broadband ISDN." IEEE Network 5.1 (1991): 64-70.

[101] Kanakia, Hemant, and David Cheriton. "The VMP network adapter board (NAB): High-performance network communication for multiprocessors." ACM SIGCOMM Computer Communication Review. Vol. 18. No. 4. ACM, 1988.

[102] Jain, Niraj, M. Schawrtz, and Theodore Bashkow. "Transport protocol processing at GBPS rates." ACM SIGCOMM Computer Communication Review. Vol. 20. No. 4. ACM, 1990.

[103] Sterbenz, James PG, and Gurudatta M. Parulkar. "AXON: Application-oriented lightweight transport protocol design." (1989).

[104] “ITS Organizational Issues – The road ahead”, unpublished, ITS SA, 2000

[105] “ITS Part of the solutions to the Transport Challenges of South Africa”. Unpublished, ITS SA, 2000

[106] “An Introduction to the Technical Specifications for UTMC Systems”, UK DETR, 1997

[107] “Technical Specifications TS001: Part 1”, UK DETR, 1997

[108] “Developing ITS Using the National ITS Architecture”, Executive Edition, ITS JPO, US DOT, 1998

[109] “Developing Freeway and Incident Management Systems Using the National ITS Architecture”, ITS JPO, US DOT, 1998

[110] “National Transportation Communications Protocol for ITS [NTCIP] 9001”, AASHTO/ITE/ NEMA, 1999

[111] “UTMC 08, Suitability of NTCIP based Communications for UK UTMC Users”, UK DETR, 1998

[112] Cable, V. “Overview. In Trade Blocs? The Future of Regional Integration (Cable and Henderson, eds.)”, Royal Institute of International Affairs, London, 1994.

[113] Wolfe, David A. "The emergence of the region state." The nation state in a global/information era: policy challenges (1997): 205-240.

[114] Sussman, Joseph M. "ITS Deployment and the “Competitive Region”." Perspectives on Intelligent Transportation Systems (ITS) (2005): 83-86.

[115] Malecki, Edward J. "Technology and economic development: the dynamics of local, regional, and national change." (1997).

[116] Ōmae, Ken’ichi. “The end of the nation state: The rise of regional economies.” Simon and Schuster, 1995.

[117] Rodríguez, D. A., and Sussman, J. “A Framework for Developing a Regional ITS System Architecture.” In Transportation Research Record 1588, TRB, National Research Council, Washington, D.C., 1997, pp. 77-85.

[118] Sussman, J. M. “Intelligent Vehicle Highway Systems.” OR/MS Today, December 1992.

[119] “ITS Architecture Documentation.” ITS America, Washington, D.C., 1996.

[120] Gary Euler and Douglas Robertson, eds. “National ITS Program Plan Synopsis.” ITS America, Washington, D.C., 1995.

[121] Rodríguez, Daniel Andrés. Developing a system architecture for intelligent transportation systems with application to San Juan, Puerto Rico. Diss. Massachusetts Institute of Technology, 1996.

[122] Espinoza, E., Fernandez, B., Giunta, J., Montaña, E., Pasteris, E., & Razquin, O. "Plan Estratégico de Transporte Multimodal para el Oasis Norte de Mendoza: Mendoza en el Corredor Bioceánico Central, Relevamiento y Estado de Situación." CIT/MIT Joint Program (1996).

[123] Yang, Xing. Designing a transportation network for Mendoza, Argentina: a strategic approach. Diss. Massachusetts Institute of Technology, 1997.

[124] Atkins, M. Stella, Samuel T. Chanson, and James B. Robinson. "LNTP-an efficient transport protocol for local area networks." Global Telecommunications Conference, 1988, and Exhibition.'Communications for the Information Age.'Conference Record, GLOBECOM'88., IEEE. IEEE, 1988.

[125] La Porta, Thomas F., and Mischa Schwartz. "Architectures, features, and implementation of high-speed transport protocols." IEEE Network 5.3 (1991): 14-22.

[126] Doeringer, W. A., Dykeman, D., Kaiserswerth, M., Meister, B. W., Rudin, H., & Williamson, R. "A survey of light-weight transport protocols for high-speed networks." IEEE Transactions on Communications 38.11 (1990): 2025-2039.

[127] Svobodova, Liba. "Implementing OSI systems." IEEE Journal on Selected Areas in Communications 7.7 (1989): 1115-1130.

[128] Braun, Torsten, and Martina Zitterbart. "Parallel Transport System Design." HPN. 1992.

[129] Zitterbart, Martina. "High-speed transport components." IEEE Network 5.1 (1991): 54-60.

[130] La Porta, Thomas F., and Mischa Schwartz. "Performance analysis of MSP: feature-rich high-speed transport protocol." IEEE/ACM Transactions on Networking (TON) 1.6 (1993): 740-753.

[131] Watson, Richard W., and Sandy A. Mamrak. "Gaining efficiency in transport services by appropriate design and implementation choices." ACM Transactions on Computer Systems (TOCS) 5.2 (1987): 97-120.

[132] FRAME (2004). “European ITS Framework Architecture”, www.frame-online.net.

[133] Bossom, R. A. P., and P. H. Jesty. "Supporting the European ITS Framework Architecture." Proceedings 9 th World Congress on ITS, Chicago. ITS America. 2002.

[134] Jesty, Peter H., and Richard AP Bossom. "Involving stakeholders in ITS architecture creation." Proceedings 10 th World Congress on ITS, Madrid. ERTICO (ITS Europe). 2003.

[135] Mochizuki, Mizuo, Akihira Suzuki, and T. TAJIMA. "UTMS System Architecture." Proceedings of 6th world congress on Intelligent Transport Systems (ITS), held Toronto, Canada, November 8-12, 1999. 1999.

[136] VERTIS: "System Architecture for ITS in Japan," http/www.iijnet.or.jp/VERTIS/, 1999

[137] Koutaro Kato, et. al.: "System Architecture Development Method," 6th World Congress on ITS, 1999

[138] UTMS Japan Web Site: http://www.utms.or.jp/

[139] Bošnjak, I. “Intelligent Transportation Systems.” Faculty of Traffic Science, Zagreb, 2005.

[140] Mandžuka, Sadko, Božica Horvat, and Pero Škorput. "Development of ITS in Republic of Croatia." 19th ITS World Congress. 2012.

[141] Project: SEE ITS - Objectives, 2012, Available at: http://www.seeits.eu/Default.aspx

[142] Yokota, T., Weiland, R. “ITS System Architectures for Developing Countries.”, Technical Note 5, Transport and Urban Development Department, World Bank, 2005

[143] “Why you need an ITS Architecture?” 2011, Available at: http://www.frame-online.net

[144] “The Intelligent Transport Systems (ITS) Practitioners’ Guide to Europe, RTI Focus.”, London, 2011

[145] “Action Plan for the Deployment of Intelligent Transport Systems in Europe”, COM (2008) 886 final, 2008

[146] Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport, Official Journal of the European Union, 2010., L 207, 1 – 13

[147] Horvat, Božica. Directives of the European Union in the field of ITS. Diss. Fakultet prometnih znanosti, Svaučilište u Zagrebu, 2011.

[148] Mandžuka, S. “Intelligent Transportation System - Experiences in the Republic of Croatia”, ITS Workshop, Ministry of Sea, Transport and Infrastructure, Zagreb, 2009

[149] Bičanić, Davor. Improving maintenance of telematics systems on motorways in Croatia. Diss. Fakultet prometnih znanosti, Sveučilište u Zagrebu, 2012.

[150] Bošnjak, Ivan, Sadko Mandžuka, and Ljupko Šimunović. "Concept and Implementation of Regional ITS Architecture." 5. hrvatskog kongresa o cestama. 2011.

[151] Mandžuka, S. “Electronic Payment in Traffic and Transport - Challenges and Perspectives of Regional Development”, 6th ITS Croatia Forum, Zagreb, 2011

[152] Rijavec, R., Mitsakis, E., Niculescu, M., & Kernstock, W. "Intelligent Transport Systems deployment and integration in South East Europe." Proceedings of ISEP, Ljubljana (2013).

[153] Webster, F. V., and P. H. Bly. "The demand for public transport. Part II. Supply and demand factors of public transport." Transport Reviews 2.1 (1982): 23-46.

[154] Pucher, J., Korattyswaropam, N., Mittal, N., & Ittyerah, N. "Urban transport crisis in India." Transport Policy 12.3 (2005): 185-198.

[155] Goldman, Todd, and Roger Gorham. "Sustainable urban transport: Four innovative directions." Technology in society 28.1 (2006): 261-273.

[156] Hensher, David A. "The imbalance between car and public transport use in urban Australia: why does it exist?." Transport Policy 5.4 (1998): 193-204.

[157] Kognitio - http://kognitio.com/

[158] Fojtík, David, Petr Podešva, and Jan Gebauer. "Storing high volumes of data in MS SQL Server Express." Carpathian Control Conference (ICCC), 2015 16th International. IEEE, 2015.

[159] Romanchuk, Vasyl, Taras Andrukhiv, and Ihor Kahalo. "The main causes of failures sql server." (2012).

[160] Qi, Chunxia. "On index-based query in SQL Server database." Control Conference (CCC), 2016 35th Chinese. IEEE, 2016.

[161] Győrödi, C., Győrödi, R., Pecherle, G., & Olah, A. "A comparative study: MongoDB vs. MySQL." Engineering of Modern Electric Systems (EMES), 2015 13th International Conference on. IEEE, 2015.

[162] Boicea, Alexandru, Florin Radulescu, and Laura Ioana Agapin. "MongoDB vs Oracle-Database Comparison." EIDWT. 2012.

[163] Lu, Hongjun, Rudy Setiono, and Huan Liu. "Effective data mining using neural networks." IEEE transactions on knowledge and data engineering 8.6 (1996): 957-961.

[164] Dou, Edward Yong. "UniShuttle-a small-scale intelligent transport system in the connected mobility digital ecosystem." (2015).

[165] Day, Mark, Jonathan Rosenberg, and Hiroyasu Sugano. A model for presence and instant messaging. No. RFC 2778. 2000.

[166] Pereira, Francisco C., Filipe Rodrigues, and Moshe Ben-Akiva. "Using data from the web to predict public transport arrivals under special events scenarios." Journal of Intelligent Transportation Systems 19.3 (2015): 273-288.

[167] Lathia, Neal, Jon Froehlich, and Licia Capra. "Mining public transport usage for personalised intelligent transport systems." Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE, 2010.

[168] Froehlich, Jon, and John Krumm. Route prediction from trip observations. No. 2008-01-0201. SAE Technical Paper, 2008.

[169] Sánchez Rico, María Teresa. "Data mining, optimization and simulation tools for the design of Intelligent Transportation Systems." (2015).

[170] Berkhin, Pavel. "A survey of clustering data mining techniques." Grouping multidimensional data. Springer Berlin Heidelberg, 2006. 25-71.

[171] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[172] Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth. "From data mining to knowledge discovery in databases." AI magazine 17.3 (1996): 37.

[173] Yndurain, Elena, Daniel Bernhardt, and Celeste Campo. "Augmenting mobile search engines to leverage context awareness." IEEE Internet Computing 16.2 (2012): 17-25.

[174] https://www.moviatrafik.dk

[175] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. Classification and regression trees. CRC press, 1984.

[176] Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine." Computer networks and ISDN systems 30.1 (1998): 107-117.

[177] Chen, Jason R. "Making clustering in delay-vector space meaningful." Knowledge and information systems 11.3 (2007): 369-385.

[178] Cheung, D. W., Han, J., Ng, V. T., & Wong, C. Y. "Maintenance of discovered association rules in large databases: An incremental updating technique." Data Engineering, 1996. Proceedings of the Twelfth International Conference on. IEEE, 1996.

[179] Chi, Y., Wang, H., Philip, S. Y., & Muntz, R. R. "Catch the moment: maintaining closed frequent itemsets over a data stream sliding window." Knowledge and Information Systems 10.3 (2006): 265-294.

[180] Cost, Scott, and Steven Salzberg. "A weighted nearest neighbor algorithm for learning with symbolic features." Machine learning 10.1 (1993): 57-78.

[181] Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." IEEE transactions on information theory 13.1 (1967): 21-27.

[182] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the royal statistical society. Series B (methodological) (1977): 1-38.

[183] Abbas, Osama Abu. "Comparisons Between Data Clustering Algorithms." Int. Arab J. Inf. Technol. 5.3 (2008): 320-325.

[184] Peterson, Leif E. "K-nearest neighbor." Scholarpedia 4.2 (2009): 1883.

APPENDIX A

Connection Manager configuration for Automated Data Acquisition

```
  </DTS:ConnectionManager>
  <DTS:ConnectionManager
    DTS:refId="Package.ConnectionManagers[PATTY.MERMAID.LOCAL.IBI_Data.ibi_cdm]"
    DTS:CreationName="OLEDB"
    DTS:DTSID="{24BA20CF-CE7F-4F3B-992D-38EFA90542B6}"
    DTS:ObjectName="PATTY.MERMAID.LOCAL.IBI_Data.ibi_cdm">
    <DTS:ObjectData>
      <DTS:ConnectionManager
        DTS:ConnectionString="Data Source=PATTY.MERMAID.LOCAL;User
ID=ibi_cdm;Initial Catalog=IBI_Data;Provider=SQLNCLI11.1;Persist Security Info=True;Auto
Translate=False;">
        <DTS:Password
          DTS:Name="Password"
          Sensitive="1"

Encrypted="1">AQAAANCMnd8BFdERjH0AwE/Cl+sBAAAAK1MYMhtQik6uP20k5bqUlwAA
AAAIAAAARABUAFMAAAAQZgAAAAEAACAAAAA1+Xc8LMX9TF06wPHmeHrjWIij1v1mLJ
zqf1PkgWHHOwAAAAAOgAAAAIAACAAAACi1MQ++UOStCu/4OzWaeWKGKs0ZhAgFChe
nfV2dMUwLyAAAABTXiWAEVBG4RTt82G0d/iWdw4RzT+O5uD5Ru/0F2rqhEAAAADtE4NI
/seImi6rqbjaBfogkvLgZspHEU5q0cVTNGbhgBHTlNDAnuiR3gBBxWSLixAIcWj0u5xfGcwHv
YOeuzYz</DTS:Password>
      </DTS:ConnectionManager>
    </DTS:ObjectData>

   </DTS:ConnectionManager>

  </DTS:ConnectionManagers>
```

Visual Mapper – Few important code classes listed here

Bus Controller Class

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using VisualMaps.DataAccess;

namespace VisualMaps.Web.Controllers.ApiControllers
{
    //[Authorize]
    public class BusesController : ApiController
    {
        private VisualMapsDataModel db = new VisualMapsDataModel();

        [Route("api/buses")]
        public IHttpActionResult GetBuses(int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;
            var buses = db.GetBuses(null, mins).ToList();

            if (buses == null)
            {
                return NotFound();
            }
            else
            {
                return Ok(buses);
            }

        }

        [Route("api/buses/{commaSeparatedBusList}")]
        public IHttpActionResult GetBuses(string commaSeparatedBusList, int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;
            commaSeparatedBusList = commaSeparatedBusList.Replace(" ","");
            var buses = db.GetBuses(commaSeparatedBusList, mins).ToList();

            if (buses == null)
            {
                return NotFound();
            }
            else
            {
                return Ok(buses);
            }

        }

    }
}
```

Corresponding Traffic Connector Class

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Text.RegularExpressions;
using System.Web.Http;
using VisualMaps.DataAccess;
using System.Xml;
using System.Xml.Linq;
using System.Text;
using System.Globalization;
using System.Xml.Serialization;
using System.IO;
using VisualMaps.Web.Models.CorrespondingTraffic;

namespace VisualMaps.Web.Controllers.ApiControllers
{
    public class CorrespondingTrafficController : ApiController
    {
        private static string CT_URL = 
"http://ibi.mermaid.dk/CorrespondingTraffic/Service.svc/GetCorrespondingTraffic?customerId={0}&b
usNumber={1}&stopNumber={2}&line={3}&dateTime={4}";

//http://ibi.mermaid.dk/CorrespondingTraffic/Service.svc/GetCorrespondingTraffic?customerId=2140
&busNumber=1304&stopNumber=9025200000052101&line=670&dateTime=2016-07-16T13:40

        private const int maxDeparturesPerLine = 3;

        private VisualMapsDataModel db = new VisualMapsDataModel();


        [Route("api/ct/{stopId}/{line}")]
        public IHttpActionResult GetCorrespondingTraffic(decimal stopId, string line, int mins = 
60, int maxDeparturesPerLine = 4)
        {
            mins = mins == -1 ? 99999999 : mins;
            // db.Database.CommandTimeout = 5;
            // db.Database.Connection.ConnectionTimeout = 180;
            string ct = FetchCorrespondingTraffic(2140, "1304", stopId, line, 
DateTime.Now.ToString("yyyy-MM-ddTHH:mm"), maxDeparturesPerLine);
```

```
            if (ct == null)
            {
                return NotFound();
            }
            return Ok(ct);

        }

        /// <summary>
        ///  TEMP Function
        /// </summary>
        [Route("api/ct2/{stopId}/{line}")]
        public IHttpActionResult GetCorrespondingTraffic2(decimal stopId, string line, int mins
= 60, int maxDeparturesPerLine= 4)
        {
            mins = mins == -1 ? 99999999 : mins;
            // db.Database.CommandTimeout = 5;
            // db.Database.Connection.ConnectionTimeout = 180;
            var ct = FetchCorrespondingTraffic_TEMP(2140, "1304", stopId, line,
DateTime.Now.ToString("yyyy-MM-ddTHH:mm"), maxDeparturesPerLine);

            if (ct == null)
            {
                return NotFound();
            }
            return Ok(ct);

        }

        #region Private Helper


        private string FetchCorrespondingTraffic(int customerId, string busNumber, decimal
stopId, string line, string dateTime, int maxDeparturesPerLine)
        {
            try
            {
                string url = string.Format(CT_URL, customerId, busNumber, stopId, line,
dateTime);
                string response = SharedLib.Connectiivty.RESTManager.Instance.GetString(url);
                response = System.Web.HttpContext.Current.Server.HtmlDecode(response);
                //response = SharedLib.Util.AppUtility.ExtractXmlFromTag(response, "string");
                response = response.Replace(Environment.NewLine, string.Empty);
                response = response.Replace("<string
xmlns=\"http://schemas.microsoft.com/2003/10/Serialization/\">",
string.Empty).Replace("</string>", string.Empty);

                response = response.Replace(Environment.NewLine, string.Empty);
                response = response.Replace("encoding=\"utf-16\"", "encoding=\"utf-8\"");
                XDocument xdoc =
XDocument.Load(SharedLib.Util.AppUtility.GenerateStreamFromString(response),
LoadOptions.PreserveWhitespace);


                StringBuilder html = new StringBuilder();
                StringBuilder ctHtml = new StringBuilder();
                XmlSerializer serializer = new XmlSerializer(typeof(CorrespondingTraficData));
                MemoryStream memStream = new
MemoryStream(Encoding.UTF8.GetBytes(xdoc.Descendants("CorrespondingTraficData").FirstOrDefault()
.ToString()));
                CorrespondingTraficData result =
(CorrespondingTraficData)serializer.Deserialize(memStream);

                var stopName = result.StopName;
                var stopNumber = result.StopNumber;

                //html.AppendFormat("<div><h4>Corresponding Trafic ({0})</h4></div>",
stopNumber);



                ctHtml.Append("<table cellspacing='2' cellpadding='2' class='ctTable'><tr><th
style='width: 30px;'>Type</th><th style='text-align:center'>Departure
at</th><th>Line</tH><th>Destination</th></tr>");

                List<Departure> departures = new List<Departure>();
                int totDepartures = 0;
                foreach (var tType in result.Departures.TransportType)
                {
                    foreach (var l in tType.Line)
                    {
                        foreach (var d in l.Departure)
                        {
                            d.Transport = tType._TransportType;
                            d.Line = l.line;
                            d.Destination = l.Destination;

                            totDepartures++;
                        }

                        departures.AddRange(l.Departure.Take(maxDeparturesPerLine));
                    }

                }

                foreach (var departure in departures.OrderBy(d =>
d.PlanedDepartureTime).ToList())
                {
                    ctHtml.AppendFormat("<tr><td>{0}</td><td style='font-size:16px; text-
align:center'><b>{1}</b></td><td>{2}</td><td>{3}</td></tr>",
                         GetTransportIcon(departure.Transport),
DateTime.ParseExact(departure.PlanedDepartureTime, "dd.MM.yyTHH:mm",
CultureInfo.InvariantCulture).ToString("HH:mm"), departure.Line, departure.Destination);
                }

                ctHtml.Append("</table>");

                if (totDepartures > 0)
                {
                    //html.AppendFormat("<div>Upcoming departures from <b>'{0}'</b><br
/><br></div>", stopName);
                    html.Append(ctHtml.ToString());
                }
                else
                {
                    html.Append("<h5>No Corresponding traffic available for this stop</h5>");
                }

                return html.ToString();
            }
            catch (Exception e)
            {
                return "<div> < br/>No data found at this moment";
            }
        }

        private string FetchCorrespondingTraffic_TEMP(int customerId, string busNumber, decimal 
stopId, string line, string dateTime, int maxDeparturesPerLine)
        {
            string url = string.Format(CT_URL, customerId, busNumber, stopId, line, dateTime);
            string response = SharedLib.Connectiivty.RESTManager.Instance.GetString(url);
            response = System.Web.HttpContext.Current.Server.HtmlDecode(response);
            //response = SharedLib.Util.AppUtility.ExtractXmlFromTag(response, "string");
            response = response.Replace(Environment.NewLine, string.Empty);
            response = response.Replace("<string 
xmlns=\"http://schemas.microsoft.com/2003/10/Serialization/\">", 
string.Empty).Replace("</string>", string.Empty);

            response = response.Replace(Environment.NewLine, string.Empty);
            response = response.Replace("encoding=\"utf-16\"", "encoding=\"utf-8\"");
            XDocument xdoc = 
XDocument.Load(SharedLib.Util.AppUtility.GenerateStreamFromString(response), 
LoadOptions.PreserveWhitespace);

            StringBuilder html = new StringBuilder();

            //get information summary

            //html.Append("<table><tr>");
            //foreach (var summary in xdoc.Descendants("NumberOf"))
            //{
            //    html.AppendFormat("<td>{0}</td><td>{1}</td>", 
GetTransportTypeName(summary.Attribute("TransportType").Value), summary.Value);
            //}
            //html.Append("</tr></table>");

            //html.Append("<hr />");

            var stopName = xdoc.Descendants("StopName").FirstOrDefault().Value;

            html.Append("<table style='width:100%'><tr><td colspan='4'><h4>Corresponding 
Trafic</h4></td></tr>");
            html.AppendFormat("<tr><td colspan='4'>Upcoming Arrivals for <b>'{0}'</b><br 
/></td>", stopName);
            foreach (var transport in xdoc.Descendants("TransportType"))
            {
                if (transport.Descendants("Line").Count() > 0)
                {
                    html.AppendFormat("<tr><td colspan='4'><b>{0}</b></td></tr>", 
GetTransportTypeName(transport.Attribute("TransportType").Value));
```

```
                    foreach (var dline in transport.Descendants("Line"))
                    {

//html.AppendFormat("<tr><td>{0}</td><td><u>Line:</u>{1}</td><td><u>Destination:</u>{2}</td><td>
({3})</td></tr>", dline.Attribute("subType").Value, dline.Attribute("line").Value,
dline.Attribute("destination").Value, dline.Attribute("numberofdepartures").Value);
                        html.AppendFormat("<tr>&nbsp;<td></td><td>Line: <u>{0}</u></td><td>,
Destination: <u>{1}</u></td><td></td></tr>", dline.Attribute("line").Value,
dline.Attribute("destination").Value);
                        int depart = 0;
                        foreach (var departure in dline.Descendants("PlanedDepartureTime"))
                        {
                            html.AppendFormat("<tr><td>&nbsp;</td><td><b>{0}</b> - </td><td
colspan='2'>{1}</td></tr>", ++depart, DateTime.ParseExact(departure.Value, "dd.MM.yyTHH:mm",
CultureInfo.InvariantCulture).ToString("HH:mm"));
                            if (depart == maxDeparturesPerLine)
                            {
                                break;
                            }
                        }

                        if (depart == 0)
                        {
                            html.AppendFormat("<tr><td></td><td colspan='3'>No
Departure</td></tr>");
                        }

                        html.AppendFormat("<tr><td colpan='4'><hr size='1'></td></tr>");
                    }
                }
            }
            html.Append("</table>");

            //return
xdoc.Descendants("TransportType").Where(d=>d.Attribute("TransportType").Value=="bus").FirstOrDef
ault().ToString(SaveOptions.DisableFormatting);

            return html.ToString();
        }

        private string GetTransportIcon(string t)
        {
            return string.Format("<img src='/img/transport/{0}.png' style='width:30px;'/>", t);
        }
        private string GetTransportTypeName(string transType)
        {
            string name = "Unknown Vehicles";

            switch (transType){
                    case "bus":{name= "Buses"; break;}
                    case "train":{name= "Trains"; break;}
                    case "s-train":{name= "S Trains"; break;}
                    case "metro":{name= "Metro"; break;}
                    case "ferry": { name = "Ferries"; break; }
            }

            return name;
        }
        #endregion
    }
}
```

Schedule Controller Class

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using VisualMaps.DataAccess;
using VisualMaps.Web.Models;

namespace VisualMaps.Web.Controllers.ApiControllers
{
    public class SchedulesController : ApiController
    {

        private VisualMapsDataModel db = new VisualMapsDataModel();


        [Route("api/schedules/{scheduleId}/stops")]
        public IHttpActionResult GetScheduleStops(int scheduleId, int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;
            // db.Database.CommandTimeout = 5;
            // db.Database.Connection.ConnectionTimeout = 180;
            var stops = db.GetScheduleStops(scheduleId, mins).OrderBy(js =>
js.StopSequence).ToList();
            var buses = db.GetBusesByScheduleDestination(scheduleId, null, 90);
            if (stops == null)
            {
                return NotFound();
            }

            var stopsAndBuses = new object []{stops, buses};

            return Ok(stopsAndBuses);

        }

        [Route("api/schedules/{scheduleIds}")]
        public IHttpActionResult GetSchedules(string scheduleIds, int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;
            var stops = db.GetSchedules(scheduleIds, mins);

            if (stops == null)
            {
                return NotFound();
            }
            return Ok(stops);

        }
        [Route("api/schedules/list")]
        public IHttpActionResult GetScheduleList(int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;
            var stops = db.GetSchedules(null, mins).Select(s => new
            {
                ScheduleId = s.ScheduleId,
                Name = "("+s.ScheduleId+") - " + s.FromName + (s.ViaName != null ? (" > " +
s.ViaName) : "") + " > " + s.Destination,
                Line = s.Line
            }).OrderBy(s => s.Name);

            if (stops == null)
            {
                return NotFound();
            }
            return Ok(stops);

        }


        [Route("api/schedules/{scheduleId}/buses")]
        public IHttpActionResult GetBusesByScheduleDestination(int scheduleId, int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;

            var buses = db.GetBusesByScheduleDestination(scheduleId, null, mins).ToList();

            if (buses == null)
            {
                return NotFound();
            }
            else
            {
                return Ok(buses);
            }

        }

        [Route("api/schedules/{scheduleId}/buses/{commaSeparatedBusList}")]
        public IHttpActionResult GetBusesByScheduleDestination(string commaSeparatedBusList, int
scheduleId, int mins = 60)
        {
            mins = mins == -1 ? 99999999 : mins;
            commaSeparatedBusList = commaSeparatedBusList.Replace(" ", "");
            var buses = db.GetBusesByScheduleDestination(scheduleId, commaSeparatedBusList,
mins).ToList();

            if (buses == null)
            {
                return NotFound();
            }
            else
            {
                return Ok(buses);
            }
        }

        /// <summary>
        /// 
        /// </summary>
        /// <param name="timeUnit">if type=hourly then hours, if type=daily then days</param>
        /// <returns></returns>
        [Route("api/stopHistory/{stopId}/{type}/{scheduleId}")]
        public IHttpActionResult GetStopHistory(decimal stopId, string type = "daily", int?
scheduleId = null, int timeUnit = 7)
        {

            if (type == "daily")
            {
                var stopHistory_d = db.GetStopHistory_Daily(stopId, scheduleId,
timeUnit).Select(s => new
                    {
                    //DateTime.Parse(s.Date).ToString("yyyy, MM, d") ,
                    s.Date,
                    s.LateArrivals,
                    s.EarlyArrivals,
                    s.OnTimeArrivals
                    }
                    ).ToList();
                if (stopHistory_d == null)
                {
                    return NotFound();
                }
                else
                {
                    return Ok(stopHistory_d);
                }
            }


            if (type == "hourly")
            {
                var stopHistory_h = db.GetStopHistory_Hourly(stopId, scheduleId,
timeUnit).Select(s => new
                    {
                    s.Date,
                    s.LateArrivals ,
                    s.EarlyArrivals ,
                    s.OnTimeArrivals
                    }
                ).ToList();

                if (stopHistory_h == null)
                {
                    return NotFound();
                }
                else
                {
                    return Ok(stopHistory_h);
                }
            }
```

```
                return NotFound();

            }



        }
    }
```