T.C.

ALTINBAŞ UNIVERSİTY

Institute of Graduate Studies
Electrical and Computer Engineering

TRAFFIC DETECTION USING SDN-IOT BASED MACHINE LEARNING NETWORK

Ali Khalid Hashim AL-ZUBAIDI

Master Thesis

Supervisor

Asst.Prof.Dr. Oguz KARAN

Istanbul, 2021

TRAFFIC DETECTION USING SDN-IOT BASED MACHINE LEARNING NETWORK

By

Ali Khalid Hashim AL-ZUBAIDI

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

In partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY
2021

The thesis titled “Traffic Detection Using SDN-IOT Based MACHINE LEARNING NETWORK” prepared and presented by “Ali Khalid Hashim AL-ZUBAIDI” was accepted as Master of Science Thesis in Electrical and Computer Engineering.

_____

Asst.Prof.Dr. Oguz KARAN
Supervisor

Thesis Defense Jury Members:

Asst.Prof.Dr. Oguz KARAN	Faculty of Engineering and Natural Sciences, Altinbas University	_____
Asst.Prof.Dr. Abdullahi Abdu IBRAHIM	Faculty of Engineering and Natural Sciences, Altinbas University	_____
Asst.Prof.Dr. Zeynep ALTAN	Faculty of Engineering and Architecture, Beykent University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies
____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ali Khalid Hashim AL-ZUBAIDI

Signature

DEDICATION

To My Mother who left this life fast strong and gentle soul who taught me to trust in hard work, and that so much could be done in a little, I will pray for your soul and hope we meet under the sky of God's mercy, thank you for being my first teacher.

To My Father Who earn an honest living for us and for supporting and encouraging me to believe in myself, for being my advisor all the journey of life, thank you for your endless love and prayers.

To My Sisters, (H-H-D-S-G), Thank you all for always being there when I need you Thank you for supporting and all the love.

My Wonderful Wife Thank you for being such supportive to me, thank you for standing by my side, and for your patience and love.

ACKNOWLEDGEMENTS

I would like to thank my university Altinbas for affording me the un-imaginable opportunity to complete my study here, despite all challenges which had frustrated all my efforts. I will not forget to thank Dr. Oguz KARAN, whom for without his patience, guidance and understanding, I wouldn't have made it this far, especially with my dissertation, I also remain grateful and thankful for all my teachers whom have managed and guided me throughout my time of study here, you are all special to me. Finally, and most importantly huge thank to Almighty God, for his grace in me.

ABSTRACT

TRAFFIC DETECTION USING SDN-IOT BASED MACHINE LEARNING NETWORK

AL-ZUBAIDI Ali

M.Sc. Electrical and Computer Engineering, Altinbas University

Supervisor: Asst.Prof.Dr. Oguz KARAN

Date: 16/6/2021

Pages: 75

In recent years, IoT (Internet of Things) techniques have experienced significant progress with application in different areas of daily life, including the monitoring of public spaces to achieve better planning and use of resources in this paper we use MATLAB to simulate IOT-SDN base network and train the ML network to detect optimal routes and traffic.

Keywords: SDN, IOT, ML, Traffic Detection

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

1. INTRODUCTION

1.1 BACKGROUND

In recent years, IoT (Internet of Things) techniques have experienced significant progress with application in different areas of daily life, including the monitoring of public spaces to achieve better planning and use of resources. This is the main objective that is addressed in this work, where a conceptual solution is proposed on the application of IoT methodologies to monitor the traffic of pedestrians, corridors and bicycles in the lanes enabled. By IoT we understand all those solutions that are based on processing information acquired locally through sensors (cameras, for example) in an intelligent system. These systems normally manage data from other devices in a centralized way, being able to decide to combine them with each other or with data from other sources, and who are in charge of publishing the data collected for later access through the connections established for that purpose, either on a public basis, or at a private level by the administrations involved. [2]:

Figure 1.1: Simple Computer Vision System.

1.2 MOTIVATION

If we go back to 1990, we will find the first device that can be considered belonging to the IoT, a toaster created by John Romkey that could be turned on / off via the Internet 1. Prior to this, there were already authors and professionals who 1 pointed to what is now known as IoT, yet the term IoT was not used publicly until 1999 at a conference on RFID organized by Kevin Ashton1 2. It is expected that by 2021 there will be around 33 billion IoT devices, reaching 1 billion devices by 2035 1. The IoT has numerous applications in people's daily lives; one of these is in the field of smart cities. These solutions are designed to be deployed in an urban environment, allowing improving the services offered to citizens, generating an improvement in their quality of life. By smart city is understood one that uses information and communication technologies (ICT) to offer better infrastructures and services to its inhabitants. The term smart city was used for the first time in the 1990s 3. In Spain, there is a National Plan for Smart Cities, drawn up in 2015 4, the main objective of which is to improve the efficiency with which entities offer services related to ICT technologies, in addition to the objectives of ICT technologies such as improving the quality of life of people, reduce the expenditure of resources, etc..

1.3 PROBLEM STATEMENT

As previously indicated, this project proposes a conceptual solution with a view to its possible real implementation in the future. Technologically, the problem is addressed by considering a camera that captures images related to transits through a certain point, proceeding to identify the type of element in transit (bicycles, pedestrians, corridors), at a certain moment, using deep learning techniques specifically using Convolutional Neural Networks (CNN), obviously under the coverage of the current Data Protection Law in terms of preserving the identity of the people who appear in the images. Once the pedestrian traffic data have been obtained, it is possible to publish these data for information purposes both at the user level and for use by public administrations, incorporating prediction techniques regarding their use at certain

times. These data could help the agencies in charge of maintaining these public sites to choose the ideal times to carry out maintenance work (when there are fewer people, for example), or help identify problems that generate a decrease in pedestrian traffic . The motivation for this project is given by the speed and convenience offered by IoT technologies when studying pedestrian traffic. Having a tool that offers agility when carrying out studies and management of pedestrian traffic would be beneficial, especially taking into account the growing trend of the world population 5. In this context, the contribution made by this work focuses on the one hand on the study of the different methodologies at the individual level, to carry out their integration into a global system under the IoT paradigm. [1], [2]:

Figure 1.2: Basic Principle of Video Segmentation

1.4 CONTRIBUTION

There are various image segmentation techniques, and in this work a medical image segmentation procedure based on the metaheuristics of Ant Colony Algorithms is proposed. The algorithms of this metaheuristic mimic the behavior of ants during their search for food, since they always produce optimal routes between the food source and the nest. This behavior was implemented using artificial ants in order to perform specific image processing tasks.

The main objective of the project is the development of a conceptual IoT solution for monitoring pedestrian traffic in public spaces. This objective also includes the design, implementation and testing with the developed application. Taking into account the above, under this objective the study of the basic principles to be applied, the formulation of a technological concept, as well as an experimental proof of concept with laboratory-level validation of the proposed modular integration is proposed, this being, for Therefore, the degree of maturity of the proposed project. The main objective can be broken down into a series of specific objectives: Design of the application architecture and Application of motion analysis techniques in video sequences. The application monitors the flow of people walking, running or cycling in a certain scenario, it is necessary that the application can analyze the movement of different people performing the indicated activities, after capturing and analyzing the scene by the application of convolutional neural network techniques. The application classifies between the different types of pedestrians, so it is necessary to have a powerful classifier. Convolutional neural networks are used to develop this classifier. This objective includes the learning and implementation of this type of networks and Development of a results analysis system in the cloud. The application provides a series of results on the monitoring it carries out; these results are uploaded to a cloud platform that allows an exhaustive analysis of them. The analysis proposed in this objective includes the use of prediction techniques, either

in the cloud or at the local level. Integrate the different modules and technologies to form the IoT application, including the validation of the proposed conceptual solution at the laboratory level...

1.5 THESIS ORGANIZATION

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. Section 4 is where we explain our model in details and implementation of that model in details, Section 5 is where we simulate our model and record the results obtained. Section 6 is where we conclude our work and put a future scope into perspective.

2. STATE OF THE ART

When it comes to controlling pedestrian traffic, there are a number of existing solutions that address it from different perspectives. A number of applications can be found, such as the People Counter offered by Mirame.net 6. The case of the city of Dordrecht can also be highlighted, although in this case the people counting is a solution that is part of a broader one, the city assumes the form of a Smart City with an ecosystem of different services. Details regarding both examples are provided below, founded in 2001, is responsible for offering a range of technology solutions to European customers. Among the solutions they offer is the People Counter. People Counter is an application that offers business owners different information about their clientele. It allows owners to keep track of the people entering and leaving their store, as well as different analyzes on age, gender or trends of said customers. The counting application is integrated directly into IP Standard network cameras, reducing costs since bandwidth is not consumed in the capture and analysis of the images. Dordrecht is a Dutch city located in the west of the Netherlands, with approximately 118,900 inhabitants, according to 2015 data.

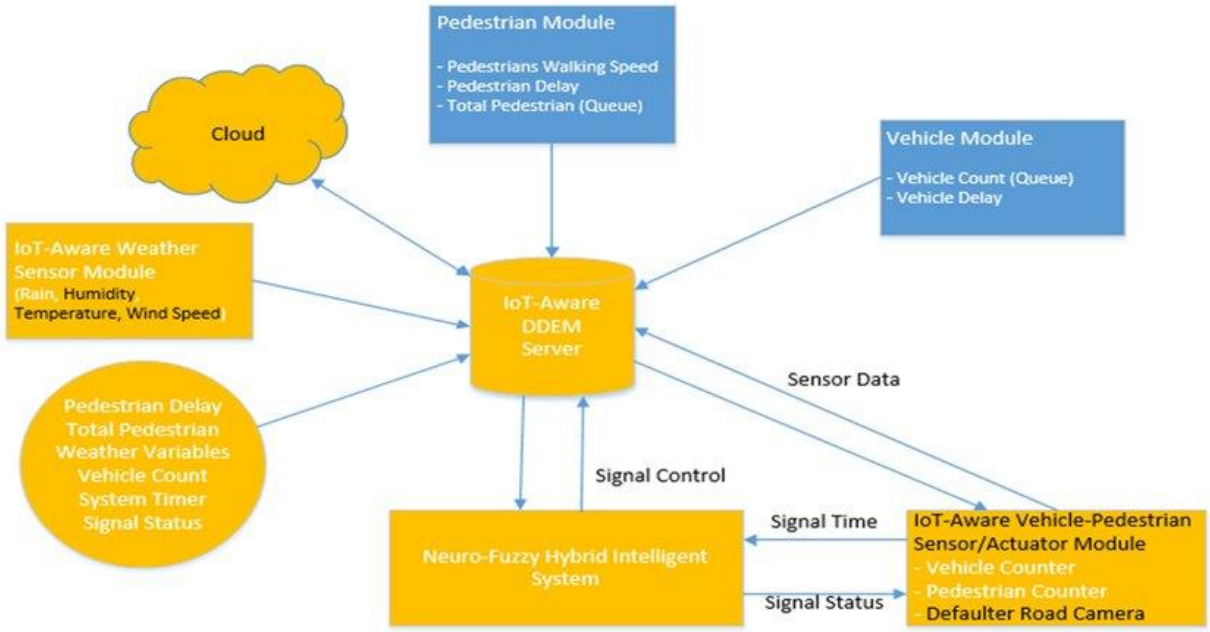

Figure 2.1: IOT Cloud Data Using Deep Learning

Vehicle and pedestrian traffic detecting the volume of both cars and pedestrians allows us to know which are the busiest streets and the times of pedestrian crossing. This project was carried out by Dimitrios Kyrytsis as part of his to develop the project, he installed at the crossroads of several streets a series of Meshlium IoT Gateways equipment developed by Libelium that received data from sensors and allowed them to reach a platform for their management. The devices detected the Mac addresses of smartphones and hands-free equipment present in cars by scanning the Wi-Fi signal. From these data and information on distances between sensors, the speed of movement of each of the devices was calculated. Depending on the speed, the devices were identified as elements being carried by three different types of users: pedestrian, cyclist and vehicle. This classification allowed us to observe which streets were the most traveled by each of the types of users. The experiment was carried out between September and October 2016 with the aim of improving city planning. As far as it has been possible to investigate these two are the closest solutions found4 in relation to the approach carried out in the present work. As can easily be deduced, the objective is similar, although the solutions provided are different, including the scope and context of application. That is why the proposed IoT solution proposes a certainly novel conceptual solution.

3. MATERIALS AND METHODS

3.1 INTERNET OF THINGS (IOT)

The Internet of Things (IoT) is a concept that is having a lot of success nowadays and everything is due to the expectations generated by being able to connect any object to the Internet and do whatever you can think of. The possibility of assigning computing capacity to everyday objects and giving them a certain intelligence to collect data from their environment and communicate with each other can create numerous applications that can completely change people's lives. Although it seems that we are talking about something that has been with us for a long time given its popularity in recent days, the reality is that it is a very recent concept. It was not until 2009, when Professor Kevin Ashton used the expression “Internet of Things (IoT)” for the first time, since then it has grown and a huge expectation has been created around the term that has grown exponentially.

In order to fully understand the concept, we must go back to the different technological evolutions that have led to what we know today as IoT and that have brought us to this point since the idea of being able to connect objects and make them intelligent existed for decades. It was not until the 1960s and, above all, the 1970s when the first communications protocols were created that would define the basis of what the Internet is today. When the Internet became popular, it did not take long for the idea of connecting objects to take advantage of this new network to return and it was in 1990 when John Romkey created the first object connected to the Internet: a toaster that could be turned on or off remotely.

Despite this vision of understanding networks representing a revolution, the communications that the Internet offered in those years were mainly wired. This, and related to the fact that the cost of the hardware was still quite high, made the concept of implementing connected objects go unnoticed for a few more years. The revolution and finally the popularization of this concept came from the hand of the

wireless connectivity that allowed increasing the growth of connected objects. This growth has been especially exponential by the emergence of other new technologies such as Wireless Sensor Networks (WSN) or Machine to Machine communications. (M2M).

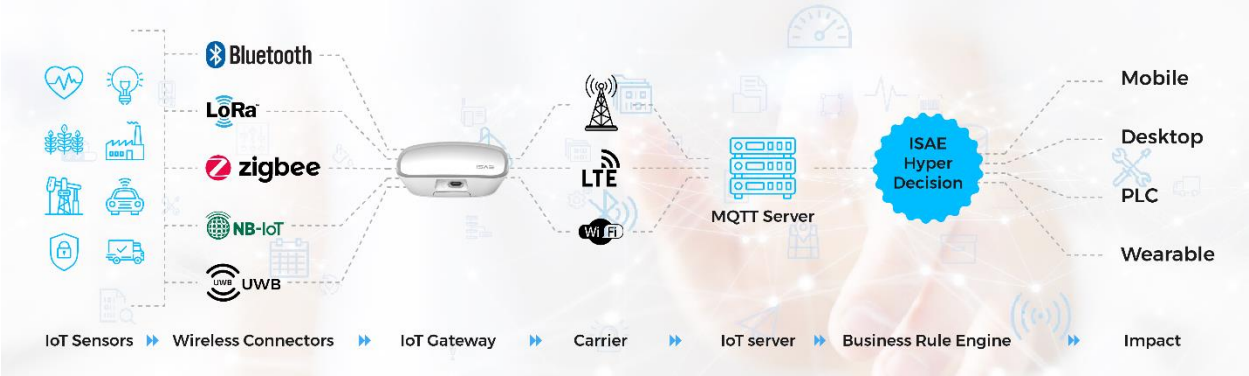

Figure 3.1: How IOT Works

3.1.1 Smart Cities

One of the main applications of the Internet of Things, which defines how each object and physical element of a city can reach it interconnected and generate useful information. These objects can be any everyday item such as cars, roads, traffic signs, traffic lights, homes or lighting. By implementing adaptable services that monitor all this information that is generated and its evaluation, it is possible to give a certain autonomy or intelligence to the city with a capacity to adapt itself to the changing environment that surrounds it.

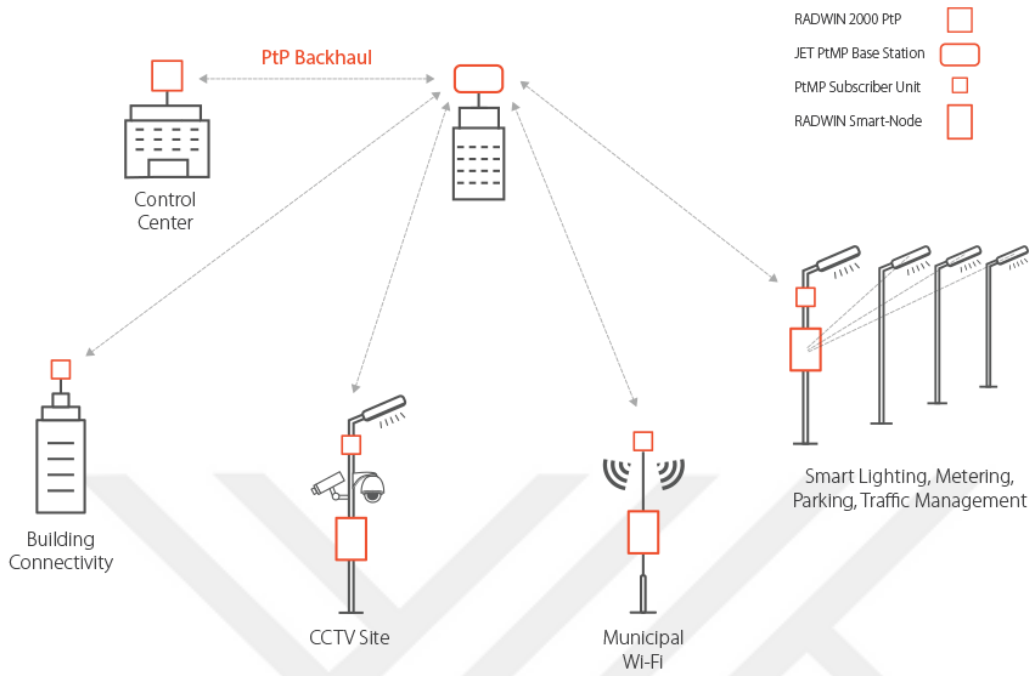

Figure 3.2: Smart Cities

3.1.2 Connected Cars

The concept of the connected car is not new as there are many examples of vehicles that are integrated into communication networks through interconnection with other mobile devices. But still, one of the challenges that remain to be met is the manufacture of vehicles that use these new technologies to facilitate driving and increase safety. The objective to be achieved in the future is the existence of autonomous driving of vehicles without any intervention from the driver. This would be achieved largely thanks to the direct connection to the network to produce and consume information in real time.

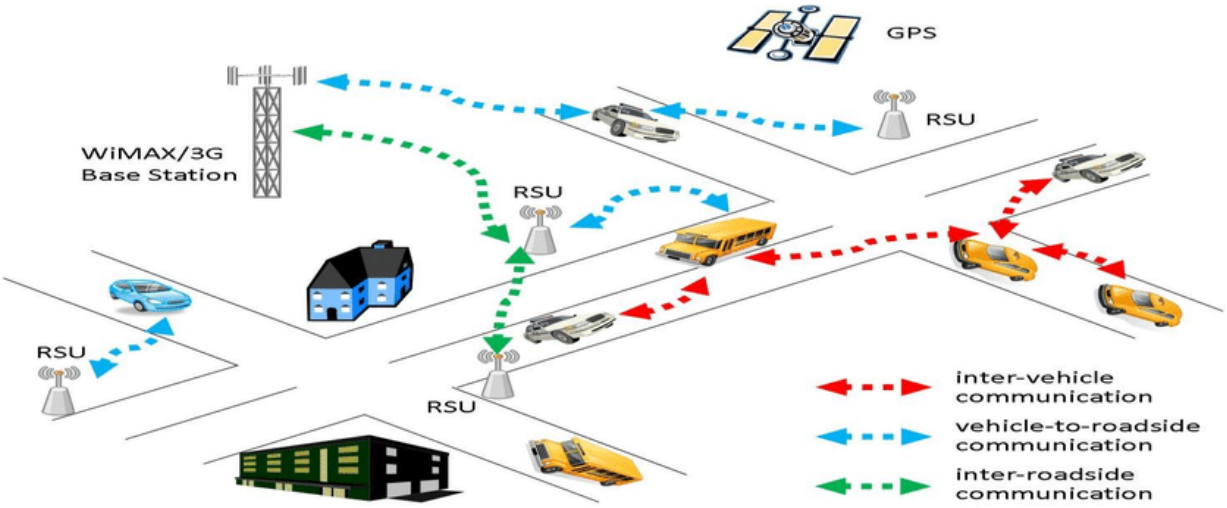

Figure 3.3: VANET Connected Cars

3.1.3 Traffic Management

Regarding the autonomous management of a city, it is first necessary to have a platform that integrates a Smart City, where all its elements consume and produce useful information through the same type of communication in a language known to all others. Second: of connected and therefore intelligent vehicles with the ability to communicate with their road infrastructure to make decisions and adapt their behavior autonomously depending on their environment. When these two applications are already available, it is possible to develop solutions for cities so that they can offer different services to users and regulate themselves to adapt to different needs such as the traffic situation and the appearance of emergencies or incidents on their roads.

3.2 SMART TRAFFIC

State Transfer technology, or commonly called REST by its acronym, describes how a system can communicate its state to other systems. Status refers to a product that is generally called a resource and access to its characteristics such as its name or description. REST is commonly associated with web service interfaces, since HTTP is by far the most widely used transport protocol in this technology thanks to its great ease of accessing resources.

A RESTful application is an application that exposes its state and functionality as a set of resources that clients can manipulate and modify according to a set of principles: Resources are unambiguously identifiable and addressable through an address, usually URI, although there are also other types of possible addresses. All resources can be manipulated through a restricted set of well-known actions, generally CRUD (creates, read, update and delete) normally represented through the HTTP methods: POST, GET, PUT and DELETE. It is not necessary to develop all the HTTP methods on the resources since, if you want to restrict or limit the access or modification of a resource, it is only necessary to implement the methods that perform the operation. The data of the resources is transferred through any of the most popular representations such as XML, JSON or HTML. The IoT platform that is used as the basis of the project has a fully defined REST API that shows the Smart City resources and their communications. Likewise, extensions of the solutions that make use of these communications are developed in the project.

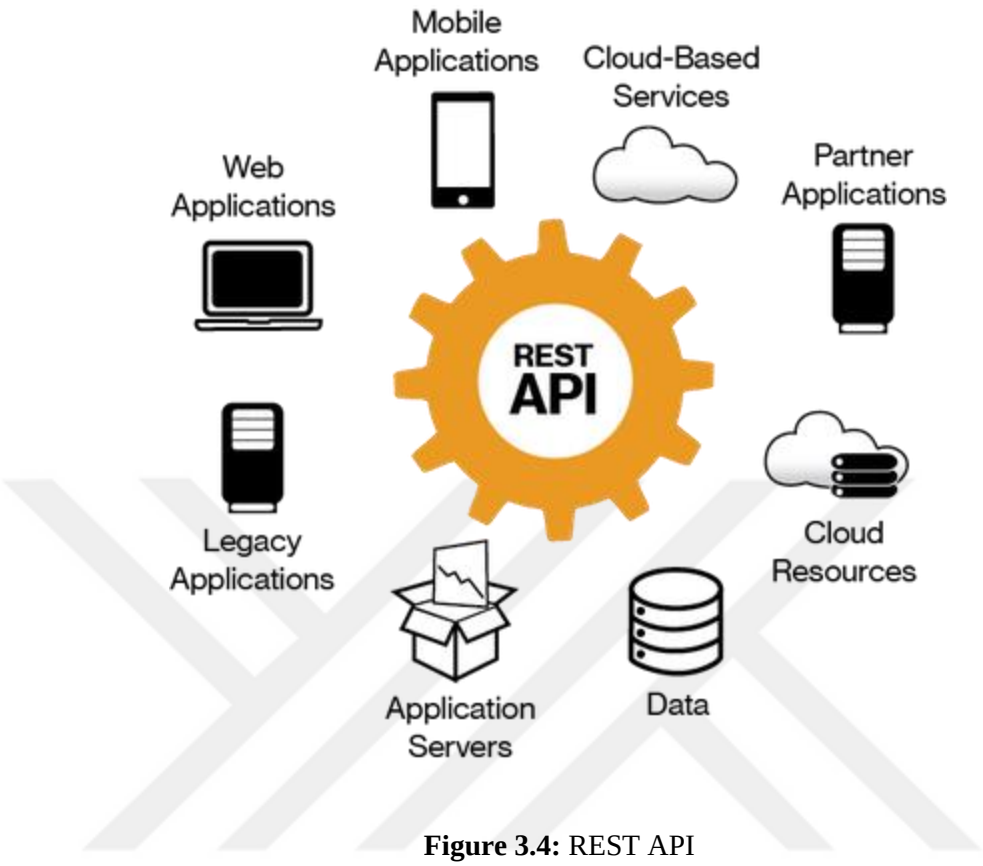

Figure 3.4: REST API

The Smart Traffic platform is the basis that is used in the project and from which it starts to develop and implement adaptive services that modify the behavior of its systems autonomously and attending to the needs of its environment at all times.

This platform is provided to me already developed by the project tutor, Joan Fons, with the aim of extending its adaptive capacities. It proposes a service prototype that integrates different resources and services in the field of smart city traffic, offering a platform that uses the concepts of the Internet of Things to integrate all types of resources that are different from each other. The elements and resources (or also called Smart Things) that it offers are the following: Smart road network: they represent the roads or streets of a map of an area or smart city. They are called within the platform with the name of Roads and they give all the information about their status in real time such as the saturation of the road, the

number of vehicles currently circulating, complications and incidents, traffic restrictions or traffic jams. Intersections they have.

Traffic Signs: represented by traffic lights, speed limits or temporary incidents, each traffic signal is positioned on a road segment called Road Segments, at an exact and specific kilometer point. For the presentation of the scenarios, a physical model is used that represents these road signs and their different modes of operation. Smart Connected Vehicles: represented by connected vehicles that have an intelligent behavior throughout the development of solutions thanks to the fact that they can connect to the road infrastructure to know the state of the roads and signaling and adapt according to their needs. In addition, they can provide information on incidents such as accidents or breakdowns. Incidents: that occurs on roads such as construction sites, accidents or blocked streets for example and, in addition, on other elements such as vehicles with technical failures that may cause some complication on the correct circulation and traffic flow. Simulator: This provides us with the interface and mechanisms to help reproduce realistic scenarios that could be applied to real life situations.

The simulator has the power to generate information about the entire infrastructure, being able and having the ability to create virtual vehicles that circulate through it, handle traffic signals such as traffic lights or generate incidents of all kinds. With this, it is intended to see and show that this is the platform from which it starts and that it is used in the project to design solutions and IoT scenarios. At the time of starting the project, the behavior of the systems was completely independent and there was no type of autonomous and self-management service working, but there were needs that suggested an adaptation of the systems to improve their situation. To do this, it must be taken into account that the limiting factor of the project and the greatest challenge it faces in its development is that it is not possible to access the implementation of each of the services or resources that make up the Smart City. Therefore, the idea is to

develop independent modules using the OSGi framework and communicate them with each of the systems to adapt them to their needs through the communication methods already available on the platform.

Finally, the annexes show how to configure the simulator and start up the solutions that will be explained later. The road infrastructure that is being used in the project under the simulator of the Smart Traffic platform is that of the Paterna Technology Park, which from now on we will name as PTPaterna. The design is fully prepared to allow both vehicles and roads to stay connected and exchange information. Undoubtedly, the choice to work on this road network and this map is not random since this infrastructure is very contained and has all the elements that are needed for the simulation, since it allows us to show, design and implement any scenario of an easier way. However, the solutions developed are not limited only to this scenario but are prepared to work under any road map without altering at any time the implementation necessary for its operation.

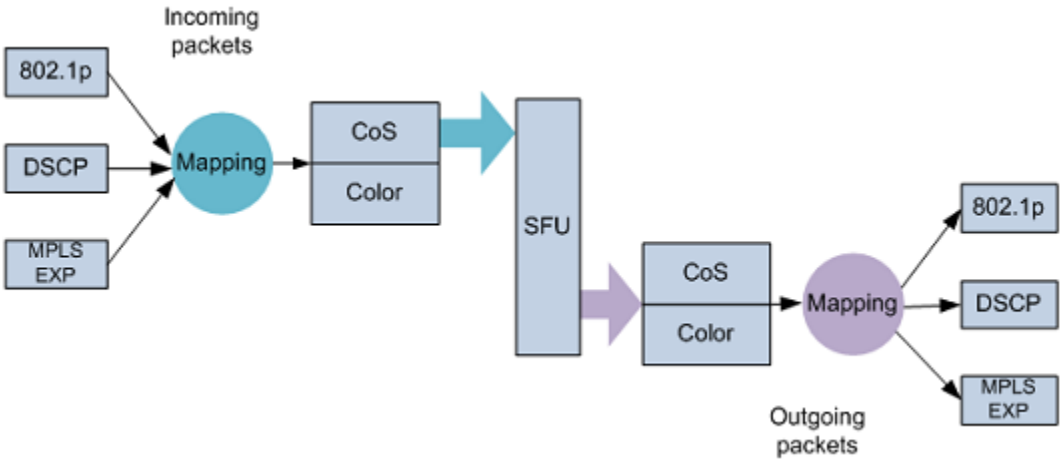

Figure 3.5: Traffic Through DPI

3.3 SOFWARE DEFINED NETWORK

More or less, the internet is made up of traditional or developed networks. Legacy networks handle routing and data forwarding, such as switches and routers. Traffic routing is done in part by a network configuration that depends on partial knowledge of the network. Legacy networking has been adopted. [Being] standardized; a network interface is typically made up of thousands of proprietary codes with no space for customization Traditional networks are perhaps the biggest problem of all because of all. Over the years, specifications for the network have been refined. As design and connectivity requirements develop the more complex the network devices become.

New initiatives are difficult to enforce in conventional environments, too. If you want to add a new system or a new service to an existing network, the necessary configuration involves having to be done on several network devices. It will take a lot of time and money to accomplish. The traditional networking world obviously falls short with the rising tide of IoT devices.

It's now the current flavor of the month: Recent times have shown a strong preference for software-defined networking. A good deal of research has been done in recent years to understand how we can take advantage of SDN's traffic management functionality and how IoT devices can use it to communicate. SDN is providing SDI by separating it into layers. A network control unit creates a model that controls the network devices.

Compared to conventional networks, SDN has a number of advantages. 6 Decoupling of control and data planes, as a result of SDN The network control manager has a separate controller that controls the forwarding switches, also known as a 'a fat controller' The forwarding devices in the data plane have no intelligent technology or settings. Human-built machines may be designed to be either intelligent or non-intelligent. I like this idea a lot. It enables administrators to build networks that are more tailored to their

needs. Additionally, this type of network has a global view as opposed to a legacy network, in which only some segments are looked at as nodes, rather than a network are examined. The controller is capable of making decisions that use network resources effectively.

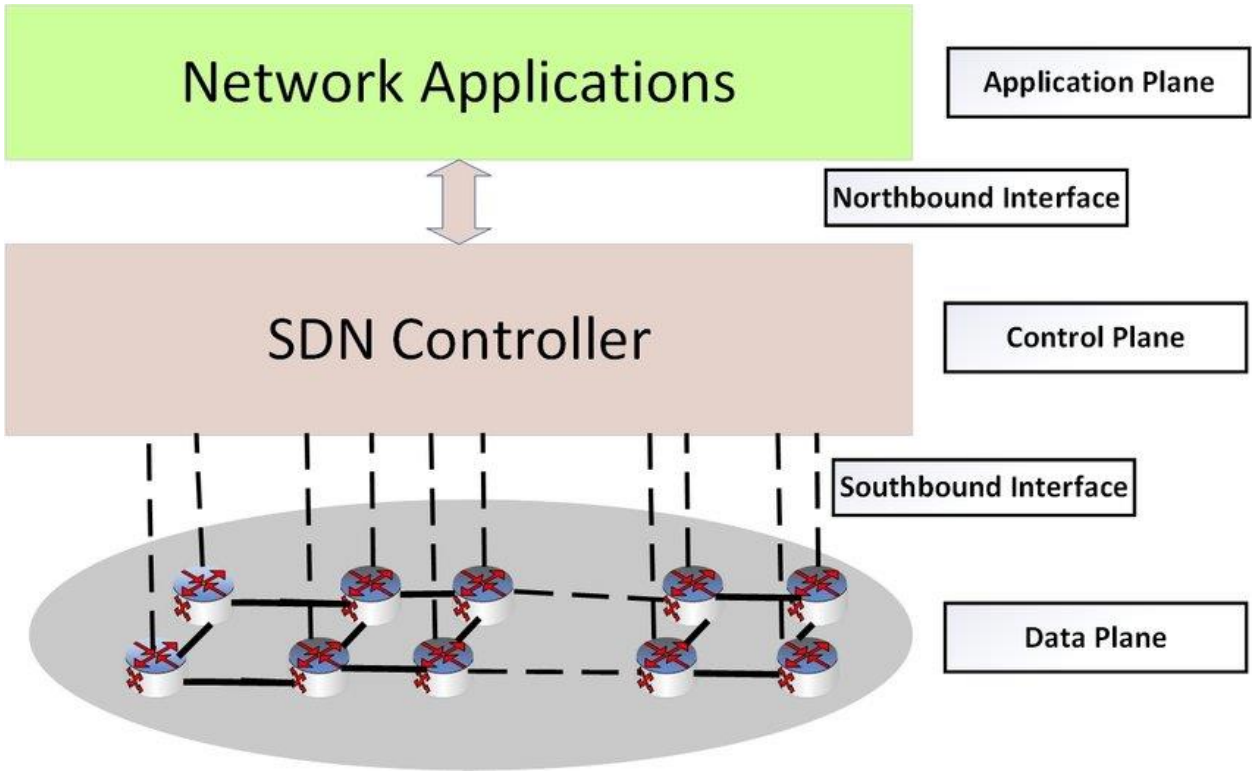

Figure 3.6: Software Defined Network SDN

3.4 SDN ARCHITECHTURE

The company has three basic layers. Control, info, and switching planes Each of these planes has its own tasks to-be-done roles. The forwarding plane houses the data. Controller refers to the whole thing. It's in the middle of the data layer and the application layer. This layer handles the thought process. An implementation of policies articulated by programs, and implemented on the data plane. Southbound APIs

communicate with the data switches and northbound APIs. That is where the apps, the apps on the network, reside.

The virtual hosts allow applications to run on both physical and virtual hosts. With full control of the network, the controller here has all the information. The product has a global perspective. Scalability problems usually mandate the use of controllers in an application. Often more than one switch controllers are typically used in networks of this size. Network architecture known as Distributed controller The controller optimizes flow to determine the packets can travel the shortest path through the network by using a global view. The controller instructs the switches to set up flow tables on them and informs the packet-forward ingresses where each packet is to be forwarded to. The Open Flow process uses a peer-to-peer networking model. The header field, action field, priority field, timer field, and counter field are included in the flow table Below SDN architecture is shown in the diagram.

The software-defined networking approach allows the network a great deal of adaptability. Policies and simply can be applied rapidly. It becomes much easier to defend against attacks as this occurs. Artificial intelligence also can be used in the software-defined networking environment. The features of SDN include: It has become much simpler to implement custom applications because of the virtualization offered by the control plane. 8 Integration is easy and seamless. In addition, it is also possible to effectively separate the traffic into different network segments, making load balancing simple.

If the switches and routers become less costly, the network costs are reduced. There are no compatibility issues to take into consideration and thus integration is simple Also, SDN has some strengths. It's a big worry that there might be a single point of failure. Since SDN is built on centralization, the network is vulnerable to going down when the brain is knocked out. Single point of failure (SPOF) also poses a huge security threat.

Targeting the controller would mostly push the network down. The additional problem is scalability. As the network grows, no one device will be able to perform all the duties of the network. The size of the network increases, more controllers are added. The tricky part of adding more controllers is deciding where to put them. The controllers do not know where they are supposed to be placed. There are a number of different ways of handling the controller placement issue, including approaches like the ones suggested in [10] and [11]. Now that SDN networks have gained more research attention, concerns about security have come up.

3.5 MACHINE LEARNING

We've created the ability for computers for learn on their own. He called it "The study of algorithms that offer computers the ability to learn without being programmed." This decision was reached based on earlier, out-of-of-the-the-the-box thinking. Tom Mitchell offered a more modern interpretation of learning: "The output of a computer program, as calculated by tasks in T, would increase with experience." machine learning, or artificial intelligence, is the scientific study of algorithms and models that use pattern and logic to help computers perform a certain task Supervised, unidirectional and semi-supervised and then reinforced learning is essentially cover the same ground. For the more information on these algorithms, please see the following lists. Supervised machine learning consists of providing “training data” and a “prediction relationship” to the system to learn. It is then influenced by new data and takes new actions based on what has been learned about the relationship. Data is “labeled' for preparation". Problems with supervised learning are classified as “regression and” while in regression problems, the model forecasts values; in classification problems, the results go from continuous to discrete. Decision trees, random forest, neural networks, and help vector machines are among the more widely used supervised learning techniques although supervised machine learning is regulated, unsupervised machine learning is uncontrolled.

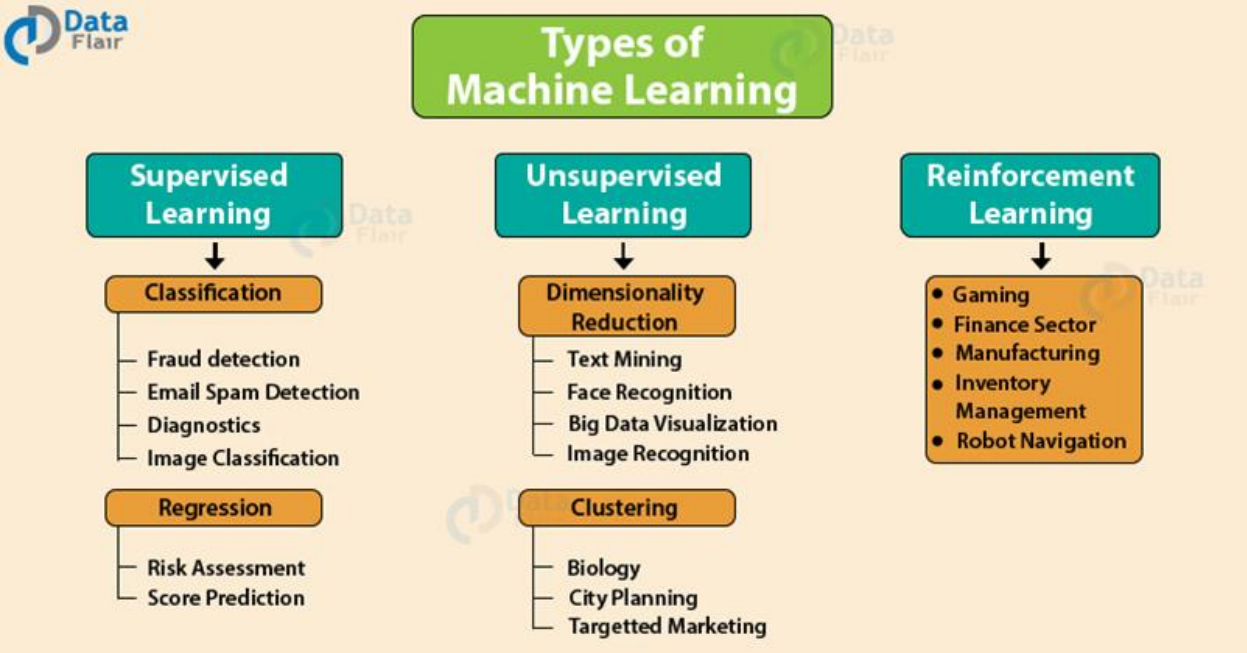

Figure 3.7: Types of Machine Learning

In this method, the inputs are absent the associated with the expected outputs. Clusters the data based on what they have in common, which are "features" These kinds of algorithms, among others, are often used in creative work. Creative ideas In between full and unsupervised learning, there is semi-auto learning. We are modeled as part of the input data and the remainder are not (unlabeled). Other common machine learning methods include transductive Support Vector Machines (SVM) and graph-based methods, among others.

It is the role of encouraging agent systems to take sequences of acts that result in greater rewards. He is a person who empathetically communicates with his environment to discover the best responses to long-term benefit. 10 Machine learning has been used in many endeavors, including bettering human life. Here are many examples, including natural language processing (NLP), diagnosis of diseases, market segmentation, and facial recognition for products. The standard machine learning application consists of

four steps: data collection, feature engineering, model choice, model training, model deployment, and model preparation. Tuning and testing the model deriving new findings from the model.

3.5.1 Data Collection and Preprocessing

One important component of developing machine learning models is data collection and planning. The machine learning problem is defined. Depending on the research issue, there are several datasets available on the Internet. Only data that has been already cleaned and preprocessed is being passed on to make sure the model is being fed with relevant information to help solve the problem.

3.5.2 Feature Extraction

The most crucial part of the machine learning process is feature selection and optimization. A feature is a general characteristic that can be observed about an event or phenomenon it may help in making a forecast or being placed in a category. The domain knowledge of the data is applied to feature engineering to build features that allow machine learning algorithms to do their job these new features are designed to enhance the model's accuracy. In data feature engineering, you can combine datasets to produce better features Redundetailing tries to reduce or strip out unnecessary and redundant attributes to increase the ability to learn. With feature selection, the algorithm's accuracy and speed are increased. Reduced number of features has in favor of low 11 computational overhead. There are three major feature selection techniques: region increasing, random feature selection, and robust feature selection. They are identical:

Since classifiers by themselves do not influence the results, feature rank techniques rank features based on the inherent features [42, 44]. Thus, they use a scoring system to give numerical values to the variables and filter out those that are not essential; there are two types of filtering: univariate and multivariate. Function dependencies are lost in univariate methods, and they must be accounted for in multivariate methods. However, while being fast, scalable, are unable to provide accurate results because there is no

interaction with the classifier. Examples of filter methods include knowledge gain, feature selection, Markov random forest (MBF), and features and classes combination (SCF)

In these methods, the indicator is used as a black box and the output as the objective. Given a search protocol, different subsets of features are constructed and examined. Training and evaluating a function classifier decides on a particular subset of features' this algorithm, unlike traditional approaches, assesses the algorithm's classification capabilities. You should think of a disadvantage as an opportunity cost savings. Sequential search algorithms and heuristic search engines are used in creative search engines.

The learning algorithm is incorporated into the code. It relates features and properties to the use of input as well as those of output. Using the independent criteria to choose the optimum subset, the system searches over 12 subsets to find the most optimal one for the defined cardinality Wrapper approaches include less computation.

Filters and wrappers are commonly used in series to classify related features. These techniques are labeled as hybrid. In most cases, filters are employed when the number of features is important. This dataset will be time-based, so we will use wrapper and embedded feature selection approaches. Tree-based methods based on a flexible classifier and strong importance features like SHAP [found in this paper] were used for our classification problem. We favor these two strategies because they have proved their worth in the market.

3.6 CHOOSING A TRAINING MODEL

Next move is picking the algorithm that provides the best results. There are several different algorithms available. In supervised, unsupervised, semi-supervised, and reinforcement learning, a machine learning algorithm is used. This is a list of popular machine learning algorithms outlined in Figure 6. These three machine learning algorithms are commonly used in traffic detection and routing.

3.6.1 Decision Tree Classifier

Neural networks use supervised learning algorithms, which use inputs that contain both data and knowledge to formulate and produce outputs. The decisions in a decision tree are linked together with branching points called nodes. A leaf represents a name, while a decision node signifies potential results from a decision and sub-as-of conditions for sub-decisions denote outcomes Classifier 54 compares the features to arrive at a leaf (label).

There are some benefits using decision trees have, such as decisions trees are easy to comprehend Training of a tree in the number of data points is inexpensive (logarithmic in terms of cost) capable of handling multiple outputs It needs less work and planning. This usually means most of the strategies involve making "pseudo" variables, making numerical variables equivalent, etc. Provides an accurate result even though the model's assumptions deviate from reality Decision-tree learners can over-decide and generalize.

This is known as “overfitting” or “parameterizing.” Substantially different decision trees are formed by minor changes in the data although there have been some attempts to optimize decision trees for concepts and problems with trivial solutions, this task remains an intractable (or difficult) problem for all and the totality of these solutions can be referred to as intractable. Hence, it follows that realistic decision tree learning algorithms employ local optimization at each node. Determining the global optimum decision tree could not be accomplished with these algorithms.

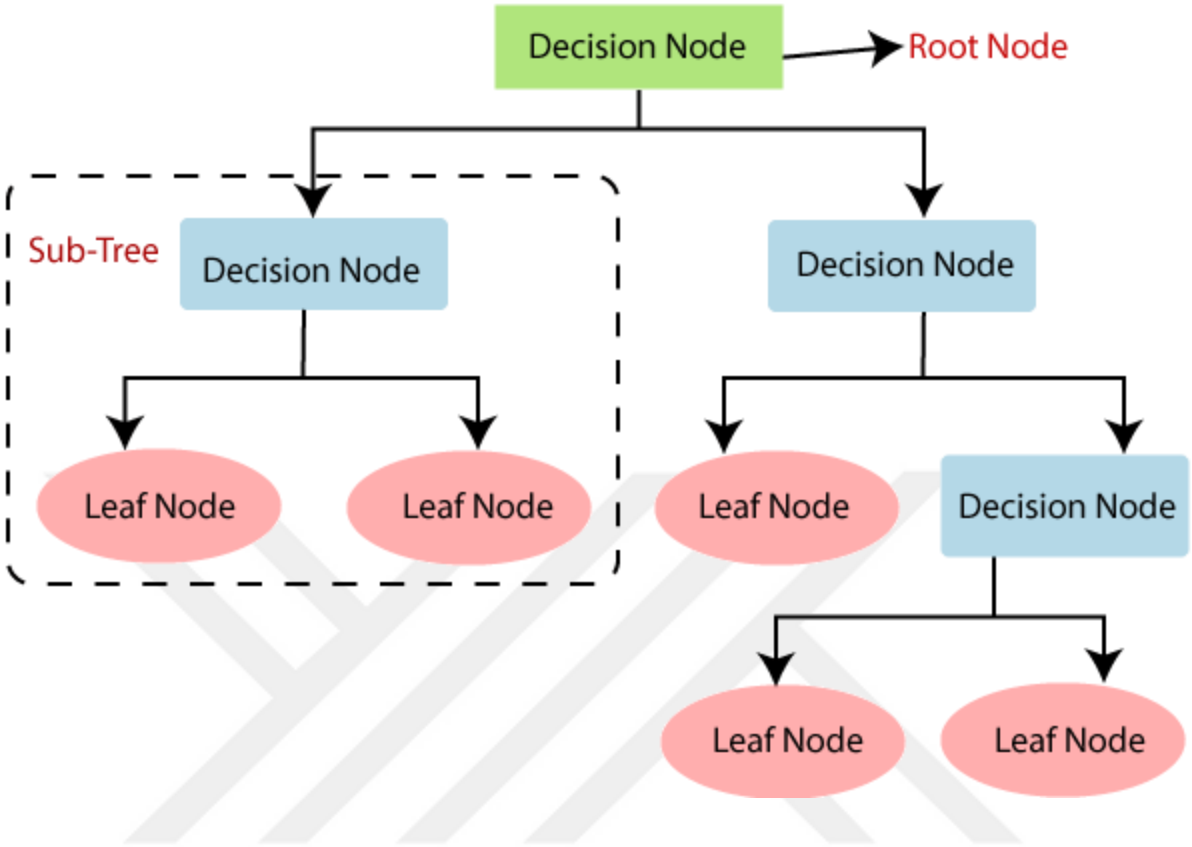

Figure 3.8: DT Classifier

3.6.2 Random Forest

A random forest is a method of combining different estimators on subsets of the dataset and then averaging the results to monitor the number of variables. Ensemble is a classification algorithm. Many experts use it as one of the most effective classifiers for artificial intelligence applications. The complexity of random forests makes them better tools for the overfitting problem.

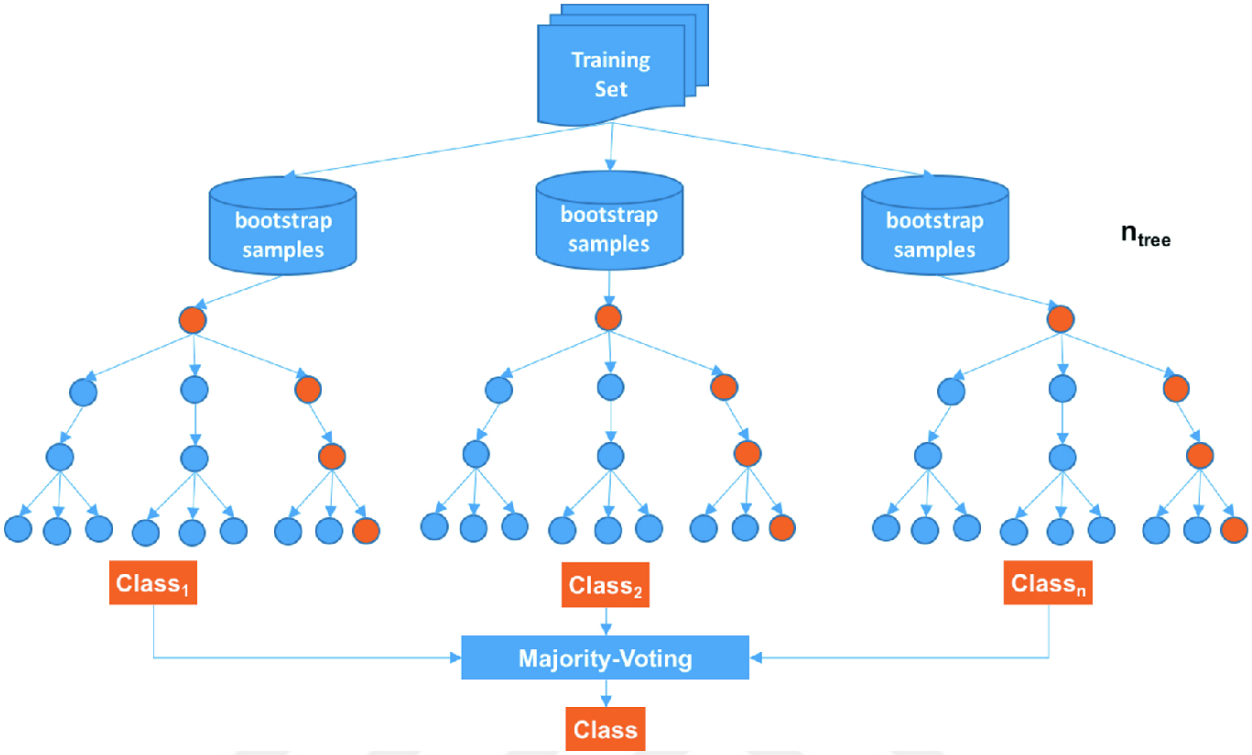

Figure 3.9: RF Classifier

3.6.3 ANN

Artificial Neural networks, also known as connectionist computing, use the structure of the nervous systems, but are not constructed from the same material as animal brains. General systems "read" to perform a task by applying heuristics their models mimic the idea of neurons enable us to draw connections between different data and their resulting output.

Neural networks can be trained using input. To understand, the system computes its inputs and desired outputs, and fine-tunes the parameters to produce the result. Neural networks have three main learning styles: supervised, unsupervised, and associative. These approaches range from unsupervised to fully-supervised to blended learning

The unsupervised and supervised learning sessions have also been covered. Complementary methods are the third method for neural networks. Either supervised or unsupervised learning methods are used. If a good seed value is discovered, the network is no longer required to be taught. This is known as prep work. Supervised learning is used to adjust the established starting points to perfection. If a model has been fine-tuned, it must be evaluated to see if it performs as expected. Accuracy is the most frequently used metric in traffic classification and routing. It can be characterized as word-based precision or count- based accuracy. The more detailed the model, the more accurate it is. The error rate may be calculated by subtracting the success rate from one. Another criterion includes the details for accuracy, recall, and F1 ranking. In general, the machine learning project's output metric selection is based on the objective of the project. Any of the algorithms discussed in this paper have long been implemented in networks. Our investigation will focus on how machine learning can be used to categorize traffic, as well as route management. The analysis is in the following manner: We will begin by examining sites that applied machine learning to solely to classify traffic. Second, we define traffic and calculate the best route using ML.

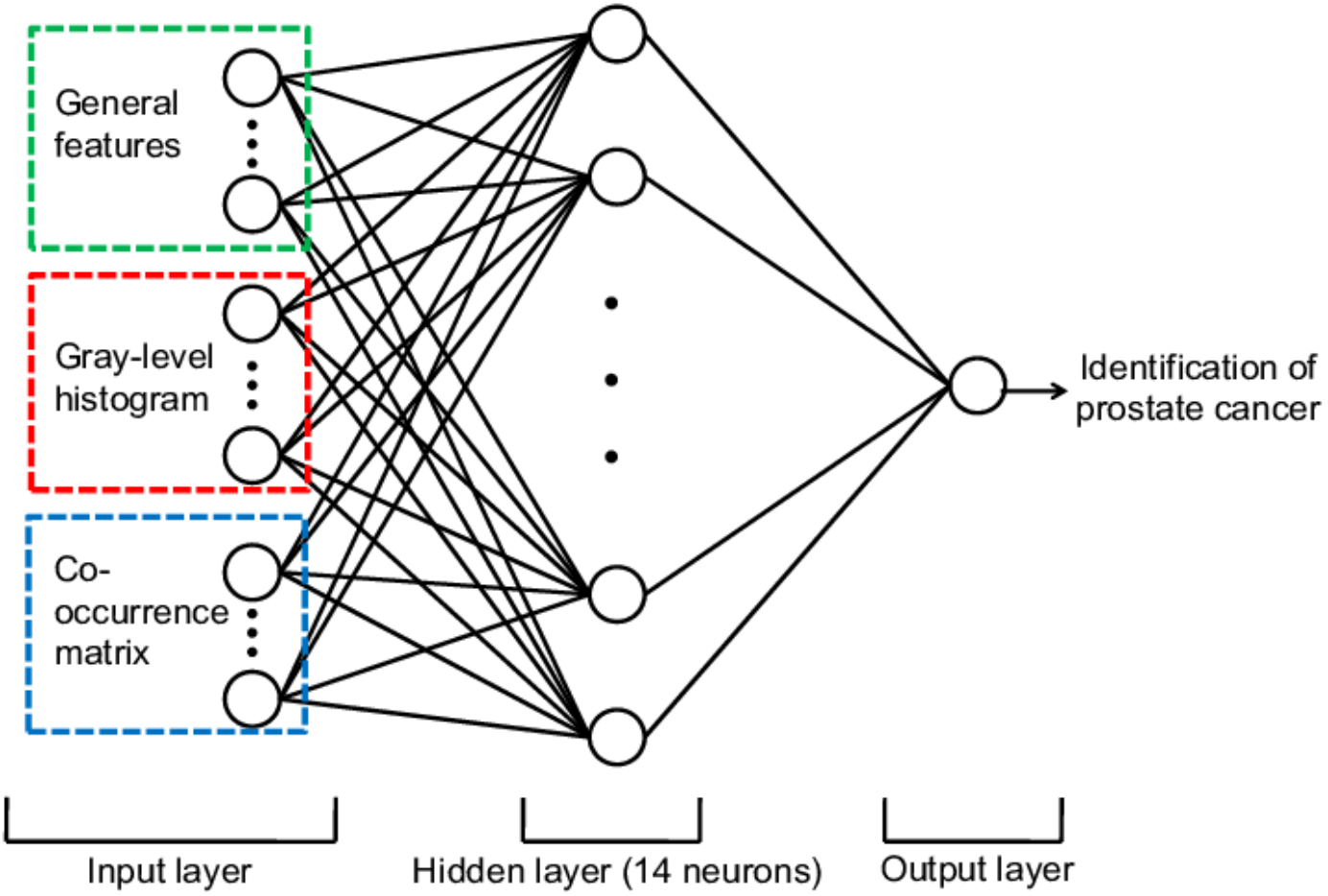

Figure 3.10: ANN Classifier

3.7 TRAFFIC CLASSIFICATION

A critical network problem for resolving issues like QoS management, it is possible to use network traffic classification as a base once you know your network traffic profile, you can distribute your network resources easily and quickly. Over the years, the general classifications of the internet protocol IP include Port Based Network Security (PBS) and Deep Packet Inspection (DPI). Identify using TCP and UDP port numbers However; this approach is not commonly used, since many apps have adopted nonstandard or ambiguous port numbers.

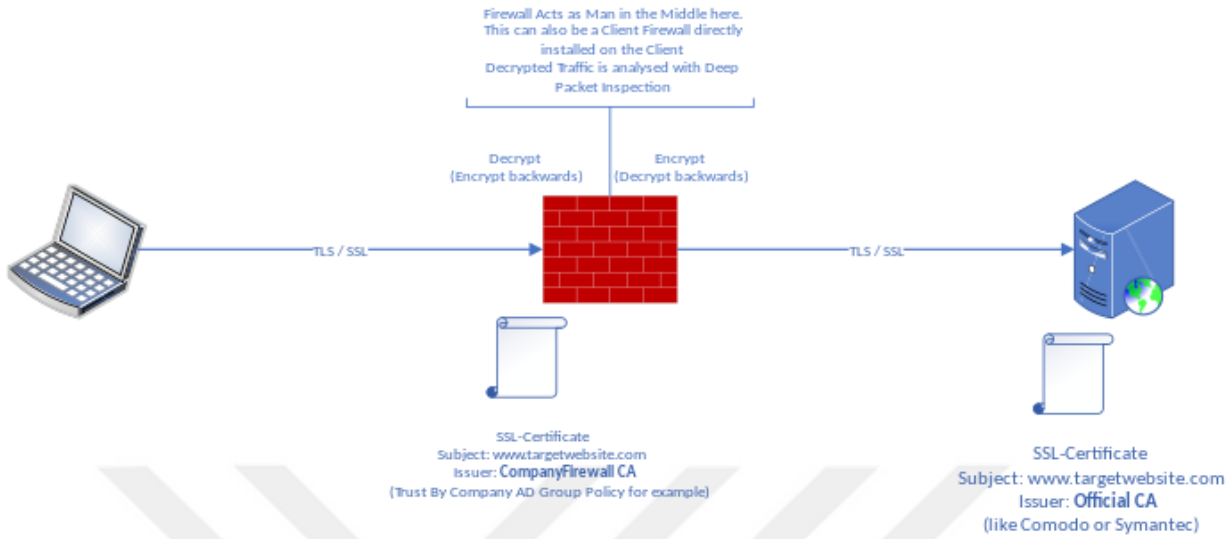

Figure 3.11: DPI Packet

In computer terminology, DPI involves a study of the packet's content to a set of patterns. As accurate as DPI needs a lot of machine resources, it' As well, because most data privacy is a prime concern, packets are now encrypted, and thus can no longer be examined for information, rendering DPI ineffective A machine learning technique that has recently been developed by researchers recently proposed employs classification is known as Data Clustering. It uses ML methods to research traffic groups for creativity you do not need to know the data inside the packets to analyze them. Leverages an “Open-Flow” protocol in SDN to obtain insights on traffic, and it automates research with low computing costs. A lot of time has been spent on trying to put traffic into categories. We have mentioned a few samples grouped the various MLC classes together based on their uses.

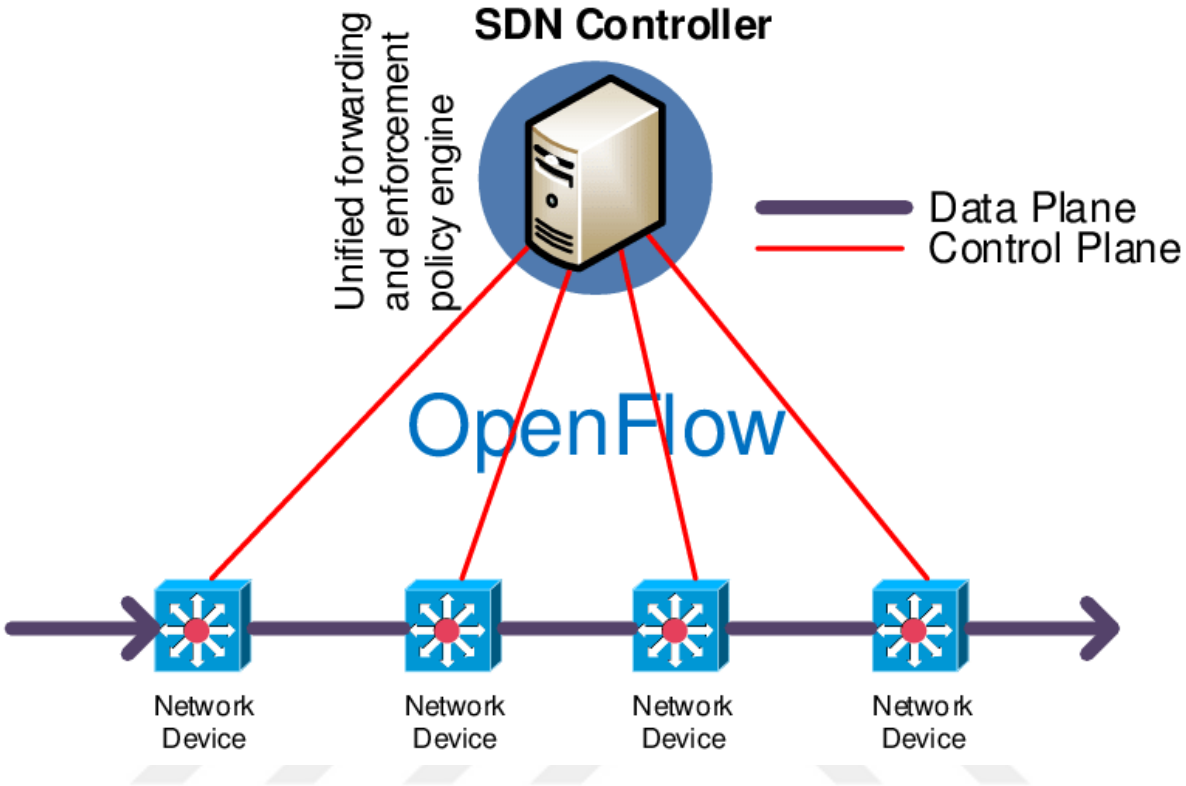

Figure 3.12: SDN Controller

4. PROPOSED METHOD

Networking is one of the most important things an algorithm does. Decentralized routing is the standard in legacy networks. As SDN works, the controller distributes routing policies on the switches by configuring flow tables in the hardware. Better quality of service (QoS) means more effectively utilizes network capital Routing decisions using the controller's global view is considered a best practice. Some heuristic methods, like ant colony optimization, are extraordinarily strong. A heuristic solution approximates the optimal solution, but incurs a great computational cost, making it impractical to search for it on a continuous basis. This disadvantage of algorithms makes impractical to run such complex flow-based algorithms in the SDN controller. In contrast to exact algorithms, machine learning, it is unnecessary for a feature network to be complete; it operates on developing relationships between features. Additionally, models can estimate near-optimally for real-time use. Several different reinforcement learning experiments have been done on routing optimization using ML. several related studies have been organized here based on the ML technique employed.

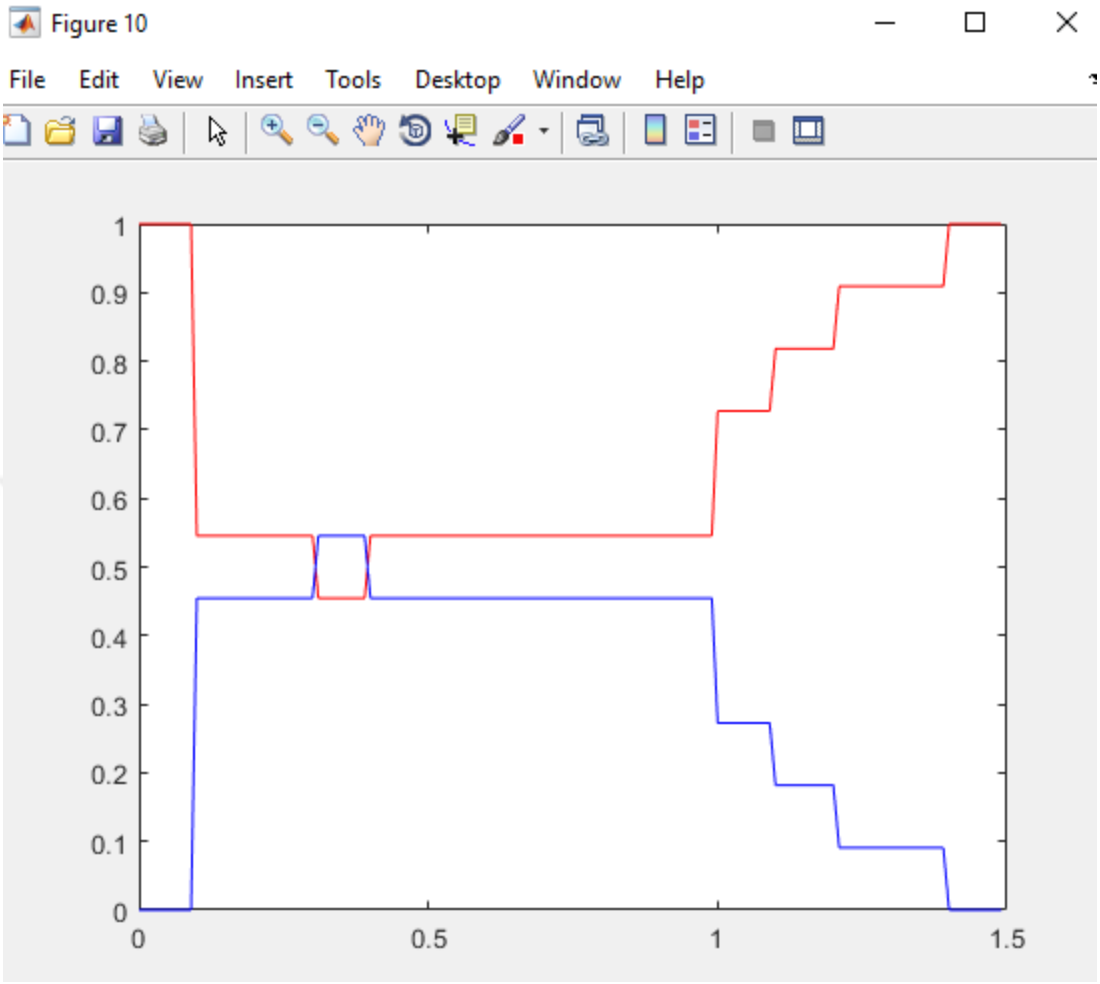

Figure 4.1: ML Initial Route Detection

4.1 OPTIMAL ROUTE DETECTION

Data science is based on supervised learning typically network and traffic measurements are used as inputs for routing algorithms. The networks are in order to analyze the click-through data for several months in order to arrive at its recommendation (LSTM-RNN). A model takes in the incoming traffic matrix at time "t" and makes a prediction for time "t+1" in the future. LSTM was built on a pre-curve training dataset with backpropagation for model training. Network traffic and network states are both input and output features to a Deep Feedforward Network that are used to train a unit to handle increased feed demand. The GEANT topology was put to use on the GEANT network One's efforts are directly

proportional to the traffic volumes, with respect to how well QoS is supported by the network and how those traffic groups are treated. One could state-of-of-the-the-the-the-art strategy that combines two methods, called supervised machine learning.

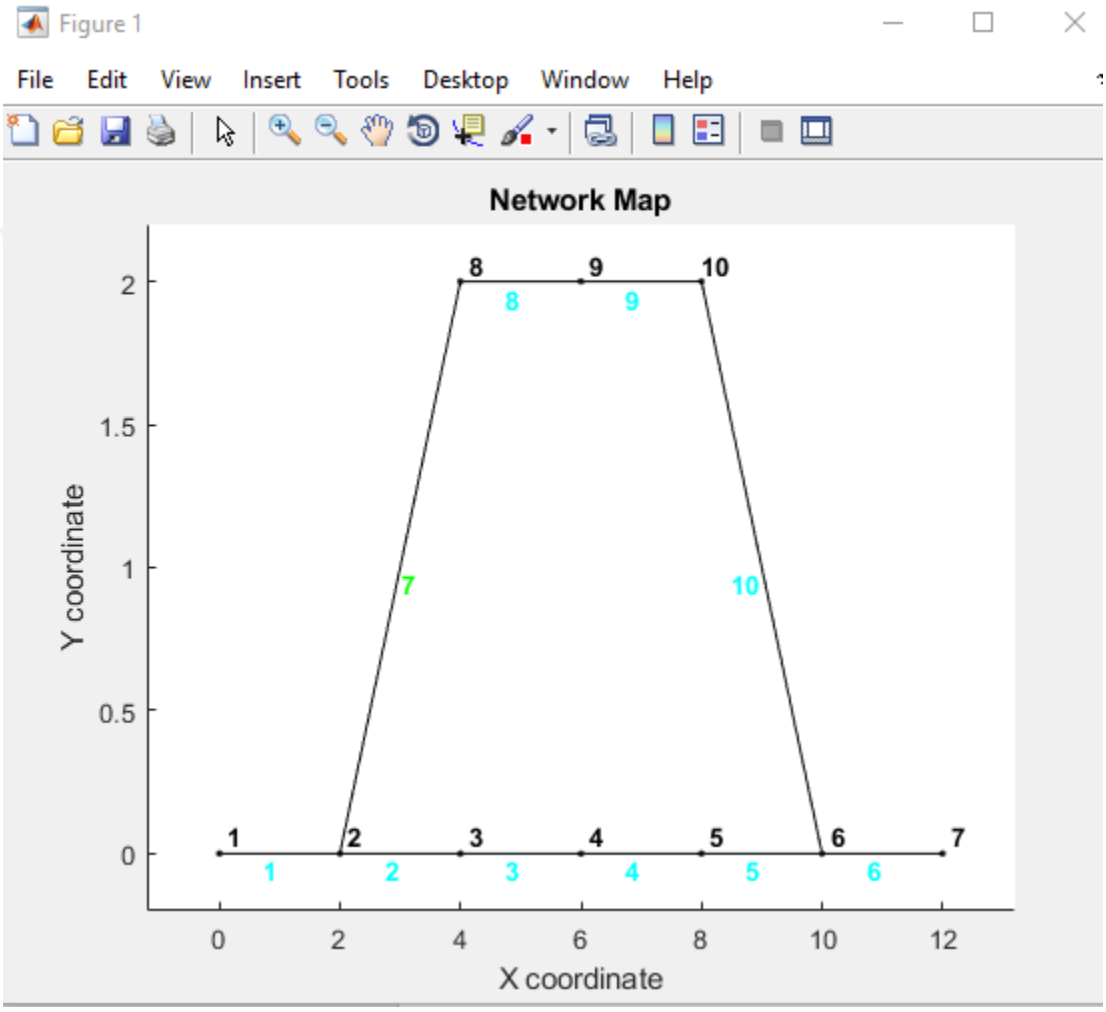

Figure 4.2: SDN – IOT Network Map

They introduced Meta-level routing using machine learning models in real-time. To create a neural network that is as accurate as possible, we use a combination of its architecture and the information fed into it to produce an architecture that emulates the performance of heuristics. Compared to other routing solutions, their proposed solution was found to be quicker. A professional in the real world decides (and is

paid) on the results of their acts. The agent's objective is to make decisions that lead to a greater profit Issue solving mostly involves finding ways to make things better. The SDN controller is the individual, and the network is the context. There are 25 network and traffic matrices from the environment when the decision is made by the agent, the network's arrival time or any of the specified network parameters which be in question.

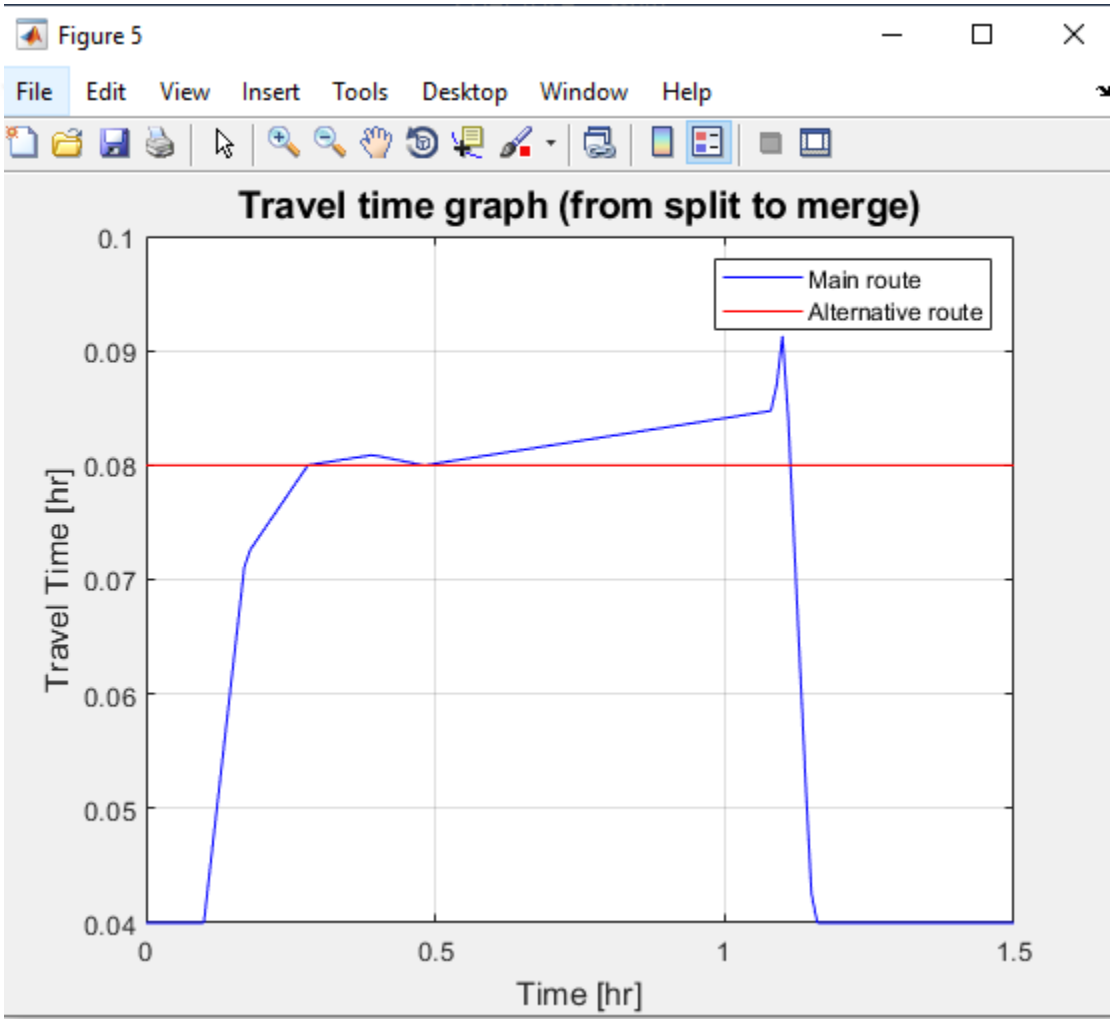

Figure 4.3: Optimal Rout Detection

Here are some examples of reinforcement learning approaches to routing: distributed routing was driven by the application of reinforcement learning techniques to SDN While, the overall architecture of the SDN

uses a distributed approach, the routing protocol was implemented in a highly localized manner. The OSP routing algorithm was incorporated into the RL protocol and thoroughly tested. They looked at the standard OSFP protocol to see how well it performs jitter and jitter delay. Employed a Routing Algorithm Based on Reinforcement Learning Multilevel SDN implementation had optimization built in from the ground up to get a better signal-to-through-noise ratio. In the figure below, three types of SDN controllers are shown; domain controllers at the top, masters in the middle, and slaves at the bottom. Each of these controllers was assigned their specific task to help the super controller minimize the burden. The slaves gave port only read access to the data plane; they didn't deal with the control plane's data. The controllers also took on the duties of traffic control, traffic management, rationing, as well as gate operation, regulating flow and flow control. Each domain controller was issued a number of flow requests and was subsequently configured with a number of policy rules. All network functions were controllable from the supervisory point of view. To achieve the best possible routing solutions, a QoS-based incentive algorithm was incorporated into the super controller. It was tried in a full Sprint GIP setting, and it worked. Compared to existing learning solutions, the solution was quick.

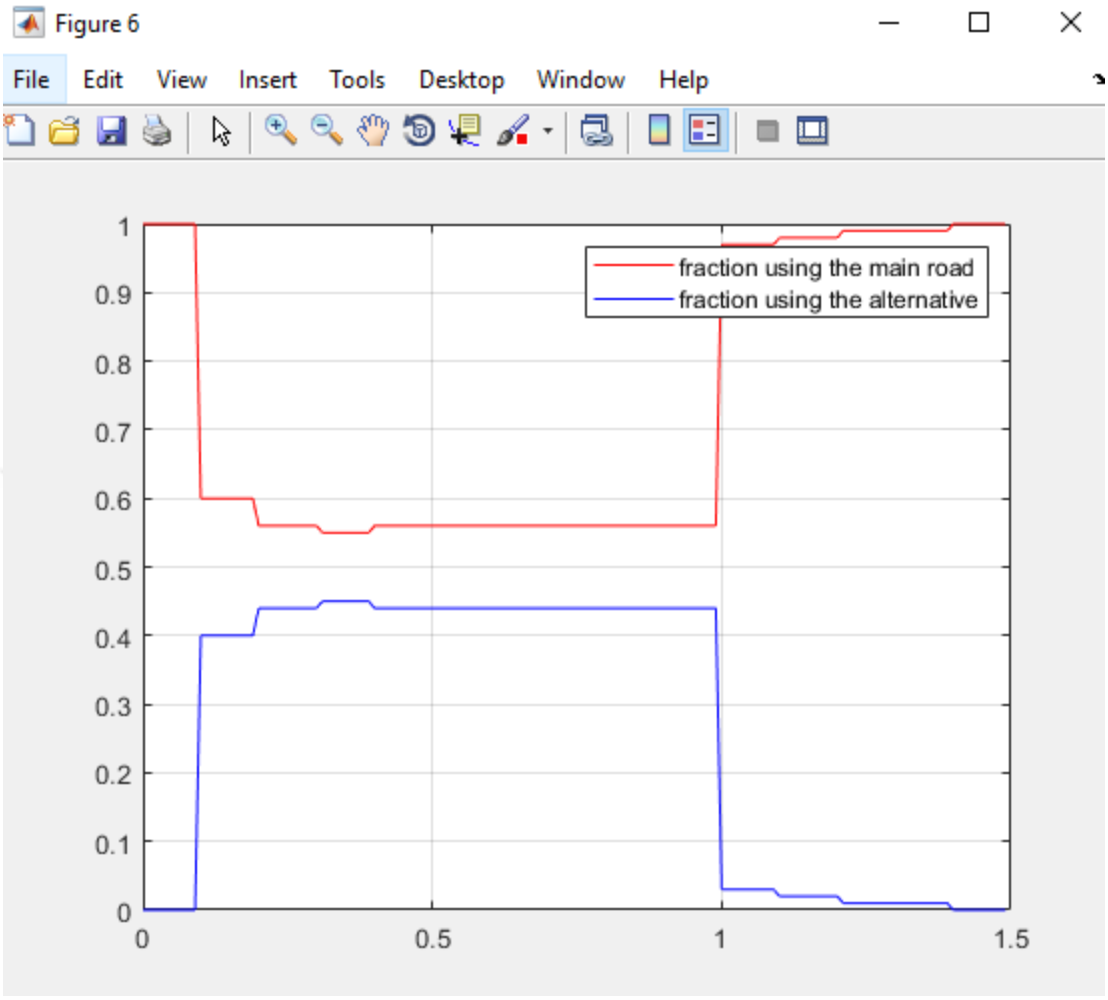

Figure 4.4: Main and Alternative Road

4.2 TRAFFIC CLASSIFICATION BASED ON OPTIMAL ROUT

Classifying traffic allowed us to pick the optimal route in this segment. Using ML and DNS improves on data center and server management scheduling and control. It took two steps to develop. The first move was taking machine learning and incorporating it into the network to separate different types of traffic on the spot. The centralized SDN controller groups all of the networks into broad categories and decides on the best route among them to each based on the global view of the network Specifics about the classification and routing were not included in this article. In addition, [38] presents another form of solution that incorporates traffic classifications and route optimization, In a SDN system, they suggested a

traffic classification with a DNN a Virtual Network Function is added to help ease the pressure on the SDN When the flow is in the flow rule tree arrives, the SDN routes the flow based on its laws. The flowchart instructs the packet to go to go to the controller. For the SDN design, the controller works out the direction and deploys these flows in the SDN switches. In addition, a flow rule is used to pass a copy of the packet to the port on which the VNF is listening.

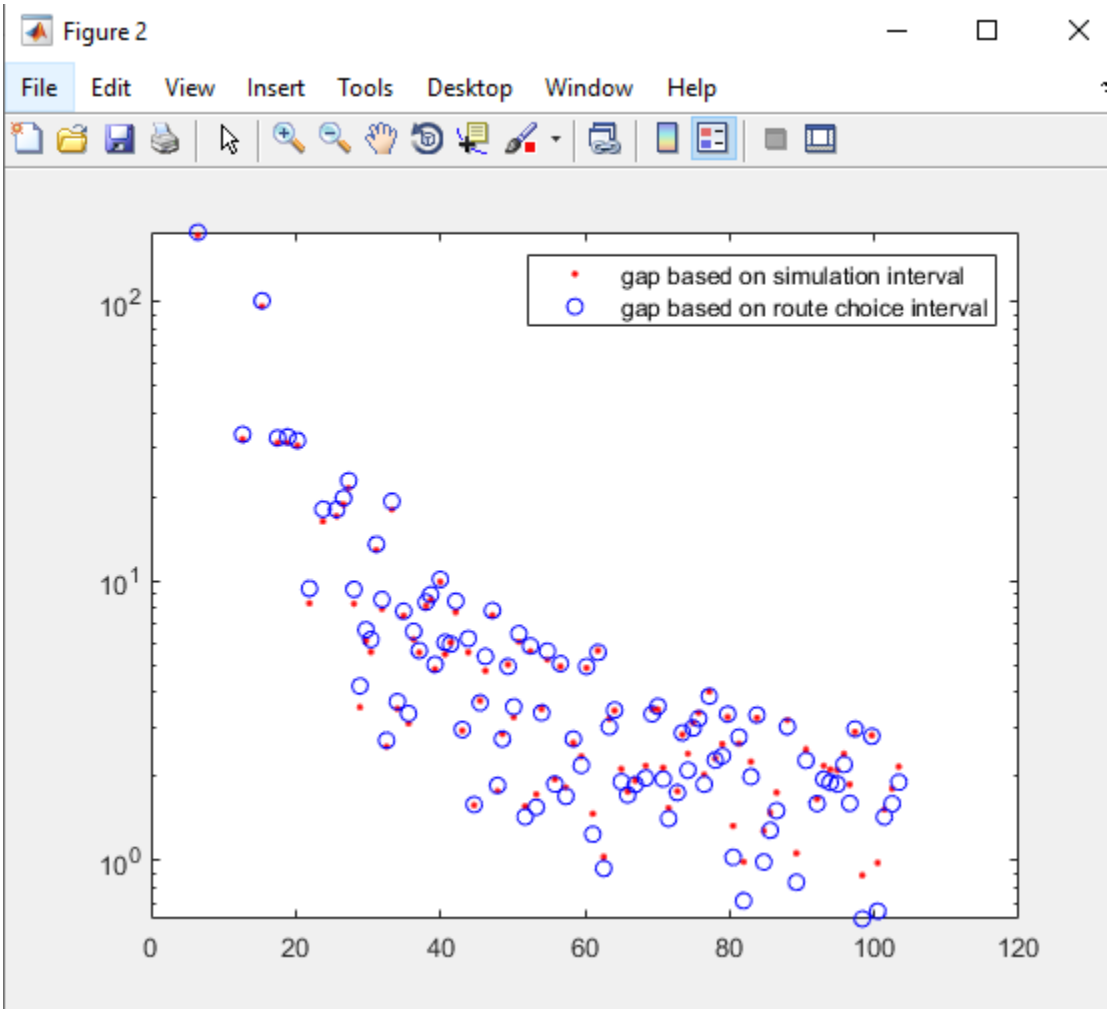

Figure 4.5: VNF Traffic of Cars Based On ML Rout Detection

The VNF uses a qualified on DNN to capture traffic and classifies traffic according to learned features A tagged class data is passed on to the SDN controller. The QoS controller now searches for a route that

gives required QoS and gives highest priority to the flow classes. It provides a way to increase the network efficiency, but does not specifically optimize for path determination by SDN controllers. By default, the controller determines the identified packet's route based on the global view of things we build a matrix that looks at the different QoS groups and decides on the direction based on the flow needs. The SDN-I has the center brain and coordinator (decider) position in the system. Many of the functions in the data plane are under its power. The basic two decision process is to be followed in our architecture is to: Traffic Classifications and Optimized Routing By extension, we keep the traffic description to avoid burdening the controller. They used to be carried out by specialized, proprietary, and dedicated devices that have been converted to a virtual type. The aim of virtual networking devices is to leverage commodity components to get the functions of the network out of the system into software that runs on the devices that hardware the data plane VNF will be hosted on a server as described, our model can offer:- Traffic classification using QoS and Optimal path determination. Herein this chapter, we define a QoS traffic class of traffic. The end goal is to apply machine learning algorithms on the selected dataset for Quality of Service (QoS) classification. Several learning algorithms are applied to traffic info. Features influence the classification is addressed as well as the numbers of features. The last but not least, we deal with the time needed for training and feature selection when employing machine learning algorithms.

4.3 DATA COLLECTION

A low-priority flow rule is installed on the edge SDN switches in our architecture. The flow from the source to a specific port is being watched by the server is on the low priority flow path. When a flow reaches the SDN switch, the switch looks at the packet and determines whether or not the packet satisfies a law. If higher-priority traffic is present, it will process the packets. If there are no other higher priority flows, it will send a copy of the packet to the SDN controller and forward the original to the port during a timeout, the VNF can take each packet, and measure time-stamp features from it. All the features have simple mathematical equations, or at most just basic ones, and hence are easily calculated and smooth.

4.3.1 Training the Algorithm

With highly dynamic SDN-I, the algorithm must be retrained to continue being successful. Training can be conducted either face-to-to-face or virtually, through the use of the Internet. The VNF server can use a database and a DPI tool for online training. As packets arrive, the server will save the new samples in a training table and compare it to the table of originals to find out which packet was accurate. It can get its data set from users during high traffic and apply training to the ML algorithm in low traffic times. Later on, the classifier will be built offline, and uploaded to the server using stored traffic from offline training. This study doesn't thoroughly explore either the two methods, which leaves the door open for the possibilities of other methods that might be added in the future.

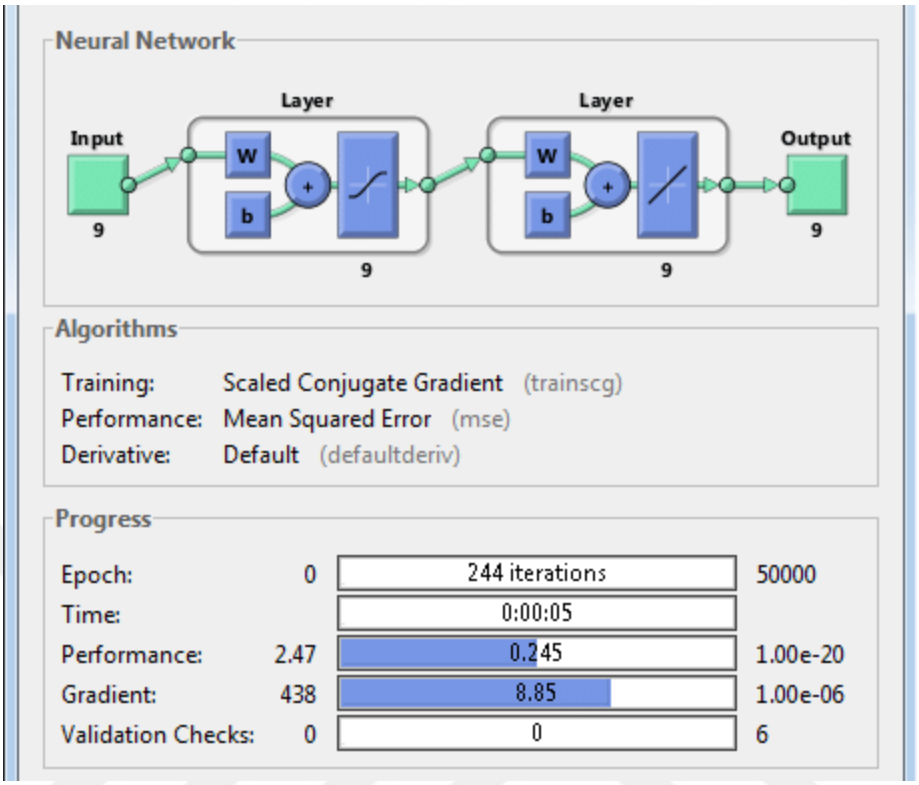

Figure 4.6: Network Training

4.3.2 Dataset

The dataset [40] was generated in the real world of (ToR). We assume most IoT data will be encrypted for protection in the future. Because of the sensitive nature of the information, we encrypted the dataset. We also chose toward dataset because it is widely used, so we can apply it as a baseline dataset. Traffic consists of 8 different styles (browsing, chat, audio streaming, video-streaming, mail, VOIP, P2P and file transfer). The dataset has a diverse mix of various qualities of service (QoS) classifications and represents a comprehensive array of data the dataset includes Facebook, Skype, Gmail, etc. The dataset was created using the architecture to the left.

The source and destination IP addresses, source and destination ports, and protocol are the same (TCP). Thus, because the traffic data collection was performed in a managed environment, the applications were

sequenced together. In the other hand, the datasets are full of errors, e.g. missing labelling and hence our classification would be compromised by such imperfections. ISC Flows and calculated all required parameters were calculated using ISC. Measurements were made for time going forwards and time going backwards. There were a variety of scenarios to analyze the data on various flow times. According to the 2004 study, flows with shorter timeouts deliver the best results.

Table 4.1: Content of the Training Dataset

Time	Browsing	Mail	Chat	Audio	Video	FT	VoIP	P2P	Total
5s	2645	497	485	1026	1529	1663	1529	2139	14508
10s	1604	282	323	721	874	864	2291	1085	8044
15s	264	213	273	50	681	556	1746	81	3864
30s	694	111	153	332	364	311	790	375	3130
60s	411	60	90	190	196	165	413	198	1723
120s	239	34	151	119	105	86	225	110	969

The datasets were all time-based, having 28 attribute times, each. More time-sensitive data can be derived from flows, while flow-based data has a higher speed of calculation. Almost real-time features can be calculated with the number of packets that constitute them, making them far more effective. These included features: is equal to the time required to send two packets. Flow AT which is the amount of time between two consecutive packets (mean, min, max, std). "Idle time which had been that which is when it began to flow (mean, min, max, std). The best time to administer an injection is right before or after flow operation (mean, max, min, std). Calibrate packets/s, packets, packets, and durations as is clear from our literature review, classifications based on function have increased over the internet, it has become more

common practice not to use classifications. Because multiple applications can have the same QoS specifications, classifying based on QoS groups is more realistic. Similarly, the statistics provided by QoS classification can also describes new applications. The dataset uses application-specific settings With regard to our work, we relabeled the data into a 2-output model, we defined a QoS to be a two-based model; we took bandwidth and latency into account Consider these conditions when classifying creative people.

Table 4.2: Dataset Classes

Application	Bandwidth Requirement	Latency Requirement
VOIP/Chat	LOW	LOW
Mail/Video	MID	MID
Audio/Video	HIGH	MID
P2P/File Transfer	HIGH	BEST EFFORT

We have a dataset of 28 variables. Because all of these feature files have the same source and destination IPs and ports, protocols, we're going to exclude these time-based features, which leave us with 23 in common. Based on the flow timeouts in Table 1, our datasets have a wide range of input/output sizes. Most of the remaining 23 features don't include traffic quality of service information. Selecting the 'most significant' features for classification is critical to correctly. To pick features, this procedure is known as feature design feature extraction. In other contexts, we feel feature sizes aren't big enough to merit feature combination selection, but our objective for this meeting is to discover different feature combinations of features that help with classifiers reducing the number of features would also result in a decreased computation time, making it imperative for real-time use. For the feature selection in the previously

mentioned earlier chapters, we employ a wrapper system based on feature importance and forward selection with SHAP [45].

4.4 RESULTS AND ANALYSIS

These findings are summarized in this section. Without prior feature selection, the ML algorithms learn everything from the training set. In the second chapter, the different feature selection techniques are compared. It's still early in the process, so we go through the selected and optimize the best and most promising features. Finally, we apply our ML algorithms to the data and inspect the output. There are 3 trials lasting for 5, 10, and 15 seconds each. Each data function has 23 items. Measurement and display of precision, F1 score, and the training period for an algorithm's F1 score See Table 11 for an example of how the various algorithms fare for the dataset with a timeout of 5 seconds. It's the highest overall Random Forest classifier score of 0.14 the decision tree is the most accurate of all the decisions are 0.48 The K-nearest neighbor has the lowest overall accuracy, but has the highest F1 score...

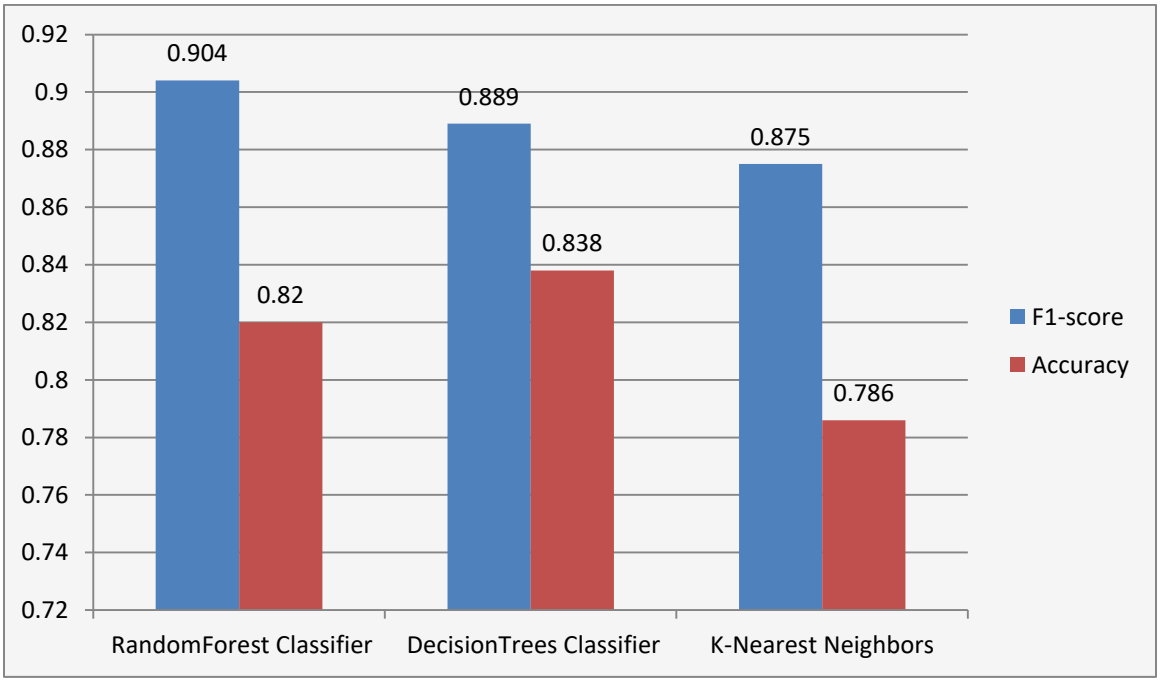

Figure 4.7: F1 and Accuracy Scores for 10 Sec Flow Timeout

5. CONCLUSION

This paper offers a conceptual approach for managing pedestrian traffic using IoT technologies. All the work has been carried out in depth on the various techniques used to identify and classify traffic and the appropriate framework to implement the method. The main goal was to assess the feasibility of the conceptual approach such that the individual integrated methods proposed provide the possibility of its practical implementation in future. The structure proposed consists of a camera which captures sequences and acts as a network sensor and transmits these sequences to a central server responsible for analysis. The strategic approach is focused on deep learning using a previously trained CNN called Alexnet, which is retrained for pedestrian classification. The neural network is classified into one of the four specified kinds of footballers (walker, two Walkers, runner and cyclist). To achieve these cut-offs, the location of the pedestrian in each picture is identified from the obtained sequence according to the detected movement by means of a combination of vision and motion detection techniques. The results of the analysis are uploaded to the Thingspeak public cloud platform, which any individual or institution may access. In addition to the data obtained by using CNN, time series were also generated during Monday 2018 with the presence of cyclists. The data was used to analyze and forecast two models: an auto regressive model with LSTM networks. The data from the time series are also submitted to the Thingspeak channel for the storage of sequence analysis data. With regard to potential activities, the following should be noted: The device uses ten videos taken in Park in the current edition. One of the potential steps will be to integrate the device effectively so that it receives video sequences from various scenarios. On the other hand, time series data capture can also be implemented to use actual data instead of randomly generated data. As discussed in the report, there is a problem when pedestrians are counted multiple times. A workaround would have to be built in the future to remind the pedestrians that it had previously intended to prevent their re-accounting. This can be approached from the classification perspective with CNN, such that, if a

step is identified, it is determined to be the same element by a similarity in the likelihood of classification. There is a problem that in the four categories outside of the area permitted for this reason, pedestrians may cross even cross sectionally or perpendicularly and so they are movement acts that are not consistent with the intended objective. In this case, consideration should be given to the direction of the optical flow vector so that on the average path the horizontal direction is shown while the direction differs from outside the path. It would also be interesting to classify additional groups, such as animals. The data on the presence of animals will enable the administration to control the spaces with a greater presence of animals. The technical approach proposed centered on the accounting of pedestrian traffic, although the same technique can also be considered in other fields, such as vehicle flow accounting, also within the smart cities framework of the future.

REFERENCES

1. Poslad, S. Ubiquitous Computing: Smart Devices, Environments and Interactions, 1st ed.; Wiley Publishing: Haboken, NJ, USA, 2009.

2. Sagl, G.; Resch, B.; Blaschke, T. Contextual Sensing: Integrating Contextual Information with Human and Technical Geo-Sensor Information for Smart Cities. Sensors 2015, 15, 17013–17035. [CrossRef]

3. Thurm, S.; Kane, Y.I. Your Apps Are Watching You. Wall Str. J. 2010, 17, 1.

4. Liu, J.; Chen, R.; Pei, L.; Guinness, R.; Kuusniemi, H. A Hybrid Smartphone Indoor Positioning Solution for Mobile LBS. Sensors 2012, 12, 17208–17233. [CrossRef] [PubMed]

5. Dhar, S.; Varshney, U. Challenges and Business Models for Mobile Location-based Services and Advertising. Commun. ACM 2011, 54, 121–128. [CrossRef]

6. Groves, P.D. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd ed.; Artech House: Norwood, MA, USA, 2013.

7. Kim, S.C.; Lee, J.K. Wireless Localization Method based on an Efficient Multilateration Algorithm over a Wireless Sensor Network and a Recording Medium in Which a Program for the Method is Recorded. U.S. Patent 8478292, 2 July 2013.

8. He, S.; Chan, S.H.G. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. IEEE Commun. Surv. Tutor. 2016, 18, 466–490. [CrossRef]

9. Zhuang, Y.; Syed, Z.; Li, Y.; El-Sheimy, N. Evaluation of Two WiFi Positioning Systems Based on Autonomous Crowdsourcing of Handheld Devices for Indoor Navigation. IEEE Trans. Mob. Comput. 2016, 15, 1982–1995. [CrossRef]

10. Zhuang, Y.; Yang, J.; Li, Y.; Qi, L.; El-Sheimy, N. Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons. Sensors 2016, 16, 596. [CrossRef]

11. Shit, R.C.; Sharma, S.; Puthal, D.; Zomaya, A.Y. Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure. IEEE Commun. Surv. Tutor. 2018, 20, 2028–2061. [CrossRef]

12. Kwak, M.; Park, Y.; Kim, J.; Han, J.; Kwon, T. An Energy-Efficient and Lightweight Indoor Localization System for Internet-of-Things (IoT) Environments. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2018, 2, 17:1–17:28. [CrossRef]

13. Cho, S.B. Exploiting Machine Learning Techniques for Location Recognition and Prediction with Smartphone Logs. Neurocomputing 2016, 176, 98–106. [CrossRef] Sensors 2019, 19, 4609 22 of 24

14. Luceri, L.; Cardoso, F.; Papandrea, M.; Giordano, S.; Buwaya, J.; Kundig, S.; Angelopoulos, C.M.; Rolim, J.; Zhao, Z.; Carrera, J.L.; et al. VIVO: A Secure, Privacy-Preserving, and Real-time Crowd-Sensing Framework for the Internet of Things. Pervasive Mob. Comput. 2018, 49, 126–138. [CrossRef]

15. Steinhoff, U.; Schiele, B. Dead Reckoning from the Pocket - An Experimental Study. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom), Mannheim, Germany, 20 May 2010; pp. 162–170. [CrossRef]

16. Woodman, O.; Harle, R. Pedestrian Localisation for Indoor Environments. In Proceedings of the 10th International Conference on Ubiquitous Computing, Seoul, Korea, 21–24 September 2008; ACM: New York, NY, USA, 2008; pp. 114–123. [CrossRef]

17. Beauregard, S.; Haas, H. Pedestrian Dead Reckoning: A Basis for Personal Positioning. In Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC'06), Hannover, Germany, 16–19 March 2006; pp. 27–35.

18. Li, F.; Zhao, C.; Ding, G.; Gong, J.; Liu, C.; Zhao, F. A Reliable and Accurate Indoor Localization Method using Phone Inertial Sensors. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; ACM: New York, NY, USA, 2012; pp. 421–430. [CrossRef]

19. Borenstein, J.; Ojeda, L. Heuristic Drift Elimination for Personnel Tracking Systems. J. Navig. 2010, 63, 591–606. [CrossRef]

20. Bachmann, E.; Yun, X.; Brumfield, A. Limitations of Attitude Estimation Algorithms for Inertial/Magnetic Sensor Modules. IEEE Robot. Autom. Mag. 2007, 14, 76–87. [CrossRef]

21. Renaudin, V.; Combettes, C. Magnetic, Acceleration Fields and Gyroscope Quaternion (MAGYQ)-Based Attitude Estimation with Smartphone Sensors for Indoor Pedestrian Navigation. Sensors 2014, 14, 22864. [CrossRef]

22. Yadav, N.; Bleakley, C. Accurate Orientation Estimation Using AHRS under Conditions of Magnetic Distortion. Sensors 2014, 14, 20008–20024. [CrossRef] [PubMed]

23. Wi-Fi Alliance, P2P Technical Group. Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.7; P2P Technical Group: Kitchener, ON, Canada, 2009. 24. Jung, S.; Hann, S.; Park, C. TDOA-based Optical Wireless Indoor Localization using LED Ceiling Lamps. IEEE Trans. Consum. Electron. 2011, 57, 1592–1597. [CrossRef]

25. Nicoli, M.; Morelli, C.; Rampa, V. A Jump Markov Particle Filter for Localization of Moving Terminals in Multipath Indoor Scenarios. IEEE Trans. Signal Process. 2008, 56, 3801–3809. [CrossRef]

26. Hazas, M.; Hopper, A. Broadband Ultrasonic Location Systems for Improved Indoor Positioning. IEEE Trans. Mob. Comput. 2006, 5, 536–547. [CrossRef]

27. Altini, M.; Brunelli, D.; Farella, E.; Benini, L. Bluetooth Indoor Localization with Multiple Neural Networks. In Proceedings of the IEEE 5th International Symposium on Wireless Pervasive Computing, Modena, Italy 14 June 2010; pp. 295–300. [CrossRef]

28. Ni, L.; Liu, Y.; Lau, Y.C.; Patil, A. LANDMARC: Indoor Location Sensing using Active RFID. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom), Fort Worth, TX, USA 26 March, 2003; pp. 407–415. [CrossRef]

29. Almaaitah, A.; Ali, K.; Hassanein, H.S.; Ibnkahla, M. 3D Passive Tag Localization Schemes for Indoor RFID Applications. In Proceedings of the IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1–5. [CrossRef]

30. Youssef, M.; Agrawala, A. The Horus WLAN Location Determination System. In Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Washington, DC, USA, 6–8 June 2005; ACM: New York, NY, USA, 2005; pp. 205–218. [CrossRef]

31. Otsason, V.; Varshavsky, A.; LaMarca, A.; de Lara, E. Accurate GSM Indoor Localization. In UbiComp 2005: Ubiquitous Computing; Lecture Notes in Computer Science; Beigl, M., Intille, S.,

Rekimoto, J., Tokuda, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3660; pp. 141–158._9. [CrossRef]

32. Chung, J.; Donahoe, M.; Schmandt, C.; Kim, I.J.; Razavai, P.; Wiseman, M. Indoor Location Sensing using GEO-Magnetism. In Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, Bethesda, MD, USA, 28 June–1 July 2011; ACM: New York, NY, USA, 2011; pp. 141–154. [CrossRef] Sensors 2019, 19, 4609 23 of 24

33. Tarzia, S.P.; Dinda, P.A.; Dick, R.P.; Memik, G. Indoor Localization Without Infrastructure Using the Acoustic Background Spectrum. In Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, Bethesda, MD, USA, 28 June–1 July 2011; ACM: New York, NY, USA, 2011; pp. 155–168. [CrossRef]

34. Zou, H.; Xie, L.; Jia, Q.S.; Wang, H. Platform and Algorithm Development for a RFID-based Indoor Positioning System. Unmanned Syst. 2014, 2, 279–291. [CrossRef]

35. Moghtadaiee, V.; Dempster, A.G. Design Protocol and Performance Analysis of Indoor Fingerprinting Positioning Systems. Phys. Commun. 2014, 13, 17–30. [CrossRef]

36. Ashraf, I.; Hur, S.; Park, Y. mPILOT-Magnetic Field Strength Based Pedestrian Indoor Localization. Sensors 2018, 18, 2283. [CrossRef] [PubMed]

37. Chen, C.; Lu, X.; Markham, A.; Trigoni, N. IONet: Learning to Cure the Curse of Drift in Inertial Odometry. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

38. Jiménez, A.R.; Seco, F.; Zampella, F.; Prieto, J.C.; Guevara, J. Improved Heuristic Drift Elimination (iHDE) for pedestrian navigation in complex buildings. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Guimaraes, Portugal, 21–23 September 2011; pp. 1–8. [CrossRef]

39. Jimenez, A.R.; Seco, F.; Zampella, F.; Prieto, J.C.; Guevara, J. Improved Heuristic Drift Elimination with Magnetically-aided Dominant Directions MiHDE for Pedestrian Navigation in Complex Buildings. J. Locat. Based Serv. 2012, 6, 186–210. [CrossRef]

40. Qiu, S.; Wang, Z.; Zhao, H.; Qin, K.; Li, Z.; Hu, H. Inertial/Magnetic Sensors Based Pedestrian Dead Reckoning by Means of Multi-Sensor Fusion. Inf. Fusion 2018, 39, 108–119. [CrossRef]

41. Ettlinger, A.; Retscher, G. Positioning using Ambient Magnetic Fields in Combination with Wi-Fi and RFID. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain, 4–7 October 2016; pp. 1–8. [CrossRef]

42. Chen, Z.; Zhu, Q.; Soh, Y.C. Smartphone Inertial Sensor-Based Indoor Localization and Tracking With iBeacon Corrections. IEEE Trans. Ind. Inform. 2016, 12, 1540–1549. [CrossRef]

43. Higuchi, T.; Yamaguchi, H.; Higashino, T. Clearing a Crowd: Context-Supported Neighbor Positioning for People-Centric Navigation. In Pervasive Computing; Lecture Notes in Computer Science; Kay, J., Lukowicz, P., Tokuda, H., Olivier, P., Krüger, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7319; pp. 325–342._20. [CrossRef]

44. Kloch, K.; Lukowicz, P.; Fischer, C. Collaborative PDR Localisation with Mobile Phones. In Proceedings of the 15th Annual International Symposium on Wearable Computers (ISWC), San Francisco, CA, USA, 12–15 June 2011; pp. 37–40. [CrossRef]

45. Wymeersch, H.; Lien, J.; Win, M. Cooperative Localization in Wireless Networks. Proc. IEEE 2009, 97, 427–450. [CrossRef]

46. Soatti, G.; Nicoli, M.; Savazzi, S.; Spagnolini, U. Consensus-Based Algorithms for Distributed Network-State Estimation and Localization. IEEE Trans. Signal Inf. Process. Netw. 2017, 3, 430–444. [CrossRef]

47. Abadi, M.J.; Gu, Y.; Guan, X.; Wang, Y.; Hassan, M.; Chou, C.T. Improving Heading Accuracy in Smartphone-Based PDR Systems using Multi-Pedestrian Sensor Fusion. In Proceeding of the 4th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Montbeliard-Belfort, France, 28–31 October 2013; p. 4.

48. Chen, P.; Kuang, Y.; Chen, X. A UWB/Improved PDR Integration Algorithm Applied to Dynamic Indoor Positioning for Pedestrians. Sensors 2017, 17, 2065. [CrossRef]

49. Abadi, M.J.; Luceri, L.; Hassan, M.; Chou, C.T.; Nicoli, M. A Collaborative Approach to Heading Estimation for Smartphone-based PDR Indoor Localisation. In Proceedings of the International

Conference in Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014. [CrossRef]

50. Stern, D.P. a Millennium of Geomagnetism. Rev. Geophys. 2002, 40. [CrossRef]

51. Chapter Five Reversals of the Earth's Magnetic Field. In The Earth's Magnetic Field: Its History, Origin and Planetary Perspective; International Geophysics; Merrill, R.T., McElhinny, M.W., Eds.; Academic Press: Cambridge, MA, USA, 1983; Volume 32, pp. 135–168.

52. Lee, J.G.; Park, C.G.; Park, H.W. Multiposition alignment of strapdown inertial navigation system. IEEE Trans. Aerosp. Electron. Syst. 1993, 29, 1323–1328. [CrossRef] Sensors 2019, 19, 4609 24 of 24

53. Kaniewski, P.; Kazubek, J. Integrated System for Heading Determination. Acta Phys. Pol. A 2009, 116, 325. [CrossRef]

54. Finlay, C.; Maus, S.; Beggan, C.; Bondar, T.; Chambodut, A.; Chernova, T.; Chulliat, A.; Golovkov, V.; Hamilton, B.; Hamoudi, M.; et al. International Geomagnetic Reference Field: The Eleventh Generation. Geophys. J. Int. 2010, 183, 1216–1230.

55. Hall, M.A. Correlation-Based Feature Selection for Machine Learning. Ph.D Thesis, The University of Waikato, Hamilton, New Zealand, 1999.

56. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA Data Mining Software: An Update. SIGKDD Explor. Newsl. 2009, 11, 10–18. [CrossRef]

57. Jalal Abadi, Marzieh, C.S.E.F.o.E.U. GPS-Less Personal Tracking Using Smartphone Sensors. Ph.D. Thesis, University of New South Wales, Sydney, Australia, 2017.

58. Kay, S.M. Fundamentals of Statistical Signal Processing; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1993.

59. Olfati-Saber, R.; Fax, J.; Murray, R. Consensus and Cooperation in Networked Multi-Agent Systems. Proc. IEEE 2007, 95, 215–233. [CrossRef]

60. Faragher, R.; Harle, R. An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications. In Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS Conference), Tampa, FL, USA, 8–12 September 2014; pp. 201–210.

61. Qualcomm Technologies Inc. LTE Direct Always-on Device-to-Device Proximal Discovery; Technical Report; Qualcomm: San Diego, CA, USA, 2014.

APPENDIX OF ALL THE CODES USED IN THIS THESIS

```
function varargout = User_Interfacefig(varargin)
% USER_INTERFACEFIG MATLAB code for User_Interfacefig.fig
%      USER_INTERFACEFIG, by itself, creates a new USER_INTERFACEFIG
or raises the existing
%      singleton*.
%
%          H = USER_INTERFACEFIG returns the handle to a new
USER_INTERFACEFIG or the handle to
%      the existing singleton*.
%
%        USER_INTERFACEFIG('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in USER_INTERFACEFIG.M with the given
input arguments.
%
%          USER_INTERFACEFIG('Property','Value',...) creates a new
USER_INTERFACEFIG or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%       applied to the GUI before User_Interfacefig_OpeningFcn gets
called.  An
%       unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to User_Interfacefig_OpeningFcn
via varargin.
%
```

```
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

%  Edit  the  above  text  to  modify  the  response  to  help
User_Interfacefig

% Last Modified by GUIDE v2.5 17-Mar-2015 14:04:23

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',  @User_Interfacefig_OpeningFcn,
...
                   'gui_OutputFcn',    @User_Interfacefig_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before User_Interfacefig is made visible.
function User_Interfacefig_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin     command line arguments to User_Interfacefig (see
VARARGIN)

% Choose default command line output for User_Interfacefig
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes User_Interfacefig wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = User_Interfacefig_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes during object creation, after setting all properties.
function V_COST1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to V_COST1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called



% --- Executes during object creation, after setting all properties.
function V_COST2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to V_COST2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% load small data
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDAT)
%%%%%%------------------------------------------------------------
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: C:\Stuff\Downloads\MS BSAN\Quarter 2\OPR
%    620\Project\test\Data.xlsx Worksheet: Small
```

```
%
% To extend the code for use with different selected data or a
different
% spreadsheet, generate a function instead of a script.

% Auto-generated by MATLAB on 2015/03/15 18:38:32

%% Import the data
[~, ~, raw] = xlsread('C:\Stuff\Downloads\MS BSAN\Quarter 2\OPR
620\Project\test\Data.xlsx','Small','A2:H4');
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) =
{''};
cellVectors = raw(:,1);
raw = raw(:,[2,3,4,5,6,7,8]);

%% Create output variable
data = reshape([raw{:}],size(raw));

%% Allocate imported array to column variable names
Appliance = cellVectors(:,1);
Job = data(:,1);
RequestHour = data(:,2);
Duration = data(:,3);
StartHour = data(:,4);
DelayCost = data(:,5);
PowerCost = data(:,6);
TotalCost = data(:,7);

%% Clear temporary variables
clearvars data raw cellVectors
%%%%%%%555555555555555555555555555555555555555555555555555555555
```

```
Schedule = [Appliance    num2cell([Job  RequestHour    Duration
     StartHour DelayCost PowerCost TotalCost])];
set(handles.uitable1,'data',Schedule);
set(handles.pushbutton4,'Visible','On');

% solve button
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% without scheduling
set(handles.edit3,'String','100');
%with scheduling
set(handles.edit4,'String','200');
%savings
set(handles.edit5,'String','100');

% display graph button
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
active = get(hObject,'Value');
if active
    set(handles.axes2,'Visible','On');
    set(hObject,'String','Hide Graph');
else
    set(handles.axes2,'Visible','Off');
```

```
    set(hObject,'String','Display Graph');
end

% reset button
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%reset values
set(handles.uitable1,'data',[]);
set(handles.axes2,'Visible','Off');
set(handles.pushbutton4,'Visible','Off');
set(handles.pushbutton3,'String','Display Graph');
% without scheduling
set(handles.edit3,'String','');
%with scheduling
set(handles.edit4,'String','');
%savings
set(handles.edit5,'String','');
%total power
set(handles.edit6,'String','');
%power utilized
set(handles.edit7,'String','');
%set slider vallue to zero
set(handles.slider1,'Value',0);

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if       ispc       &&       isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4
as a double


% --- Executes during object creation, after setting all properties.
```

```
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if       ispc       &&       isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5
as a double



% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if       ispc       &&       isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% load large data
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: C:\Stuff\Downloads\MS BSAN\Quarter 2\OPR
%    620\Project\test\Data.xlsx Worksheet: Large
%
% To extend the code for use with different selected data or a
different
% spreadsheet, generate a function instead of a script.

% Auto-generated by MATLAB on 2015/03/17 00:55:59

%% Import the data
[~, ~, raw] = xlsread('C:\Stuff\Downloads\MS BSAN\Quarter 2\OPR
620\Project\test\Data.xlsx','Large','A2:H13');
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) =
{''};
```

```
cellVectors = raw(:,1);
raw = raw(:,[2,3,4,5,6,7,8]);

%% Create output variable
data = reshape([raw{:}],size(raw));

%% Allocate imported array to column variable names
Appliance = cellVectors(:,1);
Job = data(:,1);
RequestHour = data(:,2);
Duration = data(:,3);
StartHour = data(:,4);
DelayCost = data(:,5);
PowerCost = data(:,6);
TotalCost = data(:,7);

%% Clear temporary variables
clearvars data raw cellVectors;

Schedule = [Appliance    num2cell([Job  RequestHour    Duration
     StartHour DelayCost PowerCost TotalCost])];
set(handles.uitable1,'data',Schedule);
set(handles.pushbutton4,'Visible','On');



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6
as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if       ispc       &&       isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7
as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if       ispc       &&       isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%           get(hObject,'Min')  and  get(hObject,'Max')  to  determine
range of slider
power=get(hObject,'Value');
power=round(power);
set(handles.edit6,'String',power);

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if                            isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```