NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

COMPUTER NETWORKING

MSc THESIS

Machine Learning driven road safety over smart Networks based on Electroencephalography data analysis

Panagiotis N. Kontopoulos

Supervisors **Athanasia Alonistioti,** Associate Professor

ATHENS

December 2019

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΚΤΥΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Οδική Ασφάλεια σε Έξυπνα Δίκτυα οδηγούμενη από Μηχανική Μάθηση βασισμένη στην ανάλυση δεδομένων Ηλεκτροεγκεφαλογραφίας

Παναγιώτης Ν. Κοντόπουλος

Επιβλέποντες: **Αθανασία Αλωνιστιώτη,** Αναπληρώτρια Καθηγήτρια

ΑΘΗΝΑ

Δεκέμβριος 2019

MSc THESIS

Machine Learning driven road safety over smart Networks based on Electroencephalography data analysis

Panagiotis N. Kontopoulos
S.N: M1537

Supervisors **Athanasia Alonistioti,** Associate Professor

December 2019

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Οδική Ασφάλεια σε Έξυπνα Δίκτυα οδηγούμενη από Μηχανική Μάθηση βασισμένη στην ανάλυση δεδομένων Ηλεκτροεγκεφαλογραφίας

Παναγιώτης Ν. Κοντόπουλος
Α.Μ: Μ1537

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Αθανασία Αλωνιστιώτη,** Αναπληρώτρια Καθηγήτρια

Δεκέμβριος 2019

ABSTRACT

I present a solution for road safety using EEG data provided from EMOTIV Technology and machine learning analysis to extract when a driver has a problem and cannot operate the vehicle. The solution was tested in two SDN enabled platforms EmPOWER and Mininet-WiFi. EmPOWER rests on a single platform, which consists of general-purpose hardware and operating system, and provides new features and capabilities both for the wireless and the mobile domain. It also provides three types of virtualized network resources such as forwarding nodes (OpenFlow switches), packet processing nodes (Micro Servers) and radio processing nodes (WiFi Access Points or LTE eNodeBs). Mininet-WiFi is an emulator, which creates a virtual network, kernel, switch and application code, on a single machine and allows the using of both WiFi Stations and Access Points. Moreover, EMOTIV is a platform that uses electroencephalography (EEG) sensors to measure signals of a brain and convert them to raw data, that later can be used for research and other purposes. I implemented the EmPOWER platform on a local network with a machine acting as the controller, one server acting as an edge server, one virtual machine on a cloud server with the role of the cloud server of the application, one router acting like packet processor and two routers with wireless capable hardware having the role of access points to which the clients can connect to. For the Mininet-WiFi, I run the complete setup on a single virtual machine and its topology consists from one controller server connected to OpenFlow switch, which connects the edge server, the cloud server and the Access Point. The clients are two laptops connected to the WiFi of the EmPOWER platform or two simulated nodes connected to the Mininet-WiFi's Access Point. The application consists from three deployments, a cloud server, which creates the training model and stores incidents to a database, an edge server, which classifies the data received and takes the proper actions, and a client, which send EEG and position data to the edge server and receives commands from the same server. It provides the ability to check if the operators of a vehicle, client of the above networks, has his eyes closed and therefore be able to continue his task or a two-step action is required. On the first stage, if the operator has his eyes closed, a wakeup action is taking place to his terminal. On the second stage, if the operator does not open his eyes, then an action to stop the vehicle is given and if any other operator is at close proximity, he is informed about the situation from his terminal. Moreover, the decision for the actions is taken by the edge server using the KNN algorithm, which classifies, the raw data received, to two categories, eyes close and eyes open. The communication between the servers and the client uses the MQTT protocol, which is a lightweight publish/subscribe messaging transport and is ideal for mobile applications in Internet of Things scenarios and has efficient distribution of information to one or many receivers. The results demonstrate the usage, such technologies will have to road safety by utilizing smart network topologies, user monitoring solutions and machine learning techniques. Taking into consideration the benefits of the aforementioned technologies, I conclude that such solutions show promising advantages on the area of road safety, but leave room for improvement as they are at an early stage of development.

SUBJECT AREA: Smart Network Applications

KEYWORDS: network, machine learning, classification, Internet of Things, EEG, SDN

ΠΕΡΙΛΗΨΗ

Παρουσιάζω μια λύση για την οδική ασφάλεια χρησιμοποιώντας τα δεδομένα EEG που παρέχονται από την τεχνολογία EMOTIV και την ανάλυση μηχανικής μάθησης για να εξαγάγετε, πότε ο οδηγός έχει πρόβλημα και δεν μπορεί να λειτουργήσει το όχημα. Η λύση δοκιμάστηκε σε δύο SDN ενεργοποιημένες πλατφόρμες EmPOWER και Mininet-WiFi. Το EmPOWER στηρίζεται σε μια ενιαία πλατφόρμα, η οποία αποτελείται από υλικό γενικής χρήσης και λειτουργικό σύστημα και παρέχει νέες δυνατότητες και δυνατότητες τόσο για τον ασύρματο όσο και για τον τομέα κινητής τηλεφωνίας. Παρέχει επίσης τρεις τύπους εικόνικών πόρων δικτύου, όπως κόμβοι προώθησης (switches OpenFlow), κόμβους επεξεργασίας πακέτων (Micro Servers) και κόμβους ραδιοφωνικής επεξεργασίας (σημεία πρόσβασης WiFi ή LTE eNodeBs). Το Mininet-WiFi είναι ένας εξομοιωτής, ο οποίος δημιουργεί ένα εικονικό δίκτυο, πυρήνα, διακομιστή και κώδικα εφαρμογής, σε ένα μόνο μηχάνημα και επιτρέπει τη χρήση και των δύο WiFi Σταθμών και Σημείων Πρόσβασης. Επιπλέον, το EMOTIV είναι μια πλατφόρμα που χρησιμοποιεί αισθητήρες ηλεκτροεγκεφαλογραφίας (EEG) για τη μέτρηση σημάτων ενός εγκεφάλου και τη μετατροπή τους σε ανεπεξέργαστα δεδομένα, τα οποία αργότερα μπορούν να χρησιμοποιηθούν για ερευνητικούς και άλλους σκοπούς. Έθεσα την πλατφόρμα EmPOWER σε ένα τοπικό δίκτυο με ένα μηχάνημα που λειτουργούσε ως ελεγκτής, ένας εξυπηρετητή που λειτουργούσε ως εξυπηρετητή άκρης, μία εικονική μηχανή σε ένα εξυπηρετητή σε αρχιτεκτονική τύπου σύννεφο με το ρόλο του εξυπηρετητή τύπου σύννεφου της εφαρμογής, ένας δρομολογητής που ενεργούσε ως επεξεργαστής πακέτων και δύο δρομολογητές με δυνατότητα ασύρματης επικοινωνίας που έχει το ρόλο σημείων πρόσβασης στα οποία μπορούν να συνδεθούν οι πελάτες. Για το Mininet-WiFi, τρέχω την πλήρη εγκατάσταση σε μια ενιαία εικονική μηχανή και η τοπολογία της αποτελείται από ένα εξυπηρετητή ελεγκτή συνδεδεμένο σε διακομιστή OpenFlow, ο οποίος συνδέει τον κεντρικό διακομιστή, τον εξυπηρετητή σύννεφων και το Access Point. Οι πελάτες είναι δύο φορητοί υπολογιστές συνδεδεμένοι στο WiFi της πλατφόρμας EmPOWER ή δύο προσομοιωμένοι κόμβοι συνδεδεμένοι στο Σημείο Πρόσβασης του Mininet-WiFi. Η εφαρμογή αποτελείται από τρεις υλοποιήσεις, έναν διακομιστή σύννεφο, ο οποίος δημιουργεί το μοντέλο εκπαίδευσης και αποθηκεύει συμβάντα σε μια βάση δεδομένων, έναν εξυπηρετητή άκρων, ο οποίος ταξινομεί τα ληφθέντα δεδομένα και λαμβάνει τις κατάλληλες ενέργειες, και έναν πελάτη, ο οποίος στέλνει EEG και δεδομένα θέσης στην end server και λαμβάνει εντολές από τον ίδιο διακομιστή. Παρέχει τη δυνατότητα να ελέγξει εάν οι χειριστές ενός οχήματος, πελάτη των παραπάνω δικτύων, έχει τα μάτια του κλειστά και συνεπώς να είναι σε θέση να συνεχίσει την αποστολή του ή απαιτείται δράση δύο βημάτων. Στο πρώτο στάδιο, αν ο χειριστής έχει τα μάτια κλειστά, λαμβάνει χώρα μια ενέργεια αφύπνισης στο τερματικό του. Στο δεύτερο στάδιο, εάν ο χειριστής δεν ανοίξει τα μάτια του, τότε δίνεται μια ενέργεια για να σταματήσει το όχημα και αν κάποιος άλλος χειριστής βρίσκεται κοντά, ενημερώνεται για την κατάσταση από το τερματικό του. Επιπλέον, η απόφαση για τις ενέργειες λαμβάνεται από τον κεντρικό εξυπηρετητή χρησιμοποιώντας τον αλγόριθμο KNN, ο οποίος ταξινομεί, τα ακατέργαστα δεδομένα που λαμβάνονται, σε δύο κατηγορίες, κοντά στα μάτια και τα μάτια ανοιχτά. Η επικοινωνία μεταξύ των εξυπηρετητών και του πελάτη χρησιμοποιεί το πρωτόκολλο MQTT, το οποίο είναι μια ελαφριά μεταφορά μηνυμάτων μετάδοσης / εγγραφής και είναι ιδανικό για κινητές εφαρμογές σε σενάρια του Διαδικτύου των Πραγμάτων και έχει αποτελεσματική κατανομή πληροφοριών σε έναν ή πολλούς δέκτες. Τα αποτελέσματα καταδεικνύουν τη χρήση αυτών των τεχνολογιών στην οδική ασφάλεια, χρησιμοποιώντας τοπολογίες έξυπνων δικτύων, λύσεις παρακολούθησης χρηστών και τεχνικές εκμάθησης μηχανών. Λαμβάνοντας υπόψη τα οφέλη των προαναφερόμενων τεχνολογιών, καταλήγω στο συμπέρασμα ότι τέτοιες λύσεις παρουσιάζουν πολλά υποσχόμενα πλεονεκτήματα στον

τομέα της οδικής ασφάλειας, αλλά αφήνουν περιθώρια βελτίωσης καθώς βρίσκονται σε πρώιμο στάδιο ανάπτυξης.

ACKNOWLEDGEMENTS

For the fulfillment of this Thesis, I would like to thank a number of people that helped me during its completion. First of all, I would like to thank our supervisor, Associate professor Athanasia Alonistioti who believed in me and provided me with the knowledge and support needed for this thesis. Also, I would like to thank her for allowing me to use SCAN Lab's equipment, in order to implement and test the platforms and the applications. Another person I would like to thank is Dr. Sokratis Barmpounakis, from the SCAN Lab Team at National and Kapodistrian University of Athens, for providing guidance with the network related knowledge needed to use, implement and explore the platforms. Also, I would like to thank Lina Magoula and Nikolaos Koursioumpas (Members of SCAN Lab) for the help they provided regarding the help they provided with the use and implementation of the machine learning algorithm used in this thesis. Finally, I would like to thank my family and friends for their invaluable support during all these years.

CONTENTS

LIST OF FIGURES

LIST OF IMAGES

LIST OF TABLES

PREFACE

This thesis was developed in Athens from November 2018 till November 2019. It constitutes an essential part for acquiring a degree and was conducted during my last year as Master in Science Student in the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. The concepts that are presented are state of the art technologies and are used in many new generation networks, IoT scenarios and EEG data collection. The platforms, that were used, are dynamic, continuously updated and provide freedom for development and experimentation. At this point I would like to thank my thesis supervisor, Associate Professor, Athanasia Alonistioti for giving me the opportunity to implement, configure and examine these platforms with SCAN laboratory's equipment, as well as to develop applications, which constitutes an initial effort of experimentation with the EEG data and machine learning over smart networks.

1. INTRODUCTION

The use of machine learning has opened up many opportunities to tackle problem of the past from a new perspective or tackle problems that were difficult to solve. Moreover, with the increase in computation power from the mobile devices and computers in general, decisions on a specific time sensitive problem can been taken quicker than ever before. Also, combined with the new generation of smart networks edge computing has provided a new distributed way to assist to computation problems near the edge of such networks.

1.1 The transition from traditional networks to Next Generation Networks

The growth of mobile devices, the adaptation of server virtualization and the explosion of cloud services, the shift towards edge computing made clear to the networking industry that the current traditional network architecture needs to be reexamined. Nowadays the once dominant client-server computing is replaced with dynamic computing trends where the static architecture of the conventional networks does not meet the needs of today's business environments and enterprise data centers. The rise of the cloud services, the changing traffic patterns and the explosion of "Big Data" are only some of the new computing trends that need to be addressed today. [10] Furthermore the current traditional networks have proved to be insufficient meeting the emerging demanding needs. Network architectures have enough limitations such as scalability, contradictory policies and dependence from the provider, hence the new user's and enterprise's needs cannot be fully coped.

The Software Defined Networking (SDN) is a technology that, although it is part of a long story of efforts to make the computer networks more programmable. The main concept behind the SDN is the separation of the control plane – decide how the traffic will be handled – from the data plane – forward traffic according to the controller's directives. Between the control plane and the data plane, resides a communication interface that allows direct access to the forwarding plane of network devices such as switches and routers, both physical and virtual as well as manipulation of this elements. The first standard communications interface was OpenFlow, which was created in 2008 and is still being managed from the Open Networking Foundation (ONF). OpenFlow provided the basic tools to enable dynamic flow control by establishing new open communication protocols. This enhanced the SDN concept and brought into surface a plethora of benefits.

5G Networks are the next generation of 4G LTE Networks that will not only provide faster speed and transfers, but also deliver new levels of performance and efficiency that will empower new user experiences and connect new industries. This network aims to expand on mission-critical communications and connecting the massive IoT. This is enabled by many new 5G NR air interface design techniques that focus on enhanced mobile broadband (eMBB) use cases to boost capacity and provide an elevated mobile broadband experience. [11]

1.2 Internet of Things concept and uses

The concept of Internet of Things has risen the past decade with the increase of connected devices to the internet. Basically, IoT is the internet-able nature of modern physical devices, vehicles, and buildings, referred to as connected devices or smart devices. Application for IoT go further than just consumer products, though. Large-scale projects, such as smart grids, smart homes, intelligent transportation, and smart cities are relying on massive interconnection. Furthermore, the use of such devices enables new scenarios to use technology to medical areas. The Internet of Medical Things (IoMT), (also called the Internet of health things), is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. [13]

1.3 EEG technologies and uses

An electroencephalogram (EEG) is a test used to evaluate the electrical activity in the brain. Brain cells communicate with each other through electrical impulses. An EEG can be used to help detect potential problems associated with this activity. This test tracks and records brain wave patterns by using small flat metal discs called electrodes, which are attached to the scalp with wires. The electrodes analyze the electrical impulses in the brain and send signals to a computer that records the results. More recent versions of this technology can be used in real time scenarios and provide accurate measurements of a subject, while an intense physical activity occurs. Also with the advancements in this technology, availability has reached the consumers market with many solutions readily available for everyday uses or experiments. [14]

1.4 Machine Learning in solving problems

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as image recognition and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task [12]. There are four main categories of ML supervised, unsupervised, semi-supervised and reinforcement machine learning. Supervised ML takes what has been learned in the past to new data using labeled examples to predict future events, unsupervised is used when the information used to train is neither classified nor labeled, semi-supervised falls somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training and reinforcement machine learning is a learning method that interacts with its environment by producing actions and discovers errors or rewards.[15]

1.5 Road and Car Safety

Road safety is a major category of research and development, especially in moderns days that cars become faster, there are higher number of cars on the road, than there were ever before, but also technology of smart cars progresses with high speeds towards automation of many car and road systems. Many companies have developed solutions to assist the drivers and warn them when the conditions while travelling are not ideal and help to prevent accidents. For example, by using cameras cars can assist the driver while parking or while driving and has no clear viewing angles, creation of complex safety systems for breaking or air bag deployment. These needs for improved road and car safety have shifted the efforts to enter the domain of real time video and EEG data analysis to have a more accurate prediction regarding driver fatigue, object tracking etc.

2. The 5G-EmPOWER Testbed

The 5G-EmPOWER platform was initially designed for the wireless and the wired domain of software defined networks. Subsequently it enhanced its capabilities by supporting the mobile domain. In addition it has expand its capabilities to establish the experimentation of 5G that is an emerging concept. As Dr.Riggio highlights, 5G-EmPOWER is an open Mobile Network operating System for SDN/NFV research and experimentation in heterogeneous mobile networks [9]. However, the scope of this thesis will limit mainly in the use of the platform as one of the basis networks to contact the experiments and run the client and server applications. The platform is continuously updating in order to keep up with the hot trends in software defined networking. Therefore, the architecture of the platform will be described as it was during the development of this thesis.

2.1 5G-EmPOWER Architecture

The 5G-EmPOWER Architecture comprises a Single Master - the 5G-EmPOWER Master - and multiple agents – 5G-EmPOWER Agents – running on each Access Point (AP) or Wireless Termination Point (WTP) in the language of 5G-EmPOWER. The high-level system architecture is presented on image 1.

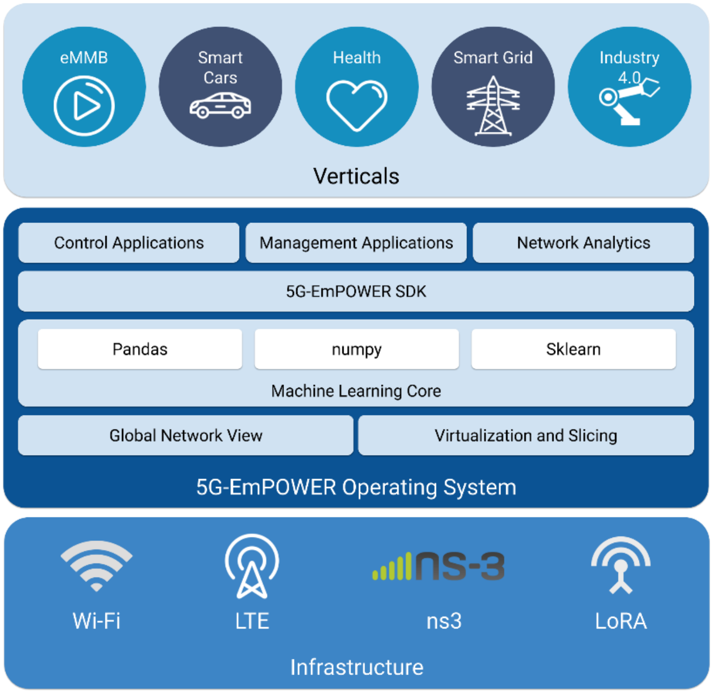

Image 1 5G-EmPOWER high level system architecture [4]

The 5G-EmPOWER master runs on top of an OpenFlow Controller (e.g. Floodlight) and has a global view of the network. This allow the controller to have knowledge of the clients, the flows and the infrastructure of the network. The 5G-EmPOWER agents handle multiple clients as a collection of logical isolated clients. Each client has the illusion that is connected to a port of a switch, due to the virtualization that the controller provides. [4] As it can been seen a REST Interface plays the role of the Northbound Interface and gives the ability to network applications, which ran on top of the controller, to request resources and data from the network or to register new rules to the network. It is important to mention that each network application runs in an isolated slice and therefore it affects all or a subset of the network. As a result a plethora of experiments can be held without interfering with each other.

Due to the fact that the whole 5G-EmPOWER platform is built on open source tools – OpenVSwitch and Click Modular Router for the datapath and Floodlight as the controller but in a newer version it has been replaced with the network operating system that EmPOWER provides – it makes it even more attractive to experiment on top of this testbed. Furthermore, except from the REST interface, the network applications can exploit the Pyretic interpreter, which is integrated in the framework in order to allow experimenters to use the programmatic capabilities that this compassable language offers.

As mentioned before the agents running on the WTPs allow multiple clients to be treated as a collection of logically isolated clients connected to different ports of a switch. This is being achieved in 5G-EmPOWER with the Light Virtual Access Point (LVAP) abstraction. [6] This leverages the network by separating association/authentication from the physical connection between a client and an AP. With LVAPs, a unique basic service set identifier (BSSID) is generated for every client that tries to associate with the WLAN. Hence every client has the impression of owning the AP. In like manner, each AP hosts an LVAP for each connected client. The power of this abstraction can be understood during the migration of an LVAP between two physical APs. This process, which is called client handover, does not require anymore the re-association and re-authentication steps that are essential in the traditional WLANs.

All in all, the whole system provides programming abstractions to developers in order to ease the development of complex network applications. The exploitation of a “logically centralized” architecture by these high-level programming abstractions gives the ability for the experimenter to control the behavior of the network. Network applications are able to register events in relation to the network’s conditions and generate triggers when a condition has changed. For instance the mobility manager scheme minimizes for the developer the interaction with all the Wi-Fi implementation details, such as directly handling the IEEE 802.11 protocol or finding workarounds for the handover process.

5G-EmPOWER WTPs build upon a programmable hypervisor. The hypervisor controls the lifecycle of the slices and is responsible for their creation, monitoring, and management.

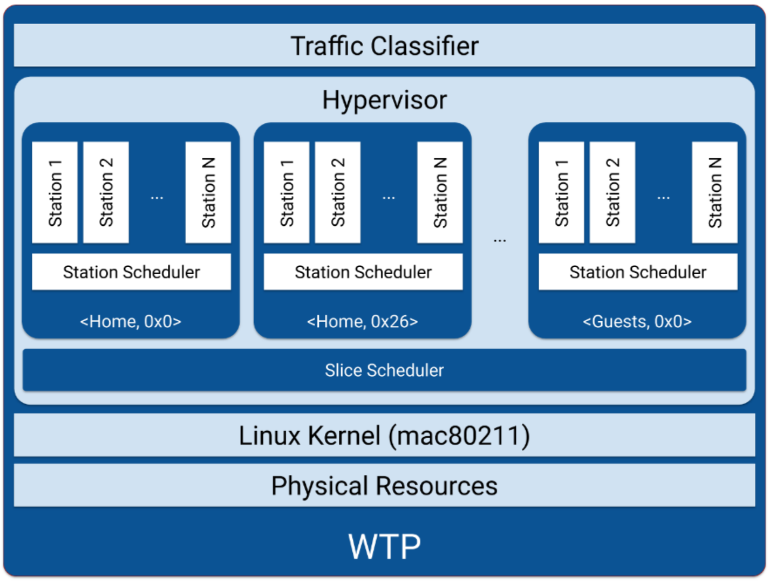

Figure 1 The high-level architecture of the hypervisor

5G-EmPOWER provides an open ecosystem where new 5G services can be tested in realistic conditions.

2.2 5G-EmPOWER Components, Modules and Abstractions

The 5G-EmPOWER testbed was built in order to provide an easy way for the administrators to handle and observe the network conditions. Furthermore, it contains abstractions that enable the developer to create his own network applications and experiment with them on top of a network slice. A plethora of components, modules and abstractions exist in the platform, much of them are outside of the scope of this thesis. However the most basic set of them will be described below in order to better explicate the platform and subsequently the application that was developed.

The 5G-EmPOWER Operating System consists of the following components:

- empower-runtime, the Python-based 5G-EmPOWER Controller. This allows network apps to control Wi-Fi APs and LTE eNBs using either a REST API or a Python API.
- empower-lvap-agent, the 5G-EmPOWER Wi-Fi agent. This agent allows controlling Wi-Fi access points using the empower-runtime. While in principle it is possible to install this agent on any Linux box you will avoid a lot of pain if you use our OpenWRT branch which includes all the necessary Kernel patches.
- empower-enb-agent, the 5G-EmPOWER LTE agent library. This agent allows controlling LTE eNBs using the empower-runtime. The agent can be integrated with any LTE stack however for the moment we officially support only srsLTE.
- empower-srsLTE, a branch of srsLTE with the 5G-EmPOWER eNB agent.
- empower-openwrt-packages, the 'empower-lvap-agent' package for OpenWRT 19.07.

- empower-openwrt, a branch of OpenWRT 19.07 including some Kernel patches necessary for the correct operation of the 'empower-lvap-agent'.
- empower-config, the configuration files for the Wi-Fi WTPs.

Wireless Termination Points (WTP)

The Wireless Termination Points are the physical points of attachment in the Radio Access Network (RAN), providing clients with wireless connectivity. In a Wifi network WTPs are similar to Access Points (APs), while in a LTE network with eNodeBs (eNBs). The WTPs are connected to the Controller through a secure channel.

Light Virtual Access Point (LVAP)

The LVAP abstraction provides a high-level interface for wireless clients' state management. This abstraction offers an efficient way of handling technology-dependent details such as association/disassociation, authentication/deauthentication, handover and resource scheduling. The creation of a new LVAP is being triggered every time a client tries to connect to the network. The WTP on the other hand, will host as many LVAPs as the number of wireless clients that are under its control. More precisely, LVAPs are the extension of VFNs on the new generation networks.

Channel Quality and Interference Maps

The channel quality map is an abstraction that provides a full view of the network state to the programmers in terms of channel quality between LVAPs and WTPs. This information is made available with the Resource Blocks. The channel quality map is available to the programmer via two data structures, the User Channel Quality Map (UCQM) and the Network Channel Quality Map (NCQM). These two data structures are 3-dimensional matrices where each entry is the channel quality (available in dBm) over one Resource Block between two entities. For the UCQM the first entity is the LVAP and the second is the WTP, whereas in the NCQM the channel quality is measured between two WTPs.

These abstractions are very important in an 5G-EmPOWER network, as they can be used for selecting the Resource Blocks that satisfy Quality of Service (QoS) requirements of an LVAP. As it will be analyzed in section 3.4 this is useful for the handover procedure - as the channel quality and interference maps provide the available resource blocks of the network, where the LVAP can be scheduled on, after a handover.

Port

In contrast with a switched Ethernet LAN, where links are deterministic and the status of a port in a switch is binary (active or not active), in a wireless network the links are stochastic. This leads to a state, where physical layer parameters that characterize the link between an LVAP and a WTP, such as transmission power, modulation, coding schemes and MIMO (Multiple Input Multiple Output) configuration must be adapted to the actual network conditions. This adaptation requires real-time coordination between LVAPs and WTPs and can only be implemented near the air interface. Therefore the port abstraction is essential as it gives the ability to the controller to establish a new control policy if the optimal operating conditions are not met.

The definition of a port is a 3-tuple <p,m,a>, where p is the transmission power, m is the available modulation and coding schemes (MCS) and a is the MIMO configuration (number of spatial streams). [5]

In summary, the port abstraction specifies the configuration of the link between a WTP and an LVAP over a certain Resource Block. In addition, a range of parameters important for the communication between the WTP and the LVAP are specified in the port configuration. Consequently a WTP will have as many port configurations as the number of LVAPs that is currently managing.

2.3 5G-EmPOWER Installation

For the installation of the Empower Platform the hardware that was used is:

- Two Soekris net5501 [1 net5501-60 (Single Core 433 MHz, 256 MB RAM), 1 net5501-70 (Single Core 500 MHz, 512 MB RAM)] in which the empower-openwrt-15.05 image was installed and have the role of WTPs (Wireless Termination Point. The network element providing client with wireless connectivity, i.e. an Access Point in IEEE 802.11 terminology.) Also two 300Mbps Wireless N PCI TL-WN951N Adapters were used, one in each soekris, as wireless interface.
- One PC (virtual machine) with Debian 8.4 Jessie (Single Core 2.4 GHz CPU, 2048 MB RAM), in which Python 3.4.2 and all required libraries (python3-tornado (Version 4.2.1), python3-sqlalchemy (Version 1.0.8), python3-construct (Version 2.5.2), protobuf (Version 3.0.0), protobuf3-to-dict (Version 0.1.2)) were installed, in order to run the empower-runtime controller.
- Two IBMs R51, with one 150Mbps High Gain Wireless USB TL-WN722N Adapter each, which have the role of clients or LVAPs [(Light Virtual Access Point). One LVAP is created for each wireless client in the network. The LVAP interface supports seamless handover and radio resource management (e.g. channel assignment)] of the network.

The controller machine is connected to a LAN network via Ethernet where a router has the role of DHCP Server. On the same network the two WTPs are connected through Ethernet connections and belong to the same subnet of the controller in order to communicate with it (between the controller and the router or the WTP and the router switch/es may reside). This topology is presented below on image 2.

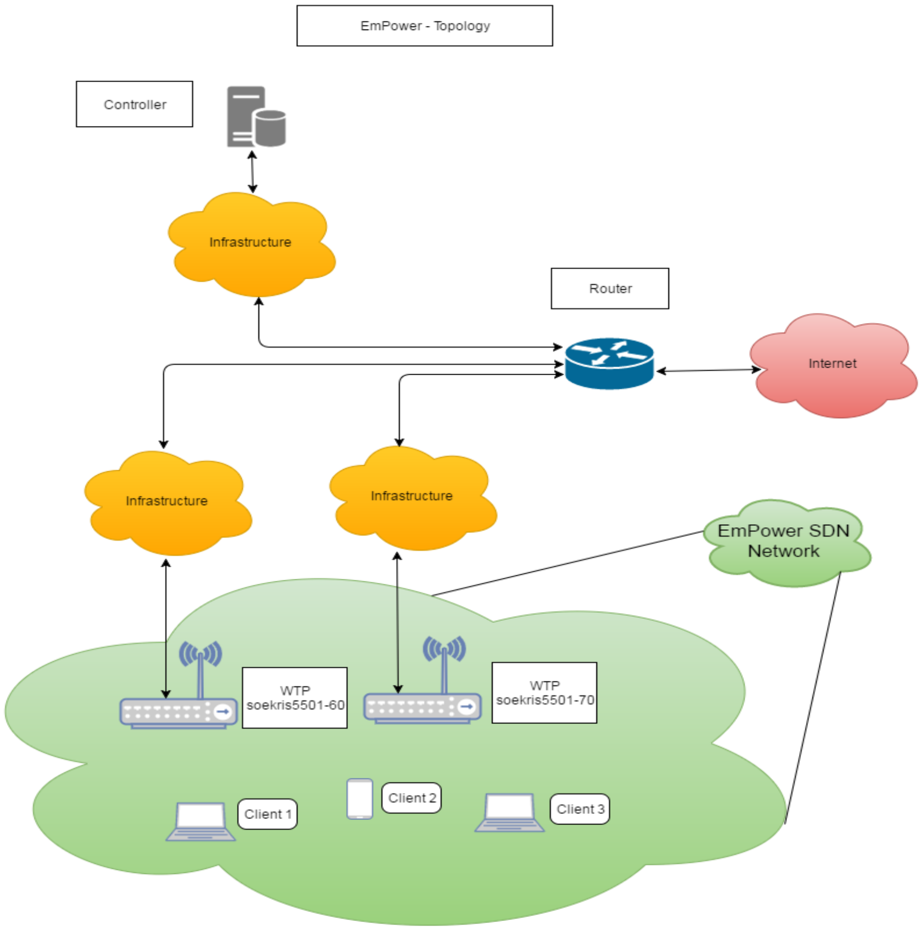

Image 2 Topology of the Empower Platform setup in SCAN Lab

3. Mininet-Wifi Definition and Deployment

Mininet-WiFi is a network emulator or perhaps more precisely a network emulation orchestration system. The project is a fork of Mininet (http://mininet.org/) which allows the using of both WiFi Stations and Access Points. Mininet-WiFi only add features and can be worked like working with Mininet, but extends it to the wireless domain. It runs a collection of end-hosts, switches, routers, and links on a single Linux kernel. It uses lightweight virtualization to make a single system look like a complete network, running the same kernel, system, and user code. A Mininet-WiFi host behaves just like a real machine and run arbitrary programs (including anything that is installed on the underlying Linux system.) The programs you run can send packets through what seems like a real Ethernet interface, with a given link speed and delay. Packets get processed by what looks like a real Ethernet switch, router, or middlebox, with a given amount of queueing. When two programs, like an iperf client and server, communicate through Mininet-WiFi, the measured performance should match that of two (slower) native machines.

In short, Mininet-WiFi's virtual hosts, switches, links, and controllers are the real thing – they are just created using software rather than hardware – and for the most part their behavior is similar to discrete hardware elements. Although, Mininet-WiFi offers many advantages for wireless network simulations, it does have some limitations, because when running multiple nodes on a single machine the performance of those nodes is restricted by that host machine.

Running on a single system is convenient, but it imposes resource limits: if your server has 3 GHz of CPU and can switch about 10 Gbps of simulated traffic, those resources will need to be balanced and shared among your virtual hosts and switches.

By default your Mininet-WiFi network is isolated from real life LAN and from the internet. However, the use of NAT option can connect Mininet-WiFi network to your LAN via Network Address Translation. Also, a real (or virtual) hardware interface can be attached to the network.

The topology for this thesis consists from:

- Two emulated moving cars (clients) that are connected to the networks eNodeBs and transmit their GPS coordinates (extracted from SUMO), while running the client application.
- Two eNodeBs for the cars to connect two, basically the Access Points for the cars to connect to the emulated wireless network.
- One OpenFlow enabled switch, which connects the controller, the edge server and the two eNodeBs. It receives the flows needed from the controller.
- The Edge Server that is responsible for the classification and everything needed for the actions of the classification
- The Cloud Server, which is responsible for the creation of the model used for classification and logging.

Basically, the same applications used for the previous scenario are used on this one and all that changes is the topology.

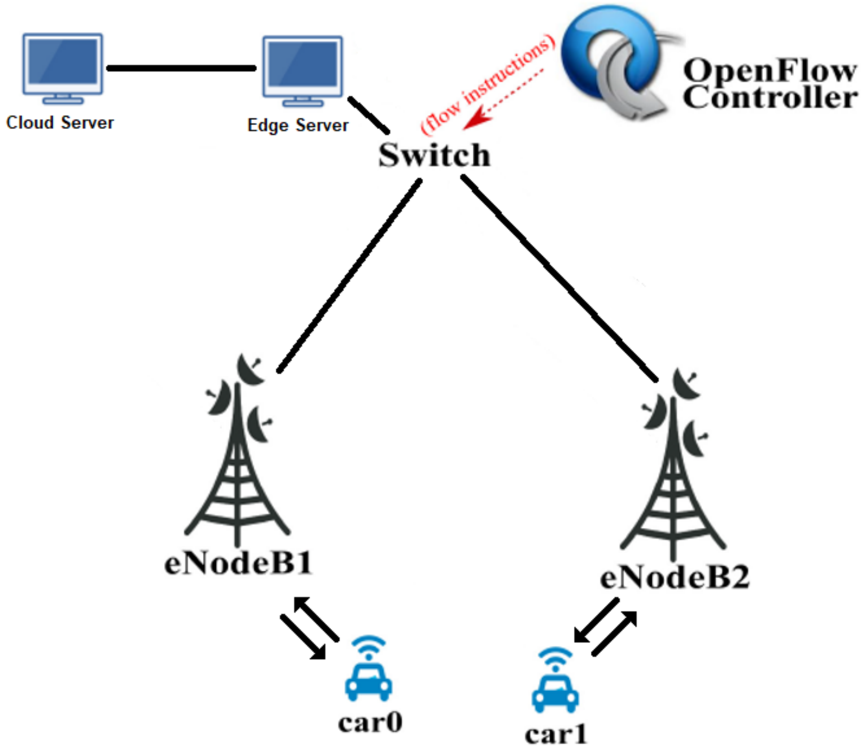

Image 3 Mininet-WiFi Emulated Topology

4. OpenDayLight

OpenDaylight (ODL) is a modular open platform for customizing and automating networks of any size and scale. The OpenDaylight Project arose out of the SDN movement, with a clear focus on network programmability. It was designed from the outset as a foundation for commercial solutions that address a variety of use cases in existing network environments. It is a highly available, modular, extensible, scalable and multi-protocol controller infrastructure built for SDN deployments on modern heterogeneous multi-vendor networks. Also, it provides a model-driven service abstraction platform that allows users to write apps that easily work across a wide variety of hardware and south-bound protocols. [16] OpenDaylight code has been integrated or embedded in many vendor solutions and apps, and can be utilized within a range of services. It is also at the core of broader open source frameworks, including ONAP, OpenStack, and OPNFV.Moreover, as part of LF Networking, ODL is driven by a global, collaborative community of vendor and user organizations that continuously adapts to support the industry's broadest set of SDN and NFV use cases.

4.1 The OpenDaylight Architecture

The core of the OpenDaylight platform is the Model-Driven Service Abstraction Layer (MD-SAL). In OpenDaylight, underlying network devices and network applications are all represented as objects, or models, whose interactions are processed within the SAL. The SAL is a data exchange and adaptation mechanism between YANG models representing network devices and applications. The YANG models provide generalized descriptions of a device or application's capabilities without requiring either to know the specific implementation details of the other. Within the SAL, models are simply defined by their respective roles in a given interaction. A "producer" model implements an API and provides the API's data; a "consumer" model uses the API and consumes the API's data. While 'northbound' and 'southbound' provide a network engineer's view of the SAL, 'consumer' and 'producer' are more accurate descriptions of interactions within the SAL. For example, protocol plugin and its associated model can either be a producer of information about the underlying network, or a consumer of application instructions it receives via the SAL.

The SAL matches producers and consumers from its data stores and exchanges information. A consumer can find a provider that it's interested in. A producer can generate notifications; a consumer can receive notifications and issue RPCs to get data from providers. A producer can insert data into SAL's storage; a consumer can read data from SAL's storage. A producer implements an API and provides the API's data; a consumer uses the API and consumes the API's data.

The ODL platform is designed to allow downstream users and solution providers maximum flexibility in building a controller to leverage services created. ODL includes support for the broadest set of protocols in any SDN platform – OpenFlow, OVSDB, NETCONF, BGP and many more – that improve programmability of modern networks and solve a range of user needs. That modularity and flexibility of OpenDaylight is why it was chosen as a controller platform for this thesis.

5. MQTT

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium. That makes it ideal for the scenarios explored in this thesis for Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium. The protocol runs over TCP/IP, or over other network protocols that provide ordered, lossless, bi-directional connections. Its features include:

- Use of the publish/subscribe message pattern which provides one-to-many message distribution and decoupling of applications.
- A messaging transport that is agnostic to the content of the payload.
- Three qualities of service for message delivery:
    - "At most once", where messages are delivered according to the best efforts of the operating environment. Message loss can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.
    - "At least once", where messages are assured to arrive but duplicates can occur.
    - "Exactly once", where messages are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.

A small transport overhead and protocol exchanges minimized to reduce network traffic.

A mechanism to notify interested parties when an abnormal disconnection occurs.

For the scope of the thesis the third quality of service was used to ensure a that no loss of message occurs and there are not duplicates send over the network. [17]

6. SUMO

SUMO is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks. "Simulation of Urban MObility", or "SUMO" for short, it allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness. [18] For this thesis OpenStreetMap data were used to run the simulations for the nodes. Afterwards, the exported data where given to SUMO in order to create the movement for the cars in the scenarios. Afterwards, the extracted data are separated to csv files for each client and then each client transmits those data, alongside the EEG data.

7. Emotiv EEG

The EPOC+ for scientific research projects and educational applications. With 14 channels, the EPOC+ gives you a dense array of high-quality data. All data are provided in CSV format. The 14 channels or sensors placed on the head of the subject are located as shown below and the 14 elements that correspond to these sensors are AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4.

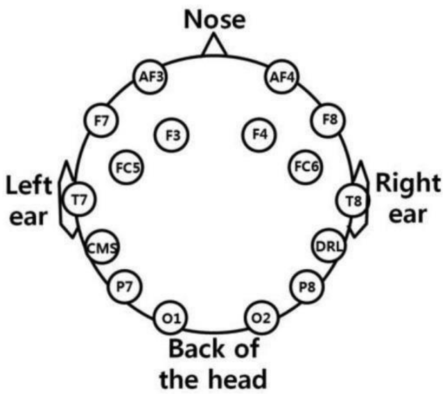

Image 4 Representation of the sensors' location on head

Also, there are 14 more elements that correspond to the quality of the connection during the data gathering process, which are CQ_AF3, CQ_F7, CQ_F3, CQ_FC5, CQ_T7, CQ_P7, CQ_O1, CQ_O2, CQ_P8, CQ_T8, CQ_FC6, CQ_F4, CQ_F8, CQ_AF4, and can take values from 0 to 4, for 0 to be no connection and 4 to be greatest quality possible. The following two points apply to all collected data:

The marker is the variable that indicates the start or end of an event / area of interest (eg at this point the user started to close their eyes), defined by a shortcut during the sampling phase. MarkerH is a variable that determines which Hardware Input (eg keyboard, mouse) MarkerH defined. In the CSV files containing the sampling values, between two non-zero markers an event occurs, such as closed or open eyes.

A graphical representation of Marker's function is shown in the following figure.

Figure 2 CSV File Representation

8. Machine Learning KNN

In this thesis for the data analysis the k-nearest neighbor algorithm (k-NN) is used. The algorithm uses Euclidean Distance between the points x, x':

$$d((x_i, x_i') = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2}$$

Figure 3 Euclidean Distance

The function counts the distance between two points on the same dimension of n-dimension space (where x and x' have n dimensions). Follows a brief step by step description of the KNN algorithm:

1. Let's say we have a collection of data for training X = {x1,x2,x3,…, xN}, where xi is a n-dimension Feature Vector
2. We know that each category that corresponds to each xi and thus we have the collection Y = {y1, y2, y3,…, yN}, where yi is the category that xi belongs to
3. Let's say we have a collection of data for evaluation Q = {q1, q2, q3, ..., qN}, where qi is a n-dimension Feature Vector
4. We choose K
    a. We calculate for q1 every Euclidean Distance with each xi and then create a list Distance_list = [d1(x1, q1), d2(x2, q1), d2(x3, q1), …, dN(xN, q1)]
    b. From the list we use the K shortest distances and thus we find the K closest neighbors xi
    c. Every xi and yi corresponds to a category, we can calculate for each K closest neighbors which yi appear the most frequently (Voting Rule) and then classify.
5. We calculate the

$$Wyi \mathrel{+}= 1 / di(\mathbf{x}i, \mathbf{q}1)$$

Figure 4 Weight of yi

6. For every category we calculate

$$\text{Yi-weight} = \textbf{\textit{yi-votes}} * \text{Wyi}$$

Figure 5 Y(i-weight) Category

7. The q1 will be classified to the category with the highest Yi-weight

9. Road Safety Application

Taking into consideration all the technologies mentioned in the previous sections, a three-part application was created to explore road safety scenarios with EEG data. For the communication of MQTT protocol, which is widely used for IoT deployments. Moreover, for the development of each sub-application Python was used to take advantage of the libraries available for data analysis and IO.

9.1 Overview and Requirments

For the entirety of this thesis many parts and technologies had to be incorporated for the creation of a fully functional system. From the communication protocols to each entity (edge server, cloud server, database server, clients) to ultimately the technologies for classification and platform to test the applications created. Every application uses python to run in order to cover the needs of such complex implementation

9.2 Client Application

The client application is the one that run on the moving nodes. The main role is to send in 'real time' the EEG data and GPS data to the Edge Server. The data are in csv format and are send every second over the network using MQTT. Furthermore, the application receives commands from the server side to inform the user of an event, in the scope of this thesis, the driver, to open his eyes and if the driver's EEG data indicate of continius non responsinveness then another command to stop the vehicle is given.

9.3 Edge Server

The Edge Server receives from the Cloud Server the training model, that is later using to classify the received EEG data. When the server receives the EEG data classifies them into two categories: eyes open, eyes closed. When the eyes of the driver are open no action is needed, as this is the default and expected state for a driver when driving a vehicle. On the other hand, if the classification indicates that the driver has his eyes closed then two actions can be set in motion depending on whether the driver opened his eyes after the first warning or not.

When the driver is found to have his eyes closed, one of two actions takes place as described below:

1. The first time a classification of eyes closed occurs, a message is send to that client's terminal to take an action (maybe play a sound or other alarms) to wake him up and resume the previous course.

2. If the above action occurs and the client continues to send data, that are classified as eyes closed, then the server starts to calculate the distance between the vehicles, using the GPS data, and warns the closest vehicles to take action (maybe stop their vehicle). At the same time, a message to stop is send to the vehicle of the driver in “danger”.

When any of the above events happens, the information of all the involved clients is send to the cloud server to catalog it to the database.

To implement the categorization algorithm, the Edge Server must extract the feature vector for each of the CSV files it receives from the terminals. The way it does that will be described on the Cloud Server below.

9.4 Cloud Server

The Cloud server has two main roles on the scope of this thesis. The first one is to train the model and then send it, with the use of MQTT, to the edge server, in order the second one to be able to do the classification. The distinction happens because in fog environments cloud servers have more computational power to train and run new models, whereas the edge server can handle up to a certain point the computational needs of real time classification to many users simultaneously. The second role is to communicate with the Database to store the critical events that are send from the edge server detection.

The Cloud Server (backhaul is responsible for the development of the trainingset which will ultimately be sent to EdgeServer. Next, the development steps of the set will be analyzed:

- Each CSV file in the Training_Set file will have to calculate the entropy of the values for each of the 14channels.
- In addition, the name of the experiment (EyesOpened / Closed) is also saved each time. The experiment name can very easily be extracted from the CSV file name. The final entropy structure is graphically illustrated in the following table.

Table 1 Feature Vector of each CSV received/send

2.EyesOpened1_10.csv	17.EyesOpened1_24.csv	18.EyesOpened1_27.csv	2.EyesOpened1_10.csv	…	20.EyesClosed1_29.csv
Feature Vector of 1st CSV	Feature Vector of 1st CSV	Feature Vector of 1st CSV	Feature Vector of 1st CSV	…	Feature Vector of n CSV

- The new CSV created needs to be sent back to the EdgeServer.

For the training set, used for the training of the classifier an indicative form is shown to the table 2.

Table 2 Training set indicative form

Experiment Name	Entropy/Feature Vector of each channel
4.EyesClosed.csv	EntropyAF3, EntropyF7,…,EntropyAF4
. . .	. . .
11.EyesClosed1.csv	EntropyAF3, EntropyF7,…,EntropyAF4

9.5 Database

The Database is a MYSQL database to store the logs created from the events. It can be located to the same computer or on a remote host. For the scope of the thesis, the database is located to a separate VM on the same server.

9.6 CONCLUSIONS AND FUTURE WORK

In this thesis an effort was made to describe the basic concepts of the SDN, EEG data collection and use, IoT communication scenarios. SDN provides the perspective to operators, developers, researchers and especially users that is needed to for the scenarios presented above. This new option of combining the use of new smart networks, EEG technologies and machine learning to tackle real life problems.

This thesis presented a combination of variety of technologies from different areas of interest. From the networks perspective, two solutions have been presented. The 5G-EmPOWER for testing on real hardware and Mininet-WiFi for emulated scenarios, in order to explore the use of the applications to different situations. For the communication MQTT protocol was used because of its versatility and light-weight nature on IoT scenarios. For the python applications, each one was designed for a specific task and the goal was to show, that distribution of computational power and compartmentation of each task can provide advantages to complex problems and provide the speed needed to tackle them. From Machine Learning's prespective K-NN was used on EEG data for its simplicity on identifying the classes needed. Moreover, SUMO was used, as a simple way to create the movement of the nodes with real coordinates on real geographical areas.

The applications were created during my research in order to experiment with the platforms and to test them in real life conditions. They acts as a distributed application on top of the platforms. It creates the infrastructure to collect the EEG and GPS data from the clients and send to the server to analyze them. In a second stage, it provides the ability to classify and take actions on the received data. This allows for on the fly classification and proper action initiation when needed. For the future, a combination between EEG data and image processing will help greatly to improve the identification of driver's status.

In conclusion, the wholesome deployment of all these technologies has proved to be a useful toolkit for investigation in road safety and other type of critical scenarios. Although they are continuously evolving and EEG data collection has not reached a really mature stage yet, it is one of the simplest solutions for experimentation and research.

TABLE OF TERMINOLOGY

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Electroencephalography	Ηλεκτροεγκεφαλογραφία
Machine Learning	Μηχανική Μάθηση
Server	Εξυπηρετητής
Switch	Διακομιστής

ABBREVIATIONS - ACRONYMS

5G	5th generation mobile networks
A-CPI	Application-Controller Plane Interface
Ajax	Asychronous Javascript and XML
AP	Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identifier
CQM	Channel Quality Map
CSS	Cascading Style Sheets
D-CPI	Data-Controller Plane Interface
DPCF	Data Plane Control Function
HTML	Hyper Text Markup Language
ISM	Industrial, scientific and medical band
IoT	Internet of Things
LVAP	Light Virtual Access Point
MCS	Modulation and Coding Schemes
MIMO	Multiple Input Multiple Output
NBI	North Bound Interface
NCQM	Network Channel Quality Map
NE	Network Element
NFV	Network Function Virtualization
NOS	Network Operating System
ONF	Open Networking Foundation
OSI	Open Systems Interconnection model
OSS	Operations Support System
QoS	Quality of Service
RDB	Resource Data Base
REST	Representational State Transfer
RSSI	Received Signal Strength Indication
SDN	Software Defined Networking
UCQM	User Channel Quality Map
WLAN	Wireless Local Area Network

WTP	Wireless Termination Point
XML	Extensible Markup Language
EEG	Electroencephalography
k-NN	k-nearest neighbors algorithm
SUMO	Simulation of Urban MObility
MQTT	MQ Telemetry Transport
MQ	Message Queuing

REFERENCES.

[1] Open Networking Foundation, SDN architecture Issue 1, ONF TR-502, June 2014.
[2] Claude Chaudet and Yoram Haddad, Wireless Software Defined Networks: Challenges and Opportunities, IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems (COMCAS), October 2013, At Tel Aviv, Israel; https://www.researchgate.net/publication/261021392_Wireless_Software_Defined_Networks_Challenges_and_opportunities
[3] Nick McKeown et al., OpenFlow: Enabling Innovation in Campus Networks, March 2008; http://archive.openflow.org/documents/openflow-wp-latest.pdf
[4] Riggio, R., T. Rasheed, and F. Granelli, "EmPOWER: A Testbed for Network Function Virtualization Research and Experimentation", IEEE SDN4FNS 2013 (Software Defined Networks for Future Networks and Services), November, 2013, Conference Paper.
[5] Riggio, R., T. Rasheed, and M. Marina, "Interference Management in Software-Defined Mobile Networks", IFIP/IEEE Integrated Network Management Symposium (IM 2015), Ottawa, May, 2015, Conference Paper.
[6] L. Suresh et al., "Towards programmable enterprise wlans with odin", in Proc. of ACM HotSDN, 2012.
[7] Adam Drescher, "A Survey of Software-Defined Wireless Networks", April 2014.
[8] Understanding the SDN Architecture, sdxcentral, https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture/
[9] http://empower.create-net.org/
[10] Open Networking Foundation, Software-Defined Networking: The New Norm for Networks, ONF White Paper, April 2012
[11] https://www.qualcomm.com/invention/5g/what-is-5g
[12] Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2
[13] Toward efficient smartification of the Internet of Things (IoT) services, Future Generation Computer Systems, Volume 92, March 2019, Pages 663-673
[14] https://www.nature.com/subjects/electroencephalography-eeg
[15] https://www.analyticsvidhya.com/blog/2019/10/mathematics-behind-machine-learning/
[16] https://www.opendaylight.org/what-we-do/odl-platform-overview
[17] http://mqtt.org/faq
[18] https://sumo.dlr.de/userdoc/Sumo_at_a_Glance.html