UNIVERSITY OF THESSALY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Πρόβλεψη και ρύθμιση χρονικά μεταβαλλόμενης οδηγικής συμπεριφοράς και αξιοποίησή της από αυτο-οδηγούμενα οχήματα

Prediction and calibration of temporal driving behavior and its deployment by self-driving cars

Diploma thesis

by

Christos Markos

Head Supervisor	Second Supervisor
Dr. Dimitrios Katsaros	Dr. Aspassia Daskalopulu

Volos, 2017

Διπλωματική Εργασία για την απόκτηση του Διπλώματος του Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

……………………………

Χρήστος Μάρκος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Πανεπιστημίου Θεσσαλίας

Εγκρίθηκε από τη διμελή Εξεταστική Επιτροπή στις …/…/ 2017.

……………………………	……………………………
Δρ. Δημήτριος Κατσαρός	Δρ. Ασπασία Δασκαλοπούλου
Επιβλέπων καθηγητής	Δεύτερο μέλος επιτροπής

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisors Dr. Dimitrios Katsaros and Dr. Aspassia Daskalopulu, for their unlimited guidance and constant support throughout not only the undertaking of my diploma thesis, but also the rest of my undergraduate studies at the Department of Electrical and Computer Engineering of the University of Thessaly. I am also indebted to Mr. Athanasios Fevgas, for his assistance in overcoming several of this project's resource limitations. Last, but certainly not least, I would like to thank my family and friends, for their invaluable support, love, and encouragement during all these years that ultimately shaped who I am.

ΠΕΡΙΛΗΨΗ

Τα αυτοκινητιστικά δυστυχήματα ευθύνονται για την απώλεια περίπου 1,25 εκατομμύρια ζωών ετησίως σε όλο τον κόσμο. Οι επιζώντες επιβαρύνονται όχι μόνο με φυσικά και ψυχικά τραύματα, αλλά και με υψηλά ιατρικά και υλικά κόστη. Θανατηφόρα ή μη, τα αυτοκινητιστικά δυστυχήματα επιφέρουν τεράστιες αθροιστικές απώλειες στην κοινωνία. Η ενίσχυση της οδικής υποδομής και των οχημάτων ώστε να αυξάνουν την οδική ασφάλεια είτε παθητικά είτε ενεργά έχει συγκεντρώσει σημαντικό ερευνητικό ενδιαφέρον τις τελευταίες δεκαετίες. Τα σύγχρονα επιβατικά οχήματα είναι εφοδιασμένα με διάφορους αισθητήρες, οι οποίοι επιτρέπουν επικοινωνία μεταξύ οχημάτων, αλλά και μεταξύ οχημάτων και έξυπνης οδικής υποδομής. Αυτό επιτρέπει το σχηματισμό ad hoc δικτύων οχημάτων, επιτρέποντας ποικίλες εφαρμογές μέσω διάχυσης κρίσιμων παραμέτρων.

Ο σκοπός αυτής της έρευνας είναι να συμβάλει στις τρέχουσες έρευνες σχετικά με την οδική ασφάλεια, χρησιμοποιώντας πρόβλεψη τοποθεσίας σε ad hoc δίκτυα οχημάτων. Για αυτό το σκοπό, εξετάζουμε την επίδοση του LeZi-Update, ενός αλγορίθμου πρόβλεψης τοποθεσίας αρχικά σχεδιασμένου για κινητά δίκτυα, αυτή τη φορά δουλεύοντας σε δίκτυο οχημάτων όπου τα πρότυπα κινητικότητας διαφέρουν. Επιπλέον, επινοούμε δύο διαφορετικούς τρόπους να συνυπολογιστεί ο χρόνος της ημέρας κατά την πρόβλεψη, και εξετάζουμε το κατά πόσο επηρεάζεται η ακρίβεια της πρόβλεψης.

Τα πειράματά μας εκτελέστηκαν χρησιμοποιώντας πραγματικά δεδομένα οδήγησης, τα οποία περιλαμβάνουν χρονοσφραγίδες και συντεταγμένες GPS που λήφθηκαν από ταξί στη Ρώμη κατά τη διάρκεια ενός μήνα. Αρχικά, οι GPS συντεταγμένες ομαδοποιήθηκαν με βάση το όνομα οδού, χρησιμοποιώντας τη διαδικασία της αντίστροφης γεωκωδικοποίησης. Έπειτα, αυτές οι οδοί χωρίστηκαν σε τμήματα 30 μέτρων, και σε κάθε τμήμα ανατέθηκε ένα μοναδικό σύμβολο. Εν τέλει, ο αλγόριθμός μας αξιοποίησε αυτά τα σύμβολα σε συνδυασμό με τις παρεχόμενες χρονοσφραγίδες ώστε να πραγματοποιήσει επανειλημμένες προβλέψεις του επόμενου συμβόλου.

Σύμφωνα με τα αποτελέσματά μας, ο LeZi-Update διατηρεί υψηλή ακρίβεια πρόβλεψης σε ad hoc δίκτυα οχημάτων. Η ακρίβεια αυτή αυξάνεται με τη χρήση μεγαλύτερου μήκους συμφραζόμενων, φτάνοντας ποσοστά επιτυχίας μέχρι και 100%. Επιπλέον, η ώρα της ημέρας αποδεικνύεται πρακτικά ασήμαντος παράγοντας στην πρόβλεψη του επόμενου τμήματος δρόμου, και θα μπορούσε να αγνοηθεί εξ ολοκλήρου ώστε να μειωθεί η συνολική περιπλοκότητα της υλοποίησης. Θα θέλαμε να επεκτείνουμε την έρευνά μας όταν υπάρξει

διαθεσιμότητα πιο πλούσιων δεδομένων, τα οποία να περιέχουν περισσότερες παραμέτρους όπως ταχύτητα ή γωνία τιμονιού, ώστε αυτές να εξεταστούν ξεχωριστά ως προς την επίδρασή τους στην ακρίβεια πρόβλεψης. Επιπλέον, εφόσον έχουμε περισσότερους υπολογιστικούς πόρους στη διάθεσή μας, θα θέλαμε να χρησιμοποιήσουμε cross-validation, μια ακριβέστερη μέθοδο μέτρησης σφάλματος.

ABSTRACT

Every year, road traffic accidents claim nearly 1.25 million lives worldwide. Survivors are often burdened with not only disability and trauma, but also high medical and property damage costs. Fatal or non-fatal, these accidents befall society with immense cumulative loss. Enhancing road infrastructure and vehicles to either passively or actively increase road safety has gathered significant research focus over the last few decades. Modern passenger vehicles are equipped with various sensors that allow vehicle-to-vehicle and vehicle-to-infrastructure communication. This enables formation of vehicular ad hoc networks, in turn allowing for various applications through dissemination of numerous critical parameters.

The aim of this study is to contribute to current research on road traffic safety, using location prediction in vehicular ad hoc networks. To this end, we survey the performance of LeZi-Update, a location prediction algorithm originally designed for mobile networks, this time working with a vehicular network where mobility patterns and dynamics are different. We also devise two different ways to factor the time of day into prediction, and investigate the extent to which prediction accuracy is affected.

Our experiments were conducted using real driving data, including timestamps and GPS coordinates obtained from taxis in Rome over the span of one month. First, the GPS coordinates were clustered by street name using the process known as reverse geocoding. Then, these streets were divided into 30-meter segments, and each segment was assigned a unique symbol. Finally, our algorithm leveraged these symbols along with the provided timestamps to repeatedly perform next-symbol predictions.

According to our results, LeZi-Update maintains high prediction performance in vehicular ad hoc networks. Prediction accuracy increases when using longer contexts, reaching levels as high as 100% in our tests. Furthermore, time of day is proven to be an insignificant factor in predicting the next street segment, and could be entirely ignored to reduce overall implementation complexity. Further research could be conducted upon availability of richer datasets, containing more parameters such as velocity, gear, or steering angle, to be evaluated individually regarding their effect in prediction accuracy. If more hardware resources are available, we would also like to use cross-validation as a more accurate error-measuring method.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

1 INTRODUCTION

According to a recent report published by the World Health Organization, approximately 1.25 million deaths occur every year as a result of road traffic accidents [1]. In fact, this was found to be the leading cause of death among people under 30 years of age. Furthermore, the National Safety Council estimates that 2016 may have been the deadliest year on U.S. roads since 2007 [2]. It is predicted that, unless preventive action is taken, such incidents will become the seventh leading cause of death worldwide in the next decade [1].

The effects of road traffic accidents can be observed on both an individual and a societal level. Regarding the former, people may make a full recovery from their injuries following an accident; often, the damage they undergo may lead to permanent disability or emotional trauma. Another aspect of the consequences of motor vehicle crashes is the economic cost, usually related to medical care. As far as society is concerned, there are also substantial financial losses, such as medical and emergency service costs, legal and court costs, as well as insurance and property damage costs [3].

A variety of elements have been found to cause road traffic accidents. Human factors such as sub-optimal driving, failure to check blind spots, intoxication, fatigue, distraction by scenery, mobile devices or advertising, have been attributed to the majority of crashes [4]. Defective road design, such as badly designed intersections and reduced visibility, has also been deemed a contributing factor to a substantial number of collisions. Proper vehicle design and maintenance are crucial as well; newer vehicles offer better protection in case of a crash, while well-maintained vehicles mean improved handling, and therefore a better chance of avoiding an accident.

In general, one could discern three different approaches to mitigating road traffic accidents; modifying human behavior, enhancing vehicles to assist human drivers or prevent accidents themselves, and upgrading existing road infrastructure to either passively or actively aid in maintaining safety. However, changing behavior entails education, information, as well as enforcement, largely depending on government initiatives and usually without lasting effects.

In the early years of automotive safety research, mitigation of collision or crash impact was the main focus. Some works [5] argued in favor of seat belts as the single most effective mechanism in reducing physical damage, while some [6] suggested that their use could ultimately lead to an increase in road casualties, as people tend to be less careful when feeling more protected. Other works stressed the importance of air bags [7], padded dashboards, or an enhanced roof structure [8], which could reduce injury caused by rollover and side collisions. However, the next generation of automotive safety solutions are engineered to eliminate the possibility of collision altogether.

In this work, we tackle the problem of road traffic safety using vehicular location prediction. We believe that dissemination of such information through road infrastructure or other vehicles could help prevent traffic accidents. To this end, we use real driving data to survey the performance of a location prediction algorithm originally designed for mobile networks, this time working with a vehicular network. We also investigate how the time of day might influence location prediction by devising two different ways to incorporate it into the prediction process. Our aim is to show that this model could positively contribute to current research on autonomous vehicles in conjunction with road traffic safety.

Chapters 2, 3, and 4 provide the reader with the necessary background information on self-driving cars, wireless networks and discrete sequence prediction. Chapter 5 comprises a review of the literature on location prediction, while Chapter 6 offers a detailed description of the methods employed to process our data and obtain our results. Our findings are presented and thoroughly discussed in Chapter 7. Finally, our research limitations, implications, and suggestions are documented in Chapter 8, while our references are listed in Chapter 9.

2 SELF-DRIVING CARS

2.1 Introduction

A self-driving or autonomous car is a vehicle that is designed to travel between destinations without a human driver. To qualify as fully autonomous, a vehicle must be able to navigate to a predetermined destination without human intervention, over roads that it may not have been trained to use. While such cars are not yet commercially available, highly automated cars with self-driving features are already on the road. Both types of vehicle rely on the input of various on-board sensors to manage navigation, safety, and handling. In the case of partially automated vehicles, this input is processed by the on-board driver assistance systems, in order to facilitate the driving experience. These systems are more thoroughly presented in section 2.2, while a clearer distinction between vehicular automation levels is made in section 2.3.

While fully automated vehicles are expected to become commercially available within the next few years, they have in fact been envisioned since the early 20th century. The world's first driverless car was presented by engineer Francis P. Houdina in 1925. Even though the driver's seat was unoccupied, this car was actually radio-controlled by a second car following close behind. During a demonstration in Manhattan, however, the vehicle's control system failed to establish communication, eventually causing it to crash into another vehicle. The GM Firebird II, released in 1956, featured a sophisticated guidance system which was meant to be used with what was imagined as the future of roadway infrastructures. Electronic roadways were expected to have embedded detector circuits, which would guide autonomous vehicles based on the determined location and velocity of other vehicles. Based on this idea, a driverless car that could move at 130 km/h on a specific trajectory was demonstrated in 1971, with research claiming that a switch to automation could prevent nearly 40% of road traffic accidents. Unfortunately, electronic roadways never became widespread enough for this type of driverless car to be publicly adopted.

A great deal of progress towards truly autonomous vehicles has been made since the 1980s, notably with Carnegie Mellon University's Navlab and ALV projects in 1984 and Mercedes-Benz and Bundeswehr University Munich's EUREKA Prometheus Project in 1987. Numerous automaker companies and research organizations have since presented their own working prototypes. Interest in autonomous vehicles has increased to the point that even companies from seemingly unrelated industries have invested in this domain. Google's self-driving car project, started in 2009, has traveled over two million miles without a driver in the US. As of recently, the project has been assigned to Google's subsidiary, Waymo, as it is estimated to be ready for production in the near future.

2.2 Advanced Driver Assistance Systems

Advanced Driver Assistance Systems (ADAS) are systems that prepare vehicles for unsafe road conditions and alert drivers to dangerous maneuvers. ADAS are able to identify pedestrians [9], road signs, or discern a police car from a taxi or regular passenger car. These systems are available in various forms; their features are built-in or can later be installed as an add-on package. Some of their features are presented in Table 1.

Table 1 – ADAS Features

Feature	Description
Adaptive cruise control	Automatic control of speed and distance in relation to the proceeding vehicle in the same lane.
Blind spot monitoring	Human driver warning regarding other vehicles located to the vehicle's side and rear.
Lane departure warning	Human driver warning in case of deviation from lane boundaries.
Automatic parking	Parallel, perpendicular, or angle parking by seizing control of the steering angle and speed.
Drowsiness monitoring	Prevention of accidents caused by human driver fatigue. Detection using steering input, lane monitoring camera, eye/face monitoring, or body sensors.
Tire pressure monitoring	Air pressure information reporting, usually via gauge or low-pressure warning light.
Night vision	Improvement of human driver's perception in darkness or poor weather, through the use of infrared cameras and active illumination techniques.
Intelligent speed adaptation	Human driver warning or automatic speed reduction in case of speed limit violation.

In general, ADAS features rely on a combination of inputs generated by various installed sensors, as shown in Figure 1. The latter provide ADAS with the necessary awareness of their surroundings, offering advantages such as high resolution, identification and classification of objects, and the ability to obtain measurements at any time during the day. However, the performance of these sensors is affected by the amount of available light, as well as weather conditions; for instance, heat may cause image degradation due to noise [10]. The most commonly used sensors are:

- *Radar*. Emits radio waves that bounce of an object, determining its distance from the sensor, as well as direction and speed.
- *Ultrasonic*. Emits ultrasonic sound waves that bounce off an object, determining its distance from the sensor.
- *Light Detection and Ranging (LIDAR)*. Emits light in the form of a pulsed laser that bounces off an object, determining its distance from the sensor.
- *Time of Flight*. Using a camera, this method measures the time it takes for an emission of infrared light to bounce off an object and return to the sensor, determining the distance to that object.
- *Structured Light*. A known pattern is projected on to an object, and the deformation of the former is captured by a camera and analyzed to determine the distance to that object.

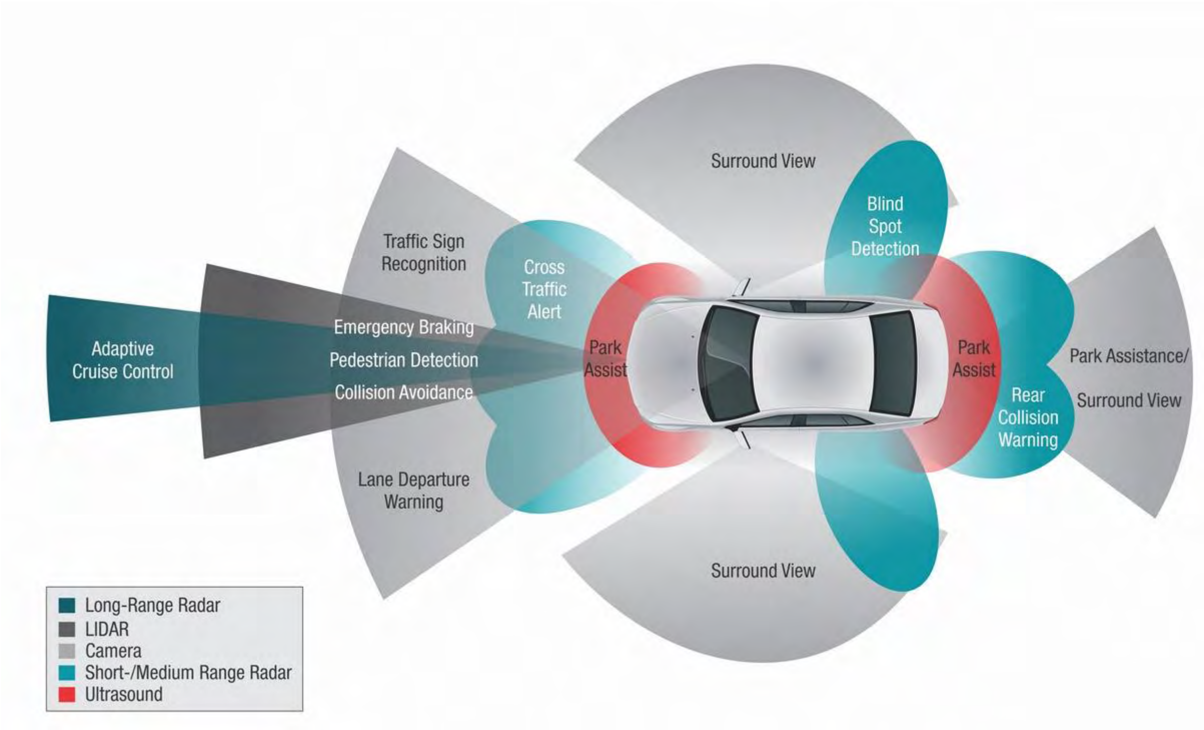

Figure 1. ADAS features and their enabling sensors. [11]

Even though ADAS applications are still at a relatively early stage, manufacturers and their suppliers estimate that they could gradually become the main feature distinguishing automotive brands, and most importantly, one of their biggest revenue sources. Moreover, it is the same technologies that enable today's ADAS that could also be used to create fully autonomous vehicles, which are now a major focus of research and development. Hence, any ADAS technology that gains early support could have an advantage when self-driving cars reach the market.

2.3 Classification

According to SAE International's J3016 standard [12], driving automation can be classified into six levels, ranging from no automation (level 0) to full automation (level 5), as shown in Table 2.

Table 2 – SAE Levels

SAE Level	Name	Description
0	No Automation	The dynamic driving task is entirely performed by the human driver, even in the presence of ADAS.
1	Driver Assistance	Steering, acceleration/deceleration are controlled by both the human driver & ADAS, but ADAS are expected to execute the remaining aspects of the dynamic driving task.
2	Partial Automation	Steering, acceleration/deceleration are controlled by ADAS, while the human driver is expected to execute the remaining aspects of the dynamic driving task.
3	Conditional Automation	ADAS perform all aspects of the dynamic driving task, but the human driver is expected to intervene upon request.
4	High Automation	ADAS perform all aspects of the dynamic driving task, even if the human driver fails to intervene upon request.
5	Full Automation	The dynamic driving task is entirely performed by ADAS.

The dynamic driving task is defined as the set of operational (steering, accelerating, decelerating, vehicle and road monitoring) and tactical (deciding when to change lanes, make a turn, use signals, etc.) aspects of driving, excluding all strategic aspects (choosing destinations and routes). In the first three levels, the driving environment is monitored by the human driver, whereas in the remaining levels, this is done by ADAS. The importance of the SAE levels is that they serve as general guidelines in determining how technologically advanced a vehicle is, making distinctions that could matter in cases like car insurance, which is expected to undergo major changes when self-driving cars become the norm [13].

2.4 Benefits

Perhaps the most anticipated benefit of self-driving cars is the potential decline of road traffic accidents caused by human driver error. It is estimated that widespread use of autonomous vehicles could eliminate as much as 90% of all road traffic accidents in the U.S., saving thousands of lives and preventing billions of dollars in annual damage and health costs [14]. Moreover, self-driving cars could allow for a smoother driving experience with higher speed limits, increased roadway capacity and minimal traffic congestion, due to decreased need for safety gaps between vehicles. Car sharing could reduce the need for parking space in urban areas, allowing for more parks, public spaces, or housing. This is expected to make dense cities much more efficient and habitable.

Self-driving cars could effectively reduce stress caused by driving. It has been shown that more driving may lead to decreased tolerance for others, as well as lower overall productivity. Furthermore, daily car commuting has been found to raise blood sugar, cholesterol, depression risk, and to negatively affect fitness and sleep quality. According to [15], traffic jams, road constructions, and long driving distances are key stress factors; in particular, the elements of unpredictability and loss of control. By either partially or fully assuming control of the vehicle, self-driving cars could remove a substantial amount of anxiety from the human driver.

Self-driving cars could increase mobility for those who are hindered by health-related issues. According to the World Health Organization [16], over a billion people are estimated to be living with disability; that is, about 15% of the world's population. People with disabilities generally have poorer health, lower academic achievements, fewer economic opportunities and higher rates of poverty than people without disabilities. This is mostly attributed to the lack of services available to them and the many obstacles they face in their daily lives. Therefore, self-driving cars could be detrimental in improving those people's quality of life.

Minimization of congestion, stop-and-go, and idling could lead to a drastically lower environmental footprint [17]. Environmental concerns have continuously grown in the past few decades, and numerous densely populated cities have increasingly invested towards reducing pollution caused by the burning of fossil fuels. Beijing, largely considered the most congested city in the world, has recently implemented a driving restriction policy to address traffic congestion and air pollution. However, despite leading to a significant drop of nearly 30% in the daily average concentration of particulate matter, these measures have been met with high levels of non-compliance and accusations of being unjust. It is clear that any satisfactory solution to the problem of traffic-induced pollution would require a non-restrictive basis, and self-driving cars may prove to be a step in the right direction.

2.5 Limitations

The safety performance of self-driving cars could, in essence, be perfect. However, they would have to compensate for additional, non-driver factors. In [18], pedestrians were deemed responsible for 80% of pedestrian crashes at intersections. A person suddenly stepping in front of a self-driving car might inevitably cause a crash, despite the vehicle responding optimally, due to the car's stopping distance which is subject to braking restrictions. Hence, an expectation of zero fatalities would probably not be realistic. Furthermore, it is unlikely that any significant increase in road traffic safety could be achieved until widespread adoption of self-driving cars. In fact, it might even deteriorate during the transition period when self-driving cars and regular cars would coexist, at least as far the latter are concerned.

As previously stated, a significant percentage of crashes is attributed to vehicular failures. It would be reasonable to expect that self-driving cars could mean mitigation for some of them. For instance, lighting failures might not affect safety in terms of vehicle control, as self-driving cars could depend on multiple inputs. Still, the same might not hold true for brake or tire faults; in fact, there would be no reason to expect otherwise, given the additional complexity of the vehicle. Similarly, self-driving cars will probably be able to detect and avoid road anomalies and debris. However, they might have difficulty handling flooded roads or other environmental factors. Weather conditions such as fog, snow, or heavy rain might still hinder the provision of sufficient information for safe travel.

3 WIRELESS NETWORKS

3.1 Introduction

Wireless networking is one of the fastest growing technologies supporting pervasive computing. Technological advances during the last two decades have produced mobile hosts and wireless networking, and therefore, a need for wireless mobile units to communicate with each other. Wireless mobile networks have traditionally been based on the concept of cells and relied on effective infrastructure support. In infrastructure mode, nodes communicate through a base station that acts as a wireless access point hub. Access points are typically fixed, have a wired or fiber network connection, and provide service to client nodes within range. Wi-Fi networks are commonly deployed in infrastructure mode, as it holds the advantage of scalability, improved reach, and centralized security management. Other common examples of infrastructure-based wireless networks are GSM and WLAN.

In recent years, the extensive availability of wireless communication and handheld devices has resulted in research on self-organizing networks that do not need a pre-established infrastructure. In ad hoc networks, as they are known, nodes may only communicate peer to peer, with no requirement for central access points. Wireless devices can discover others in a certain radius and communicate with them directly; therefore, ad hoc mode can be used to establish a network without wireless infrastructure. Ad hoc networks can be classified into two categories: static and mobile. In static ad hoc networks, a node's position may not be altered once it becomes part of the network, while in mobile ad hoc networks (MANETs), nodes are free to move arbitrarily.

3.2 Vehicular ad hoc networks (VANETs)

Vehicular ad hoc networks (VANETs) are a variation of MANETs in which the nodes are vehicular. They work with little or no permanent infrastructure and are characterized by high mobility, fixed road networks, predictable speed and traffic patterns, and minimal power or storage constraints. The primary requirements in VANETs are reliability and fast dissemination, as opposed to other communication systems where high message throughput is the main concern [19]. VANETs in the US will employ the "Wireless Access for Vehicular

Environments" (WAVE) standard, which will enable secure, high speed (up to 27 MB/s), short range (up to 1000 meters), and low latency wireless communication. WAVE uses a region of the 5.9 GHz band, and is based on the IEEE 802.11p standard.

VANETs rely on real-time communication among vehicles, pedestrians, and roadside sensors located along transportation systems, using advances in wireless communications, computing, and vehicular technologies. For this communication to be possible, On-Board Units (OBUs) and Roadside Units (RSUs) are installed in vehicles and roads respectively. Depending on the communication mode, data is generally exchanged either between OBUs or between OBUs and RSUs. The former case is quite different than what happens in MANETs, where a mobile station may only communicate with another through a base station. This direct communication reduces message latency, which is essential for safety applications such as collision avoidance. If a vehicle cannot communicate directly with an RSU, it can relay its data to other vehicles until the data finally reaches that RSU.

In principle, VANETs could help prevent accidents, facilitate eco-friendly driving, and provide better and more accurate real-time traffic information. However, concerns about security, liability, and privacy might decelerate progress toward large-scale implementation and deployment. Self-driving vehicles could help overcome these obstacles and motivate the widespread development and implementation of VANETs.

3.3 VANET communication modes

Until recently, vehicular communication primarily consisted of exchanges between various electronic control units (ECUs) and their corresponding sensors distributed over a vehicle. Communication would usually not reach the outside world, with the exception of some vehicle-to-device communication interfaces such as workshop fault diagnosis, ECU firmware updates, and mobile phone integration to allow hands-free calling. However, manufacturers are already equipping their vehicles with the required technology to communicate with other vehicles, surrounding infrastructure, or even pedestrians. These communication modes, along with some applications being developed, are introduced in the following subsections.

3.3.1 Vehicle-to-vehicle (V2V)

In vehicle-to-vehicle (V2V), messages are transmitted between nearby vehicles. Communication may be achieved through omnidirectional radio signals, either directly (one-hop) or through intermediary vehicles (multi-hop). V2V devices use dedicated short-range communications to broadcast data such as the vehicle's location, direction and velocity to nearby vehicles, up to ten times per second. This allows vehicles to be aware of each other, in turn identifying potential hazards and alerting drivers to their presence. V2V could be used to issue warnings such as:

- *Forward Collision Warning.* Issued when approaching a decelerating or stopped vehicle.
- *Emergency Electronic Brake Light Warning.* Issued when approaching a vehicle stopped in roadway, but not visible due to obstructions.
- *Blind Spot Warning.* Issued when beginning a dangerous lane departure that could place the vehicle on the travel lane of another vehicle following the same direction. Can also detect vehicles not yet in the blind spot.
- *Do Not Pass Warning.* Issued when beginning a dangerous lane departure that could place the vehicle on the travel lane of another vehicle following the opposite direction. Can also detect vehicles not yet in the blind spot.
- *Blind Intersection Warning.* Issued when crossing paths at a blind intersection, or an intersection without a traffic signal.

Even though there have been few real-world V2V trials to date, their results have been encouraging. In 2011, BMW equipped a car and a motorcycle with V2V safety technology, with a view to improving the safety of left turns [20]. This study was motivated by the large number of cars involved in such crashes with motorcycles. A camera and a laser scanner were installed on the front of both vehicles, to detect oncoming traffic and prevent cars traveling less than 10 km/h from moving in front of oncoming motorcycles or other vehicles. In case of the driver continuing to turn in front of the oncoming vehicle, both automatic braking and visual and audio warnings were activated. If the oncoming vehicle was a motorcycle, its headlights would first adjust to increase its visibility, and then it would sound its horn if the car still proceeded to enter the intersection. In this case, the safety system would finally engage the car's brakes.

3.3.2 Vehicle-to-infrastructure (V2I)

In vehicle-to-infrastructure (V2I), messages are exchanged between vehicles and roadside units installed on nearby arterial road intersections or highway on-ramps. V2I communications are intended primarily to circumvent crash scenarios that V2V cannot address, while also allowing for a wide range of mobility and environmental benefits. An illustration of how these two communication modes work in synergy is presented in Figure 2. The following are some of the V2I safety applications currently under consideration:

- *Red Light Violation Warning.* This technology will leverage the vehicle's speed and distance from an intersection to alert drivers about possible violations of upcoming red lights.
- *Curve Speed Warning.* This technology will inform the driver to slow down if the current speed is unsafe for traveling through an upcoming road curve.
- *Stop Sign Gap Assist.* This technology will detect vehicle gaps at STOP-sign-controlled intersections and alert drivers if it is unsafe to proceed.
- *Reduced Speed Zone Warning.* This technology will warn drivers in work zones to reduce speed, change lanes, or prepare to stop.
- *Spot Weather Information Warning.* This technology will provide in-vehicle alerts to drivers about real-time weather events, leveraging information from RSU connections with weather data collection services.
- *Railroad Crossing Violation Warning.* This technology will alert drivers at controlled railroad crossings when it is unsafe to cross the railroad tracks, using RSE connections with existing train detection equipment.
- *Oversize Vehicle Warning.* This technology will warn drivers of oversized vehicles to pick an alternate route or stop, using information from RSE connections to infrastructure installed at bridges or tunnels.

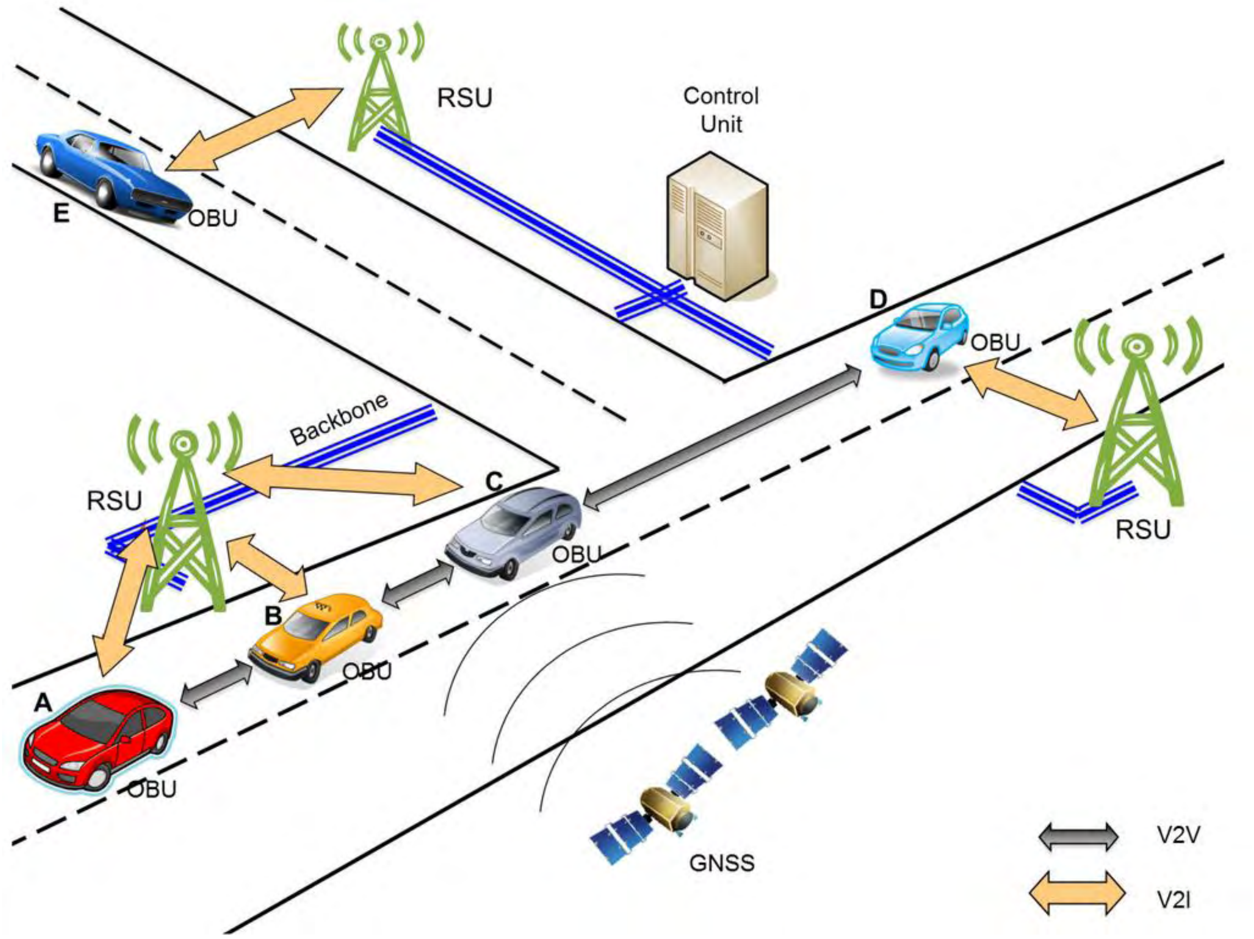

Figure 2. Combined V2V and V2I communication. [21]

In December 2016, Audi successfully launched the first V2I technology in the U.S., which allows cars to receive real-time signal information from the advanced traffic management system that monitors traffic lights via the on-board 4G LTE data connection [22]. The first available feature informs the driver about the time remaining until the signal changes to green, effectively reducing stress. This technology could be used in the future by smart cities to gain a better understanding of traffic patterns and possibly adjust traffic signal performance to minimize congestion and improve traffic flow.

3.3.3 Vehicle-to-pedestrian (V2P)

In vehicle-to-pedestrian (V2P), messages are transmitted between vehicles and pedestrians who send and receive messages using their mobile phones or other wireless devices. This will benefit a broad set of vulnerable road users, including people walking, children being pushed in strollers, people using wheelchairs or other mobility devices, passengers entering or exiting buses and trains, and bicyclists. V2P is expected to reduce road accidents by alerting vehicles of nearby pedestrians crossing the road, and vice versa. Some of the V2P applications currently in development by the US Department of Transportation [23] are:

- *Mobile Accessible Pedestrian Signal System.* This application allows for an automated call from the mobile device of a vision-impaired pedestrian to the traffic signal. Moreover, drivers attempting to make a turn are alerted to the presence of a pedestrian at the crosswalk.
- *Pedestrian in Signalized Crosswalk Warning (Transit).* This application warns bus operators when there are pedestrians in the intended path of the bus, as long as the pedestrians are within the crosswalk limits of a signalized intersection.

As shown in [24], Wi-Fi-based V2P provides satisfactory results, on condition that transmission frequency is set to a value larger than 1 Hz, and that 10 meters of GPS error are considered. However, there are still some challenges to be overcome, such as the drastic reduction of communication range when the signal is blocked by the human body, as well as the need for highly secure, high-speed wireless communication, perhaps using the WAVE standard.

3.3.4 Vehicle-to-device (V2D)

In vehicle-to-device (V2D), messages are transmitted between a vehicle and any electronic device that may be connected to the vehicle itself, such as diagnostic or programming devices, mobile phones, or computers [25]. To accomplish this, the vehicle may be equipped with hardware interfaces, or wireless interfaces such as Bluetooth or WLAN. For instance, the driver of an electric car can use a smartphone app to check the battery capacity of his vehicle while it is being topped up at a recharging station. As another example, the ParkMe app allows a driver to exit the car and use his smartphone to have the vehicle park itself.

As far as hardware interfaces are concerned, V2D communication can be used to connect external test devices for vehicle fault diagnosis, or scanning the internal communication system. It is also used to perform ECU software upgrades, after-sale feature activation, or retrieval of collected data. On the other hand, wireless V2D communication is used to connect the driver's mobile electronic devices with the vehicle. This allows for hands-free calling, or the integration of mobile navigation systems and audio players into the vehicle's corresponding systems.

V2D could particularly help prevent accidents involving bicycles, motorcycles, and other such vehicles. According to the U.S. Census Bureau [26], the number of Americans commuting by bicycle increased by 60% over the last decade, making it the largest percentage increase of all commuting modes. V2D would allow cars to communicate with bicycles, alerting them to traffic or potential dangers ahead.

4 DISCRETE SEQUENCE PREDICTION

4.1 Introduction

Discrete sequence prediction is defined as the task of predicting the next element in a sequence of data over a finite alphabet, where all possible outputs are known beforehand. In practice, this is usually achieved by providing the predictor with a dataset, then using part of that dataset to train the predictor, and part to perform prediction and evaluate its accuracy. Since the output belongs to a finite alphabet, and given that prediction is guided by a training stage, discrete sequence prediction describes a classification problem in the machine learning domain of supervised learning.

Suppose an alphabet Σ with a finite number of symbols $s_1, s_2, \dots, s_n$, where n indicates the alphabet length. A predictor collects phrases of various lengths, each consisting of symbols in Σ. These phrases represent the knowledge of the predictor, which it uses to build a model that calculates the probability for every possible outcome, based on part of the past [27]. Implementations of discrete sequence prediction commonly involve application of Markov models. A Markov model is a stochastic model used to describe randomly changing systems, under the assumption that the future only depends on some portion of the past. The latter is known as the Markov property. Markov chains, the simplest type of Markov model, are presented in the following section.

4.2 Markov chains

A Markov chain is a stochastic process satisfying the Markov property, where the system is autonomous and its state is fully observable. In probability theory, stochastic processes are collections of random variables, each corresponding to different system parameters. The system is modeled with a random variable which changes over time. PageRank, an algorithm which ranks websites in Google's search engine results according to their estimated importance, is based on a Markov process.

A time series of a random variable is said to have serial dependence if the value at some time t_1 is statistically dependent on the value at another time t_2. The term "chain" denotes the traversal of a sequence of random variables, where serial dependence only applies to neighboring events, according to the Markov property. The system's state space may be countable or continuous, and time may be discrete or continuous. The changes of state of the system are called transitions, while the probabilities corresponding to these state changes are called transition probabilities. Overall, the process is characterized by a state space, a transition matrix describing the probabilities of all transitions, as well as an initial state.

Let $S = \{1, \dots, m\}$ denote the state space of the Markov process, for some positive $m \in \mathbb{Z}$. A Markov chain is defined by its transition probabilities p_{ij} as follows:

$$p_{ij} = P(X_{n+1} = j \mid X_n = i), \qquad i, j \in S.$$

Of course, transition probabilities p_{ij} are required to be non-negative and their sum must be 1:

$$\sum_{j=1}^{m} p_{ij} = 1, \qquad \forall i.$$

The transition matrix is merely a 2D matrix where the element at the i-th row and j-th column is the transition probability p_{ij}:

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix}.$$

A discrete-time random process models a system which is in a certain state at each step, with the state changing randomly between steps. The transition probabilities p_{ij} hold whenever the process is in state i, regardless of any past events or how the process has transitioned to this state. The Markov property states that, for any time index n, for any state $i, j \in S$, and for all possible state sequences $i_0, \dots, i_{n-1}$:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i) = p_{ij}$$

This shows that the conditional probability distribution for the system at the next step depends only on the current state of the system, and not additionally on the state of the system at previous steps. Given that the probabilities are not time-dependent, the above model is defined as a *stationary* Markov chain, while the history $i, i_{n-1}, \ldots$ is termed the *context* of the predictor. Since the system changes randomly, it is generally impossible to predict with certainty the state of a Markov chain at a given point in the future. However, the statistical properties of the system's future can be predicted. In many applications, it is these statistical properties that are important.

Markov models are commonly used to model randomly changing systems in various application areas, including telecommunications, physics, chemistry, medicine, music, and game theory. Board games such as backgammon, where moves are determined only by dice, represent Markov chains. The same, however, does not hold for card games such as blackjack, where cards signify memory. For instance, let us consider the probability for a certain event in the game. In dice games, the only thing that determines the next state is the current state, in conjunction with the next roll of the dice. On the other hand, in a game of cards such as blackjack, one may increase their chances of winning by tracking which cards have been played.

Text prediction systems are another case where Markov models are vastly preferred. In this context, the predictor tries to anticipate the next block of characters, such as individual letters, syllables, words, or sentences. Prediction generally relies on the previously encountered blocks. If the system makes a correct guess, the number of keystrokes required to form a sentence decreases, thereby enhancing the speed of communication. In addition, text prediction may improve the overall quality of composed messages by correcting spelling mistakes and reordering words or sentences.

4.3 LeZi-Update

The LeZi-Update framework is an online adaptive location management scheme originally proposed for MANETs by Bhattacharya and Das [28]. The system accumulates each mobile device's movement history in a digital search tree (trie), building a universal model that allows for prediction of future locations with high accuracy. Location areas are represented by alphabetic symbols, therefore describing users' movement history as a symbol string and achieving application universality across different networks. The mobile device essentially acts as an encoder, while the system acts as a decoder. The symbols are processed in chunks, and the sequence of symbols that have occurred since the last update is finally reported in an encoded form, similarly to the dictionary-based LZ78 compression algorithm.

A new dictionary entry may only be created through concatenation of a single symbol to an already existing phrase. For example, the movement history “aaababbbbbaabccddcbaaaa…” will be parsed as “a, aa, b, ab, bb, bba, abc, c, d, dc, ba, aaa, …”, with commas indicating location updates on behalf of the encoder. Apart from storing the dictionary, the trie also holds the frequency of occurrence for each symbol, incrementing it for every suffix of each decoded phrase. The resulting trie following the above sequence of symbols is depicted in Figure 3, with corresponding frequencies shown in parentheses.

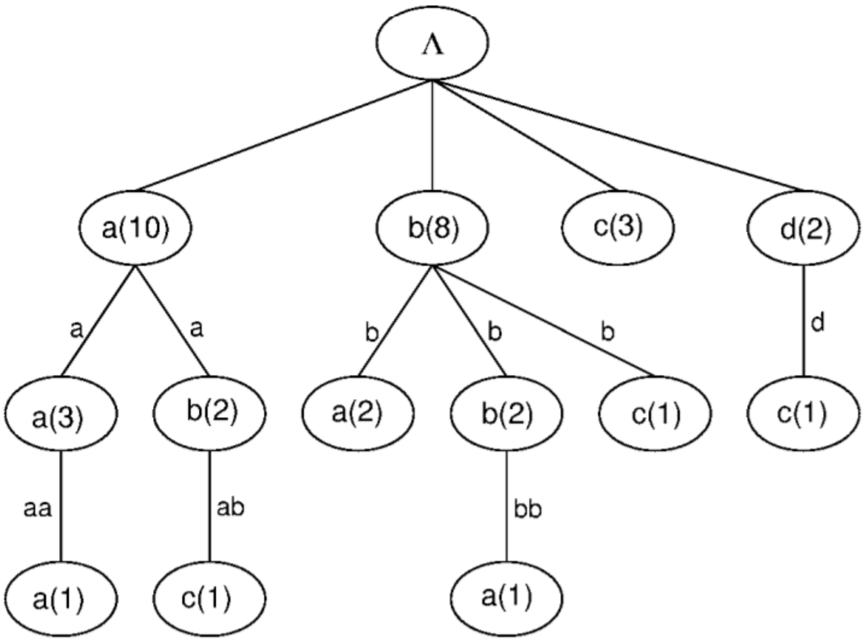

Figure 3. Trie for the sequence “aaababbbbbaabccddcbaaaa…”. [28]

The probability assignments are performed using the *exclusion* technique, often encountered in text compression schemes of the *prediction by partial match* family. Let “aaa” be the last update message received. The contexts that can be used for prediction are all suffixes of this phrase, excluding itself; that is, “aa” (order-2), “a” (order-1), and “Λ” (order-0). A list of all paths and their frequencies with respect to these orders are presented in Table 3.

Table 3 – Phrases and their frequencies at contexts "aa", "a" and "Λ"

aa (order-2)	a (order-1)	Λ (order-0)		
a | aa (1)	a | a (2)	a (5)	ba (2)	d (1)
Λ | aa (2)	aa | a (1)	aa (2)	bb (1)	dc (1)
	b | a (1)	ab (1)	bba (1)	Λ (1)
	bc | a (1)	abc (1)	bc (1)	
	Λ | a (5)	b (3)	c (3)	

Starting from the highest order, the phrase "a" occurs once out of three possible occurrences, while the rest produce null predictions. Therefore, it can be predicted with probability $\frac{1}{3}$ at context "aa" and fall back to the next order with probability $\frac{2}{3}$. At order 1, "a" occurs twice out of ten possible occurrences, hence it can be predicted with probability $\frac{1}{5}$ at context "a" and fall back to order 0 with probability $\frac{1}{2}$. Finally, "a" occurs five times out of twenty-three possible phrases, which means a probability of $\frac{5}{23}$. Therefore, the blended probability of occurrence for the phrase "a" is:

$$\frac{1}{3} + \frac{2}{3}\left\{\frac{1}{5} + \frac{1}{2}\left(\frac{5}{23}\right)\right\} = 0.5319.$$

Given that the phrase consists of one different symbol, the probability mass is entirely assigned to it. In the same manner, the probability of occurrence for phrase "bba" is calculated as:

$$\frac{1}{3} + \frac{2}{3}\left\{\frac{1}{5} + \frac{1}{2}\left(\frac{5}{23}\right)\right\} = 0.5319.$$

Since there is one 'a' and two 'b's in "bba", the individual probabilities of these symbols are computed as $\frac{1}{3}\times 0.0145 = 0.0048$ and $\frac{2}{3}\times 0.0145 = 0.0097$ respectively. This procedure of frequency blending and distribution among individual symbols is performed for all order-0 phrases. Finally, the symbols are paged by the system in decreasing order of occurrence probability.

5 LITERATURE REVIEW

In recent years, with the constantly increasing integration of ADAS in the driving task and the imminent commercialization of autonomous vehicles, the subject of location prediction in VANETs has been gathering significant scientific interest. Most studies construct models based on frequently observed patterns and describe trajectories as an ordered sequence of locations, timestamps, weather conditions, and various other factors. Krumm [29] used a Markov chain to perform short-term route prediction, considering a context of up to ten street segments, finally reaching approximately 90% prediction accuracy. Simmons et al. [30] used a hidden Markov model trained with 46 trips, predicting destinations based on knowledge of the road network. Even though in their work prediction accuracy is estimated to be around 98%, it should be noted that in 95% of all cases, the next street segment is only connected to the current one. While this could work adequately with highway networks, it might be somewhat unrealistic for big cities, where road intersections are more frequent.

Not all studies have approached the issue of location prediction using Markov models. Karbassi and Barth [31] proposed a car-sharing application that works when the driving start and end points are specified. However, such methods have been found to not perform well when location data is temporally sparse. Another practical problem is the fact that the driver cannot always be expected to specify the exact destination beforehand. Ye et. al. [32] used graph theory, representing roads as nodes and their connections as weighted edges, whose weights signify the number of times a vehicle has traversed the corresponding street segment. This allowed them to leverage the concept of point centrality, thereby assigning measurable importance to street segments, and achieving about 80% prediction accuracy.

Other works have used data mining algorithms to perform location prediction for a single moving object, by either considering all other objects in a database [33], or solely relying on the movement history of the object itself [34]. In [33], Morzy constructed a probabilistic model of all possible object locations by dividing the movement area into a grid, transforming movement paths into trajectories with respect to the grid, determining frequent trajectories, and extracting movement rules which would finally be matched with the history of an object. However, while achieving satisfactory results, this work did not involve either temporal or spatial information, which would have rendered decisions more informed.

6 METHODS

6.1 Data analysis

To test our hypothesis, we used a dataset containing mobility traces of 315 taxis in Rome, Italy [35], collected over 30 days in 2014. The dataset was provided in the form of a simple text file, comprising approximately 22 million lines of taxi IDs, timestamps, and GPS coordinates (latitude, longitude). The content of this file was then split into a new text file for every taxi, so that each could be processed separately. This dataset was chosen due to its considerable time span, the number of different trajectories, but most importantly, the high probability of trajectories displaying some level of overlap and repeatability. The last two characteristics are particularly significant in location prediction, as opposed to other types of prediction (i.e. steering angle prediction,), where longer trajectories from one point to another, usually on highways, might be preferred.

The next step of our analysis was to map each obtained geographic point to its corresponding street in Rome, a process known as reverse geocoding. Since our predictor's input is symbols, our primary aim was to be able to assign a discrete symbol to all points within a certain street segment. To this end, we employed OpenStreetMap (OSM), a free mapping service that can handle reverse geocoding requests, which were made through Nominatim. Nominatim is a search engine that specifically works with OSM, and was found to provide accurate results throughout our tests.

To the best of our knowledge, no mapping service provides reverse geocoding without limit or free of charge. Given that our project was not funded, and in order to have our millions of requests satisfied within a reasonable time frame, a local installation of Nominatim was made. This process involved downloading and compiling Nominatim, adding auxiliary data from Wikipedia, setting up a local database, and finally, importing and indexing Rome-specific OSM data. More setup details may be found in [36].

Following the local installation of Nominatim, a script was developed in Python to automate the handling of the reverse geocoding requests. This script parsed each of the 315 taxi files sequentially. For each line, the local database was queried using the extracted latitude and longitude, and the resulting street name was saved along with the previously parsed hour of occurrence and GPS coordinates. Following a line, any successive identical lines were ignored, as this was interpreted as a stationary vehicle. After execution of the script, each taxi file consisted of hours, street names, and GPS coordinates.

The previous step was necessary in order to be able to divide each street into 30-meter segments. This segment length was used to increase prediction accuracy. Had we assigned the entirety of each street with a single symbol, some accuracy would certainly have been lost; for example, a vehicle currently on a specific street would be more likely to make a turn at an intersecting street a few meters away, as opposed to a much farther one. The latter would intuitively become more probable as the vehicle approached that particular street, ultimately finding itself on the segment intersecting that street. On the other hand, public squares were treated as a single street segment, given the limited number of intersecting streets which are also in close proximity.

In turn, the Python script developed for the task of street segmentation parsed the taxi files produced by the previous script, collecting all encountered street names and grouping their corresponding GPS coordinates together. Without loss of generality, we chose integers to be our street segment symbols, due to the large expected number of street segments. We then devised an algorithm which, given both a set of points belonging to a particular street and the next available symbol, performs the segmentation as follows: as a preliminary step, the first point stored for the current street is set as the reference point P_{ref}. Then, for each point P_i, the algorithm calculates the distance between P_i and P_{ref}, and assigns the former to a temporary symbol as determined by our formula which is presented below. Since it relies on distance, however, our formula would inherently misclassify two diametrically opposite points by allocating them to the same symbol. Therefore, this assignment is only performed when the distance between P_i and the first point that has been allocated to the symbol in question is less than or equal to 30 meters. Any points that fail to satisfy this condition are examined in the next function call, and properly assigned to the symbol corresponding to the opposite street segment; there will be at most two function calls per examined street. A last step is performed in order for the final symbols to be allocated to street segments in increasing order. The Python code that describes the above algorithm is shown in Figure 4.

```python
def street_segment(gps_points, cur_symbol, f_out):
    segment_length = 30
    radius = segment_length / 2

    segments = [[]]            # Segments will either be on one or the other
                               # side of the reference point (p)

    p = gps_points[0]          # Save the reference point (p)
    gps_points.remove(p)
    segments[0].insert(0, cur_symbol)
    segments[0].insert(1, p)

    for point in gps_points:
        dist = haversine(p, point)
        symbol = cur_symbol + int(ceil(dist / radius)) / 2
        index = find_index(symbol, segments)

        if index >= 0:
            if haversine(point, segments[index][1]) > 30:
                continue       # Belongs to the other side, examine on next
                               # function call
        else:
            segments.append([])
            index = len(segments) - 1
            segments[index].insert(0, symbol)

        segments[index].insert(len(segments[index]), point)
        gps_points.remove(point)

    for segment in segments:  # Final symbol assignment
        segment.pop(0)
        for point in segment:
            f_out.write(str(point) + " " + str(cur_symbol) + "\n")
            f_out.flush()
        cur_symbol += 1

    return cur_symbol
```

Figure 4. Python code for our street segmentation algorithm.

Consider the following example. Let P_1, P_2, P_3, and P_4 be points on the same street, with their respective distances from P_{ref} having been calculated as 34, 16, 43, and 92 meters. Assuming that the next symbol to be assigned is 0, we have: $symbol_{P_1} = 0 + \frac{ceil\left(\frac{34}{15}\right)}{2} = 0 + 1 = 1$, where "ceil" represents the mathematical ceiling function, and the division remainder is discarded. Likewise, $symbol_{P_2} = 0$, $symbol_{P_3} = 1$, and $symbol_{P_4} = 3$. However, while the temporary symbol for both P_1 and P_3 was calculated as 1, P_3 is in fact diametrically opposite to P_1 with reference to P_{ref}. Therefore, the assignment for P_3 is delayed until the next function call. After performing the final symbol assignment step in the first function call, the points and their respective symbols would look as follows: $P_{ref} \rightarrow 0$, $P_1 \rightarrow 1$, $P_2 \rightarrow 0$, $P_4 \rightarrow 2$. Finally, after the second final call, we would obtain $P_3 \rightarrow 3$. A visual representation of this example is provided in Figure 5.

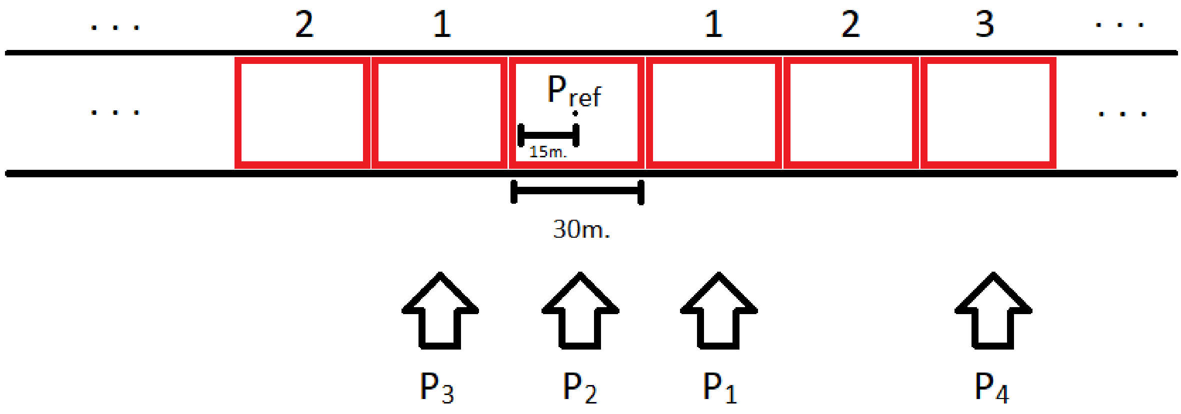

Figure 5. Illustration of our street segmentation algorithm running with 5 points.

Our algorithm employs the haversine formula [37] to determine the great-circle distance between two points on a sphere, given their GPS coordinates. The great-circle distance is defined as the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere, as opposed to a straight line through the sphere's interior. This formula has been proven to provide mathematically and computationally exact results, as opposed to the Pythagorean Theorem which, in our case, would result in minor computational error.

6.2 Prediction model

We survey the performance of the LeZi-Update prediction scheme presented in section 4.3, as well as an enhanced version where prediction is also guided by time of occurrence. LeZi-Update was chosen as the basis of our work because, given that VANETs are characterized by high mobility, there is need for fast message dissemination. The LZ78-based update scheme employed by the LeZi-Update algorithm could reduce bandwidth usage by delaying update messages sent by vehicles, as it did in the case of mobile networks. Moreover, given that VANETs have little storage constraints, smart road infrastructures would be able to accumulate large amounts of data over time, therefore achieving high prediction accuracy.

In our modification of the LeZi-Update algorithm, the encoder parses the input, which contains pairs of timestamps and symbols as described in the previous section, and forms update phrases as per the original scheme. However, these update phrases consist not only of a sequence of symbols, but also the hour of occurrence, which is appended at the end. Therefore, if the encoder encounters a different timestamp while collecting symbols from the input, then this will mark the start of the next update phrase. Each encoder, which in this case is one of 315 taxis, constructs its own trie; in contrast, the decoder builds a universal trie based on all messages received by the encoders.

In the original LeZi-Update algorithm, each trie node stores a single dictionary symbol along with its frequency of occurrence. In our modification, this frequency is represented as an array of 24 cells, each corresponding to an hour of a day. When the decoder receives an update message from an encoder, it adds the symbol phrase and its suffixes to the trie, each time incrementing the cell matching the hour of occurrence by 1. For illustration purposes, consider the trie shown in Figure 6, built after some sequence of updates. The arrows starting from the leftmost nodes "100" and "101" point to their respective frequency arrays. Only these two nodes' frequency arrays are displayed, for brevity.

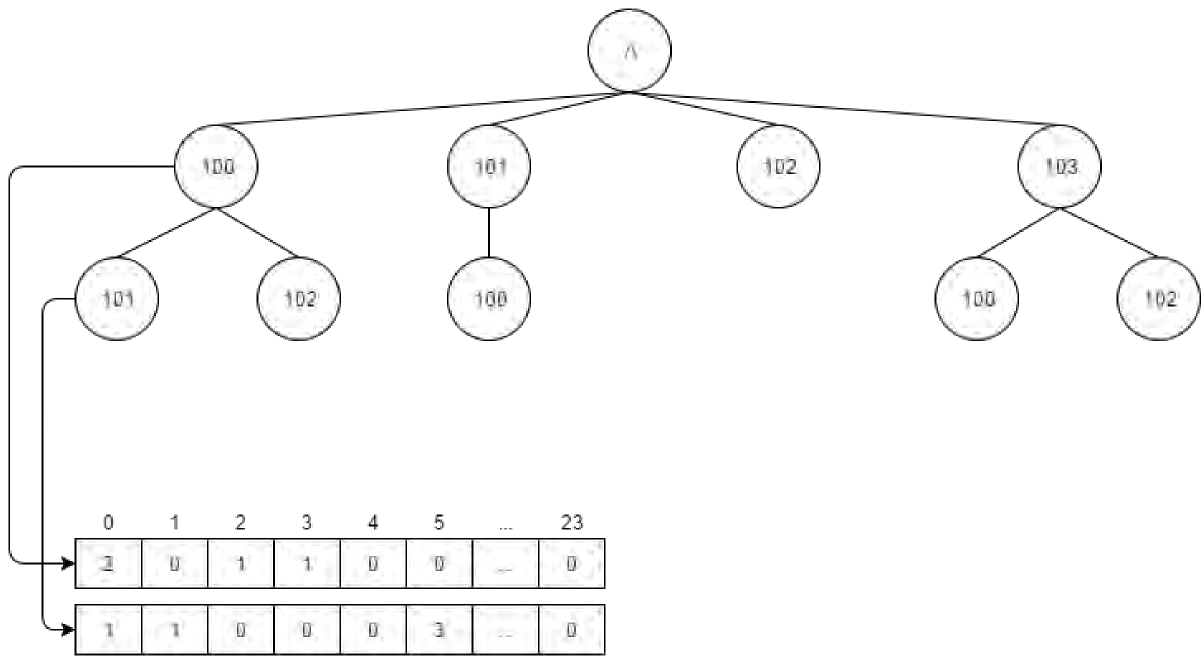

Figure 6. Example trie built by the decoder (before the update message).

Let the next received update message be "100, 101, 102, 2". In this phrase, "100", "101", and "102" are symbols, while "2" is the hour of occurrence. The decoder will then add the phrase and its suffixes to the universal trie as previously explained. The resulting trie is shown below, in Figure 7.

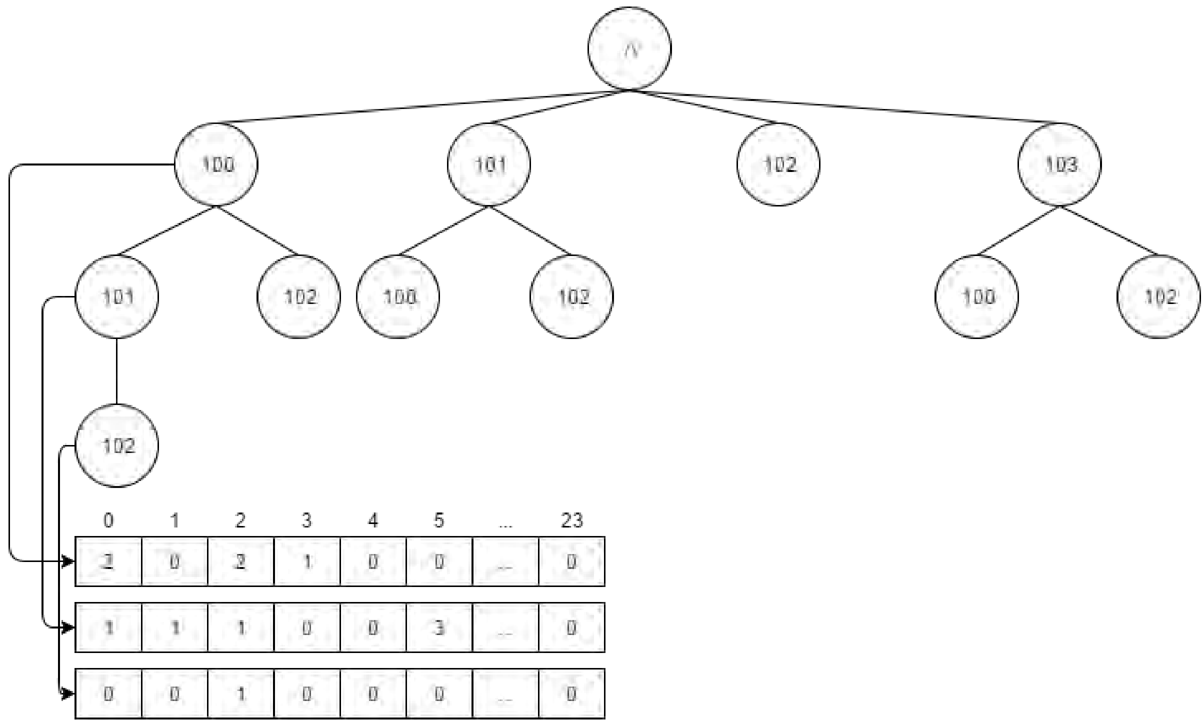

Figure 7. Example trie built by the decoder (after the update message).

The reason for storing the frequency of occurrence in such an array is that this allows us to perform *time blending* upon prediction request. Instead of merely using a universal frequency for each trie node, as is done in the original LeZi-Update algorithm, our expanded model considers the time of day when prediction is to be performed. This is achieved by "blending" the individual frequencies stored in the previously described array, for each node in the trie. Let h denote the cell matching the hour of prediction. After obtaining the entire value v from cell h, we iterate through the remaining cells as follows: in each iteration i, we add a fraction of the values at cells $h - i$ and $h + i$ to v, then reduce this fraction for the next iteration. These iterations are modular; when moving towards the last cell of the array, the next iteration will start from the first cell. Likewise, when moving towards the first cell of the array, the next iteration will examine the last cell.

While describing the process of time blending, we mentioned that only a fraction of the value in each cell was included in the final sum. Two different types of blending were devised with respect to how this fraction is reduced after each iteration; namely, the "linear" and “exponential” types. In the former, the fraction is reduced by $\frac{1}{12}$, while in the latter, it is divided by 2. The Java code for time blending, using the linear and exponential methods, is presented in Figure 8 and Figure 9 respectively. After calculating the blended frequency for each node, the phrase probabilities are calculated and then distributed to individual symbols as per the LeZi-Update algorithm. In turn, the symbol with the highest probability of occurrence is selected as our algorithm’s prediction, which is evaluated when the decoder receives the next update phrase.

```java
double f = frequency[hour];
double factor = 1.0 - 1.0 / 12;

for (int i = 1; i < 12; i++) {
    f += factor * frequency[((hour - i < 0) ? hour - i + 24 : hour - i)];
    f += factor * frequency[(hour + i) % 24];
    factor -= 1.0 / 12;
}

f += factor * frequency[((hour - 12 < 0) ? hour - 12 + 24 : hour -12)];
return f;
```

Figure 8. Java code for linear time blending.

```java
double f = frequency[hour];
double factor = 0.5;

for (int i = 1; i < 12; i++) {
    f += factor * frequency[((hour - i < 0) ? hour - i + 24 : hour - i)];
    f += factor * frequency[(hour + i) % 24];
    factor /= 2;
}

f += factor * frequency[((hour - 12 < 0) ? hour - 12 + 24 : hour -12)];
return f;
```

Figure 9. Java code for exponential time blending.

For accurate measurement of our prediction error, we employed the holdout set method. This technique has been found to provide highly accurate results when equipped with sufficient data, with the added benefit of simple implementation. The data is divided into two groups: one is used to train the model, while the second is used to measure the error of the resulting model. By holding out part of the dataset, we can directly measure the model’s true prediction error, which is how well it will predict new data. We chose 80% of our dataset as the training data, while the remaining 20% was the test data; this is a very common selection for the holdout set.

7 RESULTS

The cumulative prediction accuracy for contexts up to length 10 is presented in Figure 10. Upon first examination, one may observe that all time blending strategies have a very low hit rate for contexts of unit length. As street segments were being traversed for the first time by each of the 315 encoders, given that each builds its own trie, a large number of predictions (61%) were performed using unit-length contexts. Whenever such contexts were used, the result was essentially the most frequently encountered street segment (universally) at the time of prediction. Since the encoder could be arbitrarily far from that segment when prediction was performed, and considering that Rome has a total area of 1,285 km², the corresponding hit rates are considered reasonable.

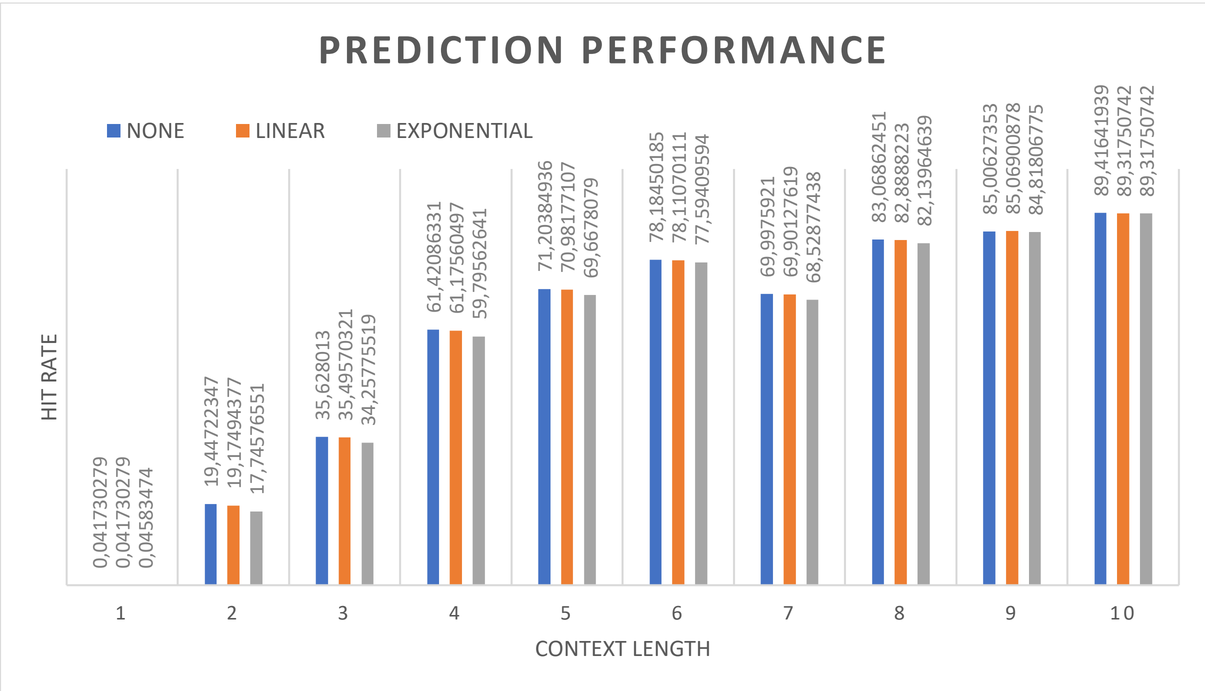

Figure 10. Prediction accuracy for contexts up to length 10.

As expected, higher orders provide increased prediction accuracy for all time blending modes. However, our results show that prediction accuracy is not very sensitive to the time of day. In fact, it seems to have a slightly negative impact on prediction, with the linear time blending mode performing better than its exponential counterpart in most cases. This performance difference seems sensible, given that the exponential mode is much stricter by blending smaller fractions of frequencies. It is possible that having more location data, as well as more features (i.e. day of the week, velocity, weather, etc.) would help trip patterns emerge more clearly, thereby refining prediction accuracy. For example, people usually commute to work or take their children to school during weekday mornings, but not on weekends. Unfortunately, currently available datasets containing such features were a bad fit for our purposes, as they usually comprised a car equipped with various sensors, performing a single trip from one point to another.

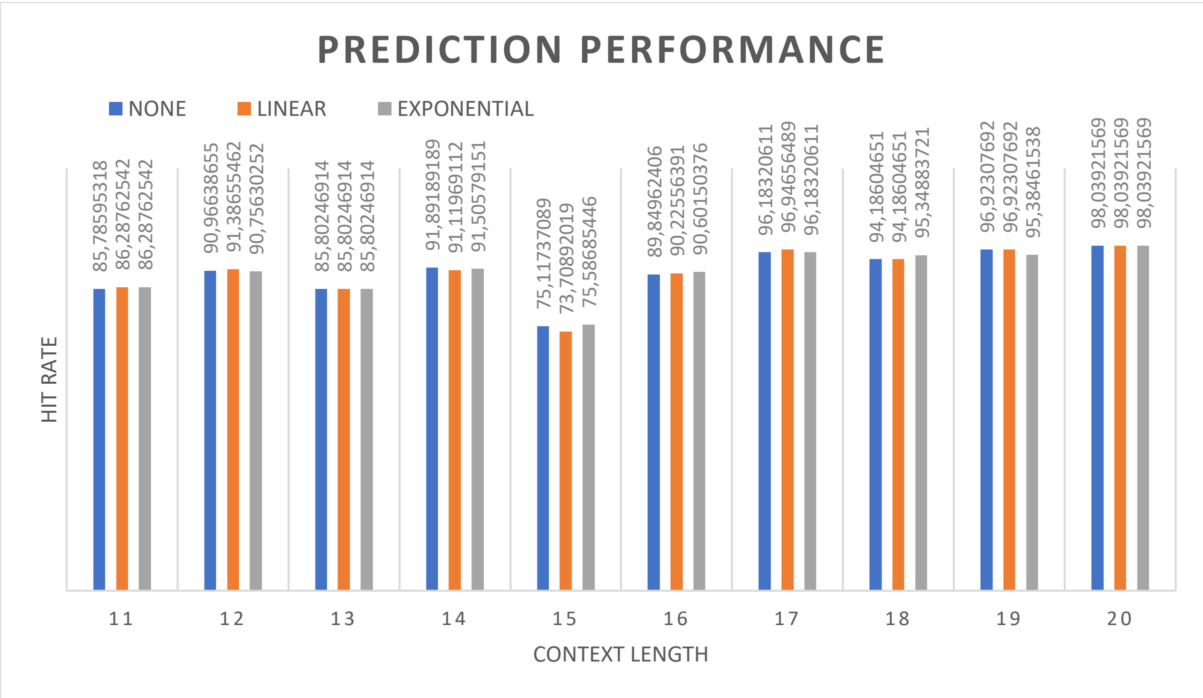

Figure 11. Prediction accuracy for contexts of length 11 to 20.

An illustration of the prediction results for contexts of length 11 to 20 is offered in Figure 11, while the results for the remaining context lengths are shown in Figure 12. We found that as the context length increases, the number of such contexts decreases steadily, as shown using a logarithmic plot in Figure 13. We chose a logarithmic plot because the data was skewed towards large values, making smaller ones indistinguishable. For instance, there were 877061 samples for context length 1, 430807 samples for context length 2, 76336 samples for context length 3, while there were only 7 samples for context length 27 and just 1 sample for context length 38. In fact, all contexts longer than 18 had fewer than 100 samples each. Given that longer contexts generally produce higher prediction accuracy, and considering that contexts with size larger than 18 had so few samples in comparison with the rest, it falls well within expectations that hit rates as high as 100% are achieved.

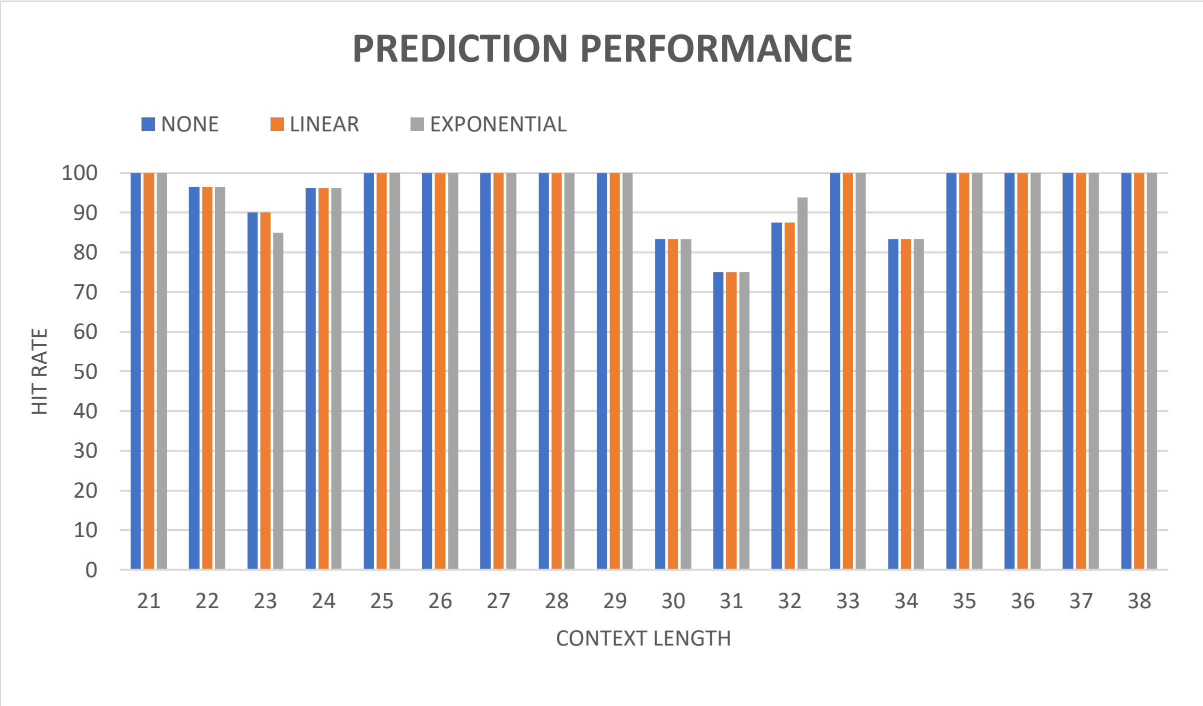

Figure 12. Prediction accuracy for contexts of length 21 to 38.

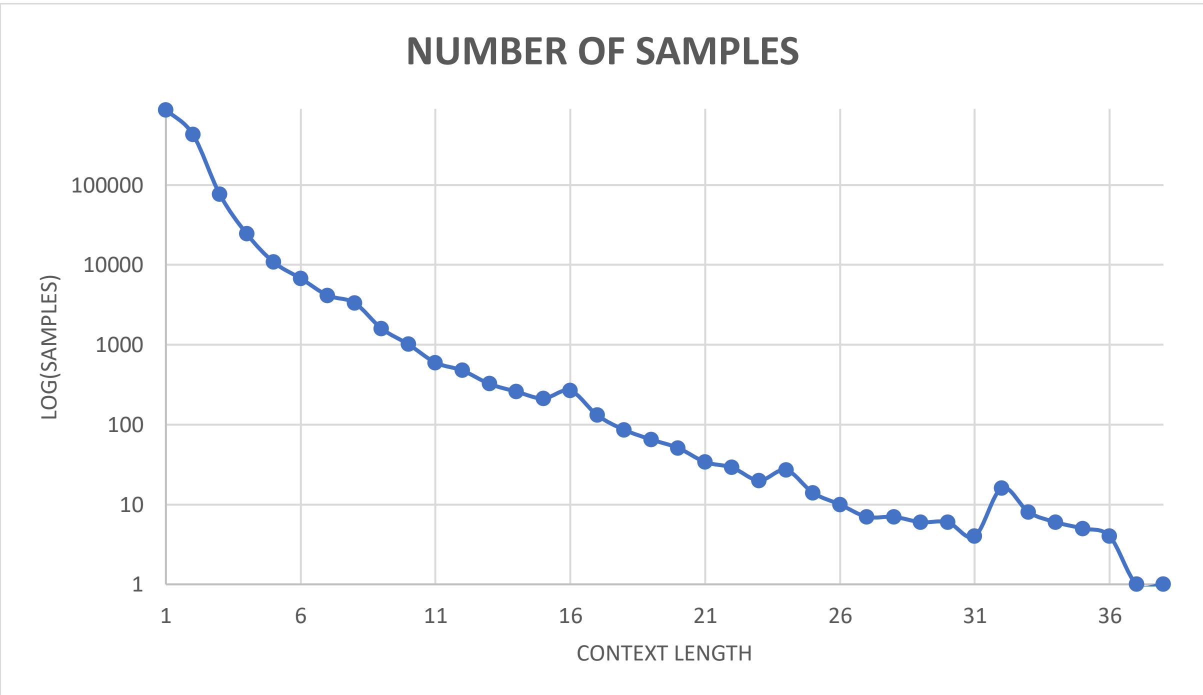

Figure 13. Logarithmic plot showing how the number of samples decreases over increasing context lengths.

8 CONCLUSION

In this work, our aim was to survey the prediction performance of LeZi-Update, an algorithm originally developed for MANETs, in a network where nodes are vehicular and therefore requirements and mobility dynamics are different. Moreover, in order to test whether the time of day affects prediction accuracy, we modified the algorithm to factor time into the prediction process in two ways. Our experiments indicate that the prediction accuracy of LeZi-Update in VANETs remains high, increasing when using longer contexts, and reaching levels as high as 100%. We also show that time of day is insignificant in predicting the next street segment, and could be ignored to reduce overall complexity.

We were presented with a number of challenges at different stages of our research. While we were able to overcome most of them, some proved to be insuperable due to either a lack of time or resources. Our predictor's accuracy, while very satisfactory, would have certainly benefited from higher digital map precision, perhaps also containing lane and intersection information. While the Google Maps Geocoding API offers optimal map precision, its free use is limited. Since our project was not funded, we could not afford to make more than 2,500 requests per day, and even if we could, use with our project would still not be permitted according to Google's terms of service. Other Geocoding APIs, such as Bing Maps REST Services and Yahoo! PlaceFinder, often proved inaccurate or lacking details during tests.

Currently available datasets containing more parameters (such as velocity, gear, steering angle, etc.) were far too large for the available hardware to handle, and did not contain sufficient location data or desirable trajectory overlap and repeatability. In our future work, we would like to blend more factors into the prediction process, and evaluate their contribution individually. We would also like to use cross-validation, a more accurate error-measuring method, which was unfortunately not employed due to being too computationally intensive for available hardware.

9 REFERENCES

[1] "Road traffic injuries", World Health Organization, 2017. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs358/en/. [Accessed: 28- May- 2017].

[2] National Safety Council Statistics Department, "NSC Motor Vehicle Fatality Estimates", 2017. Available: http://www.nsc.org/NewsDocuments/2017/12-month-estimates.pdf. [Accessed: 28- May- 2017].

[3] L. J. Blincoe, T. R. Miller, E. Zaloshnja and B. A. Lawrence, "The Economic and Societal Impact of Motor Vehicle Crashes, 2010," National Highway Traffic Safety Administration, Washington, DC, 2015. [Online]. Available: https://www.researchgate.net/publication/264862147_The_Economic_and_Societal_Impact_of_Motor_Vehicle_Crashes_2010. [Accessed: 28- May- 2017].

[4] H. Lum and J. Reagan, "Interactive Highway Safety Design Model: Accident Predictive Module," *Public Roads*, vol. 52, no. 2, 1995. [Online]. Available: https://www.fhwa.dot.gov/publications/publicroads/95winter/p95wi14.cfm. [Accessed: 28- May- 2017].

[5] A. Ensslen, W. Schwant and R. Zieglschmidt, "Can We Improve the Crash Performance of Seat Belts," *SAE Technical Paper Series*, 1985. [Online]. Available: http://papers.sae.org/851202/. [Accessed: 23- Nov- 2016].

[6] D. Bjerklie, "The Hidden Danger of Seat Belts," Time.com, 2006. [Online]. Available: http://content.time.com/time/nation/article/0,8599,1564465,00.html. [Accessed: 24- Nov- 2016].

[7] A. Zaouk, S. Kasturi, S. Punwani and H. Nagarajan, "Development of airbags for locomotive crew protection," *ASME/IEEE 2006 Joint Rail Conference*, 2006, pp. 115-120. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/1634066/. [Accessed: 23- Nov- 2016].

[8] K. Friedman, J. Hutchinson, D. Mihora, S. Kumar, R. Frieder and A. Sances, "Effect of roof strength in injury mitigation during pole impact," *Biomedical sciences instrumentation*, no. 43, pp. 69-74, 2007. [Online]. Available: https://www.researchgate.net/publication/6344485_Effect_of_roof_strength_in_injury_mitigation_during_pole_impact [Accessed: 18- Nov- 2016].

[9] D. Gerónimo, A. López, A. Sappa and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, vol. 32, no. 7, pp. 1239-1258, 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5010438/?arnumber=5010438. [Accessed: 23- Nov-2016].

[10] P. Labaziewicz, "Cars are becoming rolling sensor platforms", E2e.ti.com, 2014. [Online]. Available: https://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2014/09/25/cars-are-becoming-rolling-sensor-platforms. [Accessed: 28- May- 2017].

[11] M. Dempsey, ADAS sensors and features. 2016.

[12] J3016_201609, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. [Online]. Available: http://www.sae.org/misc/pdfs/automated_driving.pdf. [Accessed: 28- May- 2017].

[13] J. Anderson, N. Kalra, K. Stanley, P. Sorensen, C. Samaras and T. Oluwatola, "Autonomous Vehicle Technology: A Guide for Policymakers", RAND Corporation, 2016. Available: http://www.rand.org/pubs/research_reports/RR443-2.html. [Accessed: 28- May- 2017].

[14] M. Ramsey, "Self-Driving Cars Could Cut Down on Accidents, Study Says", WSJ.com, 2015. [Online]. Available: https://www.wsj.com/articles/self-driving-cars-could-cut-down-on-accidents-study-says-1425567905. [Accessed: 28- May- 2017].

[15] H. Frumkin, L. Frank and R. Jackson, *Urban sprawl and public health*, 1st ed. Washington, DC: Island Press, 2004, pp. 142-147.

[16] World Health Organization, "World Report on Disability", 2011. [Online]. Available: http://www.who.int/disabilities/world_report/2011/en/. [Accessed: 28- May- 2017].

[17] M. Sivak and B. Schoettle, "Road safety with self-driving vehicles: general limitations and road sharing with conventional vehicles", 2015. [Online]. Available: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/111735/103187.pdf. [Accessed: 28- May- 2017].

[18] C. Lee and M. Abdel-Aty, "Comprehensive analysis of vehicle–pedestrian crashes at intersections in Florida", *Accident Analysis & Prevention*, vol. 37, no. 4, pp. 775-786, 2005.

[19] W. Chen, *Vehicular Communications and Networks: Architectures, Protocols, Operation and Deployment*, 1st ed. 2015, p. 29.

[20] "BMW demonstrates left-turn drivers' aid", Articles.sae.org, 2011. [Online]. Available: http://articles.sae.org/9888/. [Accessed: 28- May- 2017].

[21] J. Sanguesa, J. Barrachina, M. Fogue, P. Garrido, F. Martinez, J. Cano, C. Calafate and P. Manzoni, Combined V2V and vehicle-to-infrastructure (V2I) communication (V2X)-based vehicular density estimation architecture, 2015.

[22] "Audi launches first Vehicle-to-Infrastructure (V2I) technology in the U.S. starting in Las Vegas", Audiusa.com, 2016. [Online]. Available: https://www.audiusa.com/newsroom/news/press-releases/2016/12/audi-launches-vehicle-to-infrastructure-tech-in-vegas. [Accessed: 28- May- 2017].

[23] "Connected Vehicles: Vehicle-to-Pedestrian Communications", https://www.its.dot.gov/. [Online]. Available: https://www.its.dot.gov/factsheets/pdf/CV_V2Pcomms.pdf. [Accessed: 28- May- 2017].

[24] J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi and J. Naranjo, "Vehicle to Pedestrian Communications for Protection of Vulnerable Road Users", in *2014 IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[25] M. Wolf, *Security Engineering for Vehicular IT Systems: Improving the Trustworthiness and Dependability of Automotive IT Applications*, 1st ed. 2009, pp. 64-66.

[26] "Biking to Work Increases 60 Percent Over Last Decade", Census.gov, 2014. [Online]. Available: https://www.census.gov/newsroom/press-releases/2014/cb14-86.html. [Accessed: 28- May- 2017].

[27] D. Katsaros and Y. Manolopoulos, "Prediction in wireless networks by Markov chains", *IEEE Wireless Communications*, vol. 16, no. 2, pp. 56-64, 2009.

[28] A. Bhattacharya and S. Das, "LeZi-Update: An Information-Theoretic Framework for Personal Mobility Tracking in PCS Networks", *Wireless Networks*, vol. 8, no. 23, pp. 121-135, 2002.

[29] J. Krumm, "A Markov Model for Driver Turn Prediction", in *SAE 2008 World Congress*, Detroit, Michigan, USA, 2008.

[30] R. Simmons, B. Browning, Y. Zhang and V. Sadekar, "Learning to Predict Driver Route and Destination Intent", in *Intelligent Transportation Systems Conference*, 2006.

[31] A. Karbassi and M. Barth, "Vehicle route prediction and time of arrival estimation techniques for improved transportation system management," in *Proceedings of the Intelligent Vehicles Symposium (IV '03)*, pp. 511–516, IEEE, June 2003.

[32] N. Ye, Z. Wang, R. Malekian, Y. Zhang and R. Wang, "A Method of Vehicle Route Prediction Based on Social Network Analysis", *Journal of Sensors*, vol. 2015, pp. 1-9, 2015.

[33] M. Morzy, "Prediction of Moving Object Location Based on Frequent Trajectories", in *Computer and Information Sciences - ISCIS 2006, 21th International Symposium*, Istanbul, Turkey, 2006.

[34] H. Jeung, Q. Liu, H. Shen and X. Zhou, "A Hybrid Prediction Model for Moving Objects", in *2008 IEEE 24th International Conference on Data Engineering (ICDE 2008)*, Cancun, Mexico, 2008.

[35] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici and A. Rabuffi, CRAWDAD dataset roma/taxi (v. 2014-07-17). [Online]. Available: http://crawdad.org/roma/taxi/20140717. [Accessed: 28- May- 2017].

[36] https://github.com/openstreetmap/Nominatim/blob/master/docs/install-on-ubuntu-16.md. [Accessed: 29- May- 2017]

[37] R. W. Sinnott, “Virtues of the Haversine,” *Sky and Telescope*, vol. 68, no. 2, 1984.