



**MARMARA UNIVERSITY**  
**INSTITUTE FOR GRADUATE STUDIES**  
**IN PURE AND APPLIED SCIENCES**



# **PACKET LOSS RATE PREDICTION FOR VEHICULAR NETWORKS WITH REGRESSION METHODS**

---

**OSMAN NURİ KOÇ**

**MASTER THESIS**

Department of Electrical and Electronics Engineering

**Thesis Supervisor**

Doç.Dr. ENGİN MAŞAZADE

ISTANBUL, 2022



**MARMARA UNIVERSITY**  
**INSTITUTE FOR GRADUATE STUDIES**  
**IN PURE AND APPLIED SCIENCES**



# **PACKET LOSS RATE PREDICTION FOR VEHICULAR NETWORKS WITH REGRESSION METHODS**

---

525018009

**MASTER THESIS**

Department of Electrical and Electronics Engineering

**Thesis Supervisor**

Doç.Dr. ENGİN MAŞAZADE

ISTANBUL, 2022

**MARMARA UNIVERSITY**  
**INSTITUTE FOR GRADUATE STUDIES**  
**IN PURE AND APPLIED SCIENCES**

OSMAN NURİ KOÇ, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended her thesis entitled “**Packet loss rate prediction for vehicular networks with regression methods**”, on March 31, 2022 and has been found to be satisfactory by the jury members.

**Jury Members**

Doç. Dr. Engin Maşazade (Advisor)  
Marmara University ..... (SIGN).....

Doç. Dr. Onur Cihan (Jury Member)  
Marmara Üniversitesi ..... (SIGN).....

Prof. Dr. Serkan Topaloğlu (Jury Member)  
Yeditepe Üniversitesi..... (SIGN).....

**APPROVAL**

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Osman Nuri KOÇ be granted the degree of Master of Science in Department of Electrical and Electronics Engineering, Electrical and Electronics Program on March 31, 2022.

**Director of the Institute**

## **ACKNOWLEDGEMENT**

First of all, I would like to express my deep and sincere gratitude to my supervisor Doç. Dr. Engin Maşazade for his invaluable supervision, support and guidance throughout the study.

I am grateful to my family for their endless support and encouragement through my study.

**March 2022**

**Osman Nuri KOÇ**

# CONTENTS

|                                                               |    |
|---------------------------------------------------------------|----|
| ACKNOWLEDGEMENT .....                                         | i  |
| CONTENTS .....                                                | ii |
| ABSTRACT .....                                                | v  |
| ÖZET .....                                                    | vi |
| ABBREVIATIONS .....                                           | vi |
| LIST OF FIGURES .....                                         | ix |
| LIST OF TABLES .....                                          | xi |
| Chapter 1 - INTRODUCTION .....                                | 1  |
| 1.1 Background .....                                          | 1  |
| 1.2 Concept .....                                             | 2  |
| 1.3 Literature Review .....                                   | 5  |
| 1.3.1 Vanet Broadcast Storm Problem.....                      | 5  |
| 1.3.2 Vanet Machine Learning-Based Studies.....               | 7  |
| 1.3.3 Vanet Transmission Power and Data Rate Adaptation ..... | 8  |
| 1.4 Contributions of the Thesis .....                         | 9  |
| 1.4.1 Vanet Broadcast Storm Problem.....                      | 10 |
| 1.4.2 Vanet Machine Learning-Based Studies.....               | 10 |
| 1.4.3 Vanet Transmission Power and Data Rate Adaptation ..... | 11 |
| 1.5 Organization of the Thesis .....                          | 11 |
| Chapter 2 - SYSTEM MODEL .....                                | 12 |
| 2.1 Machine Learning .....                                    | 12 |
| 2.2 Regression.....                                           | 14 |
| 2.2.1 Linear Regression.....                                  | 16 |

|                                  |                                                   |    |
|----------------------------------|---------------------------------------------------|----|
| 2.2.2                            | Bayesian Ridge Regression .....                   | 16 |
| 2.2.3                            | Kernel Ridge Regression .....                     | 17 |
| 2.2.4                            | Automatic Relevance Determination Regression..... | 17 |
| 2.2.5                            | K-Nearest Neighbors Regression .....              | 18 |
| 2.2.6                            | Bagging Regression .....                          | 18 |
| 2.2.7                            | Support Vector Regression .....                   | 18 |
| 2.2.8                            | NU Support Vector Regression.....                 | 20 |
| 2.2.9                            | Gaussian Process Regression .....                 | 20 |
| 2.2.10                           | Decision Tree Regression .....                    | 21 |
| 2.2.11                           | Extra Decision Tree Regression .....              | 22 |
| 2.2.12                           | Gradient Boosting Regression .....                | 22 |
| 2.2.13                           | Histogram-Based Gradient Boosting Regression..... | 23 |
| 2.2.14                           | Cat Boosting Regressor .....                      | 23 |
| 2.2.15                           | LightGBM Regressor .....                          | 24 |
| 2.2.16                           | Extreme Gradient Boosting Regressor .....         | 25 |
| 2.2.17                           | Stochastic Gradient Descent Regressor .....       | 26 |
| Chapter 3 - IMPLEMENTATION ..... |                                                   | 27 |
| 3.1                              | Ubuntu .....                                      | 28 |
| 3.2                              | Omnet++ .....                                     | 28 |
| 3.3                              | Omnet++ Installation .....                        | 29 |
| 3.4                              | Inet .....                                        | 30 |
| 3.5                              | Inet Installation .....                           | 30 |
| 3.6                              | Veins.....                                        | 30 |
| 3.7                              | Veins Installation.....                           | 31 |

|             |                                              |    |
|-------------|----------------------------------------------|----|
| 3.8         | Sumo .....                                   | 31 |
| 3.9         | Sumo Installation .....                      | 32 |
| 3.10        | Simulation Run.....                          | 32 |
| 3.11        | Python .....                                 | 32 |
| 3.12        | Scikit-Learn.....                            | 33 |
| Chapter 4 - | NUMERICAL RESULTS.....                       | 34 |
| Chapter 5 - | CONCLUSIONS FUTURE RESEARCH DIRECTIONS ..... | 45 |



# **ABSTRACT**

## **PACKET LOSS RATE PREDICTION FOR VEHICLER NETWORKS WITH REGRESSION METHODS**

According to TUIK data, an average of 1 million traffic accidents occur annually in Turkey between 2009 and 2020, and an average of 5 thousand people lose their lives. With the connected vehicle technology, it is aimed to increase the safety of all traffic elements on the roads, thus preventing traffic accidents, reducing loss of life and injuries. Connected vehicle technology will be one of the indispensable technologies in the establishment of smart cities in order to increase safety on the roads, as well as to disseminate autonomous vehicle technology and increase its reliability. Connected vehicle technology allows vehicles on the road to communicate with other vehicles on the road by sending short-range radio signals to communicate with each other. In this thesis, we propose a heuristic function that calculates the packet loss probability in the vehicle network using advanced regression methods when different vehicle density, transmission speed, and transmission power values are given. The simulation environment that is as similar as possible to real-world conditions has been prepared by using the open-source Sumo software, which is suitable for creating various flexible traffic environments that include real-world traffic elements such as roads, vehicles, pedestrians, traffic lights, and public transportation systems. Simulations are executed for different traffic conditions with the integrated operation of Omnet++, Sumo and Veins environments. The results from these simulations are translated into the dataset, which is used as input to various regression algorithms to predict the packet loss performance. Numerical results showed that the Catboost regression method gives the least error between predicted and tested results compared as compared to other regression methods. Thanks to the obtained heuristic function, when a set of transmission parameters is given, packet loss rates can be determined without simulation, and it may be possible to reduce the PLR below the target level by re-adjusting the transmission parameters.

## ÖZET

### REGRESYON YÖNTEMLERİYLE ARAÇ AĞLARI İÇİN PAKET KAYIP ORANI TAHMİNİ

TÜİK verilerine göre Türkiye'de 2009-2020 yılları arasında yılda ortalama 1 milyon trafik kazası meydana gelmekte ve ortalama 5 bin kişi hayatını kaybetmektedir. Bağlantılı araç teknolojisi ile yollardaki tüm trafik unsurlarının güvenliğinin artırılması, böylece trafik kazalarının önlenmesi, can kaybı ve yaralanmaların azaltılması hedeflenmektedir. Yollarda güvenliğin artırılmasının yanı sıra otonom araç teknolojisinin yaygınlaştırılması ve güvenilirliğinin artırılması için akıllı şehirlerin kurulmasında bağlantılı araç teknolojisi vazgeçilmez teknolojilerden biri olacaktır. Bağlantılı araç teknolojisi, birbirleriyle iletişim kurmak için kısa menzilli radyo sinyalleri göndererek yoldaki araçların yoldaki diğer araçlarla iletişim kurmasını sağlar. Bu tezde, farklı araç yoğunluğu, iletim hızı ve iletim gücü değerleri verildiğinde, gelişmiş regresyon yöntemlerini kullanarak araç ağındaki paket kaybı olasılığını hesaplayan bir sezgisel fonksiyon önerilmektedir. Yollar, araçlar, yayalar gibi gerçek dünya trafik unsurlarını içeren çeşitli esnek trafik ortamları oluşturmaya uygun açık kaynaklı Sumo yazılımı kullanılarak gerçek dünya koşullarına mümkün olduğunca benzer simülasyon ortamı hazırlanmıştır. Omnet++, Sumo ve Veins ortamlarının entegre çalışması ile farklı trafik koşulları için simülasyonlar yürütülmüştür. Bu simülasyonlardan elde edilen sonuçlar, paket kaybı oranı (PKO) performansını tahmin etmek için çeşitli regresyon algoritmalarına girdi olarak kullanılan veri kümesine çevrilmiştir. Sayısal sonuçlar, Catboost regresyon yönteminin, diğer regresyon yöntemlerine kıyasla tahmin edilen ve test edilen sonuçlar arasında en az hatayı verdiğini göstermiştir. Elde edilen buluşsal fonksiyon sayesinde bir dizi iletim parametresi verildiğinde simülasyon yapılmadan paket kayıp oranları belirlenebilmekte ve iletim parametreleri yeniden ayarlanarak PLO'yu hedef seviyenin altına düşürmek mümkün olabilmektedir.

## **ABBREVIATIONS**

**VANET:** Vehicular Ad Hoc Networks

**V2V:** Vehicle to Vehicle

**V2I:** Vehicle to Infrastructure

**IEEE:** Institute of Electrical and Electronics Engineers

**ITS:** Intelligent Transportation Systems

**BSM:** Basic Safety Messages

**OBU:** On Board Unit

**RSU:** Road Side Unit

**ML:** Machine Learning

**DL:** Deep Learning

**KNN:** K-Nearest Neighbors

**SVM:** Support Vector Machine

**CNN:** Convolutional Neural Network

**BLSTME:** Boosted Long Short-Term Memory Ensemble

**SUMO:** Simulation of Urban Mobility

**NS2:** Network Simulator 2

**NS3:** Network Simulator 3

**TraCI:** Traffic Control Interface

**TCP:** Transmission Control Protocol

**NED:** Network Definition

**Veins:** Vehicles In Network Simulation

**PRR:** Packet Reception Ratio

**MCD:** Modularity based Mobility aware Community Detection

**PLR:** Packet Loss Rate

**TLM:** Total Number of Lost Messages

**TRM:** Total Number of Received Messages

**CAM:** Cooperative Awareness Messages



## LIST OF FIGURES

|                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------|----|
| <b>Figure 1</b> Connected Vehicle Technology (Danlaw, 2022) .....                                         | 2  |
| <b>Figure 2</b> Europe ITS Channel Plan (Intelligent Transportations Systems, 2020).....                  | 2  |
| <b>Figure 3</b> The Content of BSM Message (Kim, Kim, & Jeon, 2018).....                                  | 3  |
| <b>Figure 4</b> Vanet RSU and OBU .....                                                                   | 3  |
| <b>Figure 5</b> Broadcast Storm (Najafzadeh, Ithnin, & Razak, 2014).....                                  | 5  |
| <b>Figure 6</b> Supervised Learning (Atmaca, Supervised Learning, 2021) .....                             | 13 |
| <b>Figure 7</b> Unsupervised Learning (Atmaca, Unsupervised Learning, 2021).....                          | 13 |
| <b>Figure 8</b> Vanet Signal Propagation .....                                                            | 16 |
| <b>Figure 8</b> Support Vector Machine (Noyum, Mofenjoui, Feudjio, Göktug, & Fokoue, 2021)<br>.....       | 20 |
| <b>Figure 9</b> Decision Tree Algorithm (Machine Learning Basics: Decision Tree Regression,<br>2020)..... | 22 |
| <b>Figure 10</b> Gradient Boosting Algorithm (Natekin & Knoll, 2013) .....                                | 23 |
| <b>Figure 11</b> Light Gradient Boosting Machine Algorithm (Mohammed & Kadhem, 2021)<br>.....             | 25 |
| <b>Figure 13</b> System Model Block Diagram .....                                                         | 27 |
| <b>Figure 14</b> VANETs Simulators (Herrera, Igartua, & Iza, 2016).....                                   | 28 |
| <b>Figure 15</b> Omnet++ Architecture ( CORIA & A., 2018) .....                                           | 29 |
| <b>Figure 16</b> Omnet++ Welcome Window .....                                                             | 30 |
| <b>Figure 17</b> Veins Architecture (Sommer, 2006) .....                                                  | 31 |
| <b>Figure 18</b> Sumo Environment (Behrisch & Hilbrich, 2001) .....                                       | 32 |
| <b>Figure 19</b> Scikit-learn Method Selection Guide (Cournapeau, 2007) .....                             | 33 |
| <b>Figure 20</b> Highway Scenario .....                                                                   | 34 |
| <b>Figure 21</b> Urban Scenario .....                                                                     | 34 |

|                                                               |    |
|---------------------------------------------------------------|----|
| <b>Figure 22</b> Simulation outputs dataset .....             | 36 |
| <b>Figure 23</b> Highway 100-500-1000.....                    | 39 |
| <b>Figure 24</b> Urban 100-500-1000 .....                     | 40 |
| <b>Figure 25</b> Highway Linear Regression Prediction.....    | 41 |
| <b>Figure 26</b> Urban Linear Regression Prediction .....     | 41 |
| <b>Figure 27</b> Highway CatBoost Regression Prediction ..... | 42 |
| <b>Figure 28</b> Urban CatBoost Regression Prediction.....    | 42 |
| <b>Figure 29</b> Highway CatBoost Prediction Top View .....   | 43 |
| <b>Figure 30</b> Urban CatBoost Prediction Top View .....     | 44 |

**LIST OF TABLES**

**Table 1** Simulation Parameter ..... 35

**Table 2** Highway Only ..... 37

**Table 3** Urban Only..... 38

**Table 4** Urban and Highway ..... 38



# Chapter 1 - INTRODUCTION

## 1.1 Background

VANETs (vehicular ad hoc networks) is a kind of MANETs (mobile ad hoc networks). In VANETs, vehicles can communicate with each other (V2V) and infrastructure (V2I). In general, VANETs can be defined as the communication that takes place in a dynamic network topology consisting of fast-moving and varying densities vehicles.

VANETs applications can be expressed under two main headings, which are life safety and service (Laberteaux & Hartenstein, 2010). Safety applications can be summarized as Curve speed warning, Forward Collision warning, Pre-crash awareness, Left turn assist, Lane change warning, Emergency brake lights. Similarly, service applications can be summarized as Internet access, media and connectivity, and payment services such as parking and E-toll collection.

Connected vehicle technology allows vehicles on the road to send short-range radio signals to communicate with each other so that other vehicles on the road know where they are. Thus, for example, drivers receive notifications and alerts about dangerous situations, such as blind intersection, forward collision warning, do not pass warning, platooning and adaptive cruise control messages, emergency vehicle alerts, and etc. The visual figure that gives an idea about the concept of connected vehicles can be seen in Figure 1.



**Figure 1** Connected Vehicle Technology (*Danlaw, 2022*)

The electromagnetic spectrum starts from the frequency of 3Hz and extends up to 300EHZ. The radio spectrum, which is a subset of the electromagnetic spectrum, is located between 3 kHz - 300GHz frequencies. The IEEE 802.11p standard contains a collection of standards that are important for enabling communication in a vehicle environment, i.e., one that is constantly changing. WAVE's operational frequency is set in the DSRC band of 5.85–5.925 GHz. Within this range, a channel designated as CCH (control channel) is used for system control and safety-related communications, and up to six signaling channels are utilized for the interchange of non-safety data (SCHs). The channel time is divided into synchronization intervals with a predetermined length of 100 ms, consisting of intervals of equal length alternating between the CCH and the SCH, as proposed by WAVE. 802.11p spectrum is organized into 7 channels each of which is 10 MHz wide as shown in the Figure 2.

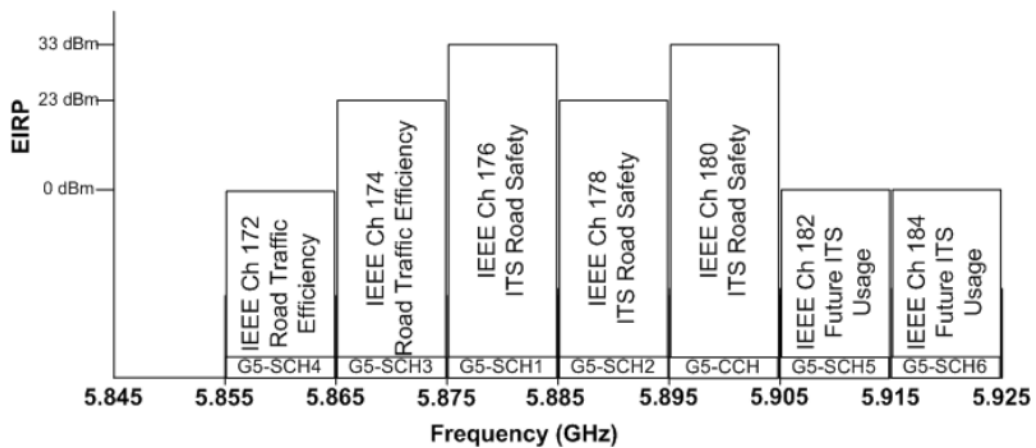


**Figure 2** Europe ITS Channel Plan (*Intelligent Transportations Systems, 2020*)

## 1.2 Concept

One of the main purposes of Vanet is to increase the safety and comfort of all elements in traffic. To achieve this purpose, vehicles constantly send messages or packets to each other and to infrastructure. The basis for the provision of security applications of vehicles connected in Vanet depends on the Basic Security Messages (BSMs). It is critical for BSM's connected vehicle technologies, known in the literature as the heartbeat message to highlight its importance. Transmitted approximately 10 times per second. The BSM

contains information such as location, speed, brake and accelerator pedal positions, as seen in Figure 3.

| Type                 | Description                                                                                                                                | Size (byte) |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| DSRCmsgID            | Data elements used in each message to define the Message type                                                                              | 1           |
| MsgCount             | It can check the flow of consecutive messages having the same DSRCmsgID received from the same message sender.                             | 1           |
| TemporaryID          | Represents a 4-byte temporary device identifier. When used in a mobile OBU device, this value is periodically changed to ensure anonymity. | 4           |
| Dsecond              | Represents two bytes of time information.                                                                                                  | 2           |
| Latitude             | Represents the geographic latitude of an object.                                                                                           | 4           |
| Longitude            | Represents the geographic longitude of an object.                                                                                          | 4           |
| Elevation            | Represents an altitude measured by the WGS84 coordinate system.                                                                            | 2           |
| PositionAccuracy     | Various quality parameters used to model the positioning accuracy for each given axis.                                                     | 4           |
| TransmissionAndSpeed | Represents the speed of the vehicle.                                                                                                       | 2           |
| Heading              | The current direction value is expressed in units of 0.0125 degrees.                                                                       | 2           |
| SteeringWheelAngle   | Represents the current steering angle of the steering wheel.                                                                               | 1           |
| AccelerationSet4Way  | It consists of three orthogonal directions of acceleration and yaw rate.                                                                   | 7           |
| BrakeSystemStatus    | Represents a data element that records various control states related to braking of the vehicle.                                           | 2           |
| VehicleSize          | Represents the length and width of the vehicle.                                                                                            | 3           |

**Figure 3** The Content of BSM Message (*Kim, Kim, & Jeon, 2018*)

These messages are sent and received via the channels reserved for IEEE802.11p by On-Board Units (OBU) devices in vehicles and Road Side Units (RSU) devices located on the roadside in infrastructure as shown in Figure 4. Thus, assistance is provided to prevent possible collisions between vehicles.

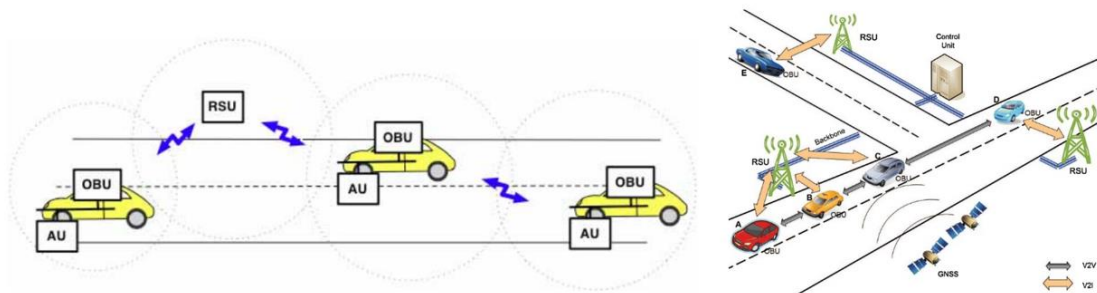


**Figure 4** Vanet RSU and OBU

Several multihop applications for automobile ad hoc networks use broadcast to find nearby neighbors or to spread important traffic information to other vehicles in a specific

geographic area. However, due to an excessive amount of broadcast packets, the traditional broadcast technique may result in the so-called broadcast storm problem, which is a scenario in which there is a high level of contention and collisions at the link layer (Wisitpongphan, et al., 2007).

There are different solution methods to the broadcast storm problem in the literature. Some of them are as follows.

### **Simple Flooding**

Broadcast is the pedestal of all Vehicular ad hoc system interchanges. Blind flooding is the most straightforward manifestation of broadcast. A node broadcasts a packet that is received by every neighboring node within its transmission range in blind flooding. (Farooq & Khan, 2014).

### **Probabilistic-based Flooding**

The probability-based strategy is becoming more appealing among the different broadcast approaches for overcoming the problem of flooding. When there are more contenders to broadcast, each node is allocated a fixed probability, reducing the odds of collision and broadcasting.

### **Counter based Flooding**

The main principle behind the counter-based approach is that as a node hears the same message from surrounding nodes an increasing number of times, the additional coverage benefit from having the node broadcasting reduces.

### **Area based Flooding**

The concept of coverage area is used in area based flooding to change the broadcast zone within the designated geographical area.

### **Location Based Flooding**

Messages are broadcast based on the geographic region of the cars via location-based approaches. Each node includes its own location in the message header, which the receiving node uses to determine the additional coverage area to broadcast. The cost of

calculating extra coverage areas is the biggest drawback of this strategy (Xu, Segupta, Jiang, & Chrysler, 2003).

### 1.3 Literature Review

In this section we present the literature review on three main issues, including common methods and solutions with the subjects of this study. First of all, we present the studies on the Vanet broadcast storm problem. Secondly, we present the studies based on Vanet machine learning-based. Finally, we introduce the studies on Vanet transmission power and data rate adaptation respectively.

#### 1.3.1 Vanet Broadcast Storm Problem

In (Babu & Newton, 2016), a new technique is proposed to solve the broadcast storm problem in vehicle ad hoc network communication. The study, it was assumed that all vehicles are on a linear highway. It was also assumed that each vehicle knows and lists the IDs of the vehicles around it. The signal message structure planned to be sent in the study was created as vehicle identification number, vehicle position, speed, time stamp and neighboring vehicles. It was stated that the broadcast storm problem was reduced with the proposed technique (Babu & Newton, 2016) .

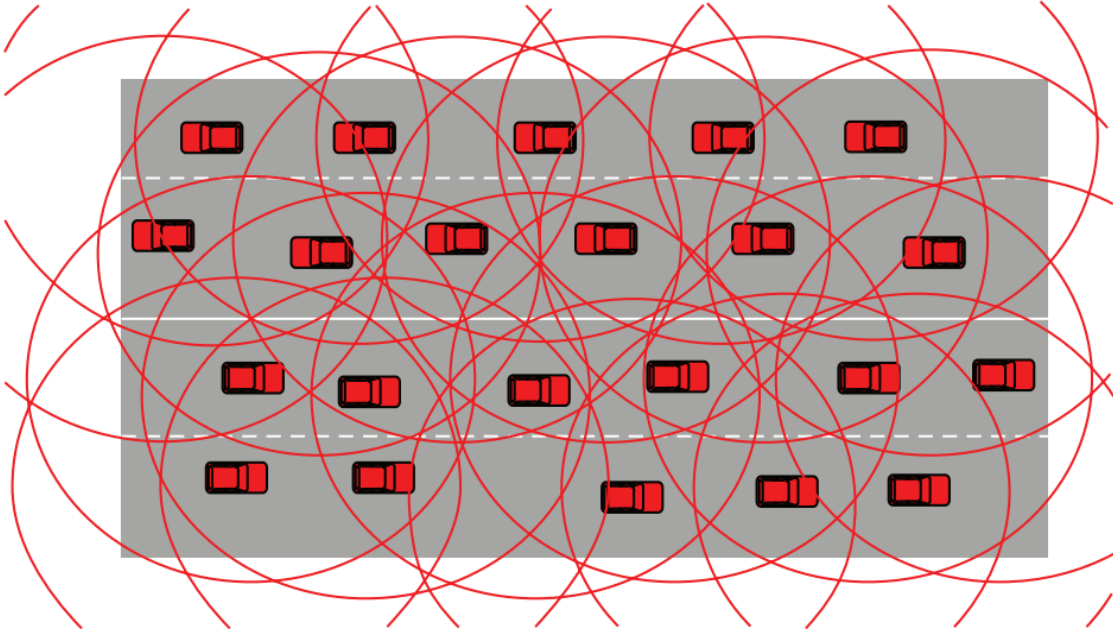


**Figure 5** Broadcast Storm (*Najafzadeh, Ithnin, & Razak, 2014*)

In (Kamakshi & Sriram, Modularity based mobility aware community detection algorithm for broadcast storm mitigation in VANETs, 2020), the authors deal with the broadcast storm problem, the MCD (Modularity based Mobility aware Community Detection) algorithm was used in vehicle ad hoc network communication. Vehicles were divided into groups according to their distance from each other. Messages to be sent were transmitted within the group and vehicles entering and leaving the group were checked periodically. Simulations were made by selecting real highways in India and America via Open Street Map. Packet collision based on CDS/SC (Connected Dominating Set and Set Cover) (Kamakshi & Sriram, Plummeting broadcast storm problem in highways by clustering vehicles using dominating set and set cover, 2019) and SAND/SHARC (Stability and Network Dynamics over a Sharper Heuristic for assignment of Robust Communities) (Herbiet, Bouvry, & Guinand, 2011) algorithms of the results obtained with the MMCD algorithm. It was stated that the rate of packet loss, the end-to-end delay decreased, while the efficiency and accessibility increased.

In (Burhan & Rehman, 2020), the authors sought a solution to the broadcast storm, and calculations were made based on the location information, speed and direction of the vehicles, and the vehicles to which the messages will be sent were selected. In the study, NS-3 was used as a network simulator and Sumo is used as a traffic simulator. It was observed that the number of messages sent in the communication channel and the delay are reduced with the solution they propose.

Another proposed solution to the problem of Vanet broadcast storm is Selective epidemic broadcast algorithm (Chitra & Sathya, 2018). This solution method created a dispatch list and a neighbor vehicle list based on the location and speed information of the vehicles. It offered a solution to the broadcast storm problem by not sending from outside the vehicles in the neighbor vehicle list. In the study, simulations were carried out at low 20, medium 40-60, high 80-100 vehicle/km traffic densities and NS-2 was used as a network simulator.

In (Feukeu & Zuva, 2020), the authors sought a solution to the broadcast storm problem. The developed algorithm has set the maximum message sending distance to 1000m (minimum safe distance). Broadcast storm is reduced with the algorithm established

based on channel busy rate, distance between vehicles, vehicle speeds and message sending time inputs. Matlab program was used as the simulation environment. Simulations were performed for 3 different vehicle densities, 10cars/lane, 40cars/lane and 80cars/lane.

Finally, in (Khan, Moharram, Alaraj, & Azam, 2014), the authors proposed a group-based solution with the help of an algorithm they named Media Mixing Algorithm. Each of the OBUs and RSUs has a unique ID initially. Then the vehicles receive the group ID given by the RSUs according to the region they are in and can send to the vehicles in the same group. The list of vehicles in the group was kept in the RSU, and RSU followed the entry and exit of the group. NS-3 was used as a network simulator.

### **1.3.2 Vanet Machine Learning-Based Studies**

Many machine learning studies with Vanet simulation have been studied based on security (So, Sharma, & Petit, 2018) (Mahmoudi, Kamel, Ben-Jemaa, Kaiser, & Urien, 2019). In (Montenegro, Iza, & Igartua, 2020), the misbehaviors of the vehicles were captured by information of the position and received signals. The classification of the vehicles with misbehave detection was made by giving the vehicle behaviors data from the Vanet simulation to the KNN ML algorithm. In another study (ZHANG, CHEN, ZENG, & XUE, 2018), data generated using Omnet++, Sumo and OpenStreetMap tools were used to train SVM from ML algorithms for misbehavior detection.

In (Kothai, et al., 2021), one of the unsupervised learning DL algorithms convolutional neural network (CNN) and boosted long short-term memory ensemble (BLSTME) were used over the Python Tensorflow tool. Using Omnet++ and SUMO, traffic congestion scenarios were created and then photographed. These photos were used for data feeding CNN-BLSTME, and then congestion prediction was made. In (Zhao, Cheng, Mao, & He, 2019), traffic accident prediction was made using CNN from DL algorithms. They used the UK Car Accident dataset to train CNN. It was stated that the accident prediction model they created was more successful than the traditional Back Propagation, but it was stated that it was too slow for real-time applications.

In (Taherkhani & Pierre, 2016), K-means was to filter messages and packet loss reduction was achieved. By controlling 50, 100, 150, 200, 250, 300, 400, 450, 500 different vehicle

densities and data rates 3, 4.5 6, 9, 12, 18, 24, 27 Mbps vehicle congestion, messages were filtered with the help of K-means. NS2 was used as network simulator and SUMO was used as traffic simulator..

### **1.3.3 Vanet Transmission Power and Data Rate Adaptation**

First of all, a patent received in the USA is the transmission power adaptation for moving things. It has been stated that the existing communication systems do not use any power adaptation system. Patent study reveals that the transmission power of the communication systems, including RSU and OBU, can be controlled. (Mountain View (CA)/ United States Patent No. US009736791B1, 2017)

In (Frigau, 2013), transmission power adaptation was studied. A neighbor vehicle list was first created by looking at the distances of the vehicles around the vehicle. Then, the transmission power was adopted by observing the vehicle density, packet reception rate, and beacon load values in its vicinity. NS-3 was used as network simulator for Vanet. It was reported that the rate of PRR (packet reception rate) increased by 40% throughput and varied between 0.4 and 0.9.

In (Wang, et al., 2022), messages from vehicles outside the distance specified for the safe braking distance were ignored. Sumo was used as a traffic simulator in the study. It was reported that in the studies carried out with the NS-2.28 network simulator, it was observed that the packet transmission rate increased by 4.5%. In the study, the simulation was handled for the number of vehicles (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100) in a single scenario. Researchers stated that the number of vehicles should be increased in future studies.

In (Deeksha, et al., 2022), it was aimed to keep the Vanet communication channel capacity constant below at a certain value with the transmission power and data rate adaptation study. Sumo was used as a traffic simulator. The communication channel busy rate variable has been determined and depending on the change in this value, the data rate varied between 12, 9, 6, 3 Mbps. The transmit power ranges from 10dBm to 23dBm. Thus, it was stated that the communication channel density was kept below a certain level.

In (Kashihara, Sahara, Kaneda, & Ohta, 2019), V2I communication channel performance enhancement was made with data rate adaptation. The data rates with the best

communication channel performance were determined for 3, 4, 5, 6, 9, 12, 18, 24, and 27 Mbps data rates with the help of the algorithm established depending on the number of lost frames. Data received (bytes) was used as the communication channel performance metric. Researchers stated that the maximum performance values of the communication channel were reached at data-rates 12,18,24,27 Mbps. Finally, RSSI level measurement was used to detect cyber Sybil attacks, which is another power control study (Yao, et al., 2019) based on Vanet, where transmission power is used.

#### **1.4 Contributions of the Thesis**

Vehicles should send BSM and other messages with the lowest packet loss rate at different traffic densities, in different regions such as urban and highway. Therefore, it is critical to determine the optimum transmit power and bandwidth in order to ensure that vehicles communicate with low packet loss rates in different regions and traffic densities. Continuously transmitting packets at high transmission power and high data rates, regardless of variable vehicle densities and different roads, may cause large number of packet collisions in the air and cannot be sent due to channel congestion. These defects prevent BSM and other messages from reaching the vehicles hence increasing the security risk and increasing traffic congestion.

The proposed approach for the problem in this paper is the selection of packet transmission power and data rates by machine learning algorithms according to varying vehicle densities and different roads. The aim of the approach is to provide Vanets communication with the lowest packet loss rate according to traffic congestion and roads. In this study, both highway and urban scenarios selected from real roads in Istanbul were used. The dataset was created by making simulations at different vehicle densities such as 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, different data rates such as 3, 6, 9, 12, 18, 24, 27 Mbps and different transmitting powers such as 20mW, 40mW, 80mW, 120mW. Next, we compare our work with the state-of-the art methods introduced in the previous literature.

#### **1.4.1 Vanet Broadcast Storm Problem**

Many solutions (Probabilistic-based Flooding, Counter-based Flooding, Area-based Flooding, Location Based Flooding) presented in the literature to the broadcast storm problem are based on the assumptions that lists of neighboring vehicles are created or data exchange between vehicles is reached without any deterioration. In the study (Babu & Newton, 2016), the problem definition was made for ideal traffic environments by making many assumptions, so there is no data about the working performance in real traffic conditions. In other studies (Kamakshi & Sriram, Modularity based mobility aware community detection algorithm for broadcast storm mitigation in VANETs, 2020) (Burhan & Rehman, 2020) (Chitra & Sathya, 2018) (Feukeu & Zuva, 2020) (Khan, Moharram, Alaraj, & Azam, 2014), it is assumed that messages or packets in the communication channel reach to the vehicles or infrastructure correctly. However, the communication channel between vehicles is open to many cyber-attacks such as GPS spoofing or BSM, CAM message spoofing. The solution method we propose in this study is not dependent on GPS or BSM, CAM messages from the surrounding vehicles. Furthermore, in (Burhan & Rehman, 2020), the number of vehicles (or density) was not changed, and only 75 vehicles were simulated. In this study, we observe the performance under varying traffic density.

#### **1.4.2 Vanet Machine Learning-Based Studies**

In the field of Vanet, many machine learning-based studies (So, Sharma, & Petit, 2018) (Mahmoudi, Kamel, Ben-Jemaa, Kaiser, & Urien, 2019) (Montenegro, Iza, & Igartua, 2020) (ZHANG, CHEN, ZENG, & XUE, 2018) were developed for the detection of cyber-attacks. Other machine learning-based studies (Kothai, et al., 2021) (Zhao, Cheng, Mao, & He, 2019) were traffic jam and accident prediction with image processing. In the machine learning study (Taherkhani & Pierre, 2016), which is the most similar to this study in the literature, message filtering was performed for different data rates and vehicle densities. In this study, in addition to data rate and vehicle densities, we aim to reduce the packet loss rate by changing the transmission power under two different traffic scenarios.

### **1.4.3 Vanet Transmission Power and Data Rate Adaptation**

In studies on transmission power and data rate adaptation, different transmission powers, data rates, and traffic scenarios have not been studied as much as in this proposed study. Studies such as (Frigau, 2013) (Wang, et al., 2022) (Deeksha, et al., 2022), assume that location data is sent between vehicles correctly. However, the communication channel between vehicles is open to many cyber-attacks such as GPS spoofing or BSM, CAM message spoofing.

In (Deeksha, et al., 2022) transmission power and data rate were adopted to keep the communication channel capacity below a certain level. Only data rates of 12, 9, 6, 3 Mbps and transmit powers of 10dBm and 23dBm were used. On the other hand, in our study, we aim to reduce the packet loss rate and all data rates allowed by the protocol were considered.

In (Kashihara, Sahara, Kaneda, & Ohta, 2019), only data rate adaptation was made. In our study, the packet loss rate change is analyzed depending on both the data rate and the transmission power, by changing the data rate and the transmit power.

## **1.5 Organization of the Thesis**

This thesis consists of five chapters; In Chapter 1, we present the introduction of the study, literature survey and the contributions of this thesis. In Chapter 2, we present the system model which gives a brief definition of machine learning and regression methods, problem formulation and the methods to solve the given problem. In Chapter 3, we present the software implementation. In Chapter 4, we present our numerical results, and finally, we devote Chapter 5 to our conclusion and address future research directions.

## **Chapter 2 - SYSTEM MODEL**

### **2.1 Machine Learning**

A machine learning method for obtaining information about the input-output relationship of a system from a set of connected input-output training samples is called Supervised Learning. An input-output training sample, also known as labeled training data or supervised data, also appears as the label of the output, input data or control. The goal of supervised learning is to create an artificial system that can learn how to match input data with output data and predict the output of the system given new input data. If the output data creates a finite set of discrete values determined by the labels of the input data, the learned combination causes the input data to be classified. If the output data accepts number values without stopping, the input data is rolled back. The learning model is frequently used to make sense of the links between input and output values. When these variables are not found as in the training data, the trained learning system should use an estimation method to find these variables. When these parameters are not directly available from the training samples, a learning system must use an estimation technique to obtain them. The biggest difference between supervised and unsupervised learning is that my supervised learning needs defined (labeled) data to learn, while my unsupervised learning doesn't need it. (Liu & Wu, 2012). Supervised and unsupervised algorithms shown in Figure 6 and Figure 7.

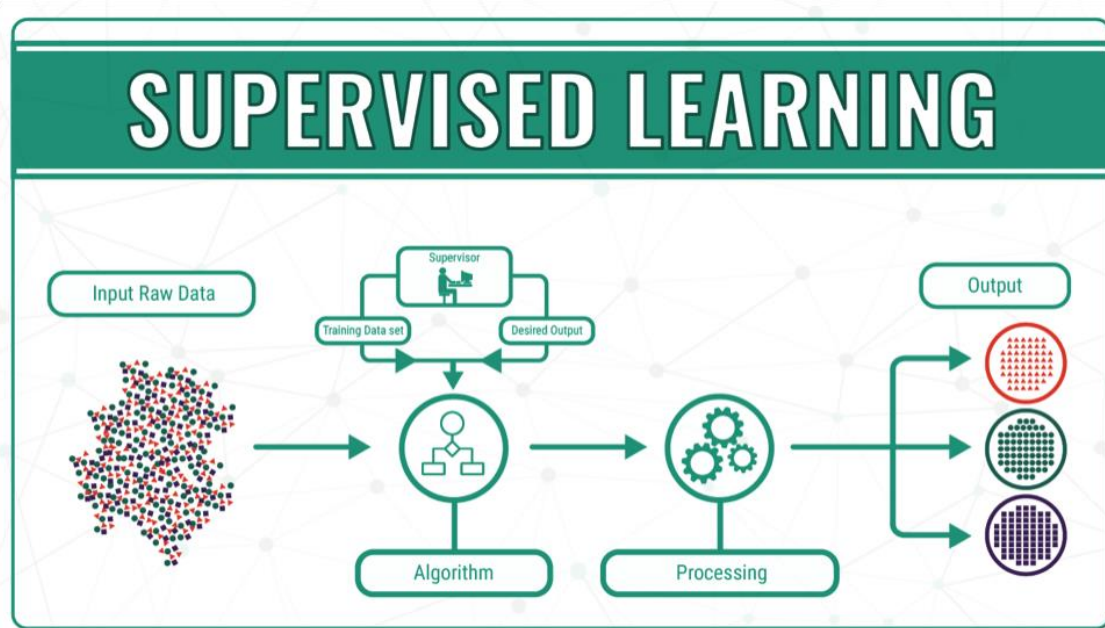


**Figure 6** Supervised Learning (*Atmaca, Supervised Learning, 2021*)

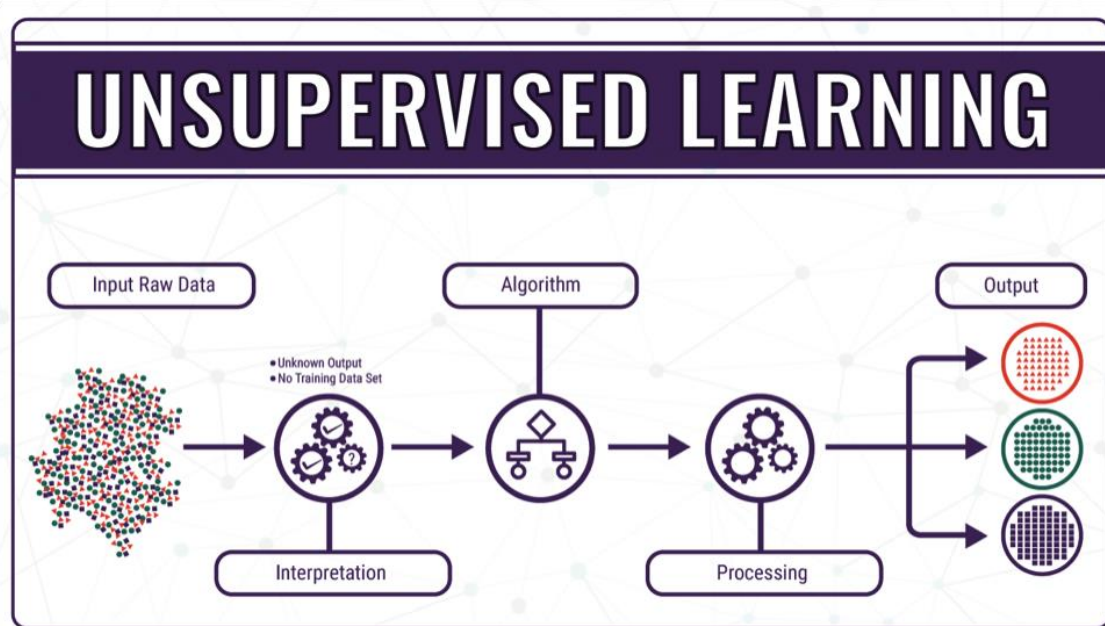


**Figure 7** Unsupervised Learning (*Atmaca, Unsupervised Learning, 2021*)

## 2.2 Regression

The analysis of the relationship between another variable (very often called the outcome or response variable) and one or more non-dependent variables (very often called predictors, covariates, explanatory variables or properties) is called regression analysis. Linear regression is the most common type of regression analysis, in which the most accurate line (or more than one linear result) is found according to mathematical operations in succession. Basic narrowest squares, for example, finds the unique line (or hyperplane) that minimizes the sum of the results by subtracting the squares between the actual data and that line (or hyperplane). This allows the researcher to estimate a contingent expectation (or mean value) of the dependent variable when the non-dependent variables have a set of values determined based on invariant mathematical operations (see linear regression). Alternative position parameters (for example, quantitative regression or Required State Analysis) or conditional expectation on a larger set of nonlinear models are estimated using slightly different approaches in less common types of regression (for example, non-parametric regression). The use of regression analysis is used for two conceptually different purposes. Firstly, it is widely used for prediction and forecasting and shares a lot of common ground with machine learning. Other reason second, regression analysis is used to find causal relationships between non-dependent and dependent variables in special cases. The most important output of the regressions independently reveals the correlations between a dependent variable and one or more groups of non-dependent variables in the data set. Before resorting to regression to find predictive or causal relationship outcomes, one should carefully demonstrate why existing correlations are capable of predicting new relationships or why a causal relationship exists between two variables. While causal relationships use observational data to make predictions, the latter is very important. Regression is an integer or floating-point estimate of true value by estimating the statistical relationships between dependent variables and independent variables of a data set.

Regression is an integer or floating-point value prediction of the real-value is made by predicting statistical relationships between the dependent variables and the independent variables of a data set.

The proposed regression model consists of four input parameters.  $x_1$  represents the urban environment (0) or highway (1),  $x_2$  represents the vehicle density,  $x_3$  represents the data transmission power, and  $x_4$  represents the transmission speed. In the expression

$$y = f(x_1, x_2, x_3, x_4) \quad (1)$$

$f(.)$  is a heuristic function to be obtained by regression, and  $y$  represents the packet loss rate (PLR) to be achieved defined as,

$$PLR = \frac{TLM}{TLM + TRM} \quad (2)$$

PLR was estimated using 18 different regression methods described in next subsections. For urban and highway scenarios, 10 different vehicle densities (100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 vehicles/ $km^2$ ) at 4 different transmission power levels (20mW, 40mW, 80mW, 120mW) and 7 different data rates (3, 6, 9, 12, 18, 24, 27Mbps) simulations were run. All the results from the vehicles and 2 RSU units were collected and the total number of lost and received packets were obtained. Packet loss rate (PLR) is expressed as the ratio the sum of the total number of lost messages (TLM) and the total number of messages received (TRM).

In Vanet, messages sent by vehicle shown in Figure 12 may not reach one, more than one vehicle, or zero vehicle. This is because vehicles are constantly moving and their distances from each other are constantly changing. For this reason, the messages that received the vehicles and were lost were taken into account. Similar formulas have been defined in other studies in the literature. (Odelstav, 2021)

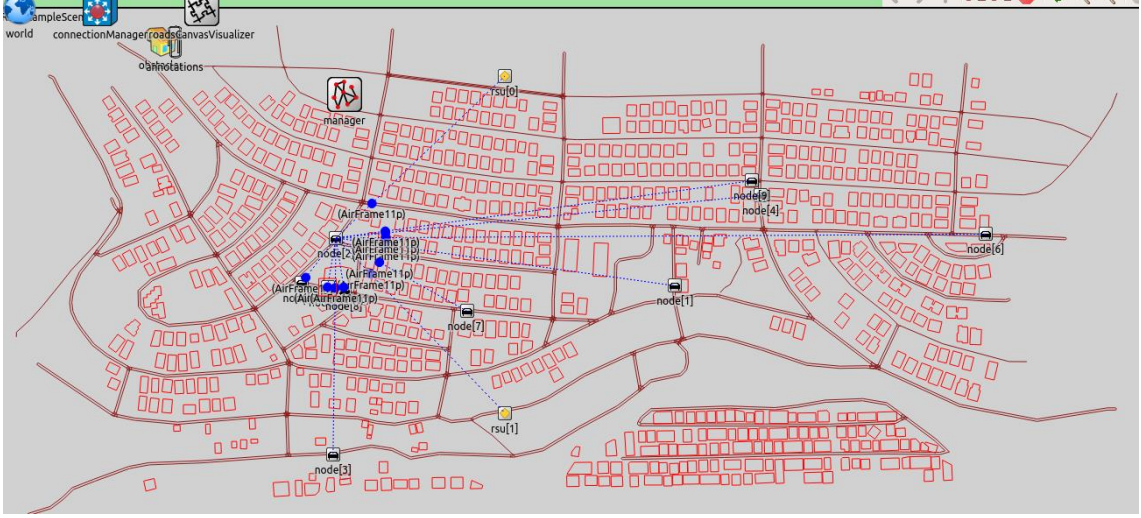


**Figure 8** Vanet Signal Propagation

### 2.2.1 Linear Regression

Linear Regression, which is commonly used in the literature, makes predictions by supposing that the connection between input and output variables is linear.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon$$

where  $x_i$  is the explanatory variable and  $y_i$  is the dependent variable. The slope of the line is  $\beta_p$ , and  $\beta_0$  is the intercept  $\epsilon$  model's error term.

### 2.2.2 Bayesian Ridge Regression

Bayesian regression formulates linear regression using probability distributors instead of making point predict, thus eliminating poorly distributed and insufficient datas. The model that predicts the regression problem probabilistically is called Bayesian Ridge regression.

Regularization parameters can be included in the estimation procedure using Bayesian regression techniques: the regularization parameter is not specified in a hard sense, but rather adapted to the data at hand.

This can be accomplished by applying uninformative priors to the model's hyperparameters. The  $l_2$  regularization used in Ridge regression and classification is identical to finding the best a posteriori estimate over the coefficients  $w$  with  $\lambda^{-1}$

precision under a Gaussian prior. Rather of explicitly setting lambda, it is possible to treat it as a random variable that may be calculated from data.

The output  $y$  is considered to be Gaussian distributed around  $Xw$  to achieve a fully probabilistic model:

$$p(y|X, w, a) = \mathcal{N}(y|Xw, a)$$

As mentioned above, Bayesian Ridge estimates a probabilistic model of the regression issue. A spherical Gaussian provides the prior for the coefficient  $w$ :

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}, \mathbf{I}_p)$$

Gamma distributions, the conjugate prior for Gaussian precision, are chosen as priors over  $a$  and  $\lambda$ . Bayesian Ridge Regression is the name given to the resulting model, which is similar to the classic Ridge.

### 2.2.3 Kernel Ridge Regression

Kernel ridge regression is a type of regression that uses the kernel Hilbert space function for nonlinear cases of ridge regression.

$(X_1, Y_1), \dots, (X_n, Y_n)$ :  $X$  arbitrary data,  $Y \in \mathbf{R}$ .

Kernelization of ridge regression: positive definite kernel  $k$  for  $X$

$$\min_{f \in H} \sum_{i=1}^n |Y_i - f(X_i)|^2 + \lambda \|f\|_H^2$$

### 2.2.4 Automatic Relevance Determination Regression

Regression Automatic Relevance Determination is a regression method of bayesian-based model weights according to the Gaussian distribution.

ARD Regression is similar to Bayesian Ridge Regression, although it can result in more sparse coefficients  $w$ . ARDRegression has a different prior  $w$  over since it does not assume that the Gaussian is spherical. Instead, an axis-parallel, elliptical Gaussian distribution is assumed for the distribution  $w$  over. This means that each coefficient  $w_i$  is selected from a Gaussian distribution with a precision  $\lambda_i$  and is centered on zero:

$$p(w|\lambda) = \mathcal{N}(w|0, A^{-1})$$

With  $\text{diag}(A) = \lambda = \{\lambda_1, \dots, \lambda_p\}$

Each coordinate of  $w_i$  has its own standard deviation  $\lambda_i$ , unlike Bayesian Ridge Regression. The same gamma distribution produced by hyperparameters  $\lambda_1$  and  $\lambda_2$  is chosen as the prior over all  $\lambda_i$ .

### 2.2.5 K-Nearest Neighbors Regression

K-Nearest Neighbors (KNN) regression is a non-periodic regression model that groups variables in a data set with similar properties.

Training values  $\{x_i, y_i\}$ ,  $x_i$  attribute-value representation of values,  $y_i$  real-valued target. Compute distance  $D(x, x_i)$  to every training value  $x_i$ , select  $k$  closest instances  $x_{i1} \dots x_{ik}$  and output the mean of  $y_{i1} \dots y_{ik}$  :

$$\hat{y} = f(x) = \frac{1}{k} \sum_{j=1}^k y_{ij},$$

### 2.2.6 Bagging Regression

Bagging regression is a regression method that makes predictions by dividing the dataset into random subsets and looking at the fit of the subsets to the basic regression. It is more beneficial to use it in datasets where there is high variance.

Given a training set  $D = \{(x_1, y_1), \dots (x_n, y_n)\}$ , sample  $T$  sets of  $n$  element form  $D$  (with replacement)  $D_1, D_2, \dots D_T \rightarrow T$  quasi replica training sets;

Train a machine on each  $D_i, i = 1, \dots, T$  and obtain a sequence of  $T$  outputs  $f_1(x), \dots f_T(x)$ .

$$\bar{f}(x) = \sum_{i=1}^T f_i(x),$$

The average of  $f_i$  of for  $i = 1, \dots, T$ .

### 2.2.7 Support Vector Regression

The support vector regression draws boundary lines to the samples in the dataset and makes a prediction by determining the hyperplane line to best fit the dataset.

Training dataset of  $n$  points of the form

$$(x_1, y_1), \dots, (x_n, y_n),$$

Where the  $y_i$  are either 1 or -1, indicating which class the point  $x_i$  belongs to. A  $p$ -dimensional real vector is represented by each  $x_i$ . Identify the "maximum-margin hyperplane" that splits the group of points  $x_i$  where  $y_i = 1$  from the group of points where  $y_i = -1$ , and is defined so that the distance between the hyperplane and the nearest point  $x_i$  from either group is maximized.

Any hyperplane can be expressed as a set of  $x$  points that satisfy certain conditions

$$w^T x - b = 0$$

Where  $w$  is the normal vector to the hyperplane (not necessarily normalized). This is similar to Hesse normal form, only  $w$  isn't always a unit vector. The offset of the hyperplane from the origin along the normal vector  $\frac{b}{\|w\|}$  is determined by the parameter  $w$ .

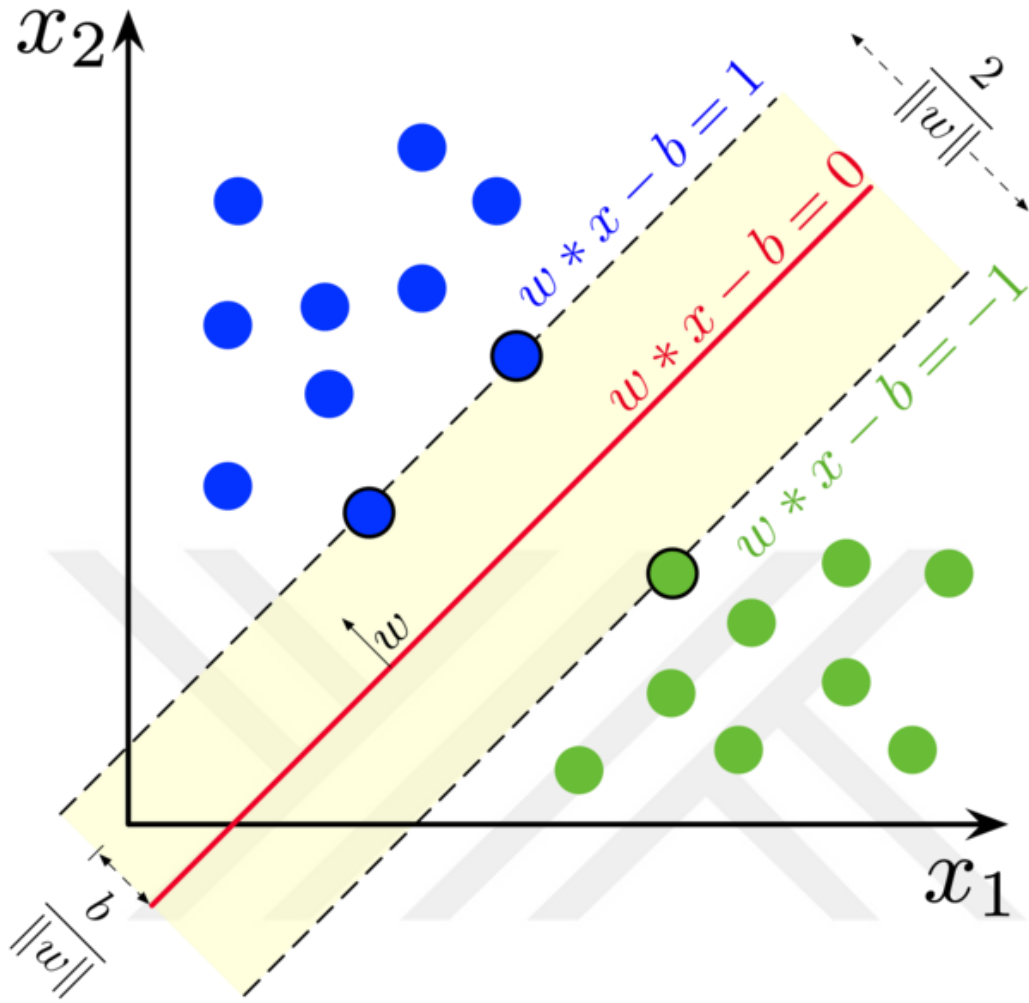


**Figure 9** Support Vector Machine (Noyum, Mofenjoui, Feudjio, Göktug, & Fokoue, 2021)

### 2.2.8 NU Support Vector Regression

The difference between nu support vector regression and support vector regression use nu parameter to control the number of support vectors.

### 2.2.9 Gaussian Process Regression

The main feature of the Gaussian process regression is that it is non-periodic, so instead of calculating the probability distribution parameters of a particular function, it calculates the probability distribution over all acceptable functions suitable for the dataset.

Input:  $X$  (inputs),  $y$  (targets),  $k$  (covariance function),  $\sigma_n^2$  (noise level),  $x_*$  (test input)

$$L := \text{cholesky}(K + \sigma_n^2 I)$$

$$\mathbf{a} := L^T / (L/\mathbf{y})$$

$$\bar{f}_* := \mathbf{k}_*^T \mathbf{a}$$

$$\mathbf{v} := L/\mathbf{k}_*$$

$$\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T \mathbf{v}$$

$$\log p(\mathbf{y}|X) := -\frac{1}{2} \mathbf{y}^T \mathbf{a} - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$$

**return:**  $\bar{f}_*$ (mean),  $\mathbb{V}[f_*]$ (variance),  $\log p(\mathbf{y}|X)$  log marginal likelihood

### 2.2.10 Decision Tree Regression

One of the most widely used and useful models for supervised learning is the Decision Tree. Decision trees are among the most used supervised learning methods. The most important reason for this is that regression and classification are methods that give very good results in both application areas. As the name suggests, the tree structure-based method includes three types of nodes. the first root node represents the first node and symbolizes the raw sample. Internal nodes represent data collection. On the other hand, Leaf nodes, solve the decision problem of the developed solution method.

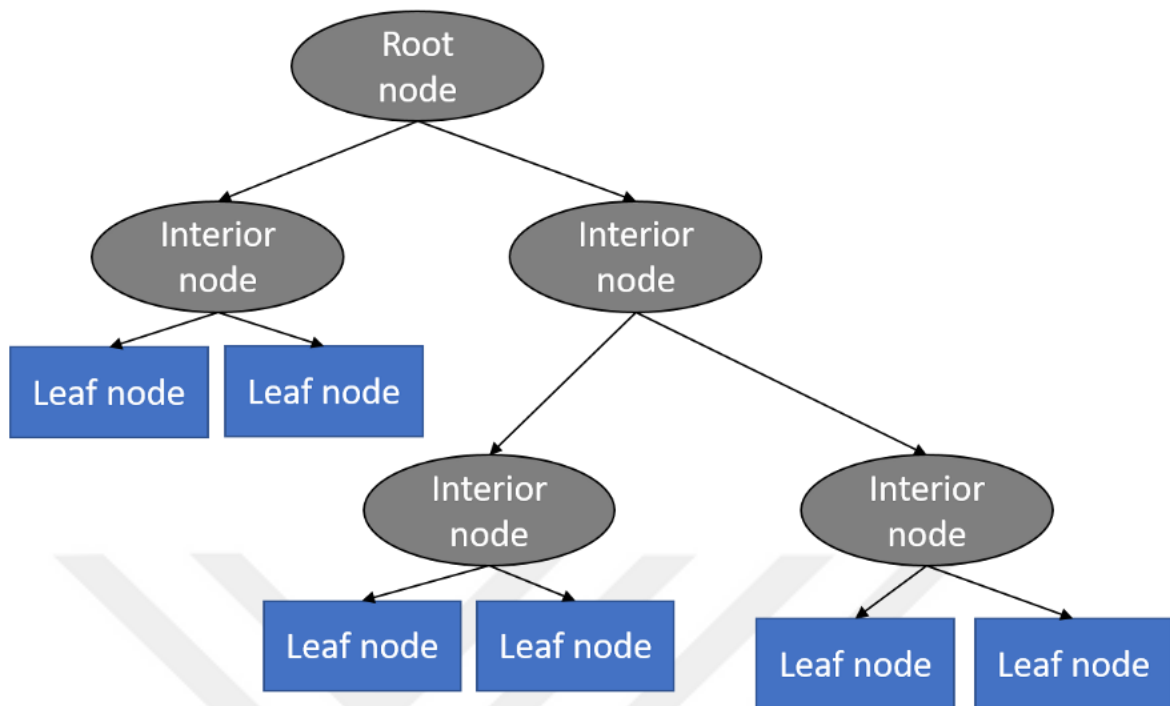


**Figure 10** Decision Tree Algorithm (*Machine Learning Basics: Decision Tree Regression, 2020*)

#### 2.2.11 Extra Decision Tree Regression

The extra word in the extra trees algorithm concept does not prune the decision trees produced from the data set. Predictions are made by averaging the generated decision trees.

#### 2.2.12 Gradient Boosting Regression

The boosting gradient algorithm works sequentially on an ensemble basis. Combining weak learners to increase prediction accuracy. gradient boosting algorithm shown in Figure 10.

---

**Algorithm 1** Friedman's Gradient Boost algorithm

---

**Inputs:**

- input data  $(x, y)_{i=1}^N$
- number of iterations  $M$
- choice of the loss-function  $\Psi(y, f)$
- choice of the base-learner model  $h(x, \theta)$

**Algorithm:**

- 1: initialize  $\hat{f}_0$  with a constant
  - 2: **for**  $t = 1$  to  $M$  **do**
  - 3:   compute the negative gradient  $g_t(x)$
  - 4:   fit a new base-learner function  $h(x, \theta_t)$
  - 5:   find the best gradient descent step-size  $\rho_t$ :  
$$\rho_t = \arg \min_{\rho} \sum_{i=1}^N \Psi [y_i, \hat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)]$$
  - 6:   update the function estimate:  
$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t)$$
  - 7: **end for**
- 

**Figure 11** Gradient Boosting Algorithm (*Natekin & Knoll, 2013*)

### 2.2.13 Histogram-Based Gradient Boosting Regression

This method optimizes if there are missing values in the dataset by scattering these values across trees, increasing the accuracy of the prediction.

### 2.2.14 Cat Boosting Regressor

The CatBoosting method is a combination of decision trees and gradient boosting. The gradient boosting method reduces errors by trying the decision trees sequentially and learning from the mistakes of the previous trees. This process continues until the selected loss function reaches the minimum point it is able to reach.

Catboost algorithm:

Ordered boosting

```

Input :  $\{(x_k, y_k)\}_{k=1}^n, I;$ 
 $\sigma \leftarrow$  random permutation  $[1, n];$ 
 $M_i \leftarrow 0$  for  $i = 1..n;$ 
for  $t \leftarrow 1$  to  $I$  do
    for  $i \leftarrow 1$  to  $n$  do
         $r_i \leftarrow y_i - M_{\sigma(i)-1}(x_i);$ 
    for  $i \leftarrow 1$  to  $n$  do
         $\Delta M \leftarrow \text{LearnModel}((x_j, r_j): \sigma(j) \leq i);$ 
         $M_i \leftarrow M_i + \Delta M;$ 
return  $M_n$ 

```

### 2.2.15 LightGBM Regressor

LightGBM is one of the gradient Boosting versions. It reduces computational complexity by using discrete variables. It differs from other boosting methods by using leaf-wise in the learning of decision trees.

**The LightGBM algorithm**

**Input:** Training data  $D = \{(\chi_1, y_1), (\chi_2, y_2), \dots, (\chi_N, y_N)\}$ ,  $\chi_i \in \mathcal{X}$ ,  $\mathcal{X} \subseteq \mathbb{R}$ ,  $y_i \in \{-1, +1\}$ ; loss function:  $L(y, \_(\chi))$ ; iterations:  $M$ ; sampling ratio of large gradient data:  $a$ ; sampling ratio of small gradient data:  $b$ ;

1. Merge mutually exclusive features (i.e. features never take nonzero values simultaneously) of  $\chi_i$ ,  $i = \{1, \dots, N\}$  by exclusive feature bundling (EFB) method;
2. Initialize  $\_0(\chi) = \arg \min_c \sum_i L(y_i, c)$ ;
3. **for**  $m = 1$  to  $M$  **do**
4. Compute absolute values of gradients:

$$r_i = \left| \frac{\partial L(y_i, \theta(x_i))}{\partial \theta(x_i)} \right|_{\partial \theta(x) = \theta_{m-1}(x)}, i = \{1, \dots, N\}$$

5. Resampled dataset by Gradient-based One-Side Sampling (GOSS) method:  
 $\text{topN} = a \times \text{len}(D)$ ;  $\text{randN} = b \times \text{len}(D)$ ;  
 $\text{sorted} = \text{GetSortedIndices}(\text{abs}(r))$ ;  
 $A = \text{sorted}[1 : \text{topN}]$ ;  $B = \text{RandomPick}(\text{sorted}[\text{topN} : \text{len}(D)], \text{randN})$ ;  
 $D_+ = A \cup B$ ;
6. Compute information gains:

$$Vj(d) = \frac{1}{n} \left( \frac{\left( \sum_{x_i \in A_l} r_i + \frac{1-a}{b} \sum_{x_i \in B_l} r_i \right)^2}{N_l^j(d)} + \frac{\left( \sum_{x_i \in A_r} r_i + \frac{1-a}{b} \sum_{x_i \in B_r} r_i \right)^2}{N_r^j(d)} \right)^{nt}$$

7. Get a new decision tree  $\theta_m(X)'$  on set  $D_+$
8. Update  $\theta_m(X) = \theta_{m-1}(X) + \theta_m(X)'$
9. **end for**
10. **return**  $\theta'_M(X) = \theta_M(X)$

**Figure 12** Light Gradient Boosting Machine Algorithm (Mohammed & Kadhem, 2021)

### 2.2.16 Extreme Gradient Boosting Regressor

Extreme Gradient Boosting Regressor is one of the optimized version of gradient boosting, it faster gradient boosting by pruning its decisions trees.

### **2.2.17 Stochastic Gradient Descent Regressor**

SGD stands for Stochastic Gradient Descent. SGD fit a linear model by minimizing a regularized empirical loss. The loss gradient is estimated each sample at a time, and the model is updated along the way using a decreasing strength schedule.



## Chapter 3 - IMPLEMENTATION

The Figure 13, we also present the block diagrams of the system set up for simulations. The tool for generating PLR datasets was built with Omnet++ software and Veins library for ad hoc network simulation. Two different scenarios were created, highway and urban.

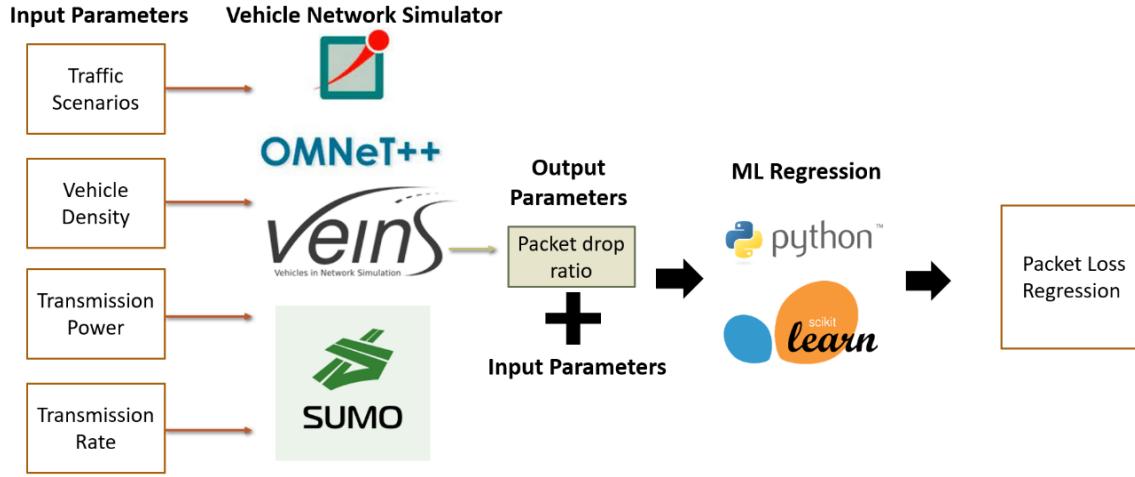


**Figure 13** System Model Block Diagram

Testing of newly developed protocols, usage scenarios, and wireless communication technologies in the real world is costly and time-consuming. So hard to perform repeatable wireless communication tests with a large number of Vanets dynamically moving. Vanets, simulation should have two pillars, network, and mobility. There are different simulators used in the literature as shown in Figure 2. This paper uses Omnet++, SUMO and Veins.

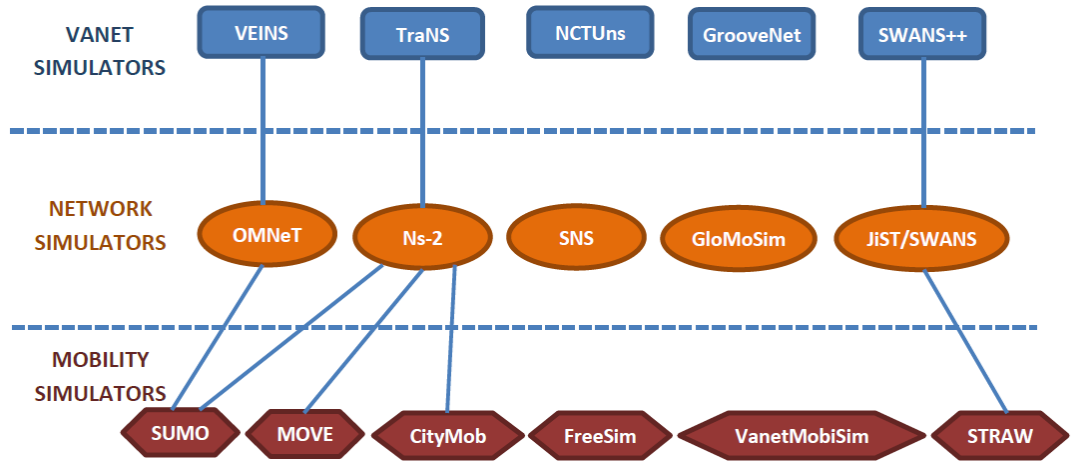


**Figure 14** VANETs Simulators (*Herrera, Igartua, & Iza, 2016*)

### 3.1 Ubuntu

Virtualbox (Oracle Corporation, 2007) was used as the virtual machine platform on windows. Ubuntu (Canonical, 2004) 18.04 version, which is an open source operating system, was used. After the ISO file was downloaded, Ubuntu 18.04 was installed by following the steps in the reference (O'Carroll, 2019).

### 3.2 Omnet++

Omnet++ is a platform where communication scenarios such as wired wireless networks, wireless temporary networks, Internet protocols, performance modeling, photonic networks, etc. allow simulated. Omnet++ has an Eclipse-based IDE and runtime graphical interface. Omnet++ is free for academic usage (Omnet++, 2006). Omnet++ consists of connections, messages, channels, collection classes, NED modules, parameters and omnetpp.ini configuration file. Omnet++ architecture as shown in Figure 14. Simulation variables are changed via omnetpp.ini and relationships between modules are defined in Ned files.

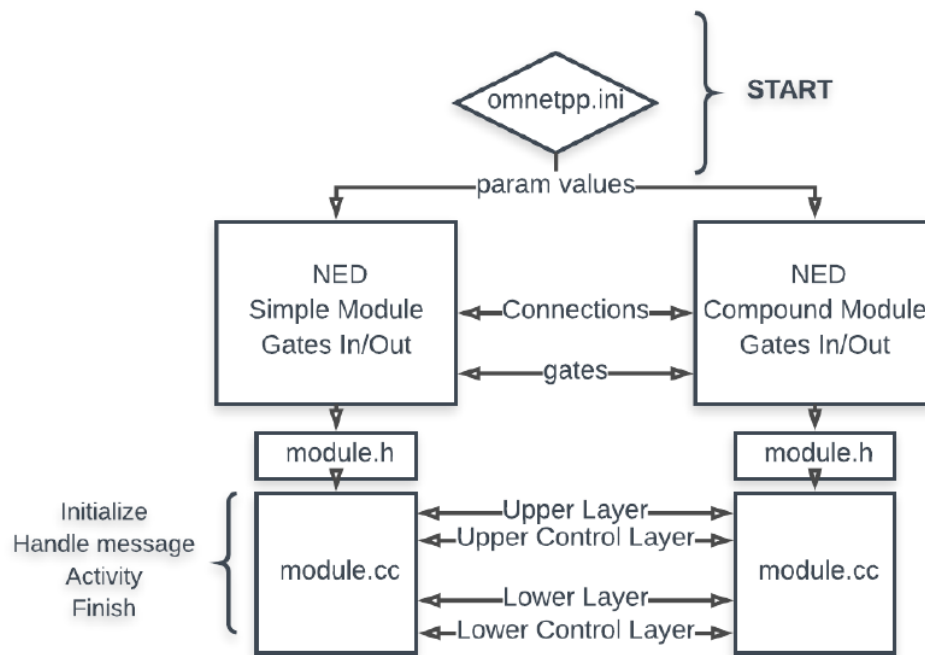


**Figure 15** Omnet++ Architecture ( *CORIA & A., 2018*)

### 3.3 Omnet++ Installation

After the Ubuntu installation is complete, a terminal window should be opened and the following lines of code should be executed.

```
$ tar xvfz omnetpp-5.6.2-src.tgz
$ cd omnetpp-5.6.2
$ . setenv
$ gedit ~/.bashrc
export PATH=$HOME/omnetpp-5.6.2/bin:$PATH
$ ./configure
$ make

After the above command lines are executed, the following command line is written and
the Omnet++ IDE is opened.

$ omnetpp
```

If all the installations are done correctly, after this command line, you should see the landing page with Figure 16.

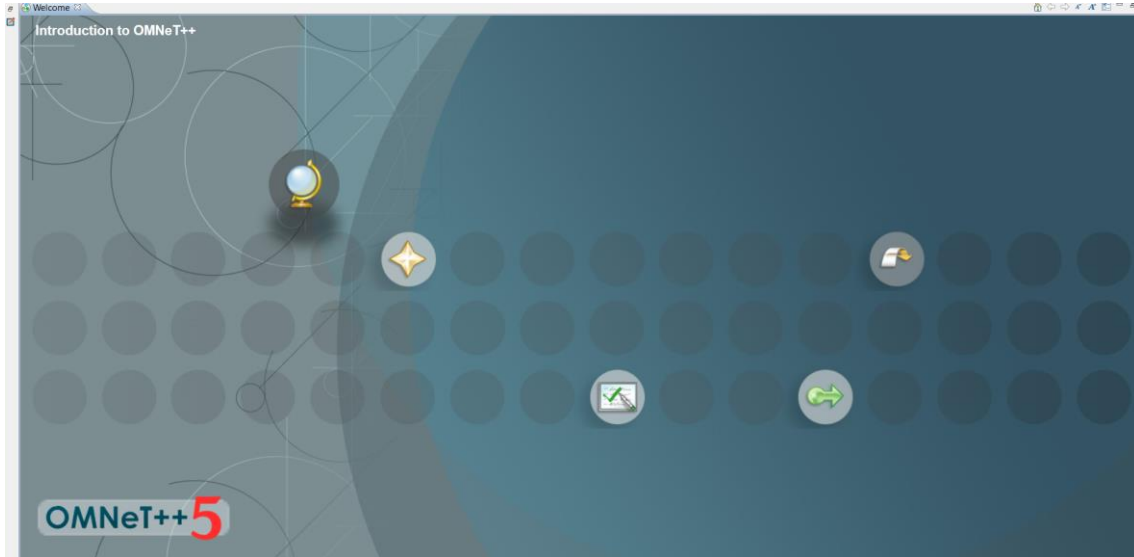


**Figure 16** Omnet++ Welcome Window

### 3.4 Inet

The Omnet++ simulation environment, the Inet framework is an open-source model library. For academics and students working with communication networks, it provides protocols, agents, and other models. Inet comes very handy when creating and validating new protocols, as well as when experimenting with novel or unusual settings.

### 3.5 Inet Installation

In this study, Inet 4.2.1 version was used by downloading from the reference (INET Framework, 2003). Then Inet was imported to Omnet++ and then built.

### 3.6 Veins

Veins is a library of Omnet++. Veins is an open-source vehicular network simulation framework. With veins, both simulators Omnet++ and Sumo run in parallel. Omnet++ and Sumo connect over the TCP socket. Vehicle movement information is obtained from Sumo via Omnet++ TraCI (Traffic Control Interface). Thus, the network and traffic

simulator can be run in parallel at the same time (Sommer, 2006). Veins architecture as shown in Figure 17.

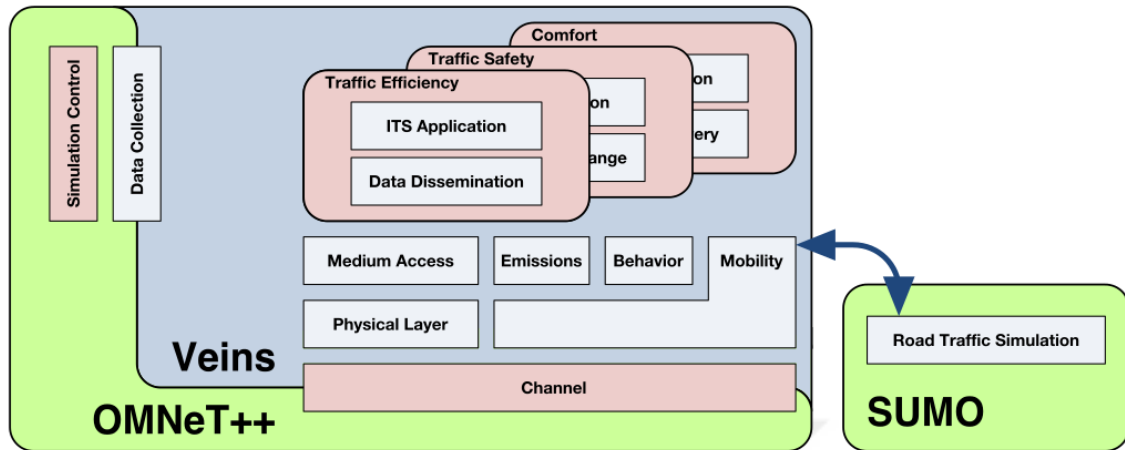


**Figure 17** Veins Architecture (Sommer, 2006)

### 3.7 Veins Installation

In this study, Veins 5.1 version was used by downloading from the reference (Veins, n.d.). Then Veins was imported to Omnet++ and then built.

### 3.8 Sumo

Sumo abbreviation means Simulation Urban Mobility. The Sumo logo and traffic environment example view are shown in the Figure 18. Sumo is an open-source traffic simulator developed by the Institute of transportation systems at the German Aerospace Center. Real-world maps, road, and traffic data can be imported from Open Street Maps into Sumo (OpenStreetMap, 2004). Areas such as residential, industrial, commercial, parks, and military bases can be defined as polygon types using the Open Street Maps tool and transferred to Sumo.



**Figure 18** Sumo Environment (*Behrisch & Hilbrich, 2001*)

### 3.9 Sumo Installation

The installation was done by following the command lines below. In this study, Sumo version 0.32.0 was used.

```
$sudo apt-get install sumo sumo-tools sumo-doc
```

```
$sudo add-apt-repository ppa:sumo/stable
```

```
$sudo apt-get update
```

```
$sudo apt-get install sumo sumo-tools sumo-doc
```

### 3.10 Simulation Run

After completing the installations, Omnet++ opens.

```
$ omnetpp
```

Omnet++ is opened with the above command line. After that, the following command line connect Sumo to Omnet++.

```
$ sudo ~/veins-master/sumo-launchd.py -v -c ~/sumo-0.32.0/bin/sumo-gui
```

Simulation parameters are defined in the omnetpp.ini file located in the veins/examples/veins/omnetpp.ini file. After editing the omnetpp.ini file according to your configuration, click the Run button in the IDE. After the simulation is completed, the results are saved as csv file with the following command line.

```
$ scavetool x *.sca -o highway2002000mW18Mbps.csv
```

### 3.11 Python

Python is an open source object-oriented, dynamically semantic, interpreted, software and programming language. Data structures along with filexible typing and dynamic linking make python ideal for Rapid Application Development and as an assembly language for combining already existing parts. Compared to other programming and software languages, Python is an easy-to-learn scripting language that prioritizes readability, which

reduces software maintenance costs. Python advocate program interchangeability and code reuse in other areas. (Rossum, 2001-2022).

### 3.12 Scikit-Learn

Scikit-learn is Python's machine learning-based library and is frequently used in literature and real-world applications. Scikit-learn is a Python software library that allows the use of machine learning algorithms based on different types of statistical or complex mathematical formulas. In addition, another important advantage of Scikit learn is that it is used free of charge and therefore there are many examples available on the internet. (Cournapeau, 2007).

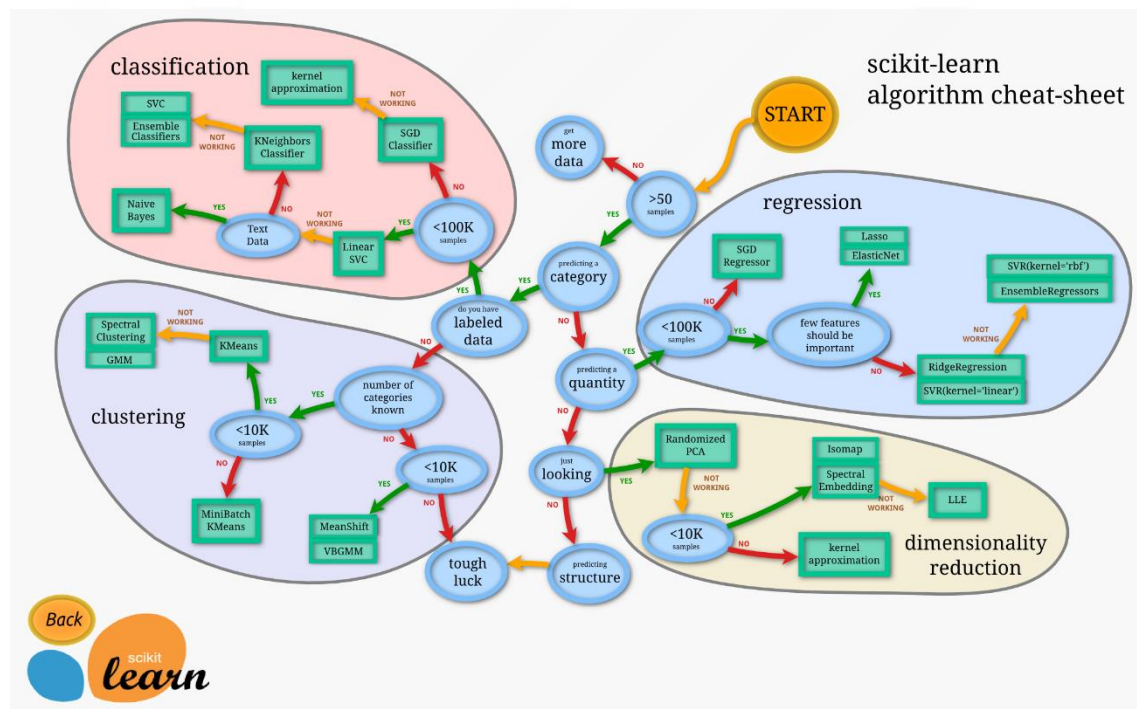


Figure 19 Scikit-learn Method Selection Guide (Cournapeau, 2007)

## Chapter 4 - NUMERICAL RESULTS

Two scenarios were created as highway and urban. The highway scenario, the region in Istanbul shown in Figure 20 below was selected from OpenStreetMap. The urban scenario, the region shown in Figure 21 below was selected from OpenStreetMap.



Figure 20 Highway Scenario

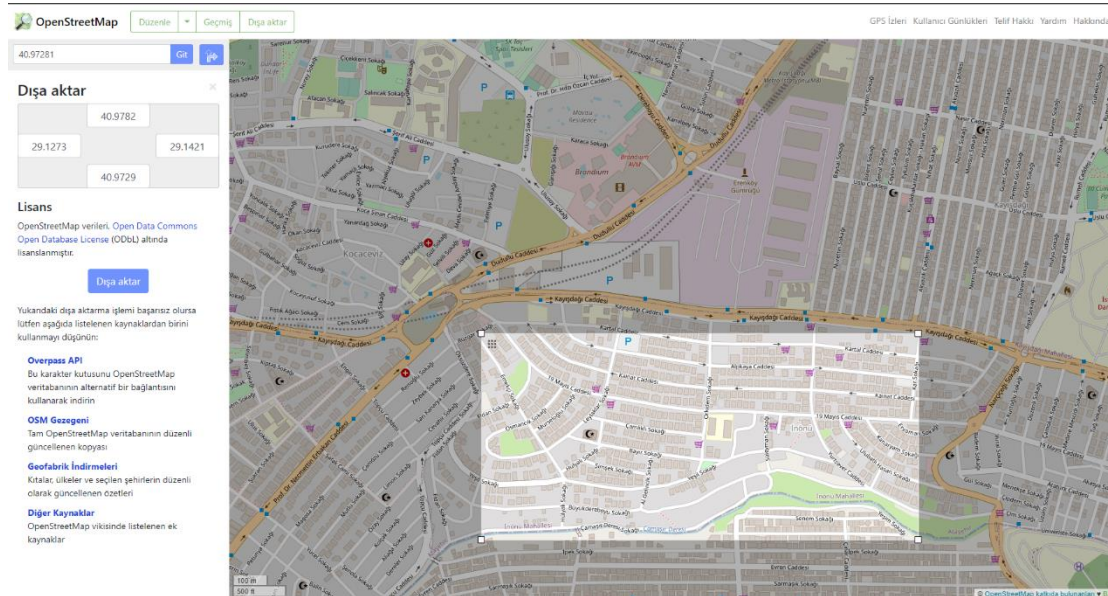


Figure 21 Urban Scenario

**Table 1** Simulation Parameter

| Parameters              | Specifications                                                     |
|-------------------------|--------------------------------------------------------------------|
| Protocol                | IEEE802.11p                                                        |
| Mac                     | IEEE802.11p/1609.4                                                 |
| Network simulator       | Omnet++, Veins                                                     |
| Traffic simulator       | Sumo                                                               |
| Frequency               | 5.9 GHz                                                            |
| Density of driving cars | 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 vehicles/ $km^2$ |
| Number of RSU           | 2                                                                  |
| Transmission power      | 20mW, 40mW, 80mW, 120mW                                            |
| Data rates              | 3, 6, 9, 12, 18, 24, 27Mbps                                        |
| Noise floor             | -98dBm                                                             |
| Scenario types          | Urban, highway                                                     |

For both scenarios (urban and highway), 10 different vehicle densities (100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 vehicles/ $km^2$ ) in 4 different transmission power levels (20mW, 40mW, 80mW, 120mW) and 7 different data rates (3, 6, 9, 12, 18, 24, 27Mbps) simulation was run. The dataset shown in Figure 22 was collected for OBU and RSUs with all the results obtained after simulations using Omnet++, Veins and Sumo, and TLM and TLR parameters and PLRs were obtained with Matlab software (Mathworks, 2020).

|    | A                  | B            | C            | D              | E         |
|----|--------------------|--------------|--------------|----------------|-----------|
| 1  | Packet loss ratio  | Area_encoded | Car Quantity | Transmit power | Data rate |
| 2  | 0.144439090552165  | 0            | 100          | 20             | 3         |
| 3  | 0.219622143103250  | 0            | 100          | 20             | 6         |
| 4  | 0.358410018024855  | 0            | 100          | 20             | 9         |
| 5  | 0.482651736897333  | 0            | 100          | 20             | 12        |
| 6  | 0.548879896786301  | 0            | 100          | 20             | 18        |
| 7  | 0.625624405328259  | 0            | 100          | 20             | 24        |
| 8  | 0.638444702519207  | 0            | 100          | 20             | 27        |
| 9  | 0.0725606785712980 | 0            | 100          | 40             | 3         |
| 10 | 0.147969493222703  | 0            | 100          | 40             | 6         |
| 11 | 0.213067160932801  | 0            | 100          | 40             | 9         |
| 12 | 0.387747345559846  | 0            | 100          | 40             | 12        |
| 13 | 0.484859408795963  | 0            | 100          | 40             | 18        |
| 14 | 0.587965413799188  | 0            | 100          | 40             | 24        |
| 15 | 0.604613765710221  | 0            | 100          | 40             | 27        |
| 16 | 0.0565632458552539 | 0            | 100          | 80             | 3         |
| 17 | 0.0745142761210627 | 0            | 100          | 80             | 6         |
| 18 | 0.142412824796502  | 0            | 100          | 80             | 9         |
| 19 | 0.238144823810083  | 0            | 100          | 80             | 12        |
| 20 | 0.392005236109764  | 0            | 100          | 80             | 18        |
| 21 | 0.526381695031021  | 0            | 100          | 80             | 24        |
| 22 | 0.548845275728837  | 0            | 100          | 80             | 27        |
| 23 | 0.0483006949297743 | 0            | 100          | 120            | 3         |
| 24 | 0.0626135097344674 | 0            | 100          | 120            | 6         |
| 25 | 0.0892091250367110 | 0            | 100          | 120            | 9         |
| 26 | 0.179979958814444  | 0            | 100          | 120            | 12        |
| 27 | 0.320435491187688  | 0            | 100          | 120            | 18        |
| 28 | 0.482296551724138  | 0            | 100          | 120            | 24        |
| 29 | 0.510109519797810  | 0            | 100          | 120            | 27        |
| 30 | 0.246594538072405  | 0            | 200          | 20             | 3         |
| 31 | 0.364344646719884  | 0            | 200          | 20             | 6         |
| 32 | 0.521735423216086  | 0            | 200          | 20             | 9         |
| 33 | 0.682142311657566  | 0            | 200          | 20             | 12        |
| 34 | 0.776044287871163  | 0            | 200          | 20             | 18        |
| 35 | 0.859543270262153  | 0            | 200          | 20             | 24        |
| 36 | 0.870916193181818  | 0            | 200          | 20             | 27        |
| 37 | 0.0966479215127754 | 0            | 200          | 40             | 3         |
| 38 | 0.245483229996504  | 0            | 200          | 40             | 6         |
| 39 | 0.358176008187440  | 0            | 200          | 40             | 9         |

**Figure 22** Simulation outputs dataset

Total loss and total received parameters were created by collecting all the results of the vehicles and 2 RSU units. For each vehicle quantity, 28 packet loss results were generated. Datasets were obtained from 280 results for both scenarios. Datasets from all scenarios were divided into 80% train 20% test and then exported to various machine learning algorithms. The results are shown in Table 2, Table 3, and Table 4. Table 3 is the results

of the dataset of 560 rows after combining both scenarios. RMS MAE shows the errors between test data and predicted data. R2\_score indicates how well the data set fits the machine learning algorithm, the value of 1 representing 100%.

**Table 2** Highway Only

| Metod                              | RMS   | MAE   | R2_score |
|------------------------------------|-------|-------|----------|
| LinearRegression                   | 0.067 | 0.054 | 0.925    |
| RidgeRegression                    | 0.07  | 0.058 | 0.92     |
| BayesianRidgeRegression            | 0.067 | 0.054 | 0.925    |
| KernelRidgeRegression              | 0.106 | 0.086 | 0.816    |
| AutomaticRelevanceDetermination    | 0.067 | 0.054 | 0.925    |
| KNeighborsRegressor                | 0.057 | 0.044 | 0.946    |
| BaggingRegressor                   | 0.044 | 0.032 | 0.967    |
| SupportVectorRegressor             | 0.063 | 0.054 | 0.935    |
| NuSupportVectorRegressor           | 0.045 | 0.031 | 0.966    |
| GaussianProcessRegressor           | 0.04  | 0.03  | 0.973    |
| DecisionTreeRegressor              | 0.056 | 0.036 | 0.947    |
| ExtraTreesRegressor                | 0.055 | 0.038 | 0.949    |
| GradientBoostingRegressor          | 0.029 | 0.024 | 0.986    |
| HistGradientBoostingRegressor      | 0.031 | 0.024 | 0.984    |
| CatBoostRegressor                  | 0.015 | 0.011 | 0.996    |
| LightGBMRegressor                  | 0.031 | 0.024 | 0.984    |
| XGBRegressor                       | 0.031 | 0.022 | 0.983    |
| StochasticGradientDescentRegressor | 0.139 | 0.125 | 0.682    |

**Table 3** Urban Only

| Metod                              | RMS   | MAE   | R2_score |
|------------------------------------|-------|-------|----------|
| LinearRegression                   | 0.076 | 0.061 | 0.884    |
| RidgeRegression                    | 0.078 | 0.063 | 0.877    |
| BayesianRidgeRegression            | 0.076 | 0.061 | 0.883    |
| KernelRidgeRegression              | 0.112 | 0.087 | 0.75     |
| AutomaticRelevanceDetermination    | 0.076 | 0.062 | 0.883    |
| KNeighborsRegressor                | 0.048 | 0.036 | 0.952    |
| BaggingRegressor                   | 0.039 | 0.018 | 0.968    |
| SupportVectorRegressor             | 0.052 | 0.042 | 0.944    |
| NuSupportVectorRegressor           | 0.036 | 0.024 | 0.973    |
| GaussianProcessRegressor           | 0.039 | 0.026 | 0.969    |
| DecisionTreeRegressor              | 0.030 | 0.014 | 0.981    |
| ExtraTreesRegressor                | 0.032 | 0.016 | 0.978    |
| GradientBoostingRegressor          | 0.037 | 0.018 | 0.972    |
| HistGradientBoostingRegressor      | 0.039 | 0.022 | 0.969    |
| CatBoostRegressor                  | 0.03  | 0.014 | 0.982    |
| LightGBMRegressor                  | 0.039 | 0.022 | 0.969    |
| XGBRegressor                       | 0.029 | 0.014 | 0.982    |
| StochasticGradientDescentRegressor | 0.151 | 0.12  | 0.547    |

**Table 4** Urban and Highway

| Metod                              | RMS   | MAE   | R2_score |
|------------------------------------|-------|-------|----------|
| LinearRegression                   | 0.077 | 0.063 | 0.902    |
| RidgeRegression                    | 0.077 | 0.064 | 0.901    |
| BayesianRidgeRegression            | 0.077 | 0.063 | 0.902    |
| KernelRidgeRegression              | 0.101 | 0.081 | 0.832    |
| AutomaticRelevanceDetermination    | 0.077 | 0.063 | 0.902    |
| KNeighborsRegressor                | 0.055 | 0.039 | 0.949    |
| BaggingRegressor                   | 0.041 | 0.027 | 0.971    |
| SupportVectorRegressor             | 0.064 | 0.053 | 0.931    |
| NuSupportVectorRegressor           | 0.047 | 0.034 | 0.962    |
| GaussianProcessRegressor           | 0.044 | 0.029 | 0.967    |
| DecisionTreeRegressor              | 0.056 | 0.037 | 0.948    |
| ExtraTreesRegressor                | 0.049 | 0.031 | 0.959    |
| GradientBoostingRegressor          | 0.037 | 0.028 | 0.976    |
| HistGradientBoostingRegressor      | 0.037 | 0.026 | 0.977    |
| CatBoostRegressor                  | 0.023 | 0.013 | 0.99     |
| LightGBMRegressor                  | 0.037 | 0.026 | 0.977    |
| XGBRegressor                       | 0.03  | 0.02  | 0.984    |
| StochasticGradientDescentRegressor | 0.118 | 0.098 | 0.769    |

The results of the simulation results of both scenarios where the number of vehicles is 100-500-1000 are shown in Figure 23 and Figure 24. In the highway scenario, increase in transmission power reduces packet loss, but increase in transmission power does not reduce the packet loss in the urban scenarios. Because in the highway scenario, the distance between vehicles is large and when transmission power increases, messages reach more vehicles. In the urban scenario, the distance between vehicles is shorter and when the transmission power increases, packet collisions occur in the air, so the packet loss increases.

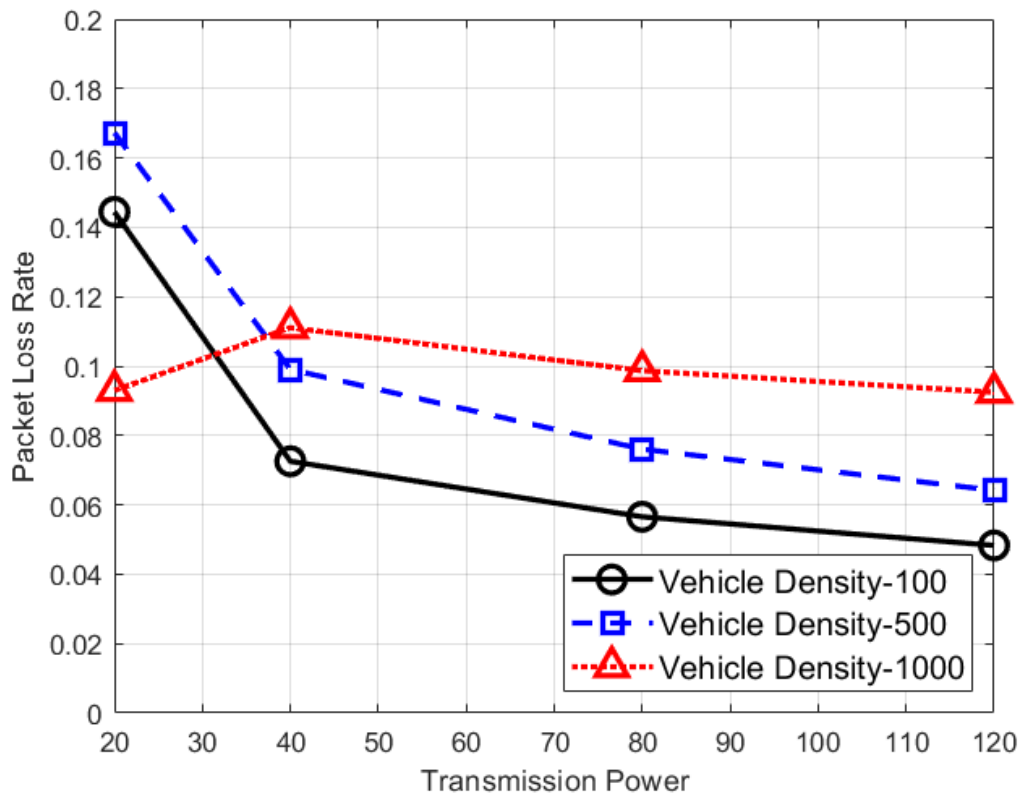


**Figure 23** PLR vs Transmission Power under Vehicle densities on Highway.

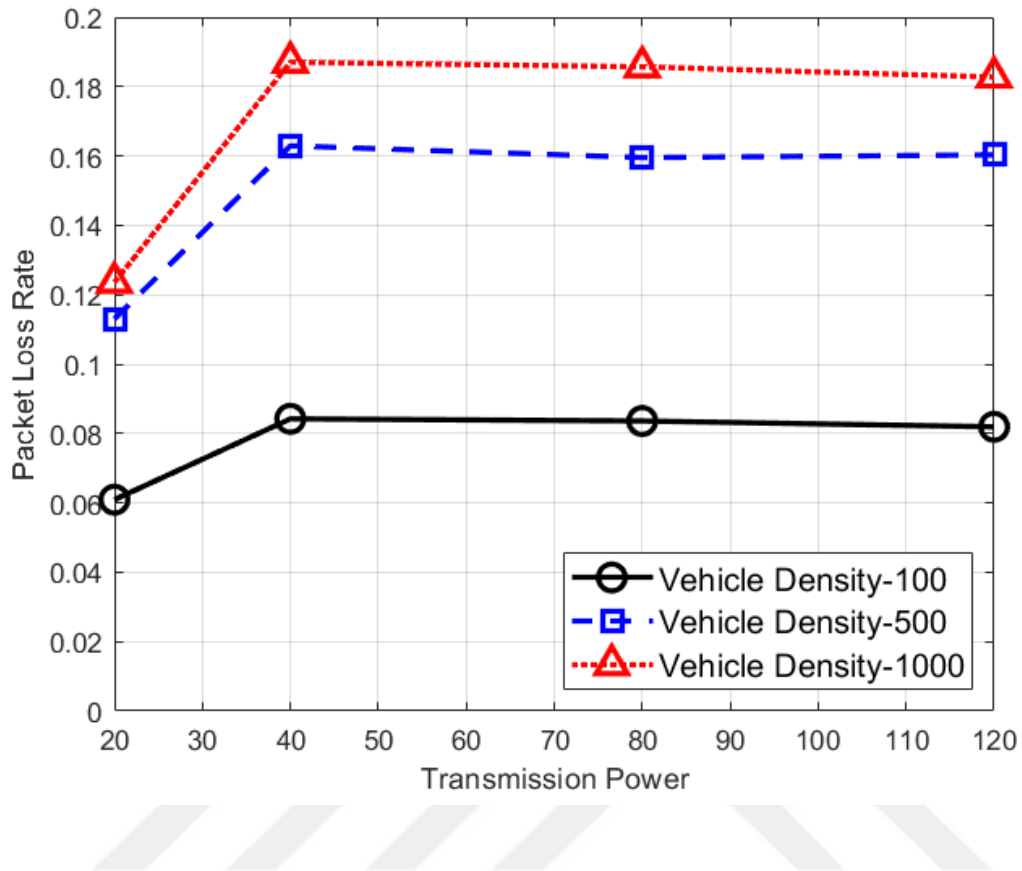


**Figure 24** PLR vs Transmission Power under Vehicle densities on Urban

Gradient boosted trees-based machine learning algorithms make predictions with lower error than linear regression algorithms as shown in Figure 25, Figure 26, Figure 27 and, Figure 28. Linear regression is a parametric model: it assumes the target variable can be expressed as a linear combination of the independent variables. Gradient boosted trees are nonparametric: they will approximate any function.

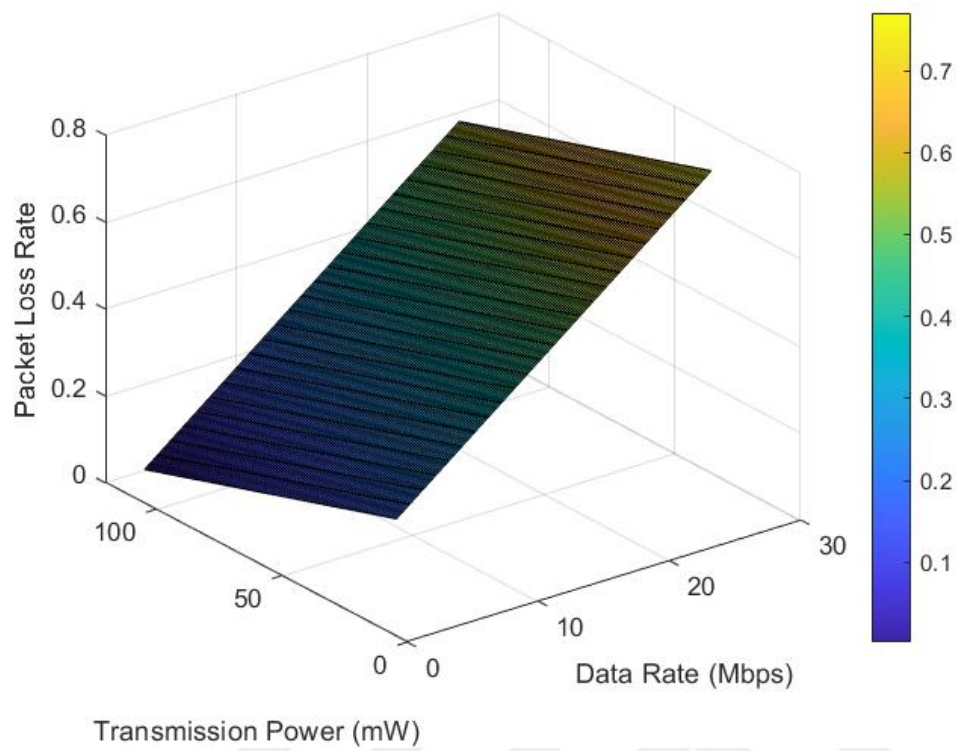


**Figure 25** Highway Linear Regression Prediction

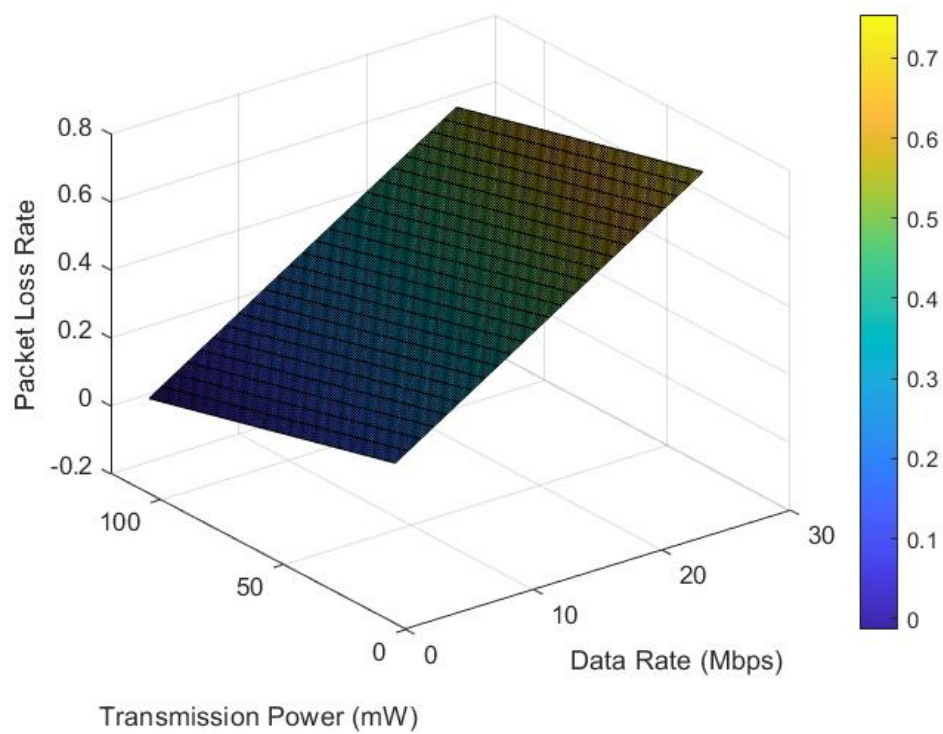


**Figure 26** Urban Linear Regression Prediction

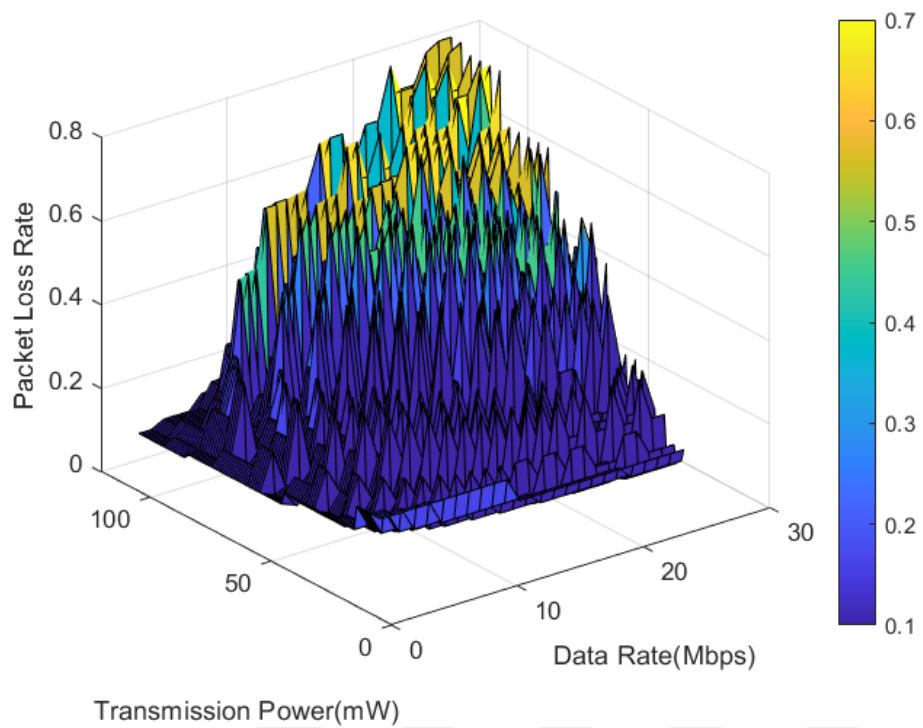


**Figure 27** Highway CatBoost Regression Prediction

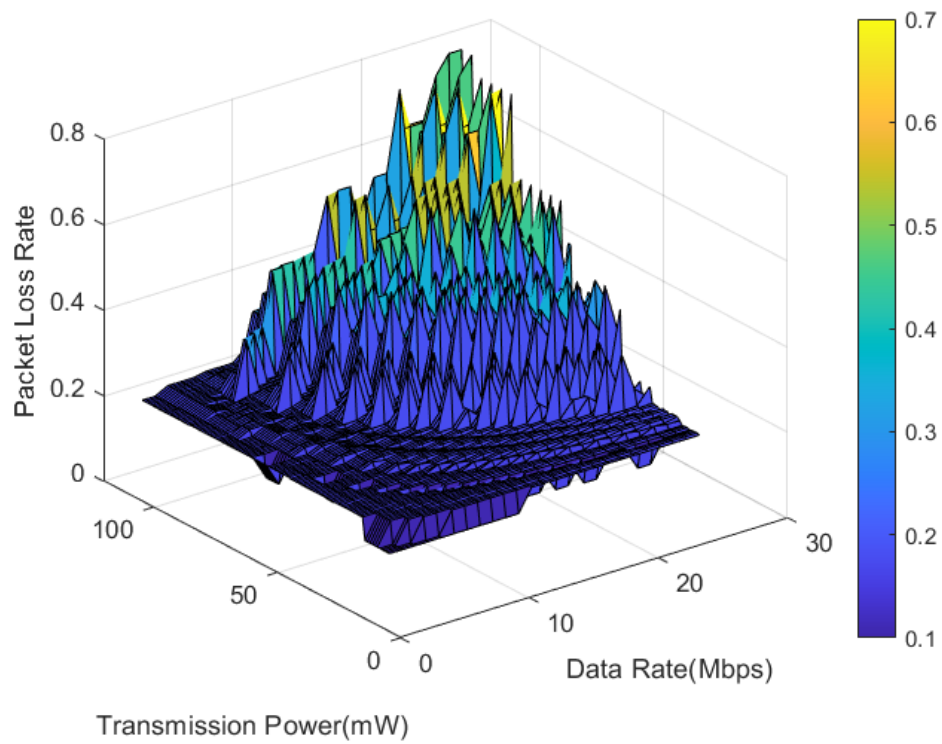


**Figure 28** Urban CatBoost Regression Prediction

Figure 29 and Figure 30 shows the packet loss rates estimated by the CatBoost method separately for highway and city, under highway and urban, 500 vehicles / km<sup>2</sup> vehicle density, transmission power between 20-120 mW and data rates between 3-27Mbps. On the color scale, dark navy blue indicates PLRs of 0.1 and below. In the highway scenario, the increase in transmission power reduces the PLR, while in the urban scenario, the increase in transmission power increases the PLR due to collisions. In the highway scenario, when the distance between the vehicles is large and the transmission power increases, the messages reach more vehicles, reducing the PLR. On the other hand, in the urban scenario, the distance between vehicles is shorter and as the transmission power increases, packet collisions occur in the environment, thus increasing the PLR. According to the results shown in Figure highway, it is possible to find more and different solutions in the highway scenario for PLR of 0.1 and below than in the urban scenario. On the other hand, in the urban scenario, results with a PLR of around 0.1 and below are seen in a much narrower range.

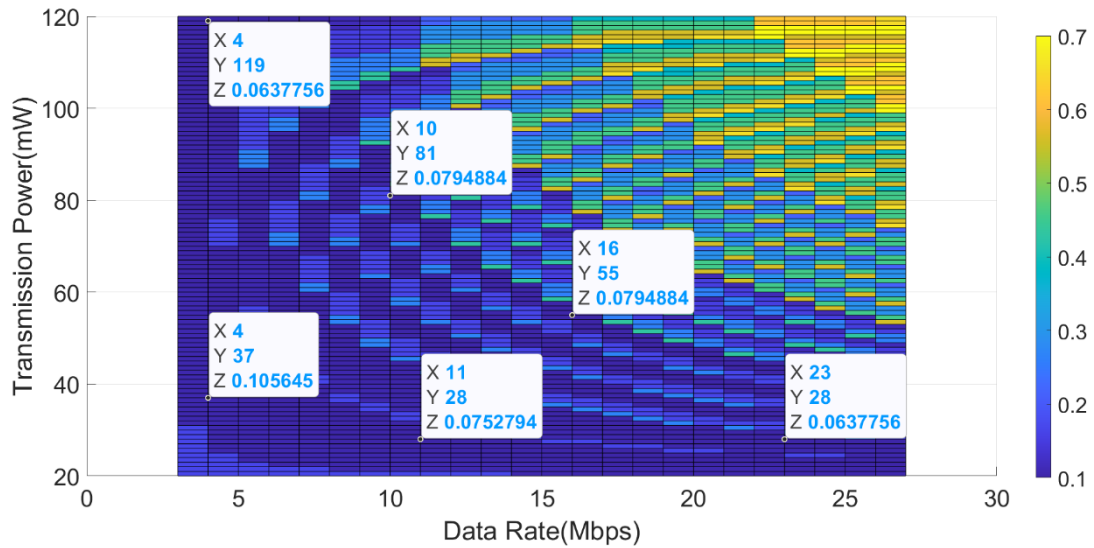


**Figure 29** Highway CatBoost Prediction Top View

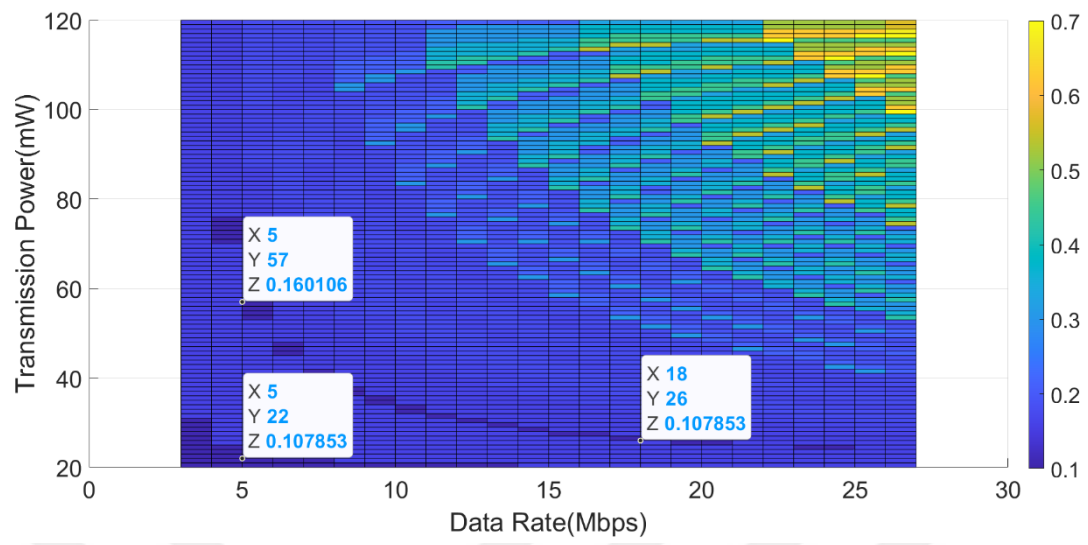


**Figure 30** Urban CatBoost Prediction Top View

## **Chapter 5 - CONCLUSIONS FUTURE RESEARCH DIRECTIONS**

In this study, we proposed a heuristic function that calculates the packet loss probability in the vehicle network using advanced regression methods when different vehicle density, transmission speed, and transmission power values were given. Numerical results showed that the Catboost regression method gave the least error between predicted and tested results compared to other regression methods. Thanks to the obtained heuristic function, when a set of transmission parameters is given, packet loss rates can be determined without simulation, and it may be possible to reduce the PLR below the target level by re-adjusting the transmission parameters. For example, in the highway scenario, since the vehicles are spread over a wide area, it is possible to see different transmission power and transmission speed results that give the probability of 0.1 packet loss, and the system administrator can choose one of these results according to the demand from the users. In the urban scenario, the transmission power and data rate solution, which gives 0.1 PLR, is extremely limited.

As a future work, the dataset can be expanded by changing the communication parameters, and more RSUs can be added to the urban scenario to further decrease the PLR and increase the number solution alternatives that can be considered by the network designer. Obtaining a mathematical expression that can give the PLR under certain conditions and constraints and comparing it with the proposed simulation environment will also take place in our future studies. In addition, simulation results can be compared with real tests on vehicles.

## REFERENCES

- CORIA, R., & A., S. (2018). *ANALYSIS OF VANET STANDARD IEEE 1609.4 MAC LAYER MULTI-CHANNEL OPERATIONS USING OMNET++ AND VEINS*. Oklahoma.
- Feukeu, E. A., & Zuva, T. (2020). Dynamic Broadcast Storm Mitigation Approach for VANETs. *Future Generation Computer Systems*, 1097-1104.
- Herbiet, G.-J., Bouvry, P., & Guinand, F. (2011). Social relevance of topological communities in ad hoc communication networks. *2011 International Conference on Computational Aspects of Social Networks*, (s. 19-24).
- (2004). Retrieved from OpenStreetMap: <https://www.openstreetmap.org/>
- Ameixieira, C. E. (2017). *Mountain View (CA)/ United States Patent No. US009736791B1*.
- Atmaca, H. (2021, April 20). *Supervised Learning*. ai2know: <https://www.ai2know.com/supervised-learning--gozetimli-ogrenme> adresinden alındı
- Atmaca, H. (2021, April 29). *Unsupervised Learning*. Retrieved from ai2know: <https://www.ai2know.com/unsupervised-learning--denetimsiz-ogrenme>
- Babu, T. S., & Newton, P. (2016). A Refined Data Dissemination Technique to reduce Broadcast Storm Problem in Vehicular Ad-hoc Networks. *International Journal of Computer Science and Engineering Communications*, 61-70.
- Behrisch, M., & Hilbrich, R. (2001). *Sumo*. Sumo: <https://www.eclipse.org/sumo/> adresinden alındı
- Bong-Kyu Park, M.-J. T.-C.-S.-H. (2005). A new approach to on-board stationkeeping of GEO-satellites. *Aerospace Science and Technology*, 722-731.
- Burhan, M., & Rehman, R. A. (2020). BSMS: A Reliable Interest Forwarding Protocol for. *3rd International Conference on Advancements in Computational Sciences (ICACS)*, 1-6.
- Canonical*. (2004, October 20). *Ubuntu*: <https://ubuntu.com/> adresinden alındı
- Chao, C., & M., B. J. (1983). On the propagation and control of geosynchronous orbits. *Journal of the Astronautical Sciences*, 31, 99-115.
- Chitra, M., & Sathya, S. S. (2018). Selective epidemic broadcast algorithm to suppress broadcast storm in. *Egyptian Informatics Journal*, 1-9.
- Cournapeau, D. (2007, June). *Scikit-learn*. scikit-learn: <https://scikit-learn.org/stable/> adresinden alındı

- Curtis, D. (2010). *Orbital Mechanics for Engineering Students*. Daytona Beach Florida: Butterworth-Heinemann.
- Danlaw. (2022). *Danlaw's DSRC Conformance Tester Earns OmniAir Certification*. Danlaw: <https://www.danlawinc.com/danlaw-dsrc-conformance-tester-earns-omniair-certification/> adresinden alındı
- Deeksha, M., Patil, A., Kulkarni, M., Shekar, N., Shet, V., & Muthuchidambaranathan, P. (2022). Multistate Active Combined Power and Message/Data Rate Adaptive Decentralized Congestion Control Mechanisms for Vehicular Ad Hoc Networks. *Journal of Physics: Conference Series*.
- Emma, B. P., & Pernicka, H. J. (2003). Algorithm for Autonomous Longitude and Eccentricity Control for Geostationary Spacecraft. *Journal of guidance, control, and dynamics*, 26(3), 483-490.
- Farooq, M. U., & Khan, K. U. (2014). Mitigating Broadcast Storm Problems in Vanets. *International Journal of Simulation--Systems, Science & Technology*.
- Frederik J. de Bruijn, S. T. (2016). Geostationary Satellite Station-Keeping Using Convex Optimization. *Journal of Guidance, Control, and Dynamics*, Vol.39, No.3.
- Frigau, M. S. (2013). Cross-layer Transmit Power and Beacon Rate Adaptation for Vanets. *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, 129-136.
- Herrera, D. L., Igartua, M. A., & Iza, C. (2016). *Simulation of basic multi-hop broadcast techniques in vehicular Ad-Hoc networks using veins simulator*. Barcelona.
- INET Framework. (2003, November 17). Retrieved from <https://inet.omnetpp.org/Download.html>
- Intelligent Transportations Systems. (2020). [https://www.its.dot.gov/cv/\\_basics/cv/\\_basics/\\_how/\\_used.htm](https://www.its.dot.gov/cv/_basics/cv/_basics/_how/_used.htm) adresinden alındı
- Kamakshi, S., & Sriram, V. (2019). Plummeting broadcast storm problem in highways by clustering vehicles using dominating set and set cover. *Sensors* 19.9.
- Kamakshi, S., & Sriram, V. (2020). Modularity based mobility aware community detection algorithm for broadcast storm mitigation in VANETs. *Ad Hoc Networks* 104.
- Kashihara, S., Sahara, T., Kaneda, S., & Ohta, C. (2019). Rate adaptation mechanism with available data rate trimming and data rate information provision for V2I communications. *Mobile Information Systems*.

- Khan, Z. S., Moharram, M. M., Alaraj, A., & Azam, F. (2014). A Group Based Key Sharing and Management Algorithm for Vehicular Ad Hoc Networks. *Hindawi Publishing Corporation*.
- Kim, J.-W., Kim, J.-W., & Jeon, D.-K. (2018). A Cooperative Communication Protocol for QoS Provisioning in IEEE 802.11p/Wave Vehicular Networks. *Sensors*.
- Kothai, G., Poovammal, E., Dhiman, G., Ramana, K., Sharma, A., AlZain, M. A., . . . Masud, M. (2021). A New Hybrid Deep Learning Algorithm for Prediction of Wide Traffic Congestion in Smart Cities. *Hindawi Wireless Communications and Mobile Computing*, 13.
- Laberteaux, K., & Hartenstein, H. (2010). VANET : vehicular applications and inter-networking technologies.
- Liu, Q., & Wu, Y. (2012). SUPERVISED LEARNING. *Encyclopedia of the Sciences of Learning*.
- Losa, D. (2007). *High vs Low Thrust Station Keeping Maneuver Planning for Geostationary Satellites*. Paris: École Nationale Supérieure des Mines de Paris.
- M.Guelman, M. (2014). Geostationary satellites autonomous closed loop station keeping. *Acta Astronautica*, 9-15.
- Machine Learning Basics: Decision Tree Regression*. (2020, July 15). Towards Data Science: <https://towardsdatascience.com/machine-learning-basics-decision-tree-regression-1d73ea003fda> adresinden alındı
- Mahmoudi, I., Kamel, J., Ben-Jemaa, I., Kaiser, A., & Urien, P. (2019). Towards a Reliable Machine Learning Based Global Misbehavior Detection in C-ITS: Model Evaluation Approach. *International Workshop on Vehicular Adhoc Networks for Smart Cities*.
- Markley, F., & Crassidis, J. (2014). Attitude Kinematics And Dynamics. *Fundamentals of Spacecraft Attitude Determination and Control* (s. 67-84). içinde New York: Springer.
- Mathworks*. (2020). Retrieved from Mathworks: <https://www.mathworks.com/>
- Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120-133.
- Mohammed, M. A., & Kadhem, S. (2021). Insider Attacker Detection Using Light Gradient Boosting Machine. *Tech-Knowledge*, 67-76.
- Montenegro, J., Iza, C., & Igartua, M. A. (2020). Detection of Position Falsification Attacks in VANETs Applying Trust Model and Machine Learning. *Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. 2020. , 9-16.

- Najafzadeh, S., Ithnin, N. B., & Razak, S. A. (2014). Broadcasting in Connected and Fragmented Vehicular. *International Journal of Vehicular Technology*.
- Natekin, A., & Knoll, A. (2013). Gradient Boosting Machines, A Tutorial. *Frontiers in neurorobotic*, 7-21.
- Noyum, V. D., Mofenjou, Y. P., Feudjio, C., Göktug, A., & Fokoue, E. (2021). Boosting the Predictive Accuracy of Singer Identification Using Discrete Wavelet Transform For Feature Extraction. *arXiv e-prints*.
- O'Carroll, B. (2019, August 15). *CODEBOTS*. Retrieved from Ubuntu 18.04 Virtual Machine Setup: <https://codebots.com/docs/ubuntu-18-04-virtual-machine-setup>
- Odelstav, A. (2021). *Evaluation of Simulated 802.11p and LTE Communication at Road Intersections and Urban Area of Self Driving Cars*.
- Omnet++. (2006). Retrieved from <https://omnetpp.org/>
- Oracle Corporation. (2007, January 17). *Virtualbox*. Retrieved from Virtualbox: <https://www.virtualbox.org/>
- Özgören, M. K. (2007). Kinematic Analysis of Spatial Mechanical Systems Using Exponential Rotation Matrices. *Journal of Mechanical Design*, 129, 1144-1152.
- Rossum, G. v. (2001-2022). *Python*. Python: <https://www.python.org/> adresinden alındı
- Satpute, S., & Emami, M. R. (2019). Concurrent Station Keeping and Momentum Management of Geostationary Satellites. *The Journal of the Astronautical Sciences*, 66, 341-360.
- Shiddharthy, R., & Gunavathib, R. (2021). A Selective reliable communication to reduce broadcasting for cluster based VANET. *Turkish Journal of Computer and Mathematics Education*, 4450-4457.
- Shrivastava, S. K. (1978). Orbital Perturbations and Stationkeeping of Communication Satellites. *American Institute of Aeronautics and Astronautics*, 15(2), 67-78.
- So, S., Sharma, P., & Petit, J. (2018). Integrating plausibility checks and machine learning for misbehavior detection in VANET. *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. (s. 564-571). IEEE.
- Sommer, C. (2006). *Veins*. Retrieved from <https://veins.car2x.org>
- Taherkhani, N., & Pierre, S. (2016). Centralized and Localized Data Congestion Control Strategy for Vehicular Ad Hoc Networks Using a Machine Learning Clustering Algorithm. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 3275-3285.

- Thomas, D. (2016). *A Comparison of GEO Satellites Using Chemical and Electric Propulsion*.  
www.colorado.edu. adresinden alindi
- Veins. (n.d.). Retrieved from <https://veins.car2x.org/download/>
- Wang, M., Chen, T., Du, F., Wang, J., Yin, G., & Zhang, Y. (2022). Research on adaptive beacon message transmission power in VANETs. *Journal of Ambient Intelligence and Humanized Computing*, 1307-1319.
- Weiss, A., Kalabic, U. V., & Cairano, S. D. (2018). Station keeping and momentum management of low-thrust satellites using MPC. *Aerospace Science and Technology*, 76, 229-241.
- Wisitpongphan, N., TONGUZ, O., PARIKH, J., MUDALIGE, P., BAI, F., & SADEKAR, V. (2007). Broadcast storm mitigation techniques in vehicular ad hoc networks. *IEEE Wireless Communications* 14.6, 84-94.
- Xu, Q., Segupta, R., Jiang, D., & Chrysler, C. (2003). Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum. *The 57th IEEE Semiannual Vehicular Technology Conference*, 2451-2455.
- Yao, Y., Xiao, B., Yang, G., Hu, Y., Wang, L., & Zhou, X. (2019). Power Control Identification: A Novel Sybil Attack. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 2588-2601.
- ZHANG, C., CHEN, K., ZENG, X., & XUE, X. (2018). *Misbehavior Detection Based on Support Vector Machine and Dempster-Shafer Theory of Evidence in VANETs*. IEEE Access 6.
- Zhao, H., Cheng, H., Mao, T., & He, C. (2019). Research on Traffic Accident Prediction Model Based on Convolutional Neural Networks in VANET. *2nd International Conference on Artificial Intelligence and Big Data* (s. 79-88). IEEE.

## CURRICULUM VITAE

**Name Surname** : Osman Nuri KOÇ

### **Education:**

#### **Bachelor of Science**

2012-2017    Erciyes University  
Faculty of Engineering  
Electric-Electronic Engineering

### **Publications:**

**Conference Presentations:** *Packet Drop Rate Prediction for Vehicular Networks with Regression Methods.* **Osman Nuri KOÇ and Engin MAŞAZADE, SIU, 2022.**