People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

A/Mira University of Bejaia
Faculty of Exact Sciences
Computer Science Department

MASTER RESEARCH PAPER

In
Computer science

Option
Artificial Intelligence

Theme

Proposal for a new trust management system in intelligent transport systems

made by:

M^r Abdelkader CHETTAB & M^s Yasmine CHETTAB

Supervised by:

M^r Sofiane AISSANI & M^r Sidali BELHOCINE

the jury is made up of:

M^r Mouloud ATMANI
M^{rs} Samira BOUKERRAM

U. A/M Bejaia, october 2021.

Acknowledgements

First of all, we would like to thank our dearest parents, siblings and friends who have stood by us and shown us an endless amount of support and encouragement.

We would like to express our deepest gratitude to our supervisor Mr. Sofiane AISSANI for accepting to supervise our work and providing us proper guidance and encouragement throughout this research as well as thank him for his patience and understanding.

We would also like to express our appreciation for co-supervisor Mr. Sidali BELHOCINE for taking time to review our work and give us thoughtful comments and recommendations during the writing of this paper.

Lastly, we thank the members of the jury for accepting to examine and evaluate our work.

Contents

List of Figures

List of Tables

General introduction

Transport has become an indispensable tool in our everyday life. But like any invention, cars came with their own set of flaws and drawbacks, two of them that we can mention are urban traffic congestion and car accidents, which count both human and material losses. The causes for car accidents are very diverse, they can be due to human carelessness, engine failure, weather conditions, roads condition, and many other external factors. Either of them has an equally high risk, and therefore should be managed.

To improve the situation, men tried putting into place a better system to regulate and control car traffic. But regardless of all the traffic laws, the panels and the work forces like policeman that were set, they were incapable of improving the quality of the traffic and assuring effectively the security and safety of both pedestrians and drivers by reducing the risks of accidents to a reasonable number (the main objective is reducing it to zero).

And now, due to the growing number of cars in the streets, more than ever, it has become a priority to improve the quality of traffic and the security of the drivers and pedestrians. With those two big problems in hand, men thought about creating self-driven cars by incorporating new technologies in them using artificial intelligence and internet of things for better decision making. By doing this, vehicles would have new features that would help them drive alone by scanning the surroundings and sharing information with other vehicles in order to coordinate the traffic more efficiently.

But this technology is still not on point, there can be faulty transmissions, bad weather reduces the sensor's accuracy, the processing demands of data keep increasing and face trouble to keep up with it in real time, the AI still can't understand all the variant obstacles that it encounters. With this said, we see that the intelligent transport systems (ITS) still have a long way to go, which is why the purpose of this paper is to introduce a new trust management system based on Machine learning that can help us determine whether or not an ITS signal is trustworthy or not.

Our work is going to be in three chapters:

- Theoretical prerequisites (ITS, ML, TM).
- State of art of trust management.
- Proposition of our method.

Chapter 1

Theoretical prerequisites

1.1 Intelligent Transportation System

1.1.1 Definition

Intelligent Transportation system (ITS) is an emerging transportation system that has been up since the mid-60s but that kept evolving through time, at first they took form as driver assistance system and automated driving systems supervised by the driver (semi-autonomous) but now they're making use of the advanced technologies that development and research has brought us, such as advanced sensors, information processing, communication and automatic control systems as a way to sense and communicate important and up-to date information regarding the surrounding we humans can't see or acquire when we're on the road, with the purpose of giving us a real-time, accurate and efficient transportation management system, that serves as a safety precautions to navigate and avoid any collision or traffic that could delay us any moment [1].

1.1.2 Evolution

ITS has gone through many transformations before reaching the level it is now on. During the past versions, and their evolution, it went through different levels of autonomy and changes. It first started with a one way communication, with the driver getting some safety assistance from the car, then the car being able to independently do some functions on its own, before it slowly grew capable to drive by itself completely, and two way communication was necessary for that, it all evolved with the help of interactive system operations and management and information networks and devices [2].

1.1.3 Deployment

The concept of problem solving ITS, has reached favorable agreement all over the world, many countries are either working on preparation of the ground for the deployment or already working on the implementation itself. Some developed countries have already deployed them and are continuously keeping up the research and development work, we can mention: United States, United Kingdom, Europe, Japan, South Korea, Singapore and Australia.

ITS models and applications differ from country to country, due to the various differences in politics, environmental structures and issues, as well as development priorities that each country upholds [3].

1.1.4 Working process

To be able to manage traffic and locate vehicles, there are a few important steps:

a) Data collection

ITS systems gather data about the transport system by observation and monitoring traffic in real time using the help of multiple devices (like sensors, camera etc.), then directly send all that information to the Traffic Management Centre (TMC) that stores all that data [4].

b) Data processing

The data collected (travel time, speed, delay, accidents on roads, conditions etc.) will be processed, verified, and transformed into a format that is useful for the operators [4].

c) Data analysis

The processed data will go through error rectification, data cleaning and synthesis [4].

d) Data transmission

Transmit the analyzed data back to the vehicle and driver with the help of a variety of electronic devices (like radio, internet, SMS, automated cell) to help provide and update him with quality information on which he can base his future route decisions on [4].

1.1.5 ITS Applications

Each application holds an important role, the Authors of [2] mention:

1. Electronic Toll Collection (ETC)

Is a fast short range system, placed in the vehicle and on the road ,That electronically collects the fee charged for using toll infrastructure (roads, bridges, tunnels) ,which helps better manage traffic flow, reduce air pollution and noise.

2. Highway Data Collection (HDC)

Sensors placed on the road, who collect and send data to traffic control centers, who then determine traffic density, and travel time, as well as weather change, with the collection of wind velocity data and the help of weather stations.

3. Traffic Management Systems (TMS)

A central control system that plans, manages and optimizes transportation, it consists of many systems: Event Management System (EMS), Traffic Control System (TCS), Traveler Information System (TIS), and Video Control System (VCS).

4. Vehicle Data Collection (VDC)

Collects data from vehicles, digital images and sensors about the quality of the road and highway, to determine rough roads, areas of high rutting, and to assist in monitoring ride quality.

5. Transit Signal Priority (TSP)

Change the transit time of traffic signals by reducing the time of the red light or prolonging the time of the green one. It's cheap and helps make transit services faster and reliable.

6. Emergency Vehicle Preemption (EVP)

Works with radio communication and GPS, eCall informs motorists of any traffic or accident ahead, with alternative recommendation routes, the nearest emergency service point is immediately located and contacted when an incident occurs, and the operator is informed about the incident (time, location, vehicle identification) and help is sent to the scene if required, other drivers are then alerted and notified of the incident .

1.1.6 ITS Technologies

Enabling technologies in intelligent transportation that improve transportation conditions, safety and services, the Authors of [2] also mention:

1. Wireless communications

Secure and fast wireless communication over short or long distance is made possible using protocols and networks like IEEE 802.11, Wireless Access for Vehicular Environment (WAVE), Dedicated Short Range Communication (DSRC) for short distance and IEEE 802.16, Worldwide Interoperability for Microwave Access (WIMAX), Third Generation (3G) and Global System for Mobile communication (GSM) for long range respectfully.

2. Computational technologies

Includes model-based process control, ubiquitous computing, artificial intelligence, and applications that use real time operating systems, costly microprocessors and hardware embedded in the vehicles for automatic running.

3. Sensing technologies

Wireless real-time and long distance sensors, installed or embedded in or surrounding the road, that are capable of detecting vehicles, speed and direction through radar and acoustics sensors and pedestrians through pressure pads.

4. Floating car data/floating cellular data

Mobile phones or GPS act as real time data collection sensors, which help locate vehicles, their direction, speed and time using triangulation. After being sent and processed from traffic centers, useful information is extracted and distributed to other drivers.

5. Inductive loop detection

Multiple loops of wire are placed under the road and connected to a control box, it helps detect vehicles, count their number, speed and sometimes even determine their class, by noticing a change in the induction (magnetic or electric change) of the wires.

6. Video vehicle detection

A weather sensitive surveillance cameras placed on the roadside or traffic light poles, which records vehicles, and analyzed through an image processor that extracts and provides information like traffic density, speed, and detect incidents.

7. Bluetooth detection

Accurate and cheap short distance communication device, that is able to calculate traffic and vehicle density, time, and supply number plate recognition.

1.2 Machine Learning

1.2.1 Definition

Machine Learning (ML) is a branch of artificial intelligence, that deals with prediction making algorithm, just as its name indicates, ML is all about how machines learn from data like humans do with experience, without being fully pre-coded and instructed on what to do, with all the knowledge acquired with these data, a predictive model is built from it, so that whenever it's confronted with new data it's capable to adapt and predict the outcome, learn from past experiences and improve with it over time as it keeps interacting with more and more data, thus lowering the errors, developing into a better model with more accuracy and higher performance rate.

1.2.2 Type of ML

There are different ways that a machines can learn in ML, but the two main categories are:

1) Supervised learning

The purpose of the algorithms in this category is to learn to predict an output when given an input, but before that, they are first trained with a known set of labeled input and expected output pairs before they're to face new and unknown data, After training it with a set of labeled examples, overtime It picks up on the pattern between the inputs and outputs and learns which feature is associated with which label, and is from then on capable of predicting it on its own, that's how we get a prediction model, afterwards if you have any new input data, let the model predict the outcome for you [5].

But those prediction models are of 2 types, each with its own purpose and requirements:

The first one is Classification, where it's predicting or classifying discrete values into categories (could be binary or more), mostly used for things like fraud and spam email detection, image classification etc.

Algorithms : among some algorithms of this type we have: K-Nearest Neighbors, Support Vector Machines, Naïve Bayes, logistic regression etc.

And second one is Regression, where it's predicting continuous or real values, mostly used for risk assessment, score prediction, pricing etc.

Algorithms : some of the algorithms of this type are: Linear and polynomial regression, Support Vector Regression, Decision Trees and random forests etc [6].

2) Unsupervised learning

The purpose of the algorithms in this category is to learn and create models from unlabeled and unclassified training datasets, the algorithm is trained with an unknown set of unlabeled input data and is expected to be able to scans through it and overtime identify patterns, similar features, that can be used to group them into subsets, form clusters and classify each data into the one they belong to. Or predict which cluster which data belongs to [5].

But the pattern and recognition models also vary between two main types:

The first one is Clustering, where it splits the dataset into groups based on similarity, in order to reveal the structure of their structure. mostly used for targeted marketing, biology etc.

Algorithms : among some algorithms of this type we have: K means, fuzzy means, k-medoids ,hierarchical clustering etc.

And second one is Dimensionality reduction, where it's Reducing the number of variables in a dataset, going from a high-dimensional space into a low-dimensional space, in order to solve or clear computational concerns, mostly used for big data visualization, text mining, face and image recognition etc.

Algorithms : some of the algorithms of this type are: Principal component analysis (PCA), Kernel PCA (KPCA), Factor Analysis, Autoencoder etc [7].

1.2.3 Life Cycle

In machine learning, models follow a particular life cycle, it counts two major steps:

1) The data preparation part

1.a) Data collection : First we collect a huge amount of raw data from lots of different sources, those data contain imperfections and errors.

1.b) Processing : We proceed to clean out the entire dataset of any flaw like inconsistent, duplicate, biased, or missing information, keeping only real unbiased and reliable information.

And Feature engineering: Will extract more information from the processed data, and transform them features, that'll help procure more relevant information. The data is then split up into one set for training and another one for evaluating purposes for the next part.

2) The model training and validating part

2.a) Learning phase : The data is trained according to one type of ML

2.b) Validation phase : Testing, evaluating the model's accuracy and validating its parameters based on the results obtained.

Performance improvement and optimization phase: Dealing with the problems that the models face, optimizing it, and reducing the learning time and size of the structure [8][9].

1.3 Trust Management

1.3.1 Definition

When two or more different distributed network entities interact, there's a fundamental but also critical question whether or not to trust the information exchanged, Trust Management (TM) is in charge of managing the trustworthiness of those entities, by verifying the source and the validity of the information, by taking into account the history of their failed and successful interactions, which could tell a lot about one entity's intentions, in ITS trust management holds a very important role in making safe automated decision for drivers. And in the case of a failed distinction of malicious information, it could lead to fatal consequences [10]. But how can it be that there is distrust in an information? Simply because data processing may take time or hackers can involve themselves in this network, by falsifying and tempering the information.

1.3.2 Categories

Trust Management can belong to different interaction patterns: between two humans (H2H), two machines (M2M) and Human with Machine (H2M).

1.3.3 Type of TMS

Varies depending on their dependency on trusted authorities, we got mainly two:

1.3.3.1 Centralized

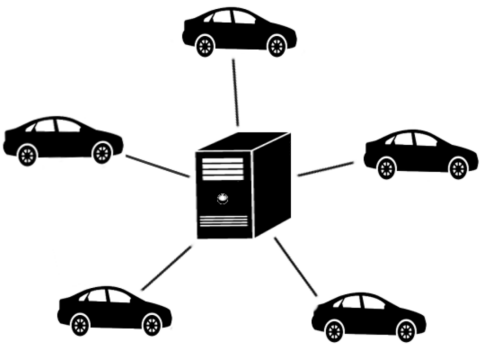

Figure 1.1: Centralized architecture.

Most common architecture used, It utilizes a mobile ad hoc network (MANET), and works by having one globally trusted server evaluating and rating every node's behavior in the network by giving out trusted credits which will later help determine whether or not to trust one node or not, and keep out falsified information from circulating around and misleading vehicles and drivers [11], But due to being limited to one central node, this system can't keep up with all demands and thus is not 100% safe and reliable.

1.3.3.2 Decentralized

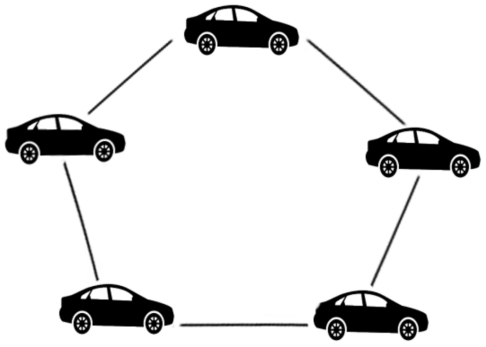

Figure 1.2: Decentralized architecture.

Has a non-centralized computing architecture and represent a system specific for VANET, VSN or IoT systems, it works as an open collaborative system and network instead, where anyone is a client but can also take the role of a server and provide services and resources instead of one global server that provides them for everyone, it evaluates trust just like a central server would have done, and keeps track of transactions with feedback. This is possible due the presence of a blockchain, which is a digital transaction record, that gets frequently updated and is distributed and available on many computers, so that any node is capable of verifying and validating the trust credits (also from past transactions) of other nodes independently and prevent any alteration and hacking done to the data [11].

1.3.3.3 Hybrid

It has both centralized and decentralized servers as their service providers. Clients can choose which server they want their service provided from [12].

1.4 Conclusion

Through this chapter, we've introduced the generalities and a few important aspects of Intelligent transport systems, machine learning and trust management.

Chapter 2

State of art of trust management

2.1 Introduction

Advancement in technologies not only extended our path but also brought us more tools and options to work with, the same way research is trying to keep up with the needs using these options, the security needs to step up keep up to par with them, as threats will always be there and be an obstacle and a challenge. The network where smart vehicles operate is very wide and large, and with internet's involvement security breaches are often found and exploited, Trust management checks out faults in the system and protects nodes and their connections, different techniques and methods are adopted, in order to achieve accurate trust results, for the reduction or overcoming of this problem. In this chapter we will study the different methods of trust management systems based on Machine learning, we will also analyze their differences and make a comparison between them depending on some chosen parameters.

2.2 Classification

In a machine learning based trust management model, you can find different type of methods used, with different type of characteristics, each of them has their own ups and downs, their own range of application, limits and so on, and all these will have a direct influence on the model itself, that's why the choice of the methods is important. Throughout researches and advancement, numerous trust evaluations and management models have been proposed, most of them being either data centric, entity centric, or hybrid models, here are some of the existing models and techniques:

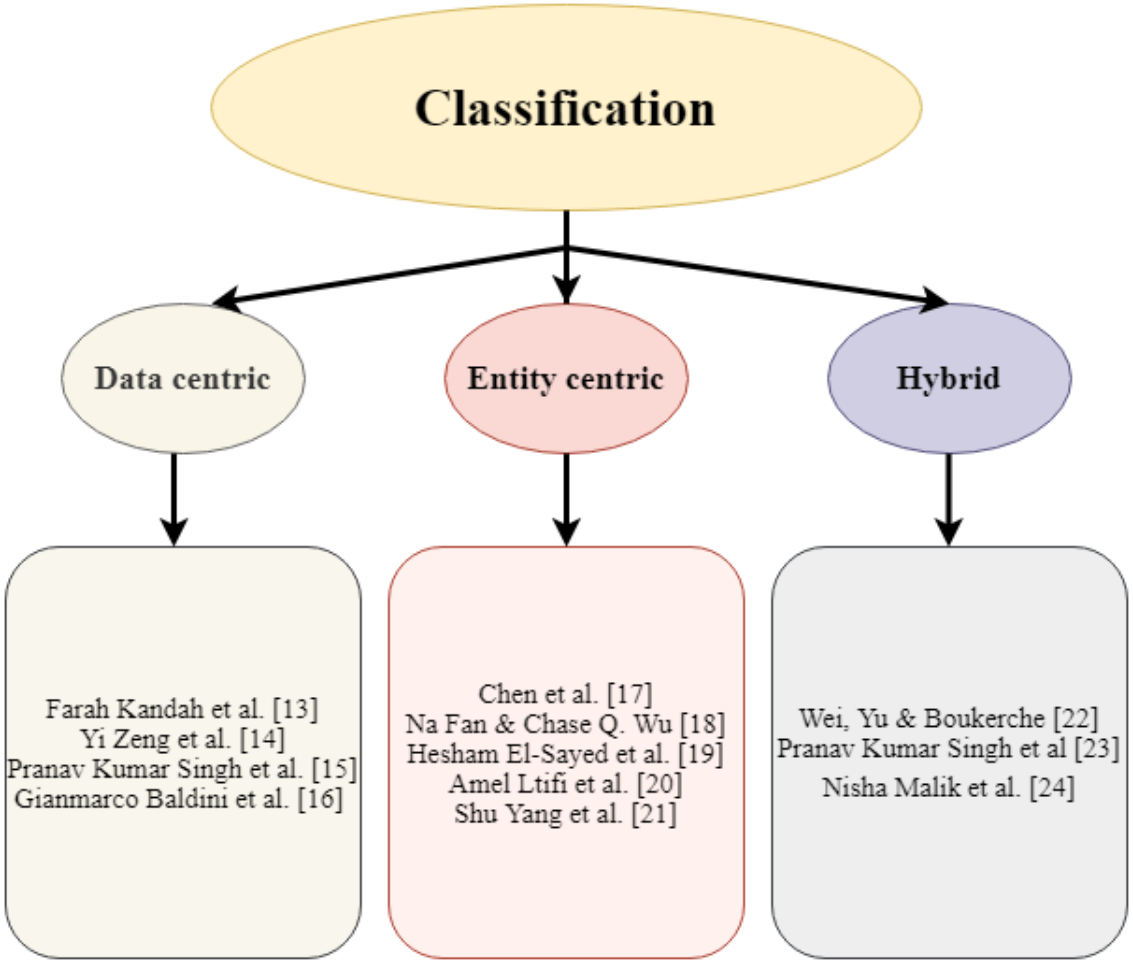

Figure 2.1: Classification of reviewed works.

2.2.1 Data centric

These models are based on data (message) for trust evaluation.

Blockchain-based Trust Management (BLAST)

Farah Kandah et al. proposed a multi-layered system, which consists of two levels of Blockchains, one at the vehicle level and another at the RSU level, vehicles will be grouped into platoons based on proximity and speed, inside each platoon neighboring Vehicles will start interacting with each other, analyze the messages and begin updating the Proof of Interaction (PoI), the platoon members will then choose a miner that will take care of creating, verifying and mining a block into the Platoon Blockchain (PB), each vehicle will then update the trust value of its neighboring vehicles according to the responses that were received. After generating a specified number of blocks in the platoon, the platoon members will elect a leader that will send the platoon blockchain to be mined into the global blockchain. RSUs will supervise a sector of the network, in which all present platoons will report and upon identifying the

platoons, the RSU will send the Genesis block (B0) to the vehicles, it includes the most up-to-date trust values from the global blockchain (GB) for each vehicle in the platoon, before adding any platoon on the global blockchain it will verify it to ensure that no incorrect and malicious behavior or data is placed onto the global blockchain, Meanwhile, each vehicle will start creating its own ID mask to hide its identity and generate new identity to be used in the next platoon, in order to not let malicious vehicles know or predict an ID and claim it for itself [13].

Deep learning intrusion detection

Yi Zeng et al. proposed a Deep Learning (DL) based end-to-end intrusion detection method, it uses Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models, Only requiring raw data, it helps examine traffic that go through the Vehicular Communication Module (DeepVCM), and automatically detect malware traffic for On-Board Units (OBUs). DeepVCM will firstly extract features from spacial range with the help of 1Dimensional CNN, the input data will be processed by the first convolutional layer and it results will be inputted into ReLU activation function who's going to be processed through the Max pooling, and at the end the Local Response Normalization (LRN) layer is added to punish abnormal responses or the outgoing layer for a better generalization effect. The output then passes through a second convolutional layer similar to the first. LSTM is applied in DeepVCM to learn characteristics from time related perspective, the input of LSTM will be the output of the CNN, the data will pass through a densely connected layer, and the softmax classifier then gets the output label at the end of the DeepVCM. In order to start the training process, we set the same hyperparameters for both first part and the second part of the DeepVCM and train them at the same time, after DeepVCM is trained, it will be automatically downloaded to OBUs and conduct malware traffic detection before traffic getting through the VCM [14].

Machine Learning Based Approach to Detect Position Falsification Attack in VANETs

Pranav Kumar Singh et al. proposed a method which focuses on detecting position falsification attacks using machine learning (ML) techniques like Support Vector Machines (SVM), Logistic Regression and feature description to detect the misbehavior. Using the VeReMi dataset, which consists of message logs for every vehicle in the simulation and a ground truth file that specifies the attacker's behavior to train and test our models, each time a message is sent it is also updated in the ground truth file which contains actual position/speed values and the attacker type. One-versus-rest binary classifiers are used to predict if the message is correct or false and by sampling a uniform distribution and comparing it to the attacker fraction parameter, Vehicles that have different position values than legitimate vehicles are classified as attackers, if an attacker sends some random position which is beyond the theoretical range of communication, then the receiver would be able to detect that [15].

Zone Keys Trust Management in Vehicular Networks based on Blockchain

Gianmarco Baldini et al. proposed an approach to improve Cooperative Intelligent Transport System (C-ITS) with the implementation of the revocation mechanism by using Blockchain and a Zone Key concept. Before entering a Roadside ITS Stations (RIS) zone or a jurisdiction which is composed of a number of RISs and a blockchain node who contains a copy of the transaction of misbehaving Vehicular ITS Stations(VISs) as well as all the enrolled C-ITS

stations, vehicles are entitled to enroll through the Enrollment Authority(EA) by sending an encrypted request for Authorization Tickets (ATs) through the RIS, the request is forwarded to the Authorization Authority(AA) to check the Revocation List before distributing a new AT. Whether it figures on the list will determine whether or not it gets the AT. The RIS collects all types of messages from VISs in the road and sends them to the Misbehavior Detection Service for analysis, if a misbehavior is detected, the blockchain records the information and distributes it to all the other blockchain nodes in the overall network, the jurisdiction revocation systems will then make a decision about whether to revoke that node and add it to the black list of revoked VISs or not [16].

2.2.2 Entity Centric

These models evaluate the trustworthiness level of an entity (node) and eliminate the dishonest ones.

Blockchain-based decentralized trust management system (DTMS)

Chen et al. proposed a framework which is managed by a consortium network and supported by a trusted execution environment (TEE), every trust evaluation round, the base station, collects and divides all the messages into groups and shares them with other base stations, the trust credits are calculated there by local nodes, who then share their ratings with the other stations, based on the quality of their contribution and their roles the global trust credit of each vehicle is calculated and temporally stored in the TEE-based storage, during that time both message senders and evaluators are encouraged by the incentive model with rewards (trust credits) or punishment depending on their behavior. A the end of the round the base stations then selects a group of trusted nodes as the validators to validate and agree upon both trust credits and reward values given out using the consensus model before uploading to the blockchain, more than half of the TEEs must agree to ensure that all the signatures are valid and validate a block and more than 2/3 of all the validators must agree to confirm a valid block [17].

On Trust Models for Communication Security in Vehicular Ad-hoc Networks

Na Fan and Chase Q. Wu proposed an integrated security scheme for communication and cooperation in VANETs, using a certain-factor (C-F) model and a fuzzy C-means clustering algorithm by dividing the messages into two clusters of true and false messages both direct and indirect reputation are measured, combined and updated at a regular interval in the communication history table. Upon the reception of a message from its neighbor, a node first checks its neighbor's combined reputation value; if it's over a threshold, it's forwarded and given a higher reputation value; else it's discarded and replaced with a lower reputation value. whenever a node behaves a cost in the form of energy consumption is included, the rewards are higher than the amount of energy consumption, so when a node cooperates and forwards a message it gets a payoff, the more a node adopts a cooperative behavior the more payoff it gets, if it doesn't it is punished in the next period by a loss greater than its payoff and will last one period of time. The RSU will store inside the history table the number of rewards and punishments for every node, and will update it at certain time intervals [18].

Machine learning based trust management framework for vehicular networks

Hesham El-Sayed et al. proposed a trust framework, where in order to calculate direct and recommended trust, various techniques are used, if a vehicle has special privileges, trust is calculated based on their roles and metrics like message forwarding/receiving time and distance between the nodes. But if it's an ordinary one, trust is calculated based on the recommendations of the neighboring nodes, using metrics like Euclidean Distance between Road Side Units (RSU)/message forwarding nodes and RSU/message receiving nodes. Decision tree classification (DT) is used to derive trust rules from the trust evaluations, whom are then used by recommending vehicular nodes (nodes with minimum distance from the RSUs and have good trust values in their communication history) in order to take appropriate decisions based on the trust rules and values obtained using the proposed trust model regarding message forwarding, which is either to allow, block or further evaluate before forwarding messages to other nodes, Artificial Neural Network (ANN) is also used by the controlling agent (a recommending node) to self-train the vehicular nodes to get the expected trust values. ANN is composed of an input layer (Message receiving nodes), a hidden layer (message forwarding nodes), an output layer (cumulative trust values from both the input and hidden nodes), and activation function (Euclidean distance and recommended trust value). When expected output is not obtained, back propagation is done, corresponding weights are adjusted and forward propagation is done again to check whether the expected output is obtained. The entire process is repeated until the expected output is obtained [19].

Smart Trust Management group vehicles for Vehicular Network

Amel Ltifi et al. Proposed Cluster-based approach which uses symmetric cryptography method to manage trust and traffic, Where central Certification Authority (CA) tasks are distributed between a set of dynamic Group leaders which are chosen according to a clustering algorithm. In this approach, each vehicle is equipped with an OBU (On Board Unit) and organized as a set of clusters, the OBU is in charge of recording, calculating, locating and sending messages, while the vehicle with the highest trust level in each cluster is selected as a group leader (GL) by the reception of a token, a GL changes periodically, it responsibility lies in the creation and update of the trust model (whom contains all the information about group members, like identifiers, trust value and cooperation's counters values) according to the members behaviors as well as managing alert message trustworthiness. The trust management system offers autonomous decision making via a component of the OBU, which is the knowledge database. When a vehicle detects an accident on the road, it sends a warning message to the leader who verifies the trust level of the vehicle sender by accessing the knowledge base and decides whether to treat the warning message or ignore it [20].

A Trust Management Scheme with Affinity Propagation

Shu Yang et al. proposed a trust-based anomaly detection scheme for intelligent vehicles (IVs) to detect abnormal vehicles and use affinity propagation to construct trustworthy cluster heads (CHs), who are responsible for intracluster trust management. Cluster and its head are generated after several rounds of iteration, each IV will periodically broadcast its self-responsibility and self-availability to the neighborhood to claim how suitable it is to become a CH. The one with the highest results will be elected CH according to the AP algorithm, this cluster algorithm helps pass messages between nodes, an UntrustDegree function as "distance measurement" is designed for this algorithm in order to find "the most trustworthy node," with

the minimal overall UntrustDegree, an IV can observe other vehicles' behaviors and give an UntrustDegree according to its knowledge. When a group of IVs pass by a RSU, RSU will proactively download/broadcast reputations to IVs. A supervisor model is proposed to alleviate cheating/ mistaking in the broadcasting process, it can receive almost the same broadcast information by sharing the same wireless channel. Each IV automatically chooses a supervisee by Supervisor Matching algorithm, and checks supervisee's related message to validate availability/ responsibility, by repeated calculation If two results have a large difference, then the IV cheated thus an alert will be released, so it would be ignored by neighbors and reported to CA. In any round if a CH was generated, it will broadcast Final Message, which represents CH's final evaluation to each cluster member [21].

2.2.3 Hybrid

Most of the recent models are hybrid models, they mainly revoke dishonest vehicle nodes and discard incorrect and falsified received messages.

Trust Based Security Enhancements for Vehicular Ad hoc Networks

Wei, Yu and Boukerche proposed a hybrid approach which Detects Misbehaving vehicles by speed variation and establishes trust with direct interactions and recommendations based on a combination of Bayesian rules to find the trust and Dempster-Shafer theory (DST) for handling uncertainty. Each vehicle that interacts with an observed vehicle, can judge if the messages are trustworthy or not and provide evidence to another observer vehicle from its perspective, trust value is calculated through the evidence provided by other vehicles independently. Each vehicle can evaluate trust of its neighbor vehicles in a distributed manner, the direct interactions and recommendations are then combined to assess trust, Bayesian rule is employed in the direct interactions, which can continuously revise the trust values through new evidence, while the DST is used to weave uncertainty of third-part recommendations into trust evaluation to improve the accuracy of trust, the trust values are then stored in the module of trust repository and used by Upper layer protocols or applications to achieve their goals [22].

Blockchain-Based Adaptive Trust Management in Internet of Vehicles Using Smart Contract

Pranav Kumar Singh et al. proposed a blockchain-based decentralized trust management scheme using smart contracts, and the concept of blockchain sharding to help reduce the work-loads and increase the transaction throughput of the blockchain network. The Traffic Authority (TA) initializes the system by deploying a Certificate Authority(CA) and assigning a number of Regional Authorities(RA) in IoV. Each RA is in charge of deploying RSUs and responsible for the maintenance and control of a blockchain shard for processing the localized transactions generated in its territory, like the registration of a new vehicle. Whenever a vehicle passes through a different region, it needs to register, the TA will then provide the vehicle with a set of private/public key pairs before it enters the road so that all the communications within that region will take place using those sets of local key pairs. Registered vehicles participate in various events at the vehicular plane, during each session, messages are exchanged among the vehicles, If any inconsistencies in the messages are detected, the vehicle will first check if there is any unclaimed reward earned for participation during a past session which they can redeem for various services and payments before proceeding to report the suspicious vehicle, the RSU

in the region will then analyze all the reports received during a session at certain random interval of time and update the trust score of vehicles depending on their behavior. When the vehicle is set to leave the region, the RA executes a transaction through the blockchain shard before also executing a transaction on the global blockchain to update the score value of the vehicle [23].

Vehicular networks with security and trust management solutions via proposed secured message exchange via blockchain technology

Nisha Malik et al. proposed a hybrid trust management system which uses two major phases, In the first one, in order to ensure secure communication, the sanitization(hiding) of sensitive data before transmission is required and done using an optimal key, that is generated through an optimization algorithm termed SLE-WOA (combination of Whale Optimization and Sea Lion Optimization Algorithm) by a request sent to the RSU, who's in charge of maintaining the keys using blockchain technology. Once the sanitized message is broadcasted among vehicles, receiver vehicles will try to access the data by requesting for the respective optimal key to the corresponding RSU. In the second phase, the RSU will do a trust evaluation using Rule based and Neural Network based models to decide whether the node is authenticated or not before providing the key. the values of Packet delivery rate (PDR), Partnerships for renewables (PFR) and Received signal strength indicator (RSSI) from the receiver node will be evaluated and if they reach beyond the threshold each respective feature attack, then node is deemed untrustworthy and forwarded to the NN model who is trained with the behavior of nodes for PDR, PFR, and RSSI and helps predict whether the node is authorized. if the RSU finds the authentication to be a success, it grants the key to desanitizate and access the data, else it simply neglects its request [24].

2.3 Comparative study

2.3.1 Comparison parameters

We will compare all the different works mentioned previously based on some parameters that found below:
-Computation power: the computation debit of the model, the lower it is the better it is.
-Memory storage used: the amount of memory storage consumption used by the model.
-Complexity: the class of complexity the model belongs to from centralized, decentralized to hybrid.
-Type of attacks countered: number and type of countered attacks.
-Detection rate: percentage of correct detection of malicious behaviors or nodes.

2.3.2 Comparison table

The table below will compare between the reviewed works :

Classification	Works	Computation power	Memory usage	complexity	Type of attacks countered	Detection rate
Data centric	[13]	Low	Low	Decentralized	Identity theft, data falsification	High
	[14]	High	Low	centralized	Varied	92-99%
	[15]	High	Medium	Decentralized	Position falsification	96-98%
	[16]	High	High	Hybrid	Varied	High
Entity centric	[17]	High	High	Hybrid	Data falsification	High
	[18]	High	High	Hybrid	Data falsification	High
	[19]	Medium-High	Low	Hybrid	Varied	90-99%
	[20]	Low-Medium	High	Hybrid	Data falsification	High
	[21]	High	High	Hybrid	Data falsification	99%
Hybrid	[22]	High	Medium	Decentralized	Inside attacks	High
	[23]	High	High	Hybrid	Data falsification	High
	[24]	High	High	Centralized	KPA, CCA, KCA,CPA	88%

Table 2.1: Comparative table.

2.3.3 Discussion

Through the study of some research papers we came across many various techniques and methods whose goal is to ensure safety in the traffic structure. Some of these methods don't only bring on their own share of advantages but also cons, different strategies seem useful against some attacks but are also more vulnerable towards others.

The following techniques in work [13] [16] [17] [23] [24] used blockchain, blockchain helps assure the safety of the data and bringing robustness to the structure but despite that it also has weak points and is vulnerable against threats like Black Hole Miners, Bad Mouthing, and Malicious Miners.

The method on deep learning in paper [14] offers a better independence from humans to select features, but it requires a lot of computation power and more data to be as equally or more efficient than the usual neural networks.

The fake position falsification method used in [15] appears to be super-efficient when it comes to position falsification type of attacks, but not very useful for any other type of attacks, which makes it non reliable in other situations and scenarios.

In the Entity centric class we find two models in work [17] [18] and one model in the hybrid class work [23] that follow a reward/punishment strategy which shows very good results in encouraging selfish nodes to adapt a cooperative behaviors, but thus far is not very helpful in stopping other malicious behaviors.

In the hybrid class paper [22] as well as in the Entity centric class paper [20] [21] these approaches use certain nodes as dynamic watchdogs in the system, they seem to grasp better control, supervision and tight security of the network and its participants.

In paper [18], the model works on reputation values, which does not necessarily assure that the computed trust values are trustworthy, there lies doubt around the values attributed towards vehicles as there is no knowledge about whether a vehicle has familiarized itself enough with another vehicle to be able to give out an appropriate trust value corresponding to it.

All the parameters we used to do the comparison with helped us show a few differences between the models used and how each of the strategies and methods impacted on them, from this we will take into consideration what could be the optimal elements to pay attention to and come up with our own approach.

2.4 Conclusion

This second chapter helped us see the different methods and strategies used in trust management, after comparing the methods from different classes we noticed that the results did not really depend on the classes in themselves but the strategies and methods used in question.

In the next chapter we will introduce a new approach which relies on multiple trust types that summarize different situations a vehicle could be in when having to compute trust, as well as their reliability to compute them. It uses k means clustering algorithm to help classify the vehicles per their respective types.

Chapter 3

Proposition of our method

3.1 Introduction

In order to help assure the security of the vehicular system, we will introduce a new trust management solution for VANET using k means clustering algorithm and a method which uses several types of trust, to determine the malice of a vehicle and from where it'll try to detect the degree of truthfulness of a message and respectively it vehicle.

This chapter will be divided in a few sections where the first part will cover the problems the method we choose to focus and work on faced in previous mentioned works, as well as introduce our own proposed method. We will then introduce the dataset generation, clustering algorithm used and at last how trust was computed in our proposed method.

3.2 Problematic

The reputation based trust computing has been applied in several works, and has been so far trusted in detecting and separating the misbehaving vehicle from the well behaved ones, But the downside of this trust is that there is no guarantee on how the trust values are assigned or exploited by malicious nodes.

This method is usually defined as a global trust value or a mixture of both direct and recommended trust, but there is no real consideration of time or quantity and priority in computing the trust, which does not assure the values you obtain are worthy of trust.

As seen some works make good use of this method but do not really mention or take into consideration certain points, this motivated us to take a closer look at them and touch on them, and made us come up with our own version that we believe will benefit the vehicular network more.

3.3 Our proposed method

3.3.1 Data generation

Introduction

A dataset is an ordered set or collection of coherent data that is associated with a particular attribute or observation and is representable in different formats (videos, images, text, sound... etc). Datasets are a very important and crucial part in the machine learning process, as the data generated in the dataset will help create and train our model to learn patterns and make accurate predictions after validation. There are different types of datasets, each one independent from the other with their own specific purpose, the training dataset is used to train the data while the testing dataset helps measure its performance, lastly the validation dataset is in charge of tuning the parameters of a classifier.

During our researches we came across many datasets, where most of them were either too detailed or poor, as we were unable to find a dataset that could fit both our goal and needs, we decided to generate our own dataset using specific tools and methods in order to achieve the abstractions we wanted to have.

The main objective of our work being to achieve a trust management system and trust computation system inside a vehicles system, we decided to override many realistic aspects of the VANETS behavior in the intention to lower the complexity of the work and focus more on our main trust management task without making it less effective in a more realistic situation. In order to get the abstraction we desired, we decide to override details like the different kinds of vehicles (bus, truck, car), the traffic time, the habits of the vehicles and current time of the system. We also decided to override the adaptation of the vehicles, thus the vehicles can't learn from their environment and improve their lies nor adopt a less suspicious behavior.

3.3.2 Our dataset

The dataset we generated includes 2 subdatasets : **1) Vehicles dataset :** a specific number of vehicles (around 400 to 600) that are going to roam and communicate with nearby vehicles. **2) Messages dataset :** a collection of messages of different types that a vehicle sends to other vehicles during communications (the size of the dataset can vary from 500000 to 600000 messages).

1) **Vehicles dataset :** The entire set of vehicles that are present in our simulation. A single vehicle is characterized by the following informations :

1. **Veh_id (int) :** The identifier number of the vehicle, it's unique and not null (plays the role of the primary key). The veh_ids are generated with the increment method.
2. **Type (int) :** A value from range 1 to 4 that defines the type of the vehicle. The type of the vehicle is a very important characteristic that is going to define the behavior and main traits of a vehicle. The vehicle type is going to have an impact on :
   (a) **The number of messages sent :** Depending on the vehicle type the message sending rate may vary.

(b) **The type of messages sent :** Different types of vehicles can send different types of messages, some message types are proper to a vehicle type.

(c) The average percentage of the different message types in the messages sent.

(d) The average percentage of the different message types in the lies made.

3. **Malice_degree (float) :** The average degree at which a vehicle is going to lie. Varies between 0 and 1.

4. **Malice_degree_matrix (matrix of float) :** The degree at which a vehicle is going to lie to each vehicle individually.

2) **Messages dataset :** The full collection of messages exchanged during our simulation. Let suppose a vehicle A sends a message M to a vehicle B. The message is characterized by the following informations :

1. **Msg_id (int) :** The identifier number of the message, it's unique and not null (plays the role of the primary key). The msg_ids are generated with the increment method. (Id of the message M from our example).

2. **Veh_id1 (int) :** The id of the vehicle message sender (Id of the vehicle A from our example).

3. **Veh_id2 (int) :** The id of the vehicle message receiver (Id of the vehicle B from our example).

4. **Pos1 ([float, float]) :** The coordinates in which the vehicle sender has sent the message (the pos of the veh A when it sent the message M).

5. **pos2 ([float, float]) :** The coordinates in which the vehicle receiver has received the message (the pos of the veh B when it received the message M).

6. **M_type (int) :** A value that ranges from 1 to 4 that defines the type of the message. The message type does not impact on the other characteristics of the message. However it does impact on its possible truth value (due to the lie probability related to the vehicle sender and the message type that we will talk about later in more details).

7. **M_truth (boolean) :** The truth value of the message, if the message is a lie the m_truth value will be 1 otherwise if the message is true the value will be 0 (in other words the truth value answers the question: is the message a lie ?).

8. **M_time (float) :** The time when the message was sent. Varies in the range [0, simulation time].

9. **M_truth_degree (float) :** Define the certitude of the IMC in defining the truth value of the message, value varies in the range [0, 1].

Intrusion detection system (IDS) : During this work we have abstracted the work of the IDS and instead used the output value of it, so the IDS in this work is a black box that takes message informations in input and outputs the certitude degree on the truth value it has assigned to the message.

These information have been chosen as parameters after careful considerations. Each of these characteristics play a part in describing the studied data, they have an impact on how vehicles behave and interact with each other, some other characteristics will impact the message type or lie rate and so on. For example both the sender and recipient's identity and position will give us some information about the parties concerned and influence on their interaction, but the type of content and other factors mentioned above will help determine the honesty of outgoing and incoming messages.

In other words we tried to lower the complexity of the model while keeping a realistic and necessary complex level to have meaningful and coherent informations in our dataset.

3.3.3 Used tools

For our data generation we used different tools namely python and some libraries :

Python : Python is an interpreted, portable, interactive, object-oriented, high-level programming language with dynamic semantics.
It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming [25].

Numpy : is a general-purpose array-processing package, it forms the basis of powerful machine learning libraries like scikit-learn and SciPy and provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python[26].

Matplotlib : Matplotlib is a plotting comprehensive library, it provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK and is used for creating static, animated, and interactive visualizations in Python [27].

Pandas : is a fast, powerful, flexible and easy to use open source Python Library and high-performant data analysis and manipulation tool. Pandas allows us to convert data from different formats, such as a CSV file, into a DataFrame object. a data structure that serves as a tabular representation of data [28].

3.3.4 Conceptualization

During the creation of our dataset we went through different stages and steps to create the different characteristics mentioned above. Some of the parameters needed a lot of tests to estimate the right values in order to get coherent and realistic data.

First and foremost We need to set up the environment of the simulation: The map.

3.3.4.1 The map

The map represents the virtual space on which the vehicles will navigate on, a map is characterized by its content and its size, in order to set it up we have to define these parameters.

- **Bounds :** The map is a determined, closed limited by bounds and entirely observable environment.
- **Content :** We opted for an empty bounded map, meaning it contains no roads, streets, buildings nor any sort of obstacles. This allows free movement in any direction and makes traveling distances less complex as the vehicles move as the crow flies .
  Our reason behind this choice relies on the fact that it offers a level of abstraction that helps reduce the complexity and number of computations of the simulation.
- **Size :** The map's size plays a relevant role in the vehicle's movements and communications in the system. If the map is too small and the vehicle number is too big the movements will lack diversity and the arbitrariness will be lower (since vehicles evolve in a small environment) therefore vehicles will more likely communicate with the same entities all along the simulation, reducing the realistic aspect of it. On the other hand if the map is too big and vehicles too few, vehicles will have a lower chance to get into communication range , lowering the communications and impoverishing the dataset. The map size and density of vehicle number in the map plays a role in the quality of the dataset.

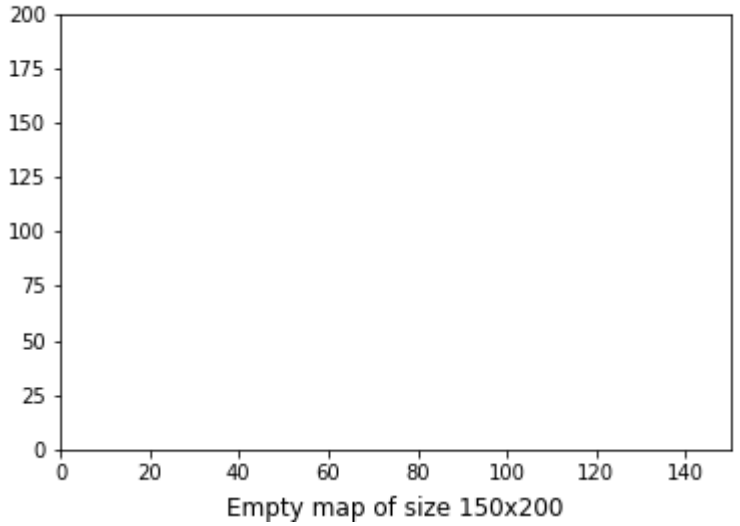

Figure 3.1: The map.

The map can be improved by adding roads, obstacles, traffic signs, centers of interests to provoke traffic in some specific areas. We really wanted to implement those features by using the Djikstra algorithm to generate itineraries or Location-based recommendation algorithms to generate networks, centers of interests and therefore traffic in targeted (or randomly generated) areas.

3.3.4.2 The actors

The actors are the components of the map and the elements that are going to evolve on it. We have 2 types of actors in our system : RSU and Vehicles.

1. **RSU :** RSUs assure the communication between vehicles, it stores, updates, sends/receives data to/from vehicles and manages interactions with the server to update the global data.
2. **Vehicles:** Entity that circulates through the map, interacts with other vehicles and RSUs by sending and receiving messages.

Just like in the environment we have set up, we have to generate and set up an optimal number of RSUs and vehicles in the system to not overflow or drought the system and dataset.

After various tests, we decided to set the number of vehicles in the network to 500 and the number of RSUs to 50, those numbers assure us that the number of computation won't be excessive while at the same time allowing a varied communication to take place between the vehicles.

After choosing the number of vehicles and RSUs in the system, we got to set their initial position on the map. We do that by using a uniform distribution to generate coordinates on the X axis and Y axis (using the method np.random.randint of numpy).

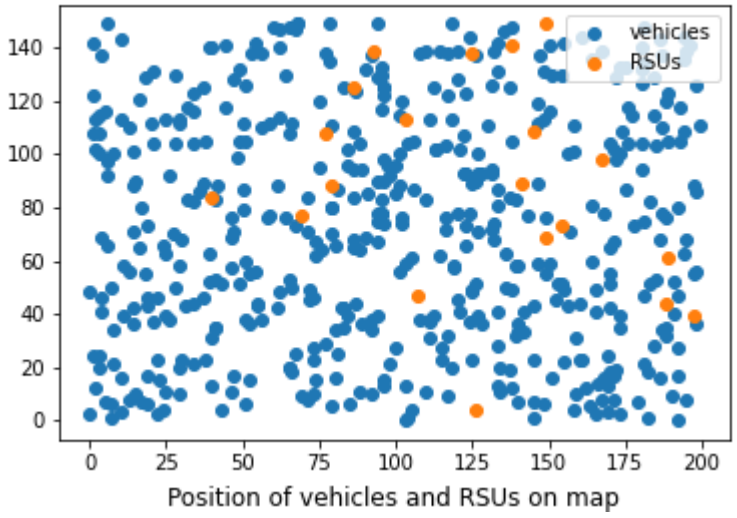

Figure 3.2: The position of the vehicles and RSUs on our map.

After setting the map bounds, vehicles and RSUs positions, the environment is ready. In the next part we will view the different methods and functions that are going to define the different behaviors of the vehicles and generate the messages according to them. We will take a closer look at the characteristics mentioned in the dataset presentation and how we used values to generate the results while modelling it in a stochastic way to simulate more realistic data.

3.3.4.3 Defining the vehicle type

The type of vehicle is what defines its main behavior, impacting on others parameters like messages type percentages, the lie probability, and message sending rate. We distinguish 4 types of vehicles : from veh_type1 to veh_type4.

Those vehicle types play a role of placeholder so far and are used to set some distinct behavior in the system. In a more specific and realistic system we could add more vehicle types and assign to each type a precise behavior inspired from reality.

Each type is characterized by :

1. **Message sending rate :** Average message number sent during 1 unit of time.
2. **Ratio of messages type :** Message type percentages in messages sent.
3. **Type wise lie probability :** The probability of lying on specific types of messages.

When creating the dataset we defined a specific percentage for each vehicle type in the system. We opted for 40%, 20%, 20%, 20% for each type of veh_t1 to veh_t4 respectively.

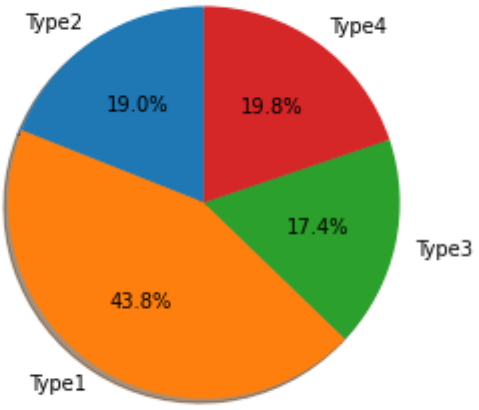

Figure 3.3: Chart of the vehicle types.

3.3.4.4 Defining the malicious degree

1. **Overall Malicious degree** Each vehicle in the system has an overall malice degree (OMD) that defines the average rate at which a vehicle is going to lie on the messages it sends. Its value is in the range [0, 1].
   For example if a vehicle A has an OMD of 0.6, the percentage of lies in all his messages will be around 60%.
   To generate the OMD of vehicles we used a gaussian distribution of mean 0.5 and standard deviation of 0.13.

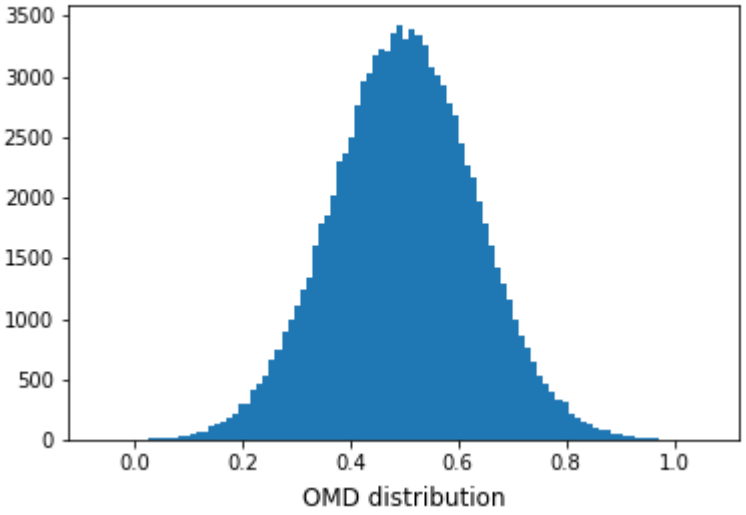

Figure 3.4: the Distribution of OMD

2. **Malicious degree for each vehicle** A vehicle may have a different malicious level towards 2 different vehicles. Let's take the previous example of vehicle A with an OMD of 0.6, vehicle A may have a different malice degree towards different vehicles, vehicle A may have a malice degree of 0.4 towards vehicle B and a malice degree of 0.8 towards C. Having 500 vehicles in our system, A may have 499 different malicious degrees depending on the vehicle it communicates with; the overall of those degrees will be around the OMD of A : 0.6.
   To generate those different malice degrees that the vehicle A will have towards the 499 other vehicles we use a gaussian distribution of mean OMD of A and standard deviation of 0.2.

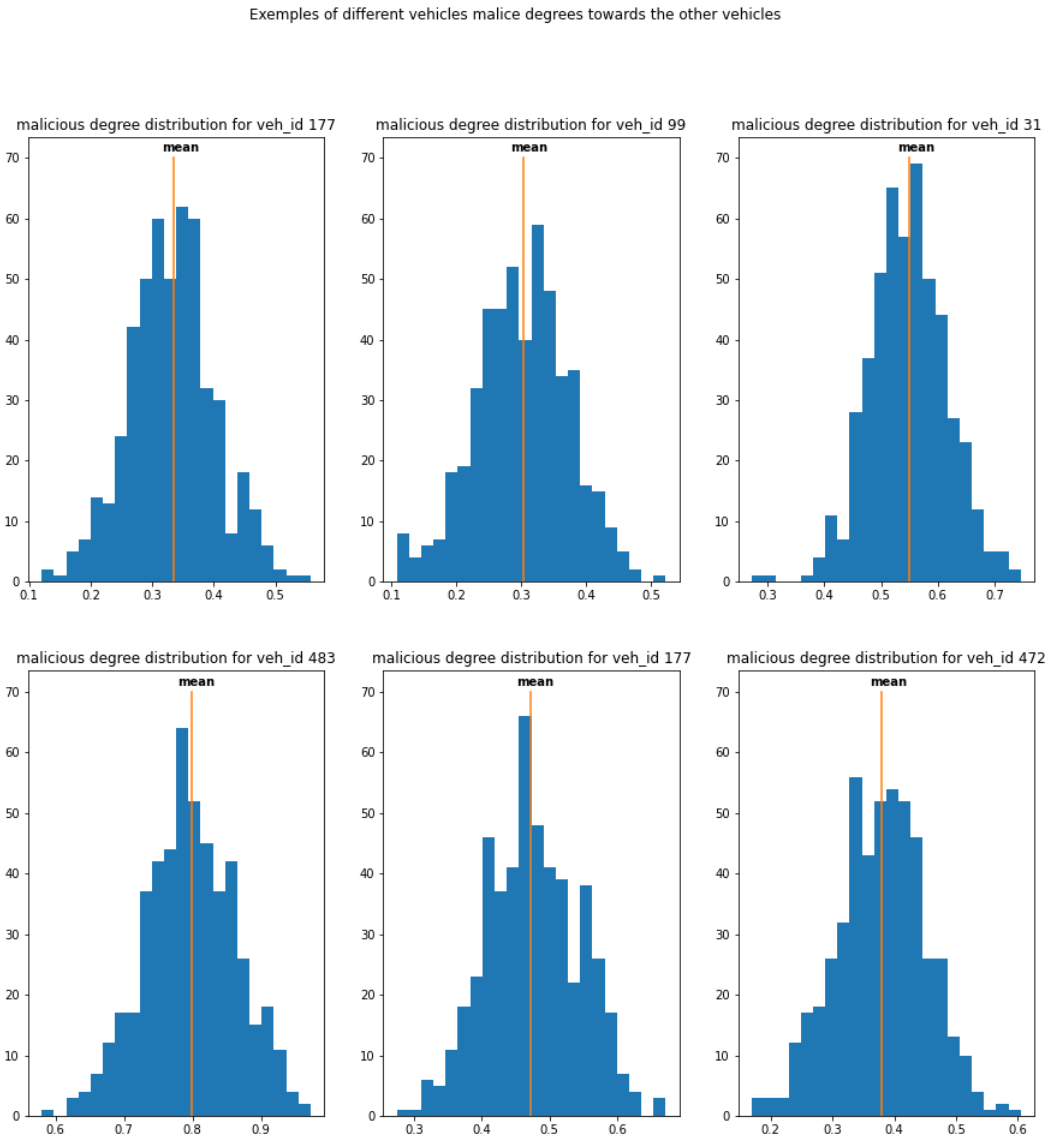

Figure 3.5: The distribution of MD towards certain vehicles.

3.3.4.5 Defining the behavior change

During this work we have implemented the concept of behavior change. The behavior of a vehicle is not the same throughout the entire simulation, the malice degree may vary by increasing or decreasing temporarily, impacting therefore on the probability of lying on the messages sent during this duration.

A single behavior change for a vehicle is characterised by:

1. **The duration (float) :** A behavior change is temporary, the durations are generated using a gaussian distribution of mean 0.33 and standard deviation 0.16. Varies from 0.16 to 0.39.

2. **Time (float) :** The time when the behavior change is starting, this time is generated using a uniform distribution throughout the simulation time. Ranges from 0 to simulation_time.

3. **Intensity (float) :** The value by which the malice degree will change during the behavior change, generated using a uniform distribution in the interval [0, 0.2].

4. **Direction (-1 or 1) :** The malice degree will either increase or decrease by intensity generated value depending on the direction generated. Bc_malice_degree = malice_degree + direction*intensity. The direction is generated using a gaussian distribution between the values -1 and 1 (it means the malice degree has an equal chance to increase or decrease during a behavior change).

We decided to set a parameter to get an overall total behavior change time around 40% of the total simulation time. That means that the sum of all the behavior change durations will be around : $0.4 \times simulation\_time$

3.3.4.6 Defining the movements

During the simulation the vehicles move through the map in a randomly generated way, which means the vehicles don't have a specific destination and they instead wander freely and randomly. We will take a closer look at how those movements are generated.

Movement : In order to generate the next position of the vehicle:

1. Generate a random angle alpha between 0 and 2pi using a uniform distribution.

2. Deduce the projection on both X and Y axis from the angle alpha to find the displacement vector.

3. Generate an intensity value that will define the intensity of the movement using a gaussian distribution of mean 20 and standard deviation 9.

4. Compute the new position :

$$vPos = vPos + vector \times intensity$$

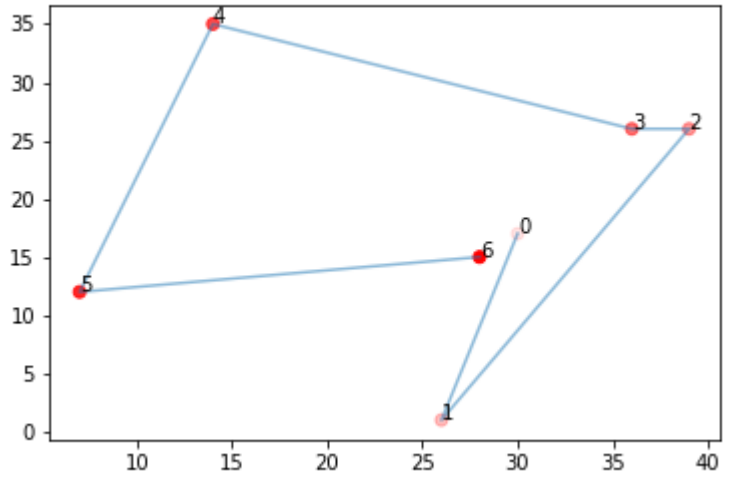

Figure 3.6: Vehicle positions and movements.

3.3.4.7 Finding the neighbours

Vehicles roam in the map changing their position every time unit. Therefore the neighbours vehicles (vehicles in range of communication) and the nearby RSUs change at every time unit. In order to send messages, finding the neighbours is a necessary step.

By using the methods find_neighbours_veh() and find_neighbours_rsu() we get nested arrays that represent the nearby vehicles/RSUS (depending on the function we use) for each vehicle in the system.

3.3.4.8 Defining the messages

The messages represent the elements that the vehicles and RSUs exchange between each other during the communications. In the same way for the vehicles, the messages are differentiated in 4 types (Message type 1 ... Message type 4). Those message types are placeholders so far that can be replaced by more realistic types in further works. As mentioned above, an abstraction has been made about the message content, size, and sending time.

To ease the explanations, let's take the following example: a vehicle A sends a message M to a vehicle B.

Type	T1	T2	T3	T4
VehT1	50%	35%	15%	0%
VehT2	40%	20%	40%	0%
VehT3	80%	10%	10%	0%
VehT4	30%	15%	15%	40%

Table 3.1: Message ratios.

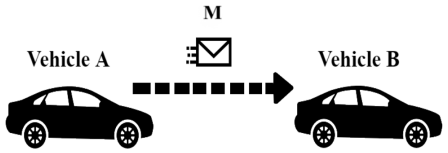

Figure 3.7: Example of Vehicle A sending Message M to Vehicle B.

During the creation of the message M, in addition of the vehicles id number that can be retrieved easily, some characteristics must be generated using a certain process :

1. Generating the message type.
2. Generating the message time.
3. Generating the message truth value and IDS accuracy.

a) Generating the message type

To define the message type during the generation, we use the ratios messages_ratio to define the message type.

Messages ratios

Depending on the vehicle type, the percentage of message type in messages sent may vary, the ratios refer to the message distribution of each type by sending vehicle type or the probability that a message is of a specific type if it is sent per a certain type of vehicle. It is represented as a matrix of 4*4, where the rows represent the vehicle type (VehT1... VehT4) and the columns the distribution of a message type (T1... T4) for a specific vehicle type.

For example :

If messages_ratios[1] = [0.5, 0.35, 0.15, 0].

It means that a vehicle of type 1(VehT1) will send 50% of messages type 1 (T1), 35% messages type 2 (T2), 15% of type 3 (T3) and 0% of type 4 (T4).

b) Generating the message time

Messages during our simulation are generated at each communication a vehicle has with its neighbours during the actual time unit. To generate the precise time when the message has been sent we used a specific method instead of generating the exact number of messages exchanged between a vehicle A and B and then generating the sending time of each message.

First we define the average arrival duration (AAD) between 2 consecutive message arrival times from the message sending rate (MSR) of the vehicle sender (depending on the vehicle type). From the AAD we generate an array of durations using a gaussian distribution of mean 1/MSR and standard deviation of 1/(3*MSR). We stop generating durations when the last value of the cumulative sum of the duration array is superior to 1 (actual time unit).

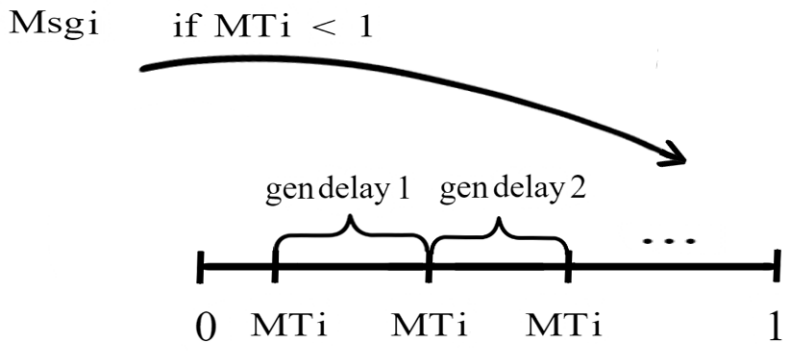

Figure 3.8: Message generation.

c) Generating the message truth value and IDS accuracy

When generating a message, we must attribute a truth value that will represent the message truthfulness. Each vehicle, as mentioned above is characterised by a lie ratio which indicates the average lie rate on each message type and is also characterised by a malice degree. Those 2 parameters are used to define the message truth value in the function is_message_malicious(v_id1, v_id2, m_t, time).

- **Lri (lie ratios array) :** The average percentage of lies in the lies made by v_id1.
- **Ri (message ratios array) :** The average percentage of message types in the messages sent by v_id1.
- **Dm (degree malice) :** Retrieve the malice degree that vehicle v_id1 has towards vehicle v_id2 at the current time / (Dmm[v_id1, v_id2], Dmm being the degree malice matrix.

- **M_t (message type) :** The type of the message sent.
- **Time :** The current time of the message sent.

Then we compute the lie probability that vehicle v_id1 have to lie on the message :

$lie\_prob = lri[v\_t1] \times dm/ri[v\_t1]$

Using this lie probability we decide if the message will be a lie or not.

The value of **IDS accuracy** is **lie_prob** if the message is a lie and **1-lie_prob** of the message is not a lie.

3.3.5 Studying and preparing the dataset :

Now that the dataset is generated, we need to prepare it in order to use it in our trust computing work.

3.3.5.1 Data overview

We generated many datasets with different parameters and values (number of vehicles, range of communication, rate of messages sent ...) in order to make comparaisons and find the best values to balance trustworthiness of the dataset and number of computations during the simulation. Therefore we chose to set the number of vehicles to 500, the range to 10 and the simulation time to 10.

Let first take a look at the current state of the dataset we chose to study :

1. **Vehicle dataset :** Number of vehicles: 500

	veh_id	veh_type	veh_malice_lvl
0	0	1	0.644378
1	1	0	0.627669
2	2	1	0.205034
3	3	2	0.601642
4	4	2	0.191192
...	...	...	...
495	495	2	0.685080
496	496	1	0.448327
497	497	1	0.486576
498	498	2	0.419473
499	499	0	0.447569

500 rows × 3 columns

Figure 3.9: Vehicle dataset.

	veh_id0	veh_id1	veh_id2	veh_id3	veh_id4	veh_id5	veh_id6	veh_id7	veh_id8	veh_id9	veh_id10	veh_id11	veh_id12	veh_id13	veh_
0	-1.000000	0.730496	0.629818	0.804761	0.787382	0.628206	0.560618	0.616250	0.595238	0.609402	0.614436	0.657731	0.698197	0.692456	0.72
1	0.828513	-1.000000	0.691414	0.648410	0.562272	0.625659	0.564057	0.674759	0.549612	0.663664	0.678216	0.550037	0.671601	0.529109	0.65
2	0.179779	0.208795	-1.000000	0.170758	0.212803	0.164749	0.180428	0.267654	0.150957	0.202407	0.233685	0.267742	0.196777	0.243498	0.13
3	0.628852	0.578800	0.614577	-1.000000	0.623228	0.580389	0.496893	0.640369	0.616397	0.522752	0.553455	0.729683	0.491444	0.539942	0.61
4	0.211872	0.184110	0.195747	0.360821	-1.000000	0.217932	0.122340	0.224352	0.105752	0.234333	0.131092	0.236623	0.286263	0.152425	0.16
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
495	0.706099	0.796713	0.550516	0.601827	0.688429	0.748538	0.603599	0.615455	0.669555	0.781684	0.620925	0.643601	0.803494	0.667066	0.58
496	0.442073	0.467731	0.443458	0.436093	0.381172	0.431252	0.503131	0.404960	0.444466	0.454078	0.444384	0.499689	0.486222	0.571662	0.62
497	0.463306	0.560329	0.577408	0.486906	0.484988	0.544628	0.505685	0.489382	0.493961	0.571814	0.416935	0.474665	0.503885	0.519634	0.33
498	0.455640	0.369852	0.556349	0.487332	0.372131	0.423340	0.379332	0.431930	0.385468	0.462562	0.414856	0.428713	0.342360	0.508566	0.43
499	0.343816	0.361855	0.461498	0.548699	0.451574	0.454031	0.485374	0.544397	0.465105	0.395010	0.291662	0.343656	0.479255	0.400476	0.37

500 rows × 500 columns

Figure 3.10: The matrix of the malice degress of the vehicles in our system.

2. **Message dataset :** Number of messages : 539273

	v_id1	v_id2	pos1	pos2	m_type	m_truth	m_truth_degree	m_time	v_id1_type
0	0	159	(200.0, 138.14076992284717)	(200.0, 146.39081064139805)	0	1	0.285008	0.051785	1
1	0	159	(200.0, 138.14076992284717)	(200.0, 146.39081064139805)	2	0	0.572488	0.092073	1
2	0	159	(200.0, 138.14076992284717)	(200.0, 146.39081064139805)	0	1	0.285008	0.138202	1
3	0	159	(200.0, 138.14076992284717)	(200.0, 146.39081064139805)	2	0	0.572488	0.169227	1
4	0	159	(200.0, 138.14076992284717)	(200.0, 146.39081064139805)	0	0	0.714992	0.230789	1
...	...	...	...	...	...	...	...	...	...
539268	499	427	(112.65080685020212, 46.60891295856189)	(106.75867898865616, 44.23466726433189)	2	1	0.480849	9.613663	0
539269	499	427	(112.65080685020212, 46.60891295856189)	(106.75867898865616, 44.23466726433189)	1	1	0.412156	9.710538	0
539270	499	427	(112.65080685020212, 46.60891295856189)	(106.75867898865616, 44.23466726433189)	0	0	0.761638	9.838507	0
539271	499	427	(112.65080685020212, 46.60891295856189)	(106.75867898865616, 44.23466726433189)	1	0	0.587844	9.942271	0
539272	499	427	(112.65080685020212, 46.60891295856189)	(106.75867898865616, 44.23466726433189)	1	0	0.587844	10.032701	0

539273 rows × 9 columns

Figure 3.11: Message dataset.

3.3.5.2 Preparing the dataset

In order to feed our trust computing function the data it needs, we must first prepare and format it. We are going to define the labels and features, separate them and create new features from the already existing one.

Defining labels and features

In this work we are trying to compute the trust we will attribute to the messages. In order to compute the trust of the messages we need the malice degree of vehicles and the different ratios. Therefore the features are the messages's trustworthiness degree and the malice degree of the vehicles while the labels are the malice degree of the vehicles and truth degree of the messages.

3.3.5.3 Classification

The objective of the classification is to divide the vehicles in a specific number of clusters (that we will need to define).

a) Clustering algorithm

There are various clustering algorithms, these algorithms take the dataset features as input, analyse the observations, process it, find a pattern and help group data that is scattered and represented as unlabeled dots into clusters based on their characteristics and their similarity, each cluster will be different from another and holding cluster member similar to each other, the clustering algorithm will classify each vehicle in specific clusters.

In our case among many clustering algorithms, we used k means in order to define which type the vehicle belongs to.

b) K means ++

A simple, fast and efficient unsupervised clustering algorithm, it has two inputs, n and k, n referring to the dataset and k referring to the number of clusters it's going to create, the term

'means' refers to the data point that is at the center of a cluster, the purpose of this algorithm to make clusters based on the smallest possible distance also known as variance between these center points.

c) Functioning

The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to a large number of samples and has been used across a large range of application areas in many different fields.

The k-means algorithm divides a set of samples into disjoint clusters , each described by the mean of the samples in the cluster. The means are commonly called the cluster "centroids"; note that they are not, in general, points from , although they live in the same space. The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum-of-squares criterion :

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Figure 3.12: K means formula.

Inertia can be recognized as a measure of how internally coherent clusters are. It suffers from various drawbacks. Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes. Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated (this is an instance of the so-called "curse of dimensionality"). Running a dimensionality reduction algorithm such as Principal component analysis (PCA) prior to k-means clustering can alleviate this problem and speed up the computations.

d) Determine the optimal number of clusters K

The dunn index is a metric for evaluating clustering algorithms, in order to find the right number of clusters, It will use the intra and inter cluster distance, following this formula :

$$Dunn\_index = \frac{Min\ inter\ cluster\ distance}{Max\ intra\ cluster\ distance}$$

We have computed the dun index for various numbers of clusters and plotted the result.

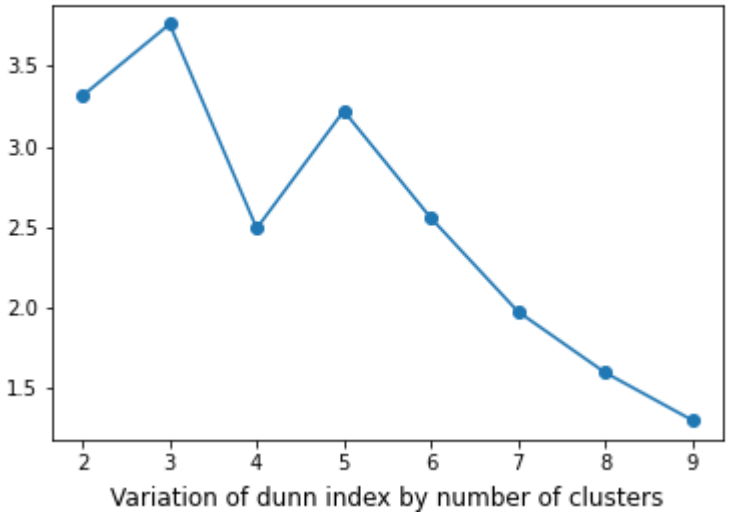

Figure 3.13: Dunn index Variation.

e) Elbow method (clustering)

In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use.

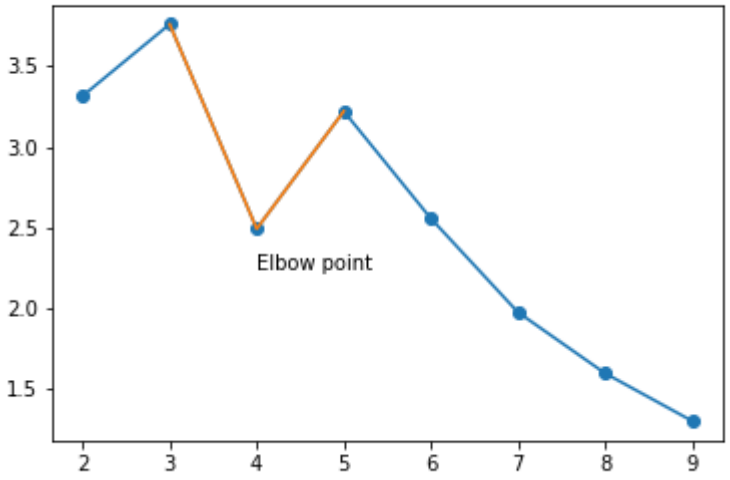

Figure 3.14: The elbow point

We notice from the plot that there is an elbow at the point 4, therefore we choose a number of clusters equal to 4.

3.3.6 Trust computing

The objective of this work is to compute the truth degree of the messages sent, in order to achieve this we need to go through different steps and obtain different values. First let's define quickly how we are going to compute the message trust degree.

3.3.6.1 The formula

The message trust degree is the value that the IDS has attributed to the chosen truth value (lie or truth) of the message. The following formula indicates how to compute it :

$$Lpi = \frac{lri.Dm}{ri}$$

So to compute the trust degree of a message we need the following parameters :

1. Informations about the vehicle sender :
   (a) Veh1_type : The type of the vehicle sender.
   (b) Dm : The malice degree of the vehicle towards the the receiving vehicle at the current time.
2. The message type.
3. Messages ratios : The different percentages of types in the messages sent by vehicles of the same type as the vehicle sender.
4. Lies ratios : The different percentages of types in the lies made sent by vehicles of the same type as the vehicle sender.

3.3.6.2 The justification

We have the following statement:

$$lp1.r1 + lp2.r2 + lp3.r3 + lp4.r4 = Dm$$

$$\sum_{i=1}^{nbrMT} lpi.ri = Dm...(1)$$

With :

- lpi : the lie probabiliy of each message type for vehicle of type T.
- ri : the different message ratios for vehicle of type T.
- Dm : the malice degree of the vehicle sender at the current time.
- T : type of the vehicle sender.

We will define 'nbrMT' as the 'Number of message types'.

We are trying to get the lpi vector:

May lri be the percentage of message type i of the lies made

$$\sum_{i=1}^{nbrMT} lri = 1$$

$$\sum_{i=1}^{nbrMT} lri.Dm = Dm...(2)$$

$$\left.\begin{array}{l}\sum_{i=1}^{nbrMT}(Lpi.ri) = Dm...(1)\\ \sum_{i=1}^{nbrMT}(Lri.Dm) = Dm...(2)\end{array}\right\} \sum_{i=1}^{nbrMT}(Lpi.ri) = \sum_{i=1}^{nbrMT}(lri).Dm$$

$$\sum_{i=1}^{nbrMT}(Lpi.ri) = \sum_{i=1}^{nbrMT}(lri).Dm$$

Since each lpi defines its respective lri (type wise), we have :

$$\forall i \in [1, nbrMT]/nbrMT \in \mathbb{N} \quad lpi.ri = lri.Dm$$

$$\Longrightarrow Lpi = \frac{lri.Dm}{ri}$$

Let's take a look at how we are going to get those parameters :

a) Messages ratios and Lies ratios

The value of those characteristics are related to the vehicle type and not the vehicle itself. Since we have many vehicles and even more messages in our system, a statistical study will return a very well estimated value of those characteristics.

We update the message ratios and lie ratios value at every income of the message batch. The more the simulation lasts, the more the number of messages used to compute those values increases, therefore the accuracy of the study gets better the more simulation the time increases.

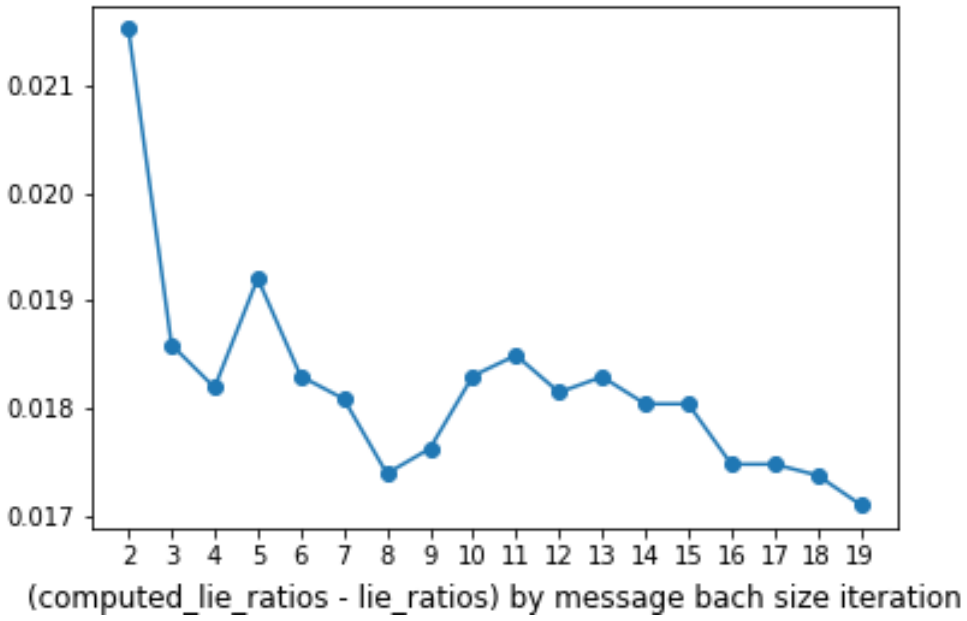

Figure 3.15: Accuracy of computed lie ratios.

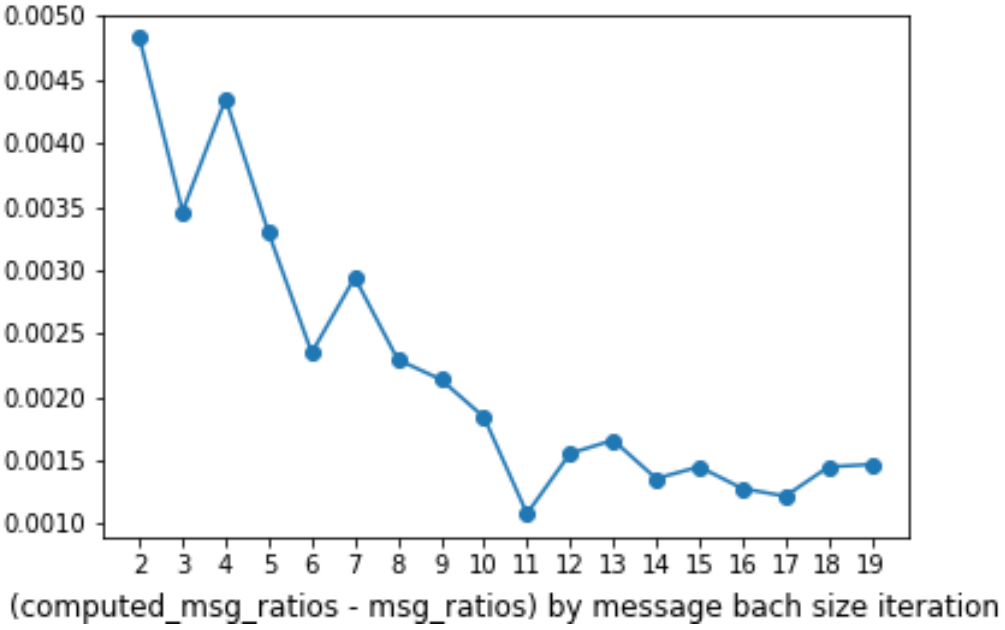

Figure 3.16: Accuracy of computed message ratios.

b) Vehicle id, type and message type

Those informations can be easily retrieved from the message sent.

c) The malice degree of the vehicle sender towards the receiving vehicle at the current time T

To compute the actual malice degree that a vehicle sender has towards another vehicle we need to keep in keep in mind that :

- Each vehicle has an OMD.
- Each vehicle can have a different malicious degree towards different vehicles (Personal MD).
- Behavior changes are temporary and change the malice degree temporarily.

3.3.6.3 Computing the Personal malice degree (PMD)

The PMD defines the malice degree that a vehicle has towards specific vehicle. To compute the PMD we need to use only the messages a vehicle sender sends to the receiving one.
May the vehicle A send messages Msgs to vehicle B, to compute the PMD that the vehicle A has towards B we need to use the messages that A sent to B only.

3.3.6.4 Managing the behavior change impact on the malice degree

The behavior change has an impact on the malice degree on the vehicle sender during a specific duration, during that period the malice degree changes and therefore the rate of lie on the messages sent during this event will also change. In order to detect the behavior, measure its impact and measure the temporary new malice degree we are going to use the messages sent during this duration.
When we study the truth degree of a message that has been sent recently, we are interested in detecting if there was or still is a behavior change active when the message got sent.
To find this temporary new malice degree we will use the messages sent during the X last time unit, deduce the lie rate and compute the temporary malice degree (we presume it's the behavior change one) Therefore computing the malice degree during the X last units of time (if it's possible to) is better than computing it on all the messages sent.

$$Last\ lie\ rate = \frac{Number\ of\ messages\ sent\ during\ last\ X\ units\ of\ times}{Number\ of\ all\ messages\ sent}$$

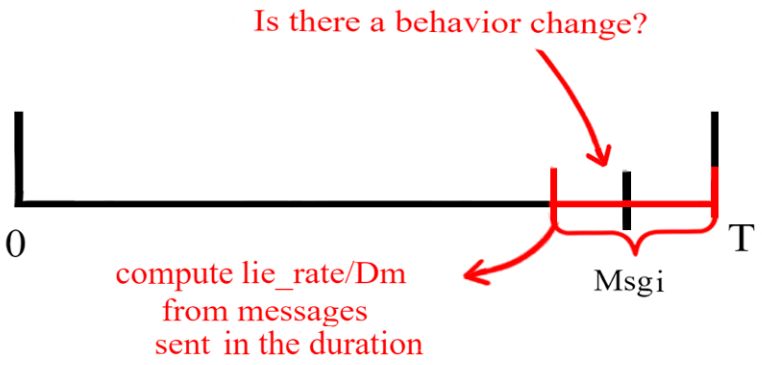

Figure 3.17: Determining the behavior change.

3.3.6.5 Computing the DM

To compute DM we are going to compute different type of trust first, then using them to compute the FINAL malice degree :

1. Local trust.
2. Covariance trust.
3. Global trust.

Reliability coefficient

During the future computations we use a reliability coefficient to determine the accuracy and truthfulness of the different types of trust and values we computed. The reliability coefficient is a value in the range [0, 1]. In order to obtain those values we use a specific function of form 1/(1+(p1/x)**p2) that we called coefficient_compute(x, p1, p2).

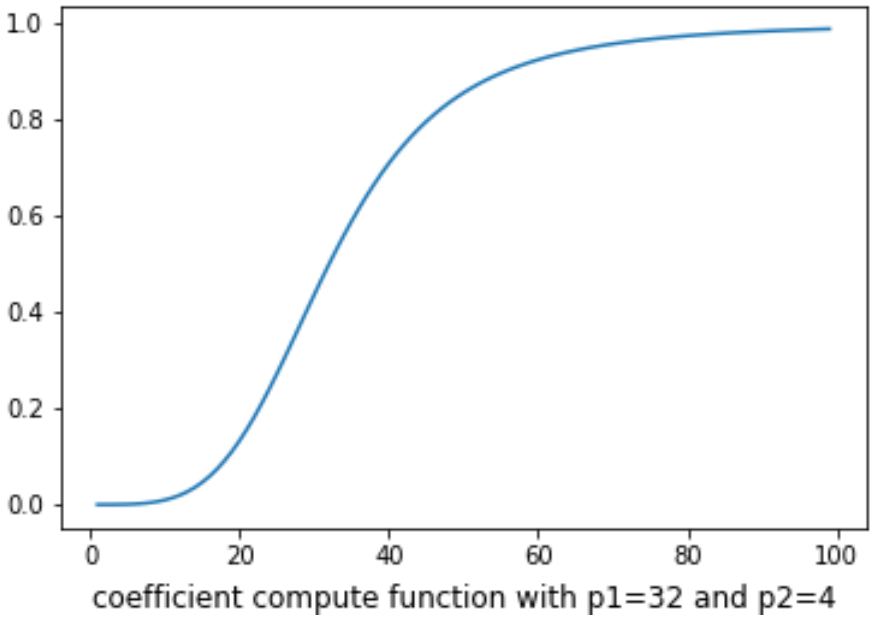

Figure 3.18: Reliability coefficient.

We will tweak the parameters p1 and p2 in order to get the result we desire according to the value of x (x being the number of messages).

1) Local trust

The local trust is based on the personal experience that a vehicle has towards an another vehicle, we use the messages exchanged between only them to compute the trust.

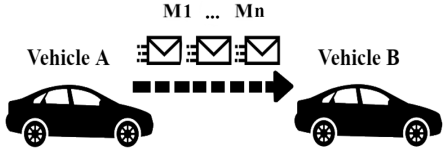

Figure 3.19: Example of Vehicle A sending multiple messages M to Vehicle B.

1. First we compute the total lie rate, which is the percentage of lies done in all the messages that A has sent to B.
2. Compute the last lie rate, which is the percentage of lies done in the messages that A has to B during the last X unit of time. May T be the time when we are computing the malice degree, to compute the last lie rate we will only use the messages sent in the duration [T-X, T].

3. The last lie rate is a more valuable information than the total lie rate (because it allows a better accuracy and detection of the behavior change), however last lie rate in reliable only if the number of messages sent in the [T-X, T] is big enough to have a decent approximation of the lie rate (therefore the malice degree). Therefore we have to compute the reliability coefficient of the last lie rate (cr_LLR : coefficient reliability of the last lie rate) using the number of the messages exchanged in the duration [T-X, T].

4. Compute the local trust by doing the weighted average of total lie rate and last lie rate with weights (1-cr_LLR) and cr_LLR respectively.

5. Compute the reliability of the local trust using the total number of messages that A has sent to B.

2) Covariance trust

2.1) Compute the trust

The covariance trust (also called friend trust) uses the malice degree that the friends of the receiving vehicle (B) have towards the sending vehicle (A).
The friends of a vehicle B are the vehicles that B has malice degree superior or equal to 0.7 towards it.
The covariance trust is the weighted mean of the different malice degree that the friends of B have towards A with the weights of each malice degree being the reliability coefficient of each friend trust based on the number of messages that A has sent to the friend.

May :

- Frds be the friends of the vehicle B.
- DmFrds the malice degrees that the friends have towards A.
- FrdsMsgNbr the number of messages that A have sent to each friend of B.
- And we define reliability_coefficient function as relCoef.

The covariance trust that B has towards A is:

$$\forall i \in [1, Frds] : \ cov(B, A) = \frac{\sum_{i=1}^{Frds}(DmFrds_i * relCoef(FrdsMsgNbr_i))}{\sum_{i=1}^{Frds}(relCoef(FrdsMsgNbr_i))}$$

2.2) Compute the reliability of the trust :

To compute the reliability of the covariance trust we will use different parameters :

- Coefficient of reliability based on the number of friends.
- The variance of the the different malice degree that the friends have towards A.
- Reliability coefficient based on the average number of message (from all the friend of B) sent to A.

The reliability is the mean of the parameters 1, 2, 3 weighted by [0.25, 0.25, 0.5] respectively.

3) The Global trust :

Global trust represents the OMD, however the global trust computation also takes into consideration the behavior change problematic that we talked about above .The global trust is computed the way as the local trust but we use all the messages that A has sent to all vehicles and not only B. The reliability of the global trust is based on the number of messages that A has sent.

3.3.6.6 Final trust :

To compute the final trust we will use all the above computed type of trusts and the computed reliabilities as the following :

May :
C_L : *the Local trust*
C_F : *the Covariance trust*
C_G : *the Global trust*
C_N : *the Neutrality* (0.5)

and their respective reliability :
R_L : *the reliability of the local trust*
R_F : *the reliability of the Covariance trust*
R_G : *the reliability of Global trust*

the formulas (1) (2) (3) represent the reliability of the new computed trust :
$R_L \times R_L + R_F \times (1 - R_L)$ ...(1)
$(1) \times (1) + R_G \times (1 - (1))$ ...(2)
$(2) \times (2) + 0.5 \times (1 - (2))$ ...(3)

the formulas (A) (B) (C) represent the new computed trust :
$C_L \times R_L + C_F \times (1 - R_L) \ = \ (A)$
$(C_L \times R_L + C_F \times (1 - R_L)) \times (1) + C_G \times (1 - (1)) \ = \ (B)$
$((C_L \times R_L + C_F \times (1 - R_L)) \times (1) + C_G \times (1 - (1))) \times (2) + C_N \times (1 - (2)) \ = \ (C)$

$(A) \times (1) + C_G \times (1 - (1)) \times (2) + C_N \times (1 - (2)) \ = (B) \times (2) + C_N \times (1 - (2)) = (C)$

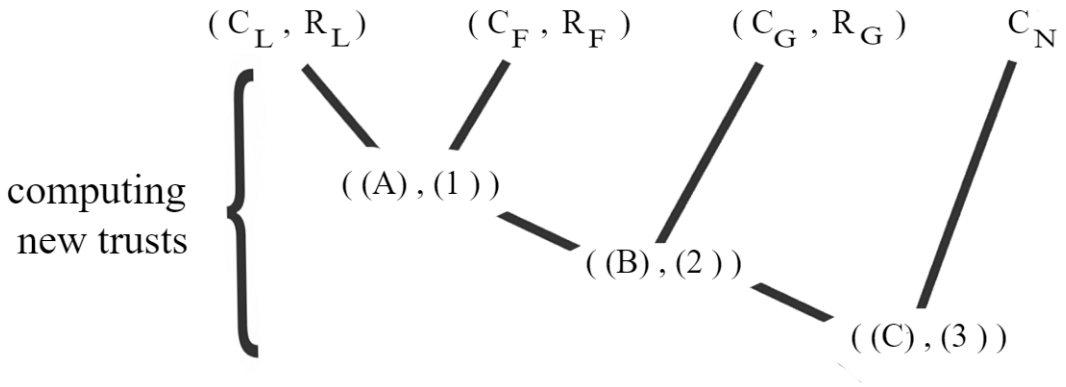

Figure 3.20: Final trust.

3.4 Conclusion

We did several abstractions in our work while still keeping it realistic, we have taken into consideration several parameters such as the fact that a vehicle can have a different malice degree depending on the vehicle it communicates with, and a change in behavior that impacts on the malice degree. In our proposed method of computing trust, several types are correlated and make the final trust value more reliable.

General conclusion and future perspectives

In this paper we studied and provided a critical review of other existing works in this increasingly important area in intelligent transportation systems, this allowed us to propose an alternative solution, that is based on multiple trust types and their reliability to properly define a reliable trust value, we made sure to give priority to vehicles with a longer communication history with our target vehicle by using a coefficient system and also took different alternatives for all the possible scenarios. The clustering algorithm k means was used to help classify vehicles into their respective types, since the type also impacts on the trust computation.

As we were unable to implement everything we initially wanted, test it performance compared to other similar models and tell whether it outperforms them or not, we will leave the rest as improvements to input into our future works. One idea is around the implementation of a cluster based architecture in our solution, this will help regroup the vehicles into small clusters with a group leader that will help manage and supervise the members more thoroughly.

References

[1] G. Dimitrakopoulos et P. Demestichas, « Intelligent Transportation Systems », IEEE Vehicular Technology Magazine, vol. 5, no 1, p. 77-84, mars 2010, doi: 10.1109/MVT.2009.935537.

[2] K. Qureshi et H. Abdullah, « A Survey on Intelligent Transportation Systems », Middle-East Journal of Scientific Research, vol. 15, p. 629-642, janv. 2013, doi: 10.5829/idosi.mejsr.2013.15.5.11215.

[3] G. Singh, D. Bansal, et S. Sofat, « Intelligent Transportation System for Developing Countries A Survey », IJCA, vol. 85, no 3, p. 34-38, janv. 2014, doi: 10.5120/14824-3058.

[4] J. Andersen et S. Sutcliffe, « Intelligent Transport Systems (ITS) - An Overview », IFAC Proceedings Volumes, vol. 33, no 18, p. 99-106, juill. 2000, doi: 10.1016/S1474-6670(17)37129-X.

[5] « What Is Machine Learning and Why Is It Important? », SearchEnterpriseAI. https://searchenterpriseai.techtarget.com/definition/machine-learning-ML (consulted april 15th, 2021).

[6] « INTEGRATED MACHINE LEARNING ALGORITHMS FOR HUMAN AGE ESTIMATION.PDF ». https://storage.ebookunlimited.club/pdf/downloads/integrated-machine-learning-algorithms-for-human-age-estimation.pdf (consulted april 15th, 2021)

[7] F. Cady, « Unsupervised Learning: Clustering and Dimensionality Reduction », 2017, p. 133-151. doi: 10.1002/9781119092919.ch10.

[8] F. Porto, « Machine Learning Life-Cycle », p. 58.

[9] « ft1paroditryolabs2019-190613170101.pdf | Machine Learning | Deep Learning ». https://fr.scribd.com/document/486767996/ft1paroditryolabs2019-190613170101-pdf (consulted april 17th, 2021).

[10] A. Ltifi, A. Zouinkhi, et M. S. Bouhlel, « Smart Trust Management for Vehicular Networks », International Journal of Electronics and Communication Engineering, vol. 10, no 8, p. 1128-1135, oct. 2016.

[11] X. Chen, J. Ding, et Z. Lu, « A Decentralized Trust Management System for Intelligent Transportation Environments », IEEE Transactions on Intelligent Transportation Systems, p. 1-14, 2020, doi: 10.1109/TITS.2020.3013279.

[12] X. Fan, L. Liu, R. Zhang, Q. Jing, et J. Bi, « Decentralized Trust Management: Risk Analysis and Trust Aggregation », ACM Comput. Surv., vol. 53, no 1, p. 2:1-2:33, févr. 2020, doi: 10.1145/3362168

[13] F. Kandah, B. Huber, A. Altarawneh, S. Medury, et A. Skjellum, « BLAST: Blockchain-based Trust Management in Smart Cities and Connected Vehicles Setup », in 2019 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, sept. 2019, p. 1-7. doi: 10.1109/HPEC.2019.8916229.

[14] Y. Zeng, M. Qiu, D. Zhu, Z. Xue, J. Xiong, et M. Liu, « DeepVCM: A Deep Learning Based Intrusion Detection Method in VANET », in 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Washington, DC, USA, mai 2019, p. 288-293. doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00060.

[15] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, et S. Nandi, « Machine Learning Based Approach to Detect Position Falsification Attack in VANETs », in Security and Privacy, vol. 939, S. Nandi, D. Jinwala, V. Singh, V. Laxmi, M. S. Gaur, et P. Faruki, Éd. Singapore: Springer Singapore, 2019, p. 166-178. doi: 10.1007/978-981-13-7561-3-13.

[16] G. Baldini, J. L. Hernandez-Ramos, G. Steri, et S. N. Matheu, « Zone Keys Trust Management in Vehicular Networks based on Blockchain », in 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, juin 2019, p. 1-6. doi: 10.1109/GIOTS.2019.8766375.

[17] X. Chen, J. Ding, et Z. Lu, « A Decentralized Trust Management System for Intelligent Transportation Environments », IEEE Trans. Intell. Transport. Syst., p. 1-14, 2020, doi: 10.1109/TITS.2020.3013279.

[18] N. Fan et C. Q. Wu, « On trust models for communication security in vehicular ad-hoc networks », Ad Hoc Networks, vol. 90, p. 101740, juill. 2019, doi: 10.1016/j.adhoc.2018.08.010.

[19] H. El-Sayed, H. A. Ignatious, P. Kulkarni, et S. Bouktif, « Machine learning based trust management framework for vehicular networks », Vehicular Communications, vol. 25, p. 100256, oct. 2020, doi: 10.1016/j.vehcom.2020.100256.

[20] A. Ltifi, A. Zouinkhi, et M. S. Bouhlel, « Smart Trust Management for Vehicular Networks », International Journal of Electronics and Communication Engineering, vol. 10, no 8, p. 8, 2016.

[21] S. Yang, Z. Liu, J. Li, S. Wang, et F. Yang, « Anomaly Detection for Internet of Vehicles: A Trust Management Scheme with Affinity Propagation », Mobile Information Systems, vol. 2016, p. 1-10, 2016, doi: 10.1155/2016/5254141.

[22] Z. Wei, F. R. Yu, et A. Boukerche, « Trust based security enhancements for vehicular ad hocnetworks », in Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications - DIVANet '14, Montreal, QC, Canada, 2014, p. 103-109. doi: 10.1145/2656346.2656353.

[23] P. K. Singh, R. Singh, S. K. Nandi, K. Z. Ghafoor, D. B. Rawat, et S. Nandi, « Blockchain-Based Adaptive Trust Management in Internet of Vehicles Using Smart Contract », IEEE Trans. Intell. Transport. Syst., vol. 22, no 6, p. 3616-3630, juin 2021, doi: 10.1109/TITS.2020.3004041.

[24] N. Malik, P. Nanda, X. He, et R. P. Liu, « Vehicular networks with security and trust management solutions: proposed secured message exchange via blockchain technology », Wireless Netw, vol. 26, no 6, p. 4207-4226, août 2020, doi: 10.1007/s11276-020-02325-z.

[25] « What is Python? Executive Summary », Python.org. https://www.python.org/doc/essays/blurb/ (consulted september 20th, 2021).

[26] « NumPy ». https://numpy.org/ (consulted september 20th, 2021).

[27] « Matplotlib: Python plotting — Matplotlib 3.4.3 documentation ». https://matplotlib.org/ (consulted september 20th, 2021).

[28] « Package overview — pandas 1.3.3 documentation ». https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html (consulted september 20th, 2021).

Abstract

During the growing development in ITS throughout the years, security breaches and attacks became a very common issue, to counter it trust management system have been put into place, but as technology kept evolving, threats did too and trust management systems started facing overwhelming amount of challenges, which put into risk the efficiency and safety of the systems and drivers, this motivated many ambitious researchers to come up with different kind of solutions to this critical yet recurrent problem, In this paper, in order to help and contribute in solving this problem, we propose a new trust management system that uses the clustering algorithm k means to help divide the vehicle into their own respective types and several types of trust along with their reliability to assure that the final trust value is trustworthy and accurate.

Key words : Trust management; Intelligent transportation System; Machine learning; VANET; K means.

Resumé

Au cours du développement croissant des STI au fil des ans, les cyberattaques et la violation des barrières de sécurité sont devenues un problème très courant. Pour contrer cela, des dispositifs de gestion de confiance ont été mis en place, mais tout comme de l'hyper croissance et évolution de la technologie, les menaces ne cesse d'augmenter et de se diversifier. Face à ces dangers, la gestion de confiance se doit de relever le défi afin de contrer ces menaces et assurer efficacement la sécurité des machines et de ses utilisateurs. Cela est aussi valable pour la conduite des véhicules intelligents où il faut assurer la sécurité des systèmes et des conducteurs. C'est pour résoudre cette problématique à la fois critique et récurrente que de nombreux chercheurs se sont dévoué à proposer différents types de solutions. Afin de contribuer à la résolution de ce problème, nous allons proposer dans cet article un nouveau système de gestion de confiance qui utilise l'algorithme de clustering k pour aider à regrouper les véhicules selon une catégorie propre à eux (types respectifs) et plusieurs niveaux de confiance selon leur fiabilité afin d'assurer que la valeur de confiance finale soit fiable et précise.

Mot clés : Gestion de confiance; Système de transport intelligent; Machine learning; VANET; K means.