International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Intelligent Accident Prevention in VANETs

A. Mari Kirthima, Rishabh Verma, Chinmayi Rajashekar Hegde, Arundhati S Shanbhag

Abstract: Accident prevention has always been an important issue for governments and car manufacturers across the world. Roughly 1.5 million people are killed in road accidents annually in India. The primary causes of accidents are broken and weathered roads, hazardous weather conditions, as well as human errors such as over speeding, distracted driving, and not following road safety rules. The traffic police work hard to enforce strict rules and maintain accident-free roads, but this hasn't proven to be efficient. A vehicular ad hoc network (VANET), as the name says, is a network consisting of nodes. These nodes depict vehicles on the road. This project aims to use this technology with K-Nearest Neighbour Classifier (KNN) to create a prototype of a system which can notify drivers of an impending accident caused by forward collisions, rear collision etc., thus enabling them to take immediate action and prevent it.

Keywords: VANET, KNN, Accident Prevention

I. INTRODUCTION

This project aims to use VANET technology with Machine Learning to create a system which can notify drivers of dangerous conditions such as forward collisions and over speeding which can lead to a potential accident. The data collected by the system can be used to take actions for preventing accidents in highways [3].

A. Vehicular Ad-hoc Networks

VANETs can be described as a sub topic under mobile ad hoc networks (MANETs). These have the potential to improve road safety and provide the traveller with comfort and infotainment. It is a wireless ad hoc network, which has high node mobility and is adapted to fast topology changes. VANET is an emerging field of technology that allows vehicles to communicate with each other or to a fixed Road Side Unit (RSU) dynamically, providing information such as speed, location etc. Vehicles are equipped with Wi-Fi hardware which does not require any changes to the existing body.

The principle of VANETs is that a vehicle can detect other vehicles in fixed vicinity enabling them to behave as a set of nodes in a highly mobile network. A large amount of research and numerous development projects have been undertaken based on VANETs in the past few years, many of which have objectives such as network security enhancement, traffic conditions optimization, improving location services, improving road safety, reducing pollution etc. VANETs can be divided based on application into two areas, safety oriented VANETs and non-safety oriented VANET applications Non safety oriented VANET applications are used to increase the comfort of the people in the car. This also includes traffic conditions optimization and entertainment. These applications aim to improve the flow of traffic and also help to resolve any congestion on the road. Infotainment applications provide entertainment through internet access. These applications of VANETs are not required to be critically reliable and have lesser critical requirements and restrictions as compared to safety oriented applications. The safety oriented applications of VANET prioritise on saving the lives of human beings. Their main functions as such are sending messages to alert vehicles to avoid accidents, to detect accidents if they occur and to alert the nearest hospital, fire station and police station. These safety related applications are also used to save the time of the drivers with congestion related information up ahead on the road. The messages dispatched can also be alerts to give way for emergency vehicles, dangerous road ahead warnings and alerts informing of roadways under progress. This project is a safety oriented application of VANET. There are mainly two types of communication in VANETs, i.e., between two vehicles (vehicle to vehicle – V2V) and between vehicle and infrastructure (vehicle to infrastructure – V2I). Vehicles mostly communicate among themselves in V2V based VANETs. In V2I based VANETs, vehicles communicate with base stations. These protocols require continuous location tracking of all vehicles on the road and a fast, responsive alert system for timely alerts and warnings in case of collisions. The proposed project uses V2I architecture [10].

B. Machine Learning

Machine Learning can be defined as computers programmed to learn from experience instead of requiring specific programming with respect to every use case. It applies the principles of Artificial Intelligence (AI). Computer programs are developed which can use datasets to learn and then this processed knowledge can be used for taking decisions. The learning process begins with observations such as examples, datasets or instructions. The main aim is to look for patterns in the data and to predict future occurrences correctly. This can help to make better decisions.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
A.MariKirthima*, Assistant Professor,Department of CSE, B.M.S. Institute of Technology and Management , Bangalore, India.
RishabhVerma, student,Department of CSE, B.M.S. Institute of Technology and Management , Bangalore, India.
ChinmayiRajashekarHegde, ,student, Department of CSE, B.M.S. Institute of Technology and Management , Bangalore, India.
Arundhati S Shanbhag, student, Department of CSE, B.M.S. Institute of Technology and Management , Bangalore, India.

These are mainly mathematical models and algorithms which are continuously tweaked and adjusted to better suit the past in order to successfully predict the future. These algorithms can be generally classified into two categories, non-supervised learning algorithms and supervised learning algorithms. Machine learning algorithms that are supervised mainly use past collected data to predict future events. Such an algorithm analyses a dataset to train itself, and then builds a function to make predictions. The system can make very accurate predictions with lots of training data. This is further categorized into regression and classification algorithms. Regression algorithms are used when the target variable is a real value, such as price. Examples are Linear Regression, Polynomial Regression etc. Classification algorithms are used when the target variable is a category such as Yes/No etc. Examples are Tree-based algorithms like ID3, Random Forest etc, and others like Logistic Regression, Support Vector Machines (SVMs) etc. Unsupervised algorithms use unlabeled data that is not classified to train themselves. A system can use unsupervised learning to understand how it can create a function to best classify the hidden pattern within the data which is not labelled. It tries to draw distinct boundaries between different categories so that an unseen example may be correctly classified. These are further classified into Clustering and Association algorithms. Clustering algorithms are used when discovery of inherent groupings in the data is needed. Examples are K-Means Clustering, EM Algorithm etc. Association rule learning algorithms are used to discover rules that describe large portions of the data. For example: the Apriori Algorithm. Semi-Supervised Algorithms that use a combination of techniques of supervised and unsupervised learning are also used. They train themselves using all kinds of data without any restriction on the structure of the data.

II. PROPOSED SYSTEM

Currently used road traffic measurement sensors like metal plates, magnetic sensors and infrared sensors are not good for obtaining current location co-ordinates of a moving vehicle on the road. Accuracy of these traditional systems can be increased by using global positioning systems (GPS) and cameras. However, detection and prediction of potential accidents is useless without an alert system capable of sending quick responses. VANETs are also used in traffic data collection models but maintaining the communication and data feed is not easy under different road conditions [1]. The prototype developed aims to use Machine Learning algorithms to learn from previous accident scenarios so that forward collisions, rear-end collisions and accidents caused due to over-speeding can be predicted and prevented. Data is collected from various vehicles on the roads, acting as nodes and transmitting their distance to a central station at regular intervals. Using the collected data, the system then applies the KNN Classifier and predicts potential accident scenarios. An alerting system then warns the corresponding vehicles of the predicted incident and enables the drivers to prevent the same.

III. IMPLEMENTATION

The proposed system consists of four main phases namely: Data Collection, Data Processing, Prediction and Alert System.

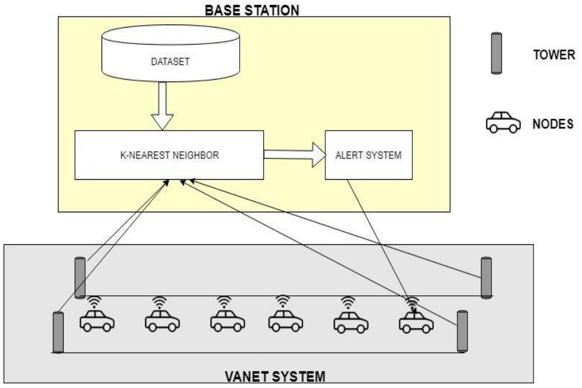

Fig 1 : Architecture of Proposed System

A. Data Collection

This phase is mainly concerned with timely collection of accurate vehicle location data. GPS has been extensively used for vehicle location detection. However, GPS has some drawbacks like signal obstruction due to obstacles and might give wrong values due to electromagnetic interference. Also, the accuracy of the position of a vehicle using GPS lies between 1000 and 3000 cm. However, a positioning system using wi-fi triangulation can be used to pinpoint the exact location of a wi-fi with an accuracy of up to 10 cm. Hence, a positioning system using wi-fi triangulation is used for obtaining the current location of the vehicle. Each vehicle moving on the road is equipped with a NodeMCU ESP8266 Wi-Fi module. Each module has an ID associated with it which helps in its identification. Vehicles moving on the road continuously broadcast their signals which are detected by strategically placed towers which act as receivers. In the towers, the NodeMCUWiFi module is used to scan and return the RSSI data regarding each WiFi hotspot within a 4m radius. A single scan takes around 60-120 milliseconds and around 4 complete scans can be performed in one second. These towers detect a vehicle's signal based on the signal strength and convert it to RSSI (received signal strength indication) data. The RSSI data is then sent to a central computer acting as a base station.

B. Data Processing

The RSSI data needs to be processed to extract location coordinates. Numpy arrays are used for storing RSSI values of each vehicle, collected from all towers prior to use in order to pinpoint the X, Y coordinates of the vehicle using polynomial regression. This numpy array is stored and used at the base station. Regression methods need to be used for location extraction. Several regression methods such as linear, multi linear and polynomial regression are available. But, RSSI data collected over time for a moving vehicle is a polynomial curve. Hence, a polynomial regression was the most apt choice for this purpose. The RSSI data collected at the base station for each node is given as input to the polynomial regression model. Machine learning module sklearn.preprocessing.PolynomialFeatures with four polynomial features is used to generate x,y coordinates. These x,y coordinates for each car is compared with the x,y coordinates of other cars travelling in the same lane to calculate the distance between the cars by finding the difference between them.

Direction of the moving vehicle is also found out by comparing a car's previous x,y coordinates with the current value of x,y coordinates.

C. Prediction Algorithm:

Several machine learning algorithms have been extensively used for action prediction in VANETs. [1] shows that KNN is the most accurate for predicting accidents when KNN, regression, Neural networks and C-means clustering method were compared [5]. KNN is memory-based approach. The classifier immediately adapts as we collect new training data. It allows the algorithm to respond quickly to changes in the input during real-time use. Hence, KNN was chosen for accident prediction. It is trained with respect to distance and direction parameters of the nodes using the data generated from the test runs. Test runs are done by moving the vehicles in same direction, opposite direction and the vehicle's location and direction parameters are continuously collected. This is used as the dataset for training KNN model. When a new instance needs to be predicted, the node's distance information along with the direction and node identifier are given as input to KNN classifier. The KNN outputs a 1 or a 0, 1 implying that the node is in danger of an impending accident, 0 implying that the node is safe. The KNN Algorithm Pseudocode:

1. Load the training and test data
2. Choose the value of K
3. For each point in test data:
   - Find the Euclidean distance to all training data points.
   - Store the Euclidean distances in a list and sort it.
   - Choose the first k points.
   - Assign a class to the test point based on the majority.
4. End

In the above algorithm, we have used K=5 i.e. 5 nearest neighbors are compared for each instance that needs to be classified.

D. Alert System:

The alert system is used to send signals to nodes which are in danger of an impending accident. Whenever the output from the KNN classifier is a '1', the alert system is notified about the node by using the node identifier. It immediately sends signals to the nodes in the form of flickering of LED lights installed on the nodes. The driver can then take the necessary precautions to evade the accident.

E. Prototype:

The prototype system hardware consists of the Arduino Mega 2560 board, a NodeMCU ESP8266 WiFi Board, NodeMCU ESP8266 WiFi Motor Shield, radio controlled cars, CAT5 Cable and MB102 Power Supply Boards.

The radio controlled vehicles mimic vehicles on a road in this prototype. The Arduino Mega board is used to collect the RSSI data by scheduling scans from all 4 towers and then sorts it on the ID. It acts as an intermediary between the base station, i.e., the computer and the VANET system. It can carry out 4 complete cycles of scans in one second. In the towers, the NodeMCUWiFi module is used to scan and return the RSSI data regarding each WiFi hotspot within a 4m radius. A single scan takes around 60-120 milliseconds and around 4 complete scans can be performed in one second. The towers and vehicles contain NodeMCU ESP8266 WiFi modules. They are used to broadcast the WiFi signals from the nodes. The modules have an ID associated with them to help with the identification. The NodeMCU ESP8266 WiFi modules plug into the NodeMCU Motor Shield which is a driver module and can be interfaced with motors, lights, beepers and buzzers using the motor control interfaces of this board. These boards are placed in each remote controlled car to control the lights using the board's motor control interfaces. This is used as an indication to warn in case of predicted collision and forms the alarm system. These also provide the power for the WiFi modules which are used to broadcast hotspot and receive any data from the base station. The WiFiboards placed on the remote controlled cars act as servers, i.e., if they receive an alert from the base station they automatically use the motor shield's interfaces to start flashing lights to warn of impending collision. A CAT5 cable offers unsurpassed data carrying capacity and speed of transmission of data. Hence, it is used to carry the data between the towers and the Arduino Mega. It supports two way connection with the help of multiple lines - Tx (Transmission) Line and Rx (Receiver) Line. The Tx line is used to schedule a scan and retrieve the data from theWiFi modules. The Rx line is used by the WiFi modules to send the scanned RSSI data back to the Arduino Mega. The MB102 board is required to provide additional power supply to each of the NodeMCUWiFi modules located in each of the towers. Each NodeMCUWiFi module requires input voltage of around 3.3v-5V & 250 mA of current. The Arduino Mega has 4 power outputs of around 3.3V-5V each. However, it cannot satisfy the large current consumption of around 1000mA of the four NodeMCUWiFi modules. Hence, a separate power supply system consisting of two MB102 boards, each powering two towers, is designed and deployed. The system hardware is programmed using C++ programming language. It has been used to program the Arduino Mega board, as well as the towers. It allows for better fine-grained control over the Arduino hardware as well as quick compilation as compared to other programming languages. The On-Board Unit (OBU) in each of the vehicles has also been programmed in C++. Each car has a different SSID and unique name which helps identify and alert the vehicle easily and quickly. Python 3.5 has been used to initiate the connection between the base station and the Arduino using a third party library, to process the collected RSSI data from the towers using polynomial regression, to convert this processed data into a structured form and to build the KNN model using this converted structured data for prediction. All the code deployed at the base station is written in Python. This helps for ease of use and easy modification of code. As the code is deployed at the base station, it can be modified and changes can be tested instantaneously. Also, this helps in compartmentalization of code. This helps to prevent any changes made in one part of the code from affecting another, properly working code section.

Any changes made in the base station will not affect the functioning of the hardware units in the VANET. It will also not affect any of the communication or alert protocols used in the VANET. Hence, if any errors occur at the base station, they are limited to the specific base station.

E. Overall Process:

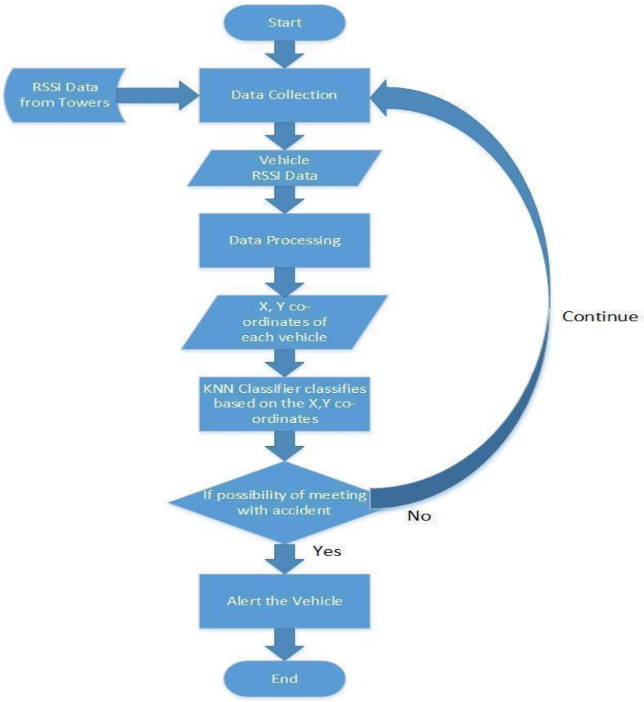

Fig 2 : Overall Process Flow Chart

Fig. 4.3.1 depicts all the major processes carried out at the Arduino and base station in one iteration. The system goes through these steps several times every second and carries out specified actions if certain conditions are satisfied. The Arduino Mega board collects RSSI data from all 4 towers and consolidates this data. This consolidated data is then transferred from Arduino Mega to the base station as a string. This process is repeated several times in one second. The data arriving at the base station from the Arduino is in the form of a string. Also, it is unstructured and contains jumbled data. This data must first be converted into a structured format, and then various RSSI values must be mapped to the vehicles and the towers. The base station proceeds to carry out this task and stores the resulting structured data in the form of a numpy array. The reason for storing this data in the form of a numpy array is to help to optimize computations over the data, to streamline it for machine learning algorithms and to take up lesser memory space for large amounts of data. Once the data has been converted into a structured form, it must then be further processed to obtain the location coordinates of each vehicle on the road. This process is carried out with the help of polynomial regression to convert RSSI values into distance in cm. The distances (in cm) of each vehicle from each tower is given as input to triangulation equations to obtain the location coordinates of each vehicle on the road. These coordinates are then passed on to the the K Nearest Neighbor Classifier. Based on a dataset, the K-Nearest Neighbour classifies the incoming data into a target value ‘Yes’ or ‘No’ where, ‘Yes’ indicates that the vehicle needs to be alerted of an impending accident. If the output from the KNN classifier is ‘No’, then the system goes through all the above specified steps again. In the case that output from KNN is ‘Yes’, the system immediately sends alert signals to the vehicles in danger.

IV. RESULTS AND ANALYSIS

The proposed system uses triangulation to locate the vehicles on the road. It is able to detect the location of vehicles (x coordinate) with an accuracy of ±90% as measured by comparing the x coordinates of the car on the prototype along with the calculated x coordinates. Vehicles that break traffic rules by over-speeding or by driving towards another car in speed are detected by the KNN algorithm. An alert signal is sent to such vehicles and all vehicles in its vicinity (30 cm radius) in such scenarios through the alert system. This is indicated in the prototype by blinking lights on the vehicles.

V. CONCLUSION

The implemented system shows how the VANET framework used along with KNN algorithm helps in predicting accidents on roads caused due to breaking traffic rules such as over speeding and wrong way driving. It also provides a GUI which can be used by the traffic department to monitor roads on the road and send help in case of an accident. The prototype is also able to successfully detect real time vehicle locations (x coordinate) with an accuracy of ±90%.

SCOPE OF FUTURE WORK

The implemented system fails to identify lane changes of a car due to inaccurate y coordinates. The flickering of RSSI data needs to be reduced is to accurately calculate y coordinates. Research needs to be carried out in this aspect. The system proposed in this paper does not take factors such as weather, road and driver conditions into account while predicting accidents. [6], [7] show that these conditions make up a huge part of accident causes. Hence, these factors also need to be taken into account while predicting accidents. The implementation costs also need to be reduced for the proposed system to be implemented in a real world scenario.

REFERENCES

1. Murat Ozbayoglu, GokhanKucukayan, Erdogan Dogdu, “A Real-Time Autonomous Highway Accident Detection Model Based on Big Data Processing and Computational Intelligence”, IEEE International Conference on Big Data (Big Data), 2016.
2. Kishwer Abdul Khaliq, Amir Qayyum, JurgenPannek, “Prototype of Automatic Accident Detection and Management in Vehicular Environment Using VANET and IoT”, 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA).
3. NejdetDogru, AbdulhamitSubasi, “Traffic Accident Detection Using Random Forest Classifier”, 15th Learning and Technology Conference (L&T), IEEE, 2018.
4. Dr.D.Selvathi, P.Pavithra, T.Preethi, “Intelligent Transportation System for Accident Prevention and Detection”, International Conference on Intelligent Computing and Control Systems ICICCS 2017.
5. YishengLv, Shuming Tang, Hongxia Zhao, “Real-time Highway Traffic Accident

Prediction Based on the k-Nearest Neighbor Method", International Conference on Measuring Technology and Mechatronics Automation, IEEE, 2009.

6. Ms. Gagandeep Kaur, Er. Harpreet Kaur, "Prediction of the cause of accident and accident prone location on roads using data mining techniques", 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2017.
7. Erik Hollnagel, "Understanding Accidents - From Root Causes to Performance Variability", IEEE 7th Human Factors Meeting Scottsdale Arizona, 2002.
8. Muhammad RizwanGhori, Kamal Z. Zamli, Nik Quosthoni, Muhammad Hisyam, Mohamed Montaser, "Vehicular Ad-hoc Network (VANET): Review", [EEE International Conference on Innovative Research and Development (ICIRD) 11- 12 May 2018.
9. Kishwer Abdul Khaliq, Amir Qayyum, J¨urgenPannek, "Prototype of Automatic Accident Detection and Management in Vehicular Environment Using VANET and IoT", 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2017.
10. Elias C. Eze, Sijing Zhang, Enjie Liu, "Vehicular Ad Hoc Networks (VANETs): Current State, Challenges, Potentials and Way Forward", Proceedings of the 20th International Conference on Automation & Computing, Cranfield University, 12-13 September 2014.

AUTHORS PROFILE

Prof. A. MariKirthima Assistant Professor Research Scholar Field of VANET BMSIT & M Bangalore, India

Mr. Rishabh Verma, UG student Field of VANET BMSIT & M Bangalore, India

Ms. Chinmayi Rajashekar Hegde UG student Field of VANET BMSIT & M Bangalore, India

Ms. Arundhati S Shanbhag UG student Field of VANET BMSIT & M Bangalore, India