Supervising Vehicle Using Pattern Recognition

Detecting Unusual Behavior Using Machine Learning Algorithms

SUBMITTED BY

Mihodi Hasan Lushan (14101222)

Manoshi Bhattacharjee (14301016)

Department of Computer Science and Engineering

BRAC University

SUPERVISOR

Dr. Tarem Ahmed

Associate Professor

Department of Electrical and Electronic Engineering

BRAC University

CO-SUPERVISOR

Mohammad Abdur Rahman (Adnan)

Lecturer

Department of Computer Science and Engineering

BRAC University

Declaration

We, hereby declare that this thesis is based on results we have found ourselves. Materials of work from researchers conducted by others are mentioned in references.

Signature of Supervisor **Signature of Authors**

Dr. Tarem Ahmed **Mihodi Hasan Lushan (14101222)**

Muhammad Abdur Rahman **Manoshi Bhattacharjee (14301016)**

ABSTRACT

Our lives are becoming busier day by day. We are consequently forced to delegate important activities to other people. In developing countries, the middle class often have paid drivers pick up their children from schools. What if the driver decides to deviate from the usual route into a seedy part of town with the child? What if it speeds and is driving recklessly? What if it gets into an accident? In our countries like us, supervising our vehicles when we are not present in it, and being notified if anyone else using it for any unwanted/illegal intention is of paramount importance in our country. Alarms are annoying, and we want to improvise the system in a smarter way for smarter monitoring. The proposed system is developed by applying Linear Regression models, kth-Nearest-Neighbor and Support Vector Machine classifier to identify a pattern and detect abnormal behavior of the vehicle.

Keywords—*Machine Learning; Pattern Recognition; KOAD Algorithm; KNN Algorithm; Linear Regression Algorithm; Support Vector Machine (SVM) Algorithm; Global Positioning System(GPS).*

ACKNOWLEDGEMENT

First of all, we would like to express our gratitude to Almighty Allah for keeping us safe and sound to initiate the research work and to put our best efforts and successfully complete it.

Secondly, we would like to thank and show our immense respect to our honorable Supervisor Dr. Tarem Ahmed and Co-Supervisor Muhammad Abdur Rahman for their enormous contributions, incomparable guidance and tireless support in conducting the research work and preparing the report. Their constant involvement and supervision ensured our gradual progress towards the completion of this thesis work. We are really grateful and humbled to have them as our supervisor.

We also thank our parents as well as our beloved friends for the patronage, moral support and aids. They helped us a lot with their valuable suggestions and direct or indirect participation which helped us maintaining a better workflow and achieving our goal.

Last but not the least, we would like to thank especially our very own BRAC University for providing us the opportunity to conduct this research.

Table of Contents

List of Figures

Chapter 1 INTRODUCTION

This chapter contains Motivation, Objectives & Goals and Thesis Report Overview which will introduce our thesis project. This chapter will give an overview of the thesis topic and other related information.

Detecting and tracking vehicle is a popular field. Tracking can minimize many issues regarding accidents, hijacking, kidnapping and many more illegal works. Monocular camera, active sensors, lidar and radars have been used to detect vehicles [1]. Image processing and IoT is working tremendously good in traffic sector. According to a patent by Johnson, Ashe and Yates far back in 1991, approximately 1.8 million of vehicles were stolen in the US costing USD 3.8 billion [2]. They showed a statistic that one vehicle is stolen in every 20 seconds. These figures may not be too different in developing countries today. Supervising vehicle is one kind of solution for these kinds of unwanted problems. There are many more ways to supervise vehicle. In this paper, we are using anomaly detection algorithm [3] for supervising vehicle. Anomaly Detection algorithm has been shown to exhibit general pattern matching characteristics, with applications outside IP networks [4-6]. Now the question is what type of anomaly can be detected? In this report, we are considering unusual behavior or unexpected pattern of vehicle as anomaly. We have developed a real time anomaly detection algorithm that maintains a dictionary of input vectors which considers GPS data, timestamp, speed and weight of the vehicle. In this modern era of technology, almost all models of vehicles have built-in sensors for determining GPS, speed and weight. Having these kinds of facilities why are we not exploiting those for ensuring our safety? Some existing systems do track our vehicles but, they are not doing their job accurately. By using those systems, we get a full overview of vehicle's movements, however, those systems don't take vigorous decisions for us. Therefore, our fundamental focus is to boost these systems or models and developing identification process of anomaly to make them smarter for integrating our algorithm.

1.1 MOTIVATION

Machine learning is the most demanding topic in computer science. More than half of all current ongoing researches are on this machine learning field. Anomaly detection algorithm is one of the most rarely used machine learning algorithm in implementation. GPS, although an old technology, being used in our daily lives always even if sometimes maybe we don't know. Our usage of GPS can be estimated in Google Maps Timeline.

Being so much available, research regarding machine learning and GPS technology to supervise our loving vehicles has not been emphasized. Our main inspiration comes from Muhammad Abdur Rahman Sir's great support and Dr. Tarem Ahmed Sir's own anomaly detection algorithm [3].

1.2 Objective and Goals

Our objectives and goals for this thesis are given bellow,

- Decreasing Vehicle theft
- Real-time pattern anomaly detection with 100% accuracy
- Decreasing hijacking or abduction or kidnapping
- Surveillance on own vehicle

1.3 Methodology

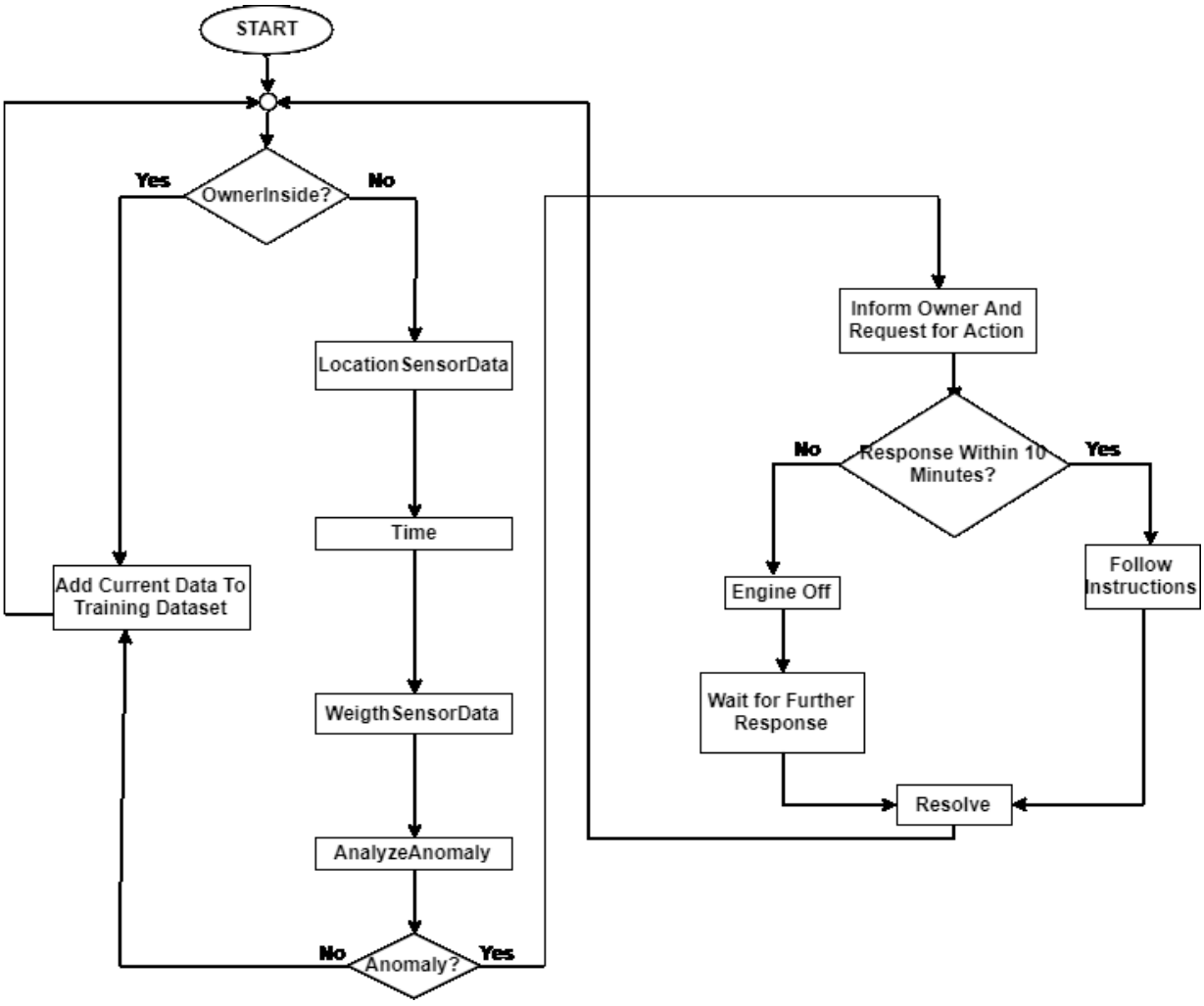

Fig. 1.1 Proposed System Methodology

We have made two mobile apps for interactions. One mobile should be fixed on the vehicle. Another app will be running in background of vehicle owner's mobile. Vehicle's app will be sending location data to our cloud server. Our cloud server receives the data and pass the data to the anomaly detection algorithm. Our python code executes the algorithm and predicts while there is any anomaly or not. If it founds any unknown behavior, it invokes the owner immediately. Otherwise, current data is stored as training data.

Owner being notified the anomaly, checks current location of his vehicle. Owner can also communicate with vehicle app to stop the vehicle engine (This feature is not yet implemented, this is our future plan).

We have our plans to change the methodology to build our own embedded system for vehicle to replace vehicle app. By building our own embedded system we can access the vehicle door open/close information with the total weight of the vehicle with passengers. This can help our algorithm being more accurate. These types of devices are available on market, but those are not fully cost efficient, customizable for our algorithm parameters.

1.4 THESIS REPORT OVERVIEW

The organization of this report is as follows. Chapter 2 is **Literature Review** presented with the study of **GPS** and **Machine Learning Algorithms** and **Related Works** similar to our work. Chapter 3 covers the details of **Algorithms** used in this thesis. Chapter 4 is designed with the **Methodology** of implementation. **Result Analysis and Comparison** of accuracy of the predicted results using all algorithms are described in Chapter 5. Lastly, Chapter 6 represents our **Future Plan and Concludes** the report.

Chapter 2 LITERATURE REVIEW

This chapter contains literature review related with Background which includes Machine Learning Algorithm (ML) and Global Positioning System (GPS).This chapter also refers Related Works. Besides, this chapter will also give information about our research activity.

2.1 BACKGROUND STUDY

We studied about the GPS and machine learning algorithms. We have studied several researches in these fields.

2.1.1 Machine Learning algorithms

Statistical analysis have been used since decades. After 1950, these analysis got light of computer science. In 1960, Bayesian methods are introduced to infer something after analyzing probabilistically [15]. After this, applications of machine learning in solving different problems evolved. In modern era, we use supervised learning algorithms, unsupervised learning algorithms and reinforcement learning algorithms in respect to our problems.

Machine learning algorithms are being used now-a-days in robotics and even in small mobile applications. The highest tech giant Google announced they are implementing machine learning even in their photo viewer application in Google Photos. Machine learning algorithms made the face recognition more accurate which enhances the field of image processing. A learning model using TensorFlow is 99 percent accurate in predicting traffic flows [17]. Increasing tools for applying machine learning algorithms are being available for visualizing large data. Researching for years, we are using commands in Matlab, Sckit-Learn, TensorFlow and others even in some IDEs like Codota.

2.1.2 GPS

Formerly GPS (Global positioning System) was known as Standard Positioning Service (SPS) [11]. For locating lost soldiers or instantly calculate some math required while sending a missile, global positioning system was announced in 1973 [12] by United States (US) Department of Defense [13]. Later in 1978 first satellite was sent [12] and in 1989, GPS-2 (Navstar) satellite was sent and GPS was opened public [14]. In 1993 launch 0f 24 satellites and constantly communication from any receiver on the earth with four satellites was ensured.

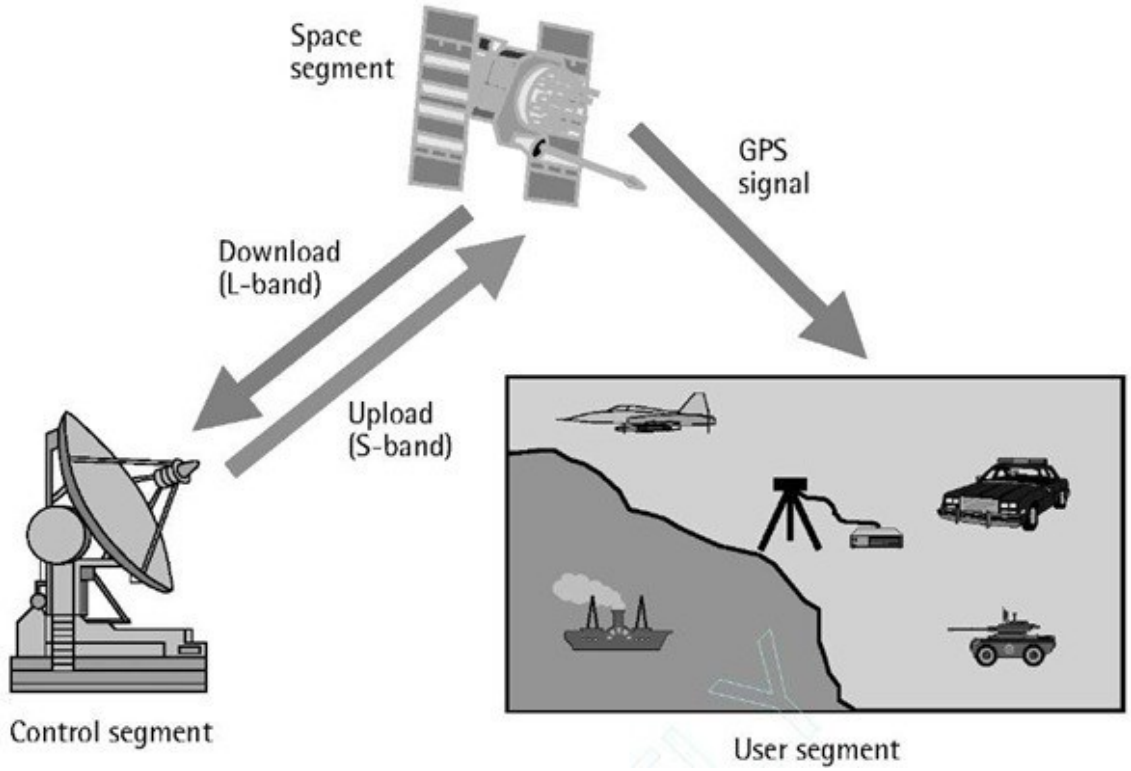

Fig. 2.1 GPS Segments [12]

Initially, accurate time locating was a leading factor. Later on, time was synced with all satellites. Scientific studies related to earthquake and synchronization of telecommunication networks GPS is used for time precision. Most common use of GPS is estimating travel time in google maps.

2.2 Related Work

We didn't find exactly same works like we did. But there are some similar works done by great researchers. We studied some of those papers and tried understanding their approach.

2.2.1 Fast Compressive Tracking System

Supervising vehicle using pattern recognition is one of the major research areas of computer vision. A lot of algorithms are used to supervise vehicle using different features. We are inspired by the work of Mekkayil and Ramasangu [1]. Our paper is closely related to their work. They have used Fast Compressive Tracking (FCT) Algorithm, MFCT algorithm and Naive Bayes classifier to select sample with Haar-like feature extraction to detect the vehicle.

2.2.2 Pattern Mining From Big Data

Another work is done [7] by Alan et al. They have worked on big historical traffic data to find anomalous traffic pattern. They have applied different models of regression and regression trees

to find daily traffic patterns. Using machine learning algorithm (ML) they have extracted big traffic data into small dataset to analyze them. In this way, by comparing small dataset with regular traffic patterns they have tried to identify abnormal traffic patterns. Moreover, we have also proposed linear regression model in our work to find out the pattern utilizing our training dataset.

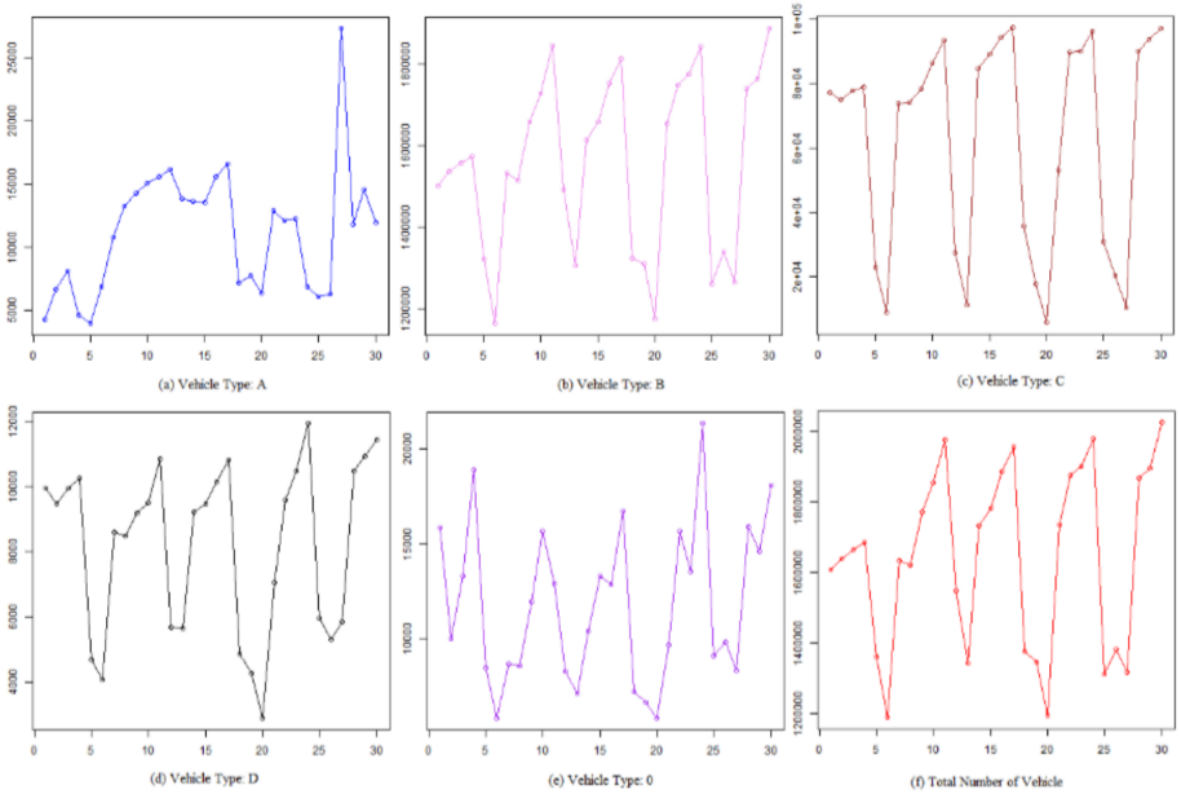

Fig. 2.2 Historical average of traffic flow for each vehicle type in April 2014, Porto City, Portugal [7]

2.2.3 Pedestrian Detection for Autonomous Vehicle

Harshitha and Manikandan in [8] have used the technique of Histograms of Oriented Gradients (HOG) feature extraction and Support Vector Machine (SVM) classifier to detect real-time pedestrian for autonomous vehicle. With the help of HOG feature extraction they have tried to find out the existence of any pedestrian from input image. The SVM classifier is used for finding a hyper route from a higher dimensional feature space. By using both of these two features they differentiated pedestrian and background separately.

2.2.4 Cluster Based Map for Vehicle Tracking

Vehicles like public transports which includes buses, taxis, cars and tracking them at a time is a challenging task. However, Tomas, Alexander and Sasi made it easy in [9]. They have used high performance cluster based method to store GPS (Global Positioning System) data of huge number of vehicle; also, to process and retrieve the data. This method is also used for supervising and

plotting the vehicle locations as cluster. After collecting the GPS data they sent them to the tracking application in regular basis and again stored them in a database. For completing the whole process they used the GPS tracking device which fits on to the vehicle. By using this device they located the GPS location, GSM data and GPRS status regularly which helps to give end users a map view with vehicles details.

2.2.5 Obstacle Detection and Classification

For tracking high speed autonomous driving Prabhakar et al. invented an algorithm [10]. They have used Faster Region-Based CNN (R-CNN), SVM classification and have taken data set of PASCAL VOC 2012 and applied ZF net (a pre-trained network model) to detect on-road obstacle such as vehicles, animals and pedestrians.

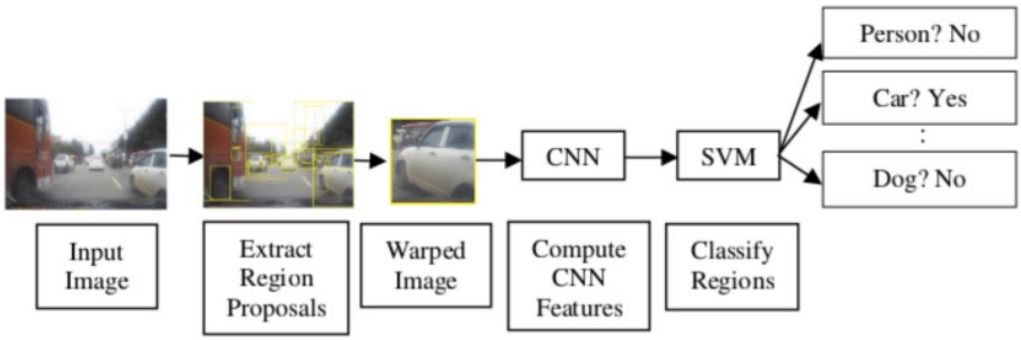

Fig. 2.3 Functionality of R-CNN [10]

Chapter 3 ALGORITHMS

This chapter contains Algorithms of kth-Nearest-Neighbor, Linear Regression and Support Vector Machine (SVM) which have been used for our research purpose. This chapter also represents Data Preparation, Parameter Selection and Algorithm Complexity.

Pattern Recognition is a branch of machine learning and Anomaly detection algorithm is available in supervised, unsupervised and reinforcement algorithms. Our problem is a classification problem and we have approached in supervised learning approach. In our research work, we worked with several algorithms like kth nearest neighbor, linear regression and support vector machine with our datasets as they are classification based algorithms. Sometimes we run the same algorithm using different tuning parameters to get a better result. Several machine learning algorithms gave us different result with percentage accuracy. Bellow, all those algorithms are described with analysis and comparison for this dataset in our observation.

3.1 DATA PRE-PROCESSING

Considering a minimum threshold of 0.0001 latitude and 0.001 longitude, we defined our targets to be usual and unusual. Within the threshold range of GPS location and following a pattern with timestamp is defined as usual. Otherwise it is defined as unusual in the training data.

asset	latitude	longitude	time	speed	HEADING	REASONS	DISTANCE_TRAVE	ODOMETER	LOADTS	STATUS
185	39.143687	-84.536837	2/5/2018 12:22	0	NNE	9	1.5	74986.9	2/5/2018 12:22	42015
185	39.141605	-84.509184	2/5/2018 12:04	26.4	N	11	0	5230.4	2/5/2018 12:06	40991
185	39.105384	-84.566204	2/5/2018 12:12	0	N	9	0	114177.1	2/5/2018 12:12	42015
185	39.137153	-84.475613	2/5/2018 12:07	0	N	12	0	18418.1	2/5/2018 12:07	8223
185	39.150476	-84.544234	2/5/2018 12:22	29.6	SSW	11	0.6	36623.2	2/5/2018 12:22	8429599
185	39.118228	-84.577774	2/5/2018 12:08	6.2	SSW	10	0	60170.6	2/5/2018 12:08	9247
185	39.185808	-84.607008	2/5/2018 11:59	46.2	W	11	0	113424.7	2/5/2018 12:02	40991
185	39.174708	-84.434502	2/5/2018 12:13	6.7	E	11	0.1	30614	2/5/2018 12:15	8429599
185	39.12189	-84.474384	2/5/2018 12:00	7.5	W	10	0.5	42375.6	2/5/2018 12:00	42015
185	39.116754	-84.439825	2/5/2018 12:07	0	SW	9	1.5	65769.3	2/5/2018 12:07	8430623
185	39.123927	-84.494294	2/5/2018 12:07	0	W	9	0.1	3564.8	2/5/2018 12:07	42015
185	39.112412	-84.516645	2/5/2018 12:06	4.2	N	10	0	110097.4	2/5/2018 12:06	42015
185	39.119678	-84.532231	2/5/2018 11:58	0	W	9	0.7	54956.1	2/5/2018 11:58	42015
185	39.112794	-84.574779	2/5/2018 12:02	0	NW	9	0.7	16847.9	2/5/2018 12:02	8430623
185	39.079762	-84.405987	2/5/2018 12:16	0	NW	9	0.8	3624.8	2/5/2018 12:16	42015
185	39.174853	-84.602353	2/5/2018 11:57	29	NNW	11	1.5	113423.5	2/5/2018 11:59	40991
185	39.077919	-84.609479	2/5/2018 12:23	4.5	W	10	0.5	5153.4	2/5/2018 12:23	8430623
185	39.152579	-84.563137	2/5/2018 11:55	26	NNW	11	0.5	51111.6	2/5/2018 11:56	8429599
185	39.072078	-84.420298	2/5/2018 12:18	20.7	NWW	11	1.8	3625.8	2/5/2018 12:20	40991
185	39.087728	-84.385864	2/5/2018 12:25	3.7	W	11	0.9	68863.8	2/5/2018 12:25	40991
185	39.109288	-84.527425	2/5/2018 12:02	0	W	9	0.4	39550.8	2/5/2018 12:02	42015
185	39.11853	-84.599155	2/5/2018 11:55	24.5	NNW	11	0	60156.6	2/5/2018 11:56	8429595
185	39.165898	-84.513416	2/5/2018 12:20	8.7	NW	10	3.1	2185.4	2/5/2018 12:20	42015
185	39.263518	-84.436499	2/5/2018 11:56	3.9	W	10	0.5	41162.9	2/5/2018 11:56	8430623
185	39.144008	-84.537706	2/5/2018 11:55	16.8	W	11	0	62850.4	2/5/2018 11:56	40991
185	39.13626	-84.535287	2/5/2018 12:20	3	NEE	10	0.4	60156.5	2/5/2018 12:20	42015
185	39.093525	-84.573564	2/5/2018 12:00	0	NW	9	0.1	46483.4	2/5/2018 12:00	8430623
185	39.1611	-84.528703	2/5/2018 12:17	13.7	SW	10	1	74984.9	2/5/2018 12:17	42015
185	39.104346	-84.562456	2/5/2018 12:11	7.2	S	10	0	4537.3	2/5/2018 12:11	42015
185	39.110564	-84.496364	2/5/2018 12:10	5.7	W	10	0.2	140281.9	2/5/2018 12:10	42015

Fig. 3.1 Dataset structure

	date	day	hour	latitude	longitude	minute	month	target	year
0	5	0	12	39.143687	-84.536837	22	2	0	2018
1	5	0	12	39.141605	-84.509184	4	2	0	2018
2	5	0	12	39.105384	-84.566204	12	2	0	2018
3	5	0	12	39.137153	-84.475613	7	2	0	2018
4	5	0	12	39.150476	-84.544234	22	2	0	2018
5	5	0	12	39.118228	-84.577774	8	2	0	2018
6	5	0	11	39.185808	-84.607008	59	2	0	2018
7	5	0	12	39.174708	-84.434502	13	2	0	2018
8	5	0	12	39.12189	-84.474384	0	2	0	2018
9	5	0	12	39.116754	-84.439825	7	2	0	2018
10	5	0	12	39.123927	-84.494294	7	2	0	2018
11	5	0	12	39.112412	-84.516645	6	2	0	2018
12	5	0	11	39.119678	-84.532231	58	2	0	2018
13	5	0	12	39.112794	-84.574779	2	2	0	2018
14	5	0	12	39.079762	-84.405987	16	2	0	2018
15	5	0	11	39.174853	-84.602353	57	2	0	2018
16	5	0	12	39.077919	-84.609479	23	2	0	2018
17	5	0	11	39.152579	-84.563137	55	2	0	2018
18	5	0	12	39.072078	-84.420298	18	2	0	2018
19	5	0	12	39.087728	-84.385864	25	2	0	2018
20	5	0	12	39.109288	-84.527425	2	2	0	2018
21	5	0	11	39.11853	-84.599155	55	2	0	2018
22	5	0	12	39.165898	-84.513416	20	2	0	2018
23	5	0	11	39.263518	-84.436499	56	2	0	2018
24	5	0	11	39.144008	-84.537706	55	2	0	2018
25	5	0	12	39.13626	-84.535287	20	2	0	2018
26	5	0	12	39.093525	-84.573564	0	2	0	2018
27	5	0	12	39.1611	-84.528703	17	2	0	2018
28	5	0	12	39.104346	-84.562456	11	2	0	2018
29	5	0	12	39.110564	-84.496364	10	2	0	2018

Fig. 3.2 Dataset Structure after pre-processing

3.2 KTH-NEAREST-NEIGHBOR

k-th Nearest Neighbor is a supervised learning algorithm. In this algorithm, it takes features and targets and plots them in a graph. All labeled featured data being segmented in target groups and then calculate 'Euclidean Distance' [18] [19] [20] between new feature data and training feature data. Tuning n_neighbor value groups in training data and the prediction varies. Suppose, we have n feature data with two dimensions. So the euclidean distance x is,

$$x = \sum_{i=0}^{n} (F_{1i} - F_{2i})^2 \tag{1}$$

3.3 LINEAR REGRESSION

Let we have training n dataset X with c columns. Each column of the dataset holds unique attribute related to algorithm. Considering X as a vector space, mathematically,

$$x_i = w_{i1} w_{i2} w_{i3} \ldots\ldots w_{ip} \quad (2)$$

Each column wc is the subspace of the vector space X. From every i'th row Xi there is a target array Ti,

$$T_i = x_i \beta \quad (3)$$

From the equation (3), β is the vector of constants. This constant is measured through the graph plotted with the dataset and being used to measure the test features and to predict test result. It uses nearest subspace approach [16].

Suppose, we want to measure test result Tt,

$$T_t = x_t \beta \quad (4)$$

We got β by plotting previous training data in equation (3).

3.4 SUPPORT VECTOR MACHINE

Support vector machine is one of the machine learning model algorithm. Support vector machine is used in classifying related problem. Using kernel mapping SVM separates the hyperplane (maximal margin) for two classes [21]. Support Vector Machine is also known as non-linear learning pattern [22]. As we want to detect a pattern in our given GPS and timestamp data, we chose SVM to check our result accuracy. Support vector machine algorithm works best with the outliers in the dataset in linear kernel. Scikit-Learn [23] provided with classifier and necessary functions with all documentation to train and test the dataset.

3.5 PARAMETER SELECTION

Vehicle GPS Data: Department of Public Services [24] contains eleven columns. In this paper, we are using Latitude, Longitude and time column as our feature vectors of the algorithms. Additionally, we created target vector using our own threshold algorithm. From the time column we get year, month, date, hour, minute, second and weekday and been passed as feature.

3.6 ALGORITHM COMPLEXITY

Algorithm efficiency is prioritized in our system because we chose cloud based real-time anomaly detection. For KNN algorithm, it takes O(ndk) time where k is fixed and n and d is the

dimension of our dataset. SVM algorithm is also dataset dependent in case of efficiency. SVM takes O(nfeatures x n2samples) to O(nfeatures x n3samples) time to run and predict the output. Running linear regression takes O(c2 x n) where c is the number of features and n is total observations.

Chapter 4 IMPLEMENTATION

This chapter describes about our proposed model training and data collecting mobile application. This chapter also describes how data collecting application has been developed and used to collect our local data.

4.1 MODEL TRAINING

Anomaly detection algorithm is a broad concept. Several algorithms of data science falls under this category. We used three algorithm, as early described kth-Nearest-Neighbor, Linear Regression and SVM.

What we had was only raw location data of several vehicles at first. We started with that data firstly. For our job done, we had to pre-process the data with our own knowledge. After studying the data we introduced and apply our pre-processing algorithm to fit it in machine learning algorithms. Later, after getting a satisfactory result, we collected our own data using our own mobile application. Our very close Hossain Al Muhee, also a student of department of computer science and engineering of BRAC University provided us with very classified and authentic data. As this was done using our own application, this is very much credible data for this research Scikit-Learn [23] is a python library to train our machine learning model with our dataset and test the data with future prediction. Scikit-Learn provides almost all necessary algorithms with proper documentation to start experimenting data science algorithms. We have used their provided functions to train the dataset to learn our model, split the dataset into 70 percent to train and 30 percent to for testing. We compared the test output with the given data to find out the accuracy.

4.2 DATA COLLECTING APPLICATION

After testing the data from United States of America, it was too urgent to train our machine with some data from Dhaka. For that, we have implemented an android application with background service. The application starts when user press START button and continuously gets location updates of the user from the background. We tried trading off in collecting more data and mobile battery. We have implemented Google Location Services API for developing the application. For storing our data, we used firebase realtime database. Using Google Location API, we set the parameter ‘ACCURACY_HIGH’ for getting exact location. As we needed continuous location stream data, it was a big concern if the network fails while driving with the application.

To solve this issue, we had to made this application forcefully save the data while offline and sync that properly with the server database while the network connection is available.

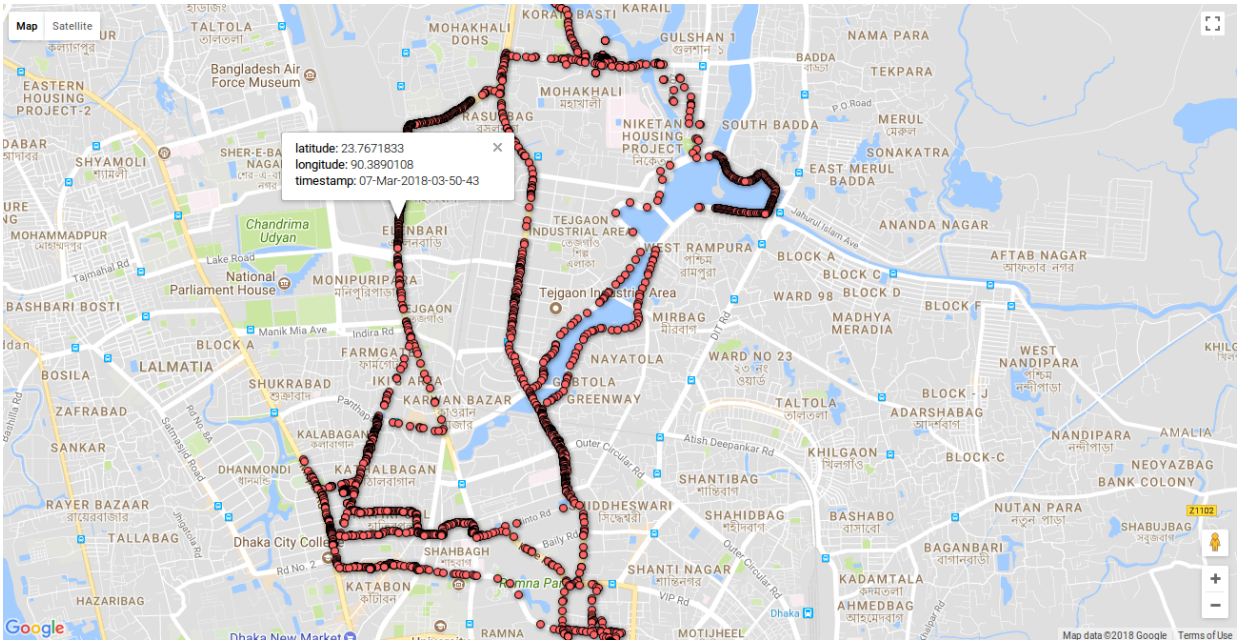

Fig. 4.1 Our Local Data Visualization

As we have our plans to migrate this application to the final application, it is necessary to have different accounts. We implemented the individual accounts and login system with sign in with Facebook or Google so that user does not need to bother about credentials.

Chapter 5 RESULTS

This chapter shows the outcomes of the training and testing datasets. The same datasets are also used for training and testing and k-NN algorithms, Linear Regression and Support Vector Machine for comparing the accuracies generated by the following algorithms. This chapter also describes Final Model Selection comparing the accuracies of the algorithms.

Changing the values of k in kth-Nearest-Neighbor algorithm changes the accuracy of the test in a large scale. k value indicates the measurement of how much data the algorithm will count for its calculation. For example if the system is made for k=3, then the algorithm assumes that possible 3 nearest data points from all plotted points in a graph should be measured. Measure is the Euclidian distance. Here the algorithmic complexity is very simple. Comparing he complexity of clustering 3 location points is always lower than clustering 10 location points. But, results are more accurate in larger clusters. We worked with a small amount of data of a single vehicle of 1 month. Our research work is a long-term-plan and our supervised algorithm works best with large scale training data. So, we are concerned about algorithmic complexity and tested all possible measures. For linear regression and SVM, we did the same.

5.1 RESULT ACCURACY

Here we will discuss and compare between the accuracy results using the same dataset in all algorithms we used.

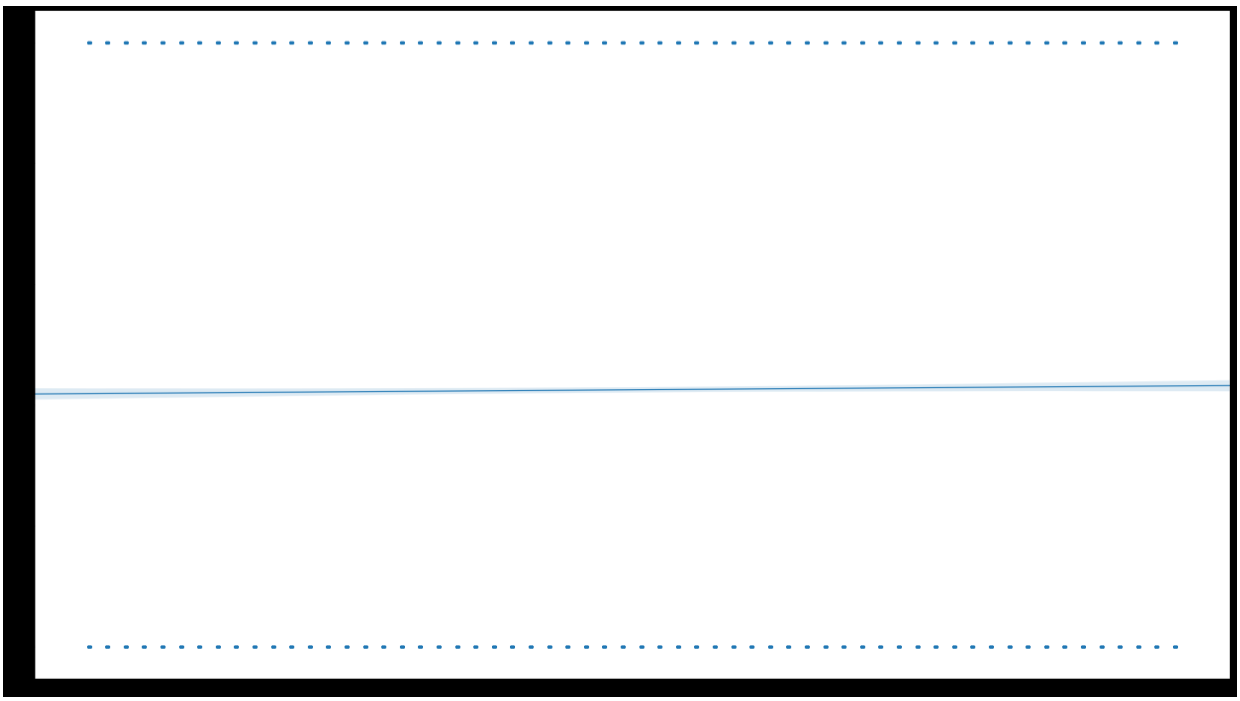

Fig. 5.1 Minute vs Target

Here, in this Fig. 8, we have plotted the minute column data from our pre-processed dataset. In this figure, minute is the variable and target is constant. Thus, target is set in Y axis and minute goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the minute data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

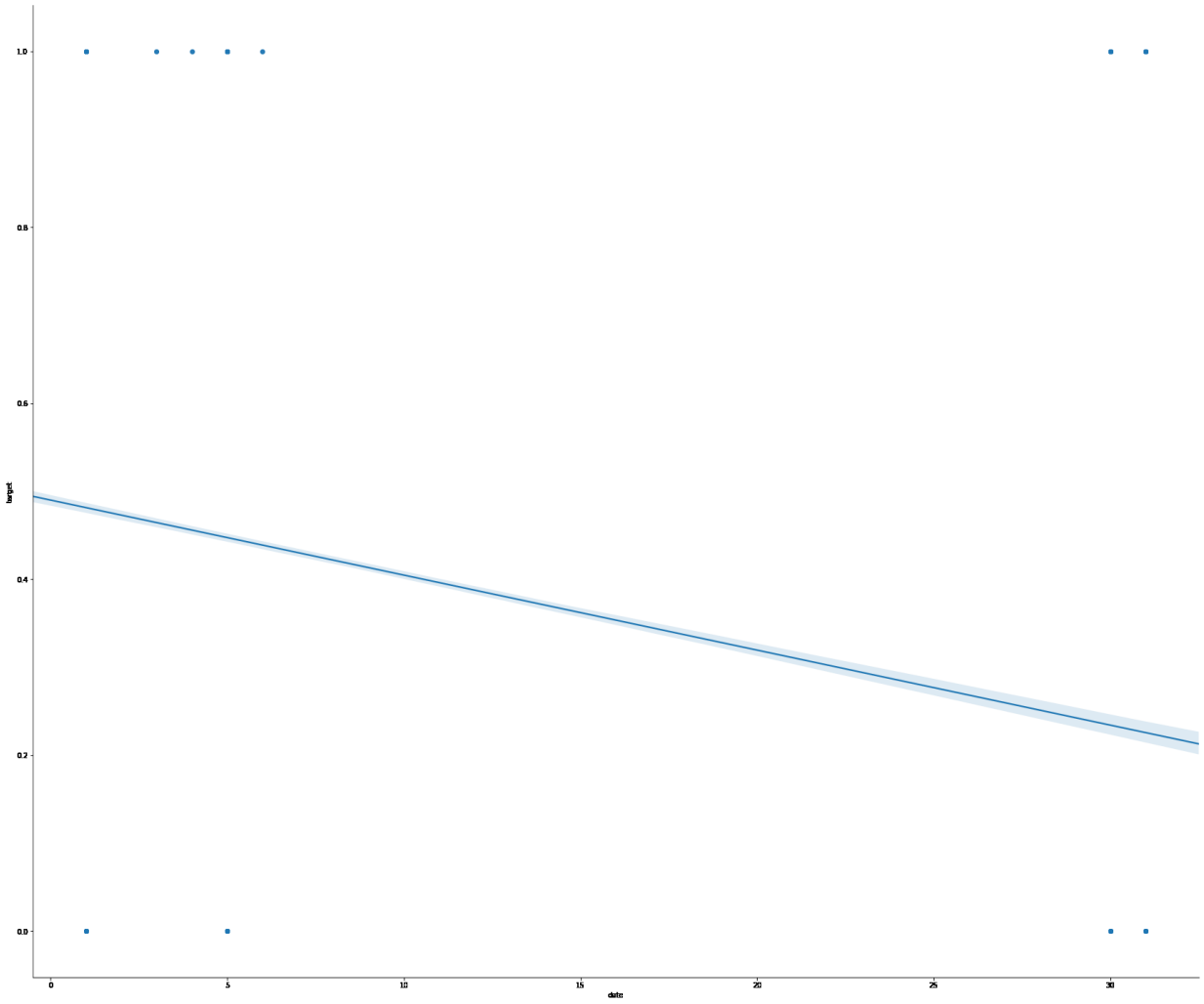

Fig. 5.2 Date vs Target

Here, in this Fig. 9, we have plotted the day column data from our pre-processed dataset. In this figure, date is the variable and target is constant. Thus, target is set in Y axis and date goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the date data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

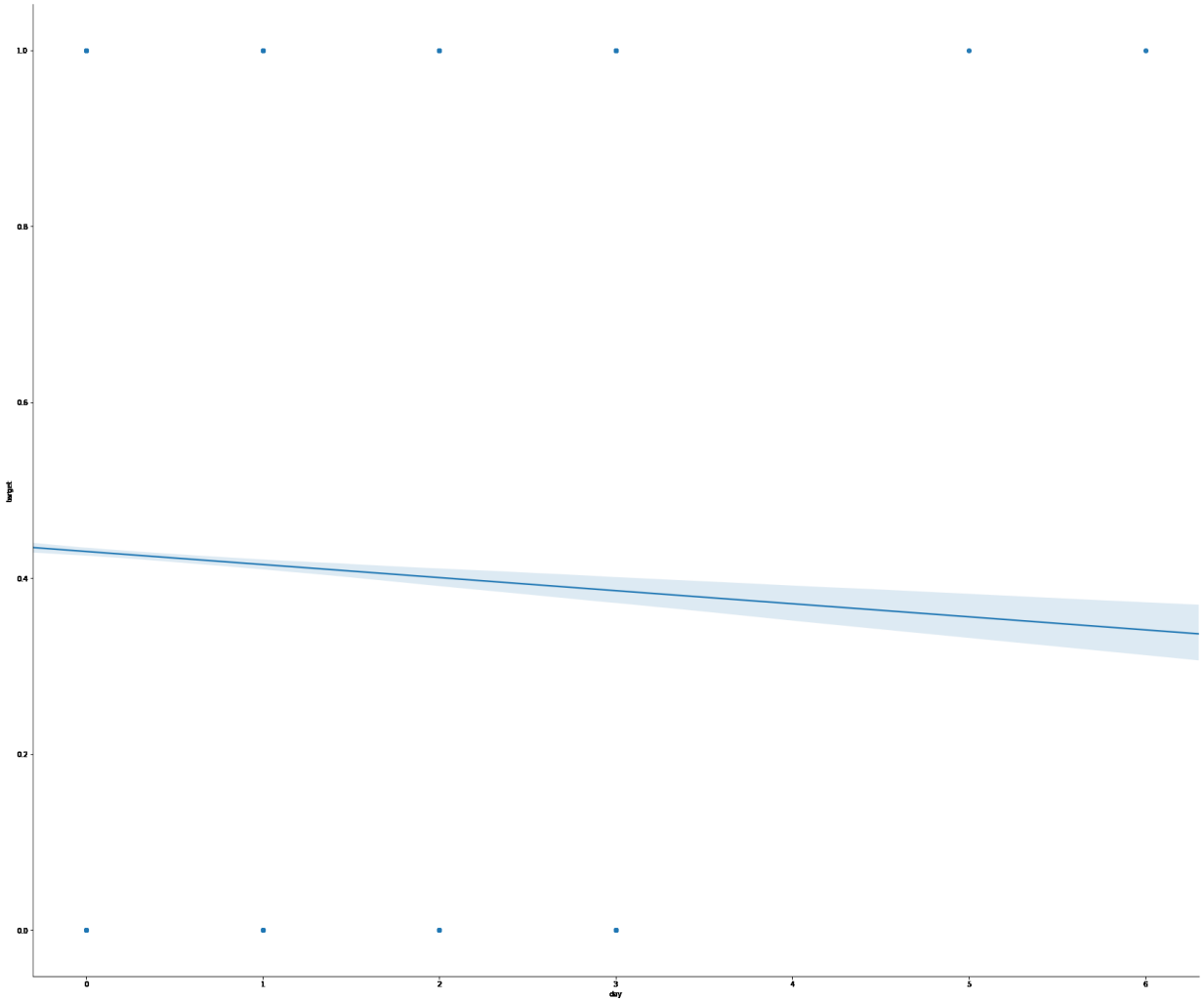

Fig. 5.3 Day vs Target

Here, in this Fig. 10, we have plotted the week day column data from our pre-processed dataset. In this figure, minute is the variable and target is constant. Thus, target is set in Y axis and week day goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the week day data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

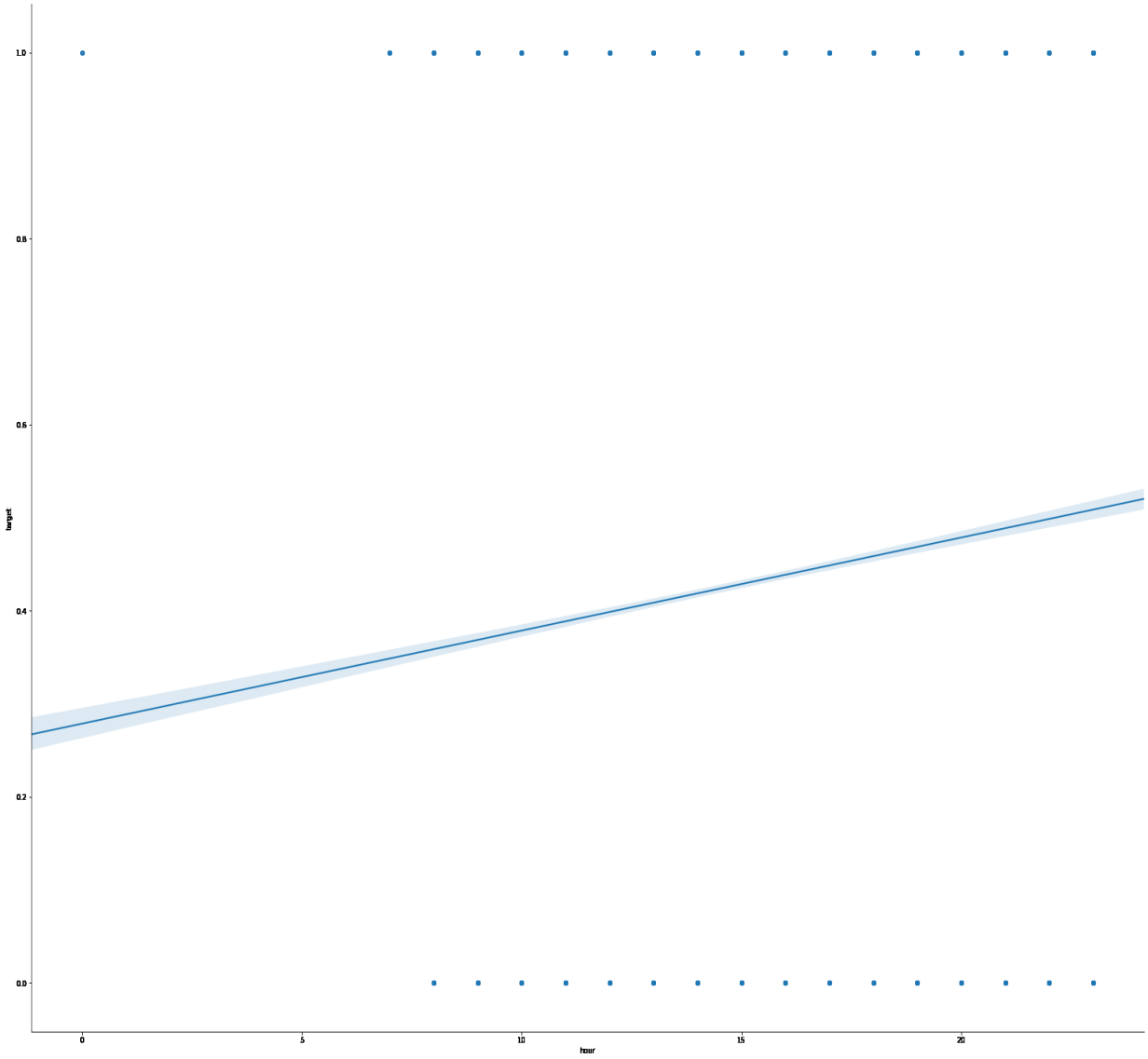

Fig. 5.4 Hour vs Target

Here, in this Fig. 11, we have plotted the hour column data from our pre-processed dataset. In this figure, hour is the variable and target is constant. Thus, target is set in Y axis and hour goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the hour data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

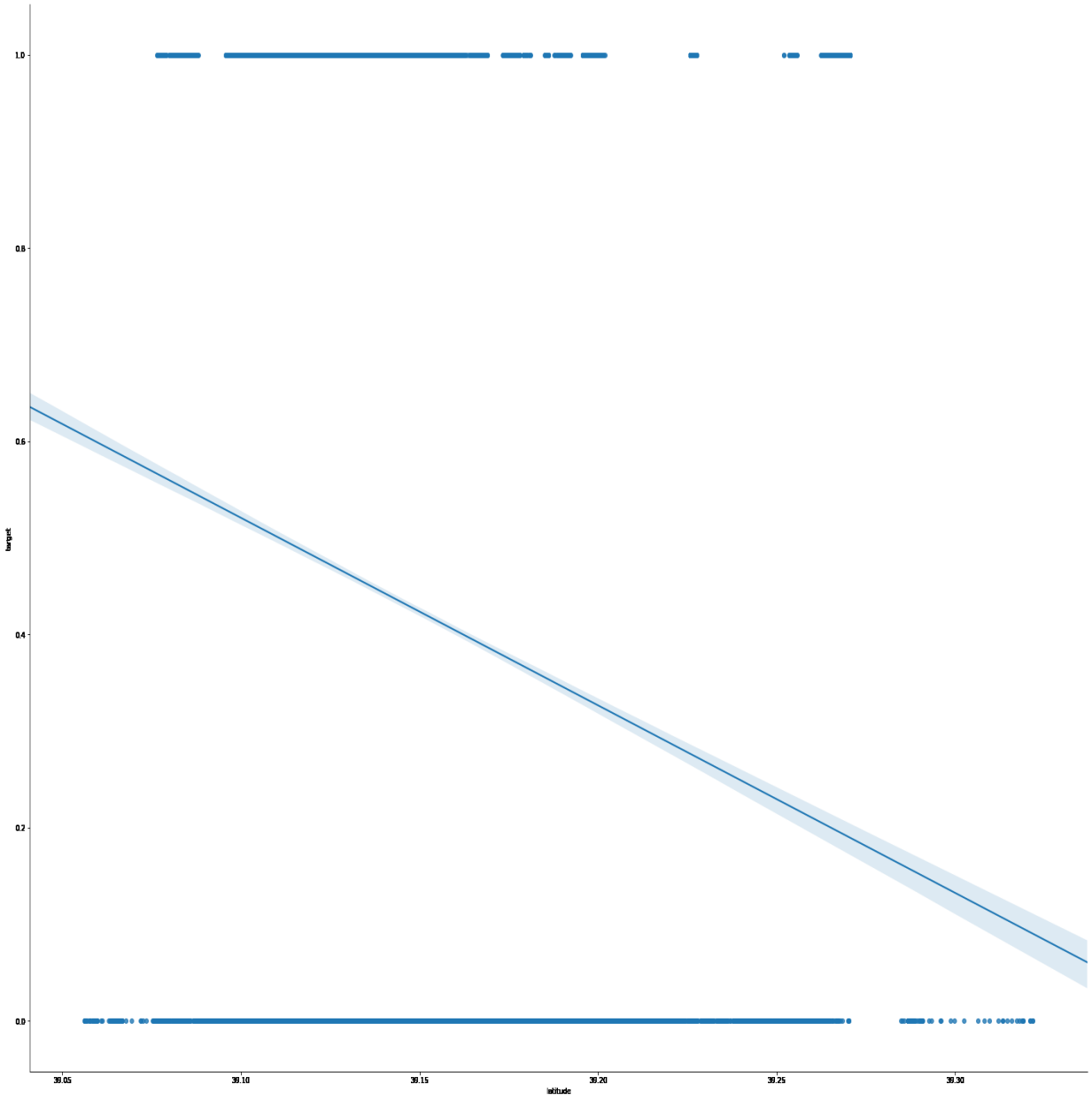

Fig. 5.5 Latitude vs Target

Here, in this Fig. 12, we have plotted the latitude column data from our pre-processed dataset. In this figure, latitude is the variable and target is constant. Thus, target is set in Y axis and latitude goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the latitude data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

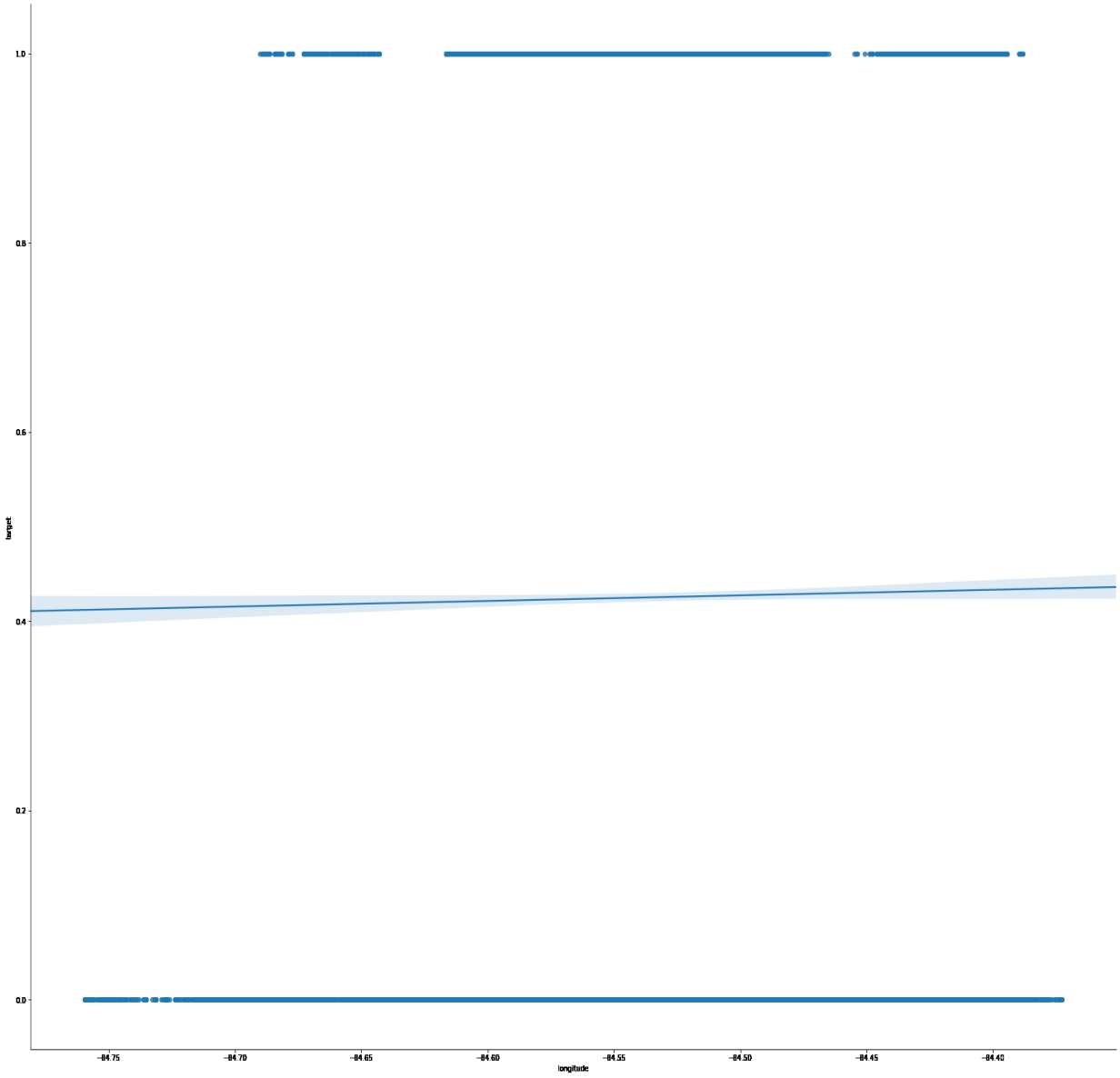

Fig. 5.6 Longitude vs Target

Here, in this Fig. 13, we have plotted the longitude column data from our pre-processed dataset. In this figure, minute is the variable and target is constant. Thus, target is set in Y axis and longitude goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the longitude data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

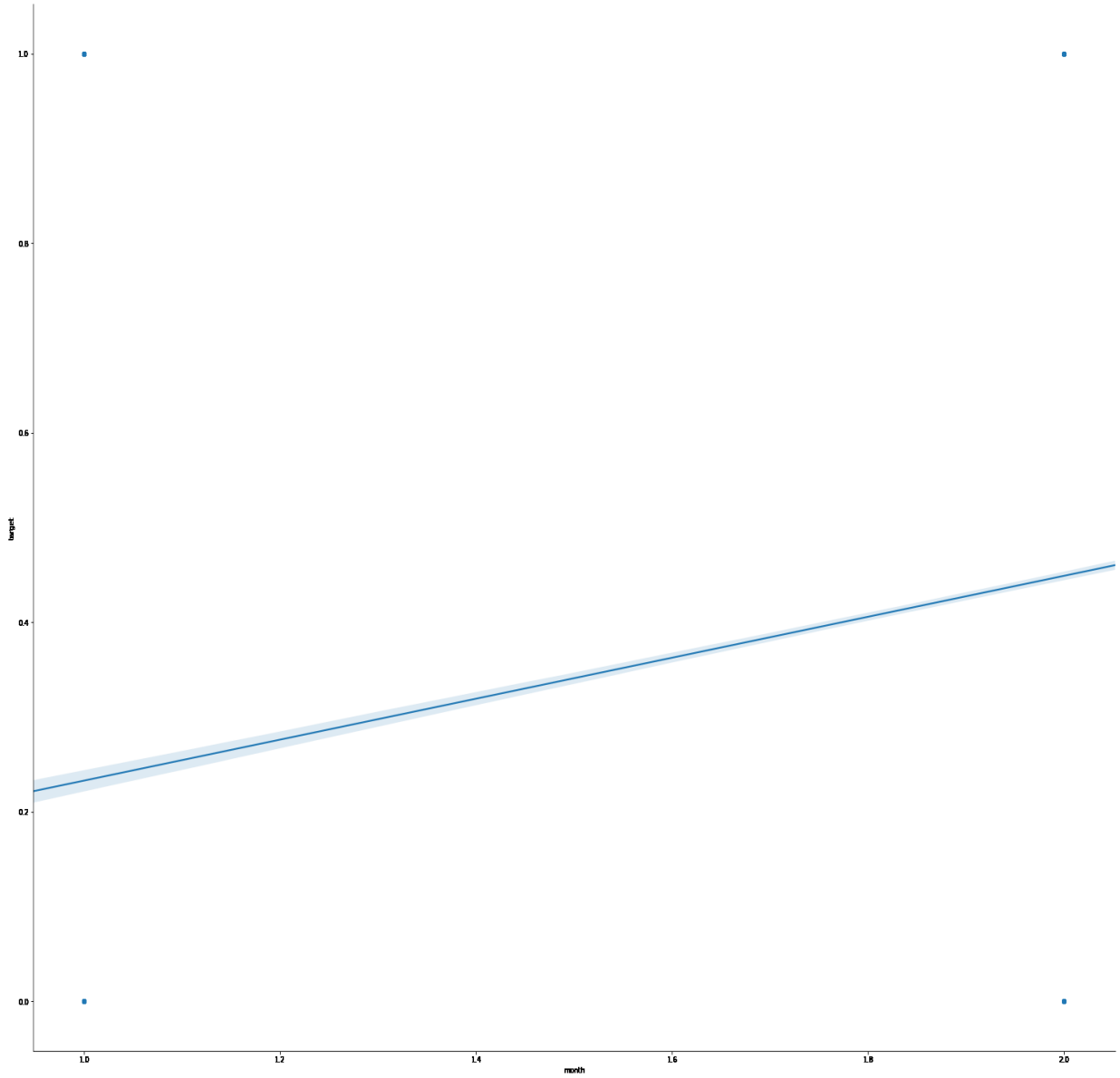

Fig. 5.7 Month vs Target

Here, in this Fig. 14, we have plotted the month column data from our pre-processed dataset. In this figure, month is the variable and target is constant. Thus, target is set in Y axis and month goes in X axis in the same manner.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the month data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

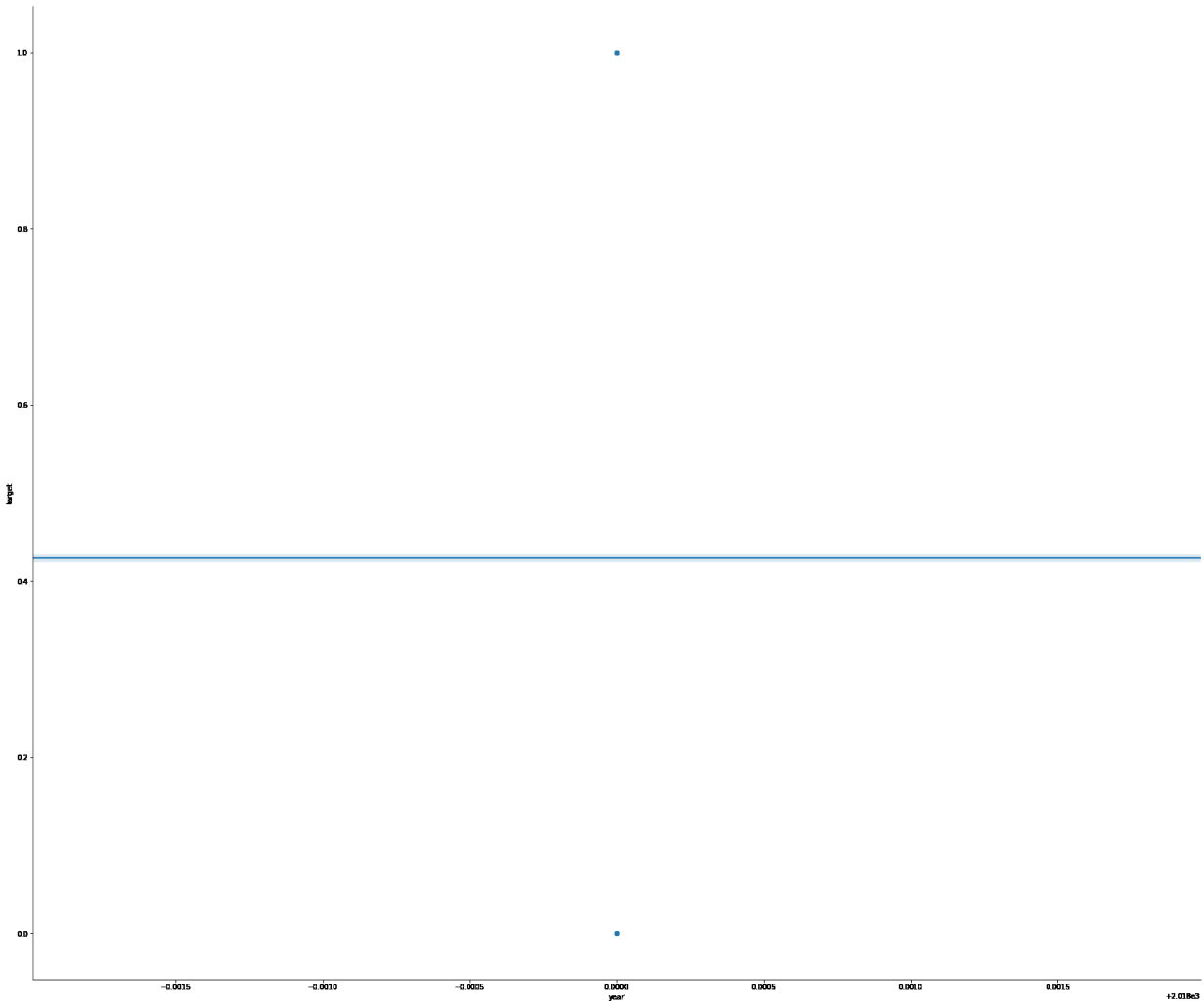

Fig. 5.8 Year vs Target

Here, in this Fig. 15, we have plotted the year column data from our pre-processed dataset. In this figure, minute is the variable and target is constant. Thus, target is set in Y axis and year goes in X axis in the same manner. As this dataset contains only one year, this plot is not that important but we keep it for the future use.

This mid separating line indicates the classification line. This classification line separates the data in to two different class. For future testing, in machine learning approach, the year data will be plotted in the same way and geometrically predict the target from the graph. Hence the prediction becomes more accurate.

5.1.1 kth-Nearest-Neighbor

We tried 10 values of kth-Nearest-Neighbor. The accuracy is given below,

TABLE I. ACCURACY OF KNN

K values	Accuracy Score
k=1	99.97
k=2	96.42
k=3	95.35
k=4	92.90
k=5	91.27
k=6	88.89
k=7	86.80
k=8	85.26
k=9	83.47
k=10	82.55

TableI contains the results of KNN algorithm over the dataset. Using K-Nearest-Neighbor algorithm, we get the following result for different values of k, where the test sample is 30 percent of total data.

5.1.2 Linear Regression

We used generic tuning parameters first and the we started tuning algorithmic parameters. We get the best accuracy of 99.76 percent in our tuning parameters [22] for this given dataset.

5.1.3 SVM

SVM also gives an 64.14 percent accuracy with more algorithmic complexity than previous two classifier.

5.2 FINAL MODEL SELECTION

As we are providing realtime online anomaly detection solution, our design is online. Our main algorithm runs in online server using python. So we prefer lower computational cost. Here we made a tradeoff between accuracy and computational cost.

Comparing all results, we chose to work with K-Nearest-Neighbor algorithm with k=5 for better result in lower computational complexity although linear regression has higher accuracy with high complexity.

Chapter 6 FUTURE PLAN AND CONCLUSION

This chapter contains the Future Plan and Conclusion which will give the summary of our thesis work and also give the indication of our future plan with our thesis project.

We have collected our data from a city of Cincinati from United States of America for research purpose. We have also collected local data from Dhaka using our own android application and GPS service. Our stored local data in no-sql format was a biggest challenge to put in machine learning models. So, first plan for the future is not mixing up technologies in the same research work.

In terms of research purpose we have submitted our research paper in IEEE Region TENSYMP. We are hopeful for our paper. Besides, commercially we want to develop our system and sell it for private use. We are working with embedded systems and learning to design PCB layout. Currently, we need two mobile applications. One application is in the vehicle and another one is for vehicle owner for monitoring. We plan to replace the vehicle application because it is easy to remove from the vehicle and get disconnected. We will integrate an embedded system with the vehicle that will be more efficient than just a mobile application. Also the cost will be lower replacing a high end mobile device with a small part of chip. We are studying on taking GPS location using GPS module and to detect vehicle door opening/closing using Cortex STM32 processor. Extensionally, if we get the total weight of the vehicle then this will train our model more efficiently.

REFERENCES

[1] Mekkayil and H. Ramasangu, "Vehicle tracking using modified fast compressive tracking algorithm," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, 2017, pp. 1-5.

[2] Johnson, S., Ashe, C. W., & Yates, J. B. (1996). U.S. Patent No. 5,557,254. Washington, DC: U.S. Patent and Trademark Office.

[3] T. Ahmed, M. Coates and A. Lakhina, "Multivariate Online Anomaly Detection Using Kernel Recursive Least Squares," IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, Anchorage, AK, 2007, pp. 625-633

[4] T. Ahmed, S. Ahmed, S. Ahmed and M. Motiwala, "Real-Time Intruder Detection in Surveillance Networks Using Adaptive Kernel Methods," 2010 IEEE International Conference on Communications, Cape Town, 2010, pp. 1-5.

[5] T. Ahmed, A. S. K. Pathan and S. Ahmed, "Adaptive algorithms for automated intruder detection in surveillance networks," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 2775-2780.

[6] T. Ahmed, S. Ahmed and F. E. Chowdhury, "Taking Meredith out of Grey's Anatomy: Automating Hospital ICU emergency signaling," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 1886-1890.

[7] I. Alam M. Ahmed, M. Alam, J. Ulisses, D. Farid, S. Shatabda and R. Rossetti, "Pattern mining from historical traffic big data," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, 2017, pp. 1-5

[8] R. Harshitha and J. Manikandan, "Design of a real-time pedestrian detection system for autonomous vehicles," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, 2017, pp. 1-4.

[9] G. Thomas, G. Alexander and P. M. Sasi, "Design of high performance cluster based map for vehicle tracking of public transport vehicles in smart city," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, 2017, pp. 1-5.

[10] G. Prabhakar, B. Kailath, S. Natarajan and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, 2017, pp. 1-6.

[11] A. Mulla, J. Baviskar, A. Baviskar and A. Bhovad, "GPS assisted Standard Positioning Service for navigation and tracking: Review & implementation," 2015 International Conference on Pervasive Computing (ICPC), Pune, 2015, pp. 1-6.

[12] S. Vatansever and I. Butun, "A broad overview of GPS fundamentals: Now and future," 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2017, pp. 1-6.

[13] Maddison, R., & Ni Mhurchu, C. (2009). Global positioning system: a new opportunity in physical activity measurement. International journal of behavioral nutrition and physical activity, 6, 1-8.

[14] Ming, L., & Vitányi, P. M. (1990). Kolmogorov complexity and its applications. In Algorithms and Complexity (pp. 187-254).

[15] R. Solomonoff. Formal theory of inductive inference, part II. Information and Control, 7(2):224–254, 1964.

[16] P. Stephen and S. Jaganathan, "Linear regression for pattern recognition," 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, 2014, pp. 1-6.

[17] Hongsuk Yi, HeeJin Jung and Sanghoon Bae, "Deep Neural Networks for traffic flow prediction," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, 2017, pp. 328-331.

[18] Jabbar M. A, Deekshatulu B.L, Chandra P, “Classification of Heart Diseases Using K-Nearest Neighbor and Genetic Algorithm”, in International Conference on Computational Intelligence: Modeling Techniques and Applications, 2013. CIMTA’13. pp. 86-94.

[19] Zhang Z, “Introduction to machine learning: k-nearest neighbor”, Big-data Clinical Tril Column. https://www.ncbi.nlm.nih.gov. Retrieved on August 1, 2017.

[20] KNN classification, ResearchGate. https://www.researchgate.net. Retrieved on August 2, 2017.

[21] M. E. Mavroforakis and S. Theodoridis, "Support Vector Machine (SVM) classification through geometry," 2005 13th European Signal Processing Conference, Antalya, 2005, pp. 1-4.

[22] A. Matsunaga and J. A. B. Fortes, "On the Use of Machine Learning to Predict the Time and Resources Consumed by Applications," 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, Australia, 2010, pp. 495-504.

[23] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[24] Vehicle GPS Data: Department of Public Services. https://data.cincinnati-oh.gov/Innovative-Government/Vehicle-GPS-Data-Department-of-Public-Services/b56d-ydmm