Active Learning Augmented Folded Gaussian Model for Anomaly Detection in Smart Transportation

[1]Venkata Praveen Kumar Madhavarapu, [1]Prithwiraj Roy, [2]Shameek Bhattacharjee, and [1]Sajal K. Das
[1]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA
[2]Department of Computer Science, Western Michigan University, Kalamazoo, MI, USA
Emails: {vmcx3, przhr, sdas}@mst.edu, shameek.bhattacharjee@wmich.edu

***Abstract*—Smart transportation networks have become instrumental in smart city applications with the potential to enhance road safety, improve the traffic management system and driving experience. A Traffic Message Channel (TMC) is an IoT device that records the data collected from the vehicles and forwards it to the Road Side Units (RSUs). This data is further processed and shared with the vehicles to inquire the fastest route and incidents that can cause significant delays. The failure of the TMC sensors can have adverse effects on the transportation network. In this paper, we propose a Gaussian distribution based trust scoring model to identify anomalous TMC devices. Then we propose a semi-supervised active learning approach that reduces the manual labeling cost to determine the threshold to classify the honest and malicious devices. Extensive simulation results using real-world vehicular data from Nashville are provided to verify the accuracy of the proposed method.**

***Index Terms*—Smart Transportation, TMC, Active Learning, Anomaly Detection**

I. Introduction

Smart transportation is an essential clog in the wheel that runs current and future smart cities, and past two decades have witnessed an explosive growth in smart transportation network and Intelligent Transportation Systems (ITS) [1]. These networks use two types of communication technologies, Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I). V2V communication is the wireless interaction and exchange of information like speed, location, and other information between the vehicles. In V2I communication, the road infrastructure consisting of IoT sensors collects data of vehicle speeds in various road segments, analyzes them, and shares the traffic information with the vehicles. The infrastructure and the vehicles communicate through Dedicated Short Range Communication (DSRC) protocol [2]. Fig. 1 illustrates the basic architecture of a smart transportation network [3].

The Traffic Message Channel (TMC) sensors are deployed on road segments to capture the ambient speeds of passing vehicles. Multiple such sensors forward information to a Road Side Unit (RSU). Numerous RSUs are deployed to cover the smart city area. RSUs sends all the TMC sensor data to an edge/fog computing module that implements data driven traffic services (e.g. driving assistance, detection of incidents, roadside assistance locator, road traffic control, and increasing efficiency of freeway systems). Naturally, the accuracy of the data collected from such TMC is of utmost importance for accurate decisions in a safety critical transportation CPS [4].

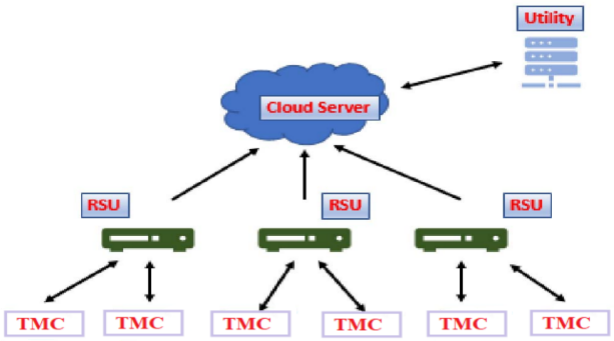


Figure 1: Architecture of a Smart Transportation System.

A. Challenges and Motivation

Incorrect or no reporting of vehicle information such as speed can result in incorrect interpretation of the traffic situation, which might lead to severe traffic jams. There are several scenarios that can produce erroneous data from TMCs. Environmental disasters like hurricanes and lightning strikes can damage the sensors. Extra moisture can hinder the ability to supply accurate data from a fraction of sensing devices. Moreover, the sensors may get stuck at a particular sample value [5] and keep reporting the same older value. Calibration errors, low battery in the sensors can also cause the reported data to be above or below the actual reading. Similarly, some sensor errors can stop data collection altogether (omission). For a large community scale IoT infrastructure, we need a scalable and lightweight device level anomaly detection technique that can quickly detect these malfunctioning IoT sensors, such that the maintenance personnel can be dispatched to replace them.

While several theories of anomaly detection [6] and device trust scoring models [3], [7]–[9] have been proposed to find the devices whose data is anomalous, there is a challenge of scalability when it comes to large community scale smart living IoT applications such as smart connected transportation. For example, [9] proposed a novel framework for identifying compromised IoT devices sending falsified data. However, it uses k-means for classification which is highly data dependent

and requires all devices to participate in the process. Similarly, the supervised machine learning approaches such as decision trees and standard Support Vector Machine (SVM) require labeling of the complete training set. The cost of labelling is very high and increases with the size of training set. This puts a tremendous burden on the infrastructure and the large scale computations also increase the carbon footprint. Ideally, for community scale IoT, we need a device level anomaly detection classifier that can reduce the labeling costs and remain consistent with large test cases.

B. Contributions

To solve the above challenges, the proposed anomaly detection model has 2 main parts. The first part is the trust scoring model which gives a score based on the recorded speeds from the TMCs. The consensus aware trust scoring model is based on Folded Gaussian distribution inspired from our earlier work [9] which is built for smart meters. The second part is the classification of anomalous TMCs. For this, we proposed an active learning based approach which is a semi-supervised learning algorithm that avoids the need for large sets of labeled data by employing a technique to identify and prioritize a limited set of labeled data. This is immensely beneficial for large community scale smart living IoT applications such as transportation systems having a large number of IoT sensing points. The detection model is verified with different experimental results using a real-world vehicular dataset from Nashville [10]. We show that the model is able to detect the traffic incidents and TMCs that are malfunctioning with an accuracy of more than 85%.

The classification from active learning will be advantageous compared to classification based on traditional clustering algorithms such as k-means and decision trees. This is because the outlying samples have lower priority and will not be considered while learning the threshold for classification. The main idea is to select specific data samples to label that will give us optimal classification threshold. Thus active learning reduces the cost of labeling needed for training the model compared to supervised learning algorithms.

The rest of the paper is organized as follows. Section II, discusses the related works. Section III defines the anomalies and their impacts. Section IV presents the threat model and the trust scoring mechanism to detect the anomalies. Section V offers the experimental results and finally Section VI concludes the paper with remarks on future research directions.

II. Related Work

Research on Smart city applications has seen rapid advancements in recent years. A large portion of this research contribution has focused on the implementation of sensor systems for transportation, communication, and infrastructure monitoring [11]–[13]. The two key challenges in large decentralized IoT networks like the smart transportation network are Quality-of-Service (QoS) and Security. While QoS focuses on the ability to provide services within an acceptable time frame, thus making it a latency critical application, security deals with resilience and mitigation of unwanted interference, whether it is environmental or created by an external adversary. Generally, anomaly detection is focused on finding perturbations that may cause by either an unexpected event or a *False Data Injection (FDI)* attack on the system. Different Intrusion Detection Systems (IDS) are deployed at key points in the distributed network to collect and analyze the network traffic to detect anomalies in the system [14].

Traditional anomaly detection schemes are based on classification, statistical inference, state-based analysis, and clustering [6]. Classification based detection schemes usually rely on Support Vector Machines (SVM), Bayesian Models, Gaussian Processes or Neural Networks [15]. However, these methods require large-scale accurate models of system behavior which might contain sensitive information (e.g., exact locations and movements of the users over time). State based methods based on Kalman Filtering [16] require realistic assumptions on the data distribution to estimate normal behavior which is a challenging task. In [3], the authors have presented a decentralized and light-weight anomaly detection approach on RSU level based on the ratio of Harmonic and Arithmetic mean to detect different types of data falsification. However, the method results in a false positive rate of 20% which is relatively high considering the fact that attacks on the system are generally rare, and a high false positive rate would disrupt the system frequently which would cost in infrastructure management.

III. System Model

We consider a set of N TMCs that collects the speeds information from the vehicles. The speed reported by i-th TMC at time slot t is represented by S_t^i. We model S_t^i as the realizations of a random variable (r.v.) S^i denoting the speed distribution of the vehicles of i-th TMC. We develop a detection model that is deployed at the cloud server to analyze the measurements of each TMC. The model will be able to detect congestion, accident, or sensor failures in real-time.

A. Anomaly or Failure Model

In this paper, we specifically investigate TMC sensor faults and failures as the causal reason for anomalies. In this work, we propose a model to detect such anomalies at the TMC level to isolate the TMCs that need inspection. Let's consider M TMCs record are anomalous of the total N TMCs. We define $\frac{M}{N} = \rho_{an} \in [0, 1)$. For example, $\rho_{an} = 0.05$, means 5% of the total number of TMCs have readings that anomalously deviate from the free-flow either due to sensor failure or congestion. A sensor failure can result in following situations:

Stuck Value Anomaly: In this type of sensor failures, the reporting value gets stuck at a value in which the sensor was last correctly working. This results in reporting of the same value which is not the actual true value.

Calibration Anomaly: If condensation builds up on the sensor equipment, it can impact the sensor calibration accuracy and result in reporting of false data. The calibration anomaly can result in increased or decreased speeds compared to the true value.

Omission Failures: In this case of sensor failure, the TMC will stop reporting. This can be easily detected as the records will be empty for that particular TMC.

Free-flow is the average speed recorded under no congestion and sensor failure. Depending on the average speed, the anomalies could be classified as deductive or additive based on its nature of deviation from the free-flow speeds. For example, for omission failure of TMC, the actual speed of information S_t^i from the i-th TMC at time t will be much lower or zero compared to normal free-flow situation. The additive anomaly is possible under the calibration error as this type of sensor failure can report any false value.

We denote δ_{avg} as the average margin of deviation from the free-flow for each TMC. It is the average of all δ_t values for a TMC in a given time frame. Note that our model does not use specific vehicle information rather uses the collection of speed information of multiple vehicles captured at the TMC level.

IV. Proposed Approach

The detection model consists of two main steps. The first step is the scoring model, and the second step is classification. In the scoring model, a trust score will be calculated depending on the vehicular readings of each TMC. The second step uses an active learning model to classify benign or non-anomalous TMCs from anomalous ones. Our method is divided into four sub-modules: (1) Trust Scoring model; (2) Selection of Sparse Manual Labels and Initial Threshold; (3) Priority Scoring of TMCs; (4) Priority Score enabled Final Threshold Selection.

A. Trust Scoring Model

The trust scoring model will be used to identify TMCs reporting the anomalous data by assigning a score depending on the speeds recorded. The trust score is calculated for each TMC over a time frame of T ($<$ 2 hours). The trust scoring model starts with discrete rating criterion that assigns a rating level to each TMC reading, by comparing proximity of its reported data S_t^i at time slot t with the historical (previous time frame) free-flow mean consensus μ_H over the time frame. The absolute difference between the S_t^i for any TMC i and the μ_H, $|S_t^i - \mu_H|$ will be used along with the historical standard deviation (σ_H). The discretized rating level for each TMC reading denoted by r_t^i is given by Table I, using the empirical rule for Gaussian distributions to assign S_t^i as belonging to one of the 4 possible rating levels. The highest rating 4 is closest in terms of proximity to μ_H, and similarly lower ratings are obtained if the TMC's data is further from the μ_H. Over the time frame T, all the discrete ratings over time frame T for each TMC i is collected to form a rating vector sequence $\vec{r_{sort}^i}$ sorted in ascending order of discrete ratings.

Table I: Discrete Rating Levels

Scenario of S_t^i	**Rating (r_t^i)**
$\lvert S_t^i - \mu_H \rvert \leq \sigma_H$	4
$\sigma_H < \lvert S_t^i - \mu_H \rvert \leq 2\sigma_H$	3
$2\sigma_H < \lvert S_t^i - \mu_H \rvert \leq 3\sigma_H$	2
otherwise	1

Most of the vehicles will be going closer to the mean free-flow speed. So, under no anomalies, the most common and highest rating level is 4 followed by all others. The sign of the discrete rating is always positive as in the folded Gaussian, the magnitude of difference $|S_t^i - \mu_H|$ is the only thing that is considered. Higher percentage of lower ratings in a time frame will give lesser weights to the lower ratings than a scenario with lower percentage of low level ratings and vice versa. First, a weight parameter x_t distributed between 1 to 4 is calculated as shown in Eqn. 1 where $K = 4$ is the total number of discrete rating levels in the system, C_T is the count of readings in a selected time frame. The final weights are achieved through Eqn. 2 where $\mu_{BR} = 4$ is the best or highest possible rating level and σ_{dr}^i denotes the standard deviation of discrete ratings of each TMC in the time frame T. σ_{dr}^i for each TMC will be different based on different observations compared to common mixture data, which captures individual differences in behavior. Therefore, the corresponding raw weight cw_t of the rating at time index t yielded from Eqn. 2, are normalized as shown in Eqn. 3.

$$x_t = 1 + \frac{(K-1)t}{(C_T - 1)} \qquad \forall \quad t \in T \tag{1}$$

$$cw_t^i = \frac{1}{\sigma_{dr}^i \sqrt{2\pi}} e^{-\frac{(x_t - \mu_{BR})^2}{2(\sigma_{dr}^i)^2}} \tag{2}$$

$$w_t^i = \frac{cw_t^i}{\sum_{t=0}^{T-1} cw_t^i} \tag{3}$$

All the density values are combined to form a weight vector $\vec{W^i}$ for each TMC i as in Eqn. 4. The aggregate weight rating R^i of the i-th TMC will be a scalar value between 1 and 4 resulting from the dot product of weight vector $\vec{W^i}$ and sorted discrete rating vector $\vec{r_{sort}^i}$ as shown in Eqn. 5.

$$\vec{W^i} = [w_1^i, w_2^i, ..., w_t^i, ...] \qquad \forall \quad t \in T \tag{4}$$

$$R^i = \vec{r_{sort}^i} \cdot \vec{W^i} \tag{5}$$

As the ratings will be positive irrespective of whether the reading is greater or lesser than the rating level 4 are treated as the same random variable. Hence, the aggregate weighted (R^i), when interpreted as a trust score will also follow a folded Gaussian shape. This meaning R^i = 4 represents the highest trust score followed by an exponential reduction of trust, as R^i decreases. We used the inverse power law inspired kernel trick to transform the R^i that ranges from 1 to 4 into a final trust value, TR^i, for each TMC i between 0 and 1, as shown

in Eqn. 6. The value of K depends on the number of rating levels (4, in our case).

$$TR^i = \frac{1}{(K)^\eta}(R^i)^\eta \tag{6}$$

B. Selection of Sparse Manual Labels and Initial Threshold

The folded Gaussian model gives a trust score (TR^i) for each TMC $i \in N$. The TMCs with lower trust scores imply anomalous behavior because they result in lower rating labels. The classification is done by determining a linear threshold that separates the anomalous TMCs from the benign ones. The TMCs with trust scores higher than the threshold will be considered as benign whereas the ones less than the threshold will be marked as anomalous. In this section, we will discuss the selection of the manual label set and initial threshold that initiates the active learning process.

Consider the trust scores of all the N TMCs. First, using winsorization we trim out $\alpha\%$ of the lowest and highest trust scores to reduce the influence of extreme points on the learning process. From the set of remaining TMCs, we pick a subset Q_b verified with no anomalies which forms the first class (denoted by blue dots of size $|Q_b| = 10$ in Fig. 2). Then, we pick a subset of TMCs of size Q_a with verified presence of congestion, stuck value anomaly, and traffic incidents (denoted by red stars of size $|Q_a| = 10$ in Fig. 2). The verification is allowed by a ground truth data set available from Nashville Police and Emergency Response Units [10].

The combination of Q_a and Q_b ($Q_a \cup Q_b$) from the training set forms the initial sparse set of TMCs of size Q that requires manual labeling. For illustration, 10 anomalous labels and 10 benign labels are shown in Fig. 2 making $|Q| = 20$ labels. For the rest of the TMCs (denoted by green marker in Fig. 2), we have scores from the training, but no information on whether they are benign or anomalous. *The challenge is to learn the accurate threshold without knowing the label status of most of the TMCs in the network.* This exemplifies the power of our approach for community scale smart living IoT applications.

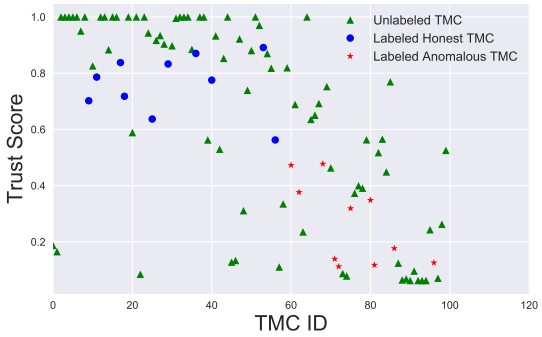


Figure 2: Initial Manual Labeling of few TMCs.

Now, we use the set Q, to calculate an initial threshold (denoted as TH_{inl}) using Support Vector Machine (SVM) with a linear kernel. The rationale for using a linear kernel is due to the fact that the scores are distributed which indicate that they are linearly separable.

C. Priority Scoring of TMCs

Given the initial threshold TH_{inl} and the sparse labeled set Q, we need to iteratively find the most appropriate training data points of N TMCs that will enable the learning of the final threshold (TH_{fnl}) which in turn will be used for classification in the test set. The selection of important data points for each iteration of active learning is achieved via priority scoring which uses *least confidence* to calculate the scores. These newly selected data points will be used to keep updating the threshold in each iteration. This process ends when the threshold remains unchanged in two consecutive iterations.

In least confidence, the data points whose scores are neither too high nor low end up with higher priority scores compared to extreme scores. For example, the data points among the highest trust scores and least trust scores have higher probability to belong to the true benign class and anomalous class respectively. However, the data points closer to the current threshold (at any iteration) cannot be certainly determined whether they belong to one class or the other. Thus, they have the least confidence or paradoxically, the highest priority score (denoted by LC). These higher priority data points play a proportionally more crucial role in the determination of the final threshold.

To calculate the priority scores, we need the probability of each TMC i belonging to the anomalous class (P_a^i) and the benign class (P_b^i). These probabilities should depend on the trust score (TR^i) of the TMC i and the threshold calculated at the j-th iteration $TH(j)$. When the trust score of the i-th TMC is equal to the threshold, there is no preference on the class membership and we can say that the $P_a^i = P_b^i = 0.5$. As the TR^i scores far away from the threshold, the probability of the i-th TMC belonging to certain class increases. The least confidence priority score ($LC^i(j)$) of TMC i and iteration j is calculated from $P_a^i(j)$ and $P_b^i(j)$ as shown in Eqn. 9. The priority score will be higher for TMCs with trust score closer to the threshold. For example, consider $TH(j) = 0.55$ and two TMCs with trust scores $TR^1 = 0.5$ and $TR^2 = 0.9$, the priority scores will be $LC^1 = 0.45$ and $LC^2 = 0.11$ respectively. So, the first TMC will be picked over the second for the set Z because of higher priority score.

$$P_a^i(j) = \begin{cases} \frac{1}{2} - \frac{TR^i - TH(j)}{2\times(1-TH(j))}, & \text{If } TR^i > TH(j) \\ \frac{1}{2} + \frac{TH(j) - TR^i}{2\times TH(j)}, & \text{Otherwise} \end{cases} \tag{7}$$

$$P_b^i(j) = 1 - P_a^i(j) \tag{8}$$

$$LC^i(j) = 1 - \max(P_a^i(j), P_b^i(j)) \tag{9}$$

D. Priority Score based Final Threshold Selection

The manual labeled set Q and initial threshold (TH_{inl}), are input to the calculation of final threshold TH_{fnl}. Active learning is an iterative approach and slowly corrects the threshold. The change in threshold leads to change in the set of appropriate data points. We represent the changing set with $Z(j)$ for iteration j.The active learning starts with TH_{inl}. It continues using the following 6 steps until we get the final threshold. The iteration for active learning in Algorithm 1 (line 4-9) is explained below:

1) The current threshold ($TH(j)$) will be used to calculate the priority score (LC^i) of each TMC i using Eqn. 9.

2) Find the set $Z(j)$ with TMCs having highest $|Q|$ priority scores calculated from step 1.

3) Manually label the unknown data points from the set $Z(j)$ using ground truth information.

4) Increment j by 1.

5) Using the trust scores of TMCs from set $Z(j-1)$, the threshold ($TH(j)$) will be calculated using SVM.

6) If $TH(j)$ is different from $TH(j-1)$, go to step 1. Otherwise the current threshold $TH(j)$ will be the final threshold TH_{fnl}.

Algorithm 1 Finding threshold using Active learning

1: **Input:** $Q, j = 1, TH(j) = TH_{inl}, TH(0) = 0$
2: **Output:** TH_{fnl}
3: **while** $TH(j) \neq TH(j-1)$ **do**
4: Calculate LC for all TMCs using $TH(j)$
5: $Z(j) =$ Top $|Q|$ TMCs with highest LC values
6: Query the unknown labels of $Z(j)$
7: $j = j + 1$
8: $TH(j) =$ SVM$(Z(j-1))$
9: **end while**
10: $TH_{fnl} = TH(j)$

1) Optimal size of Q: Q is the set of TMCs considered for manual labeling and finding new threshold in each iteration of the active learning model. The classification performance of the model is dependent on the size of Q which is a hyperparameter that can impact the final threshold TH_{fnl}. If the size of the set Q is too small, it can result in under-fitting, whereas a larger size of Q can result in over-fitting. Hence, we need to determine the optimal value of Q.

The measure of optimal size of Q can be done using an error function E that will be minimum under best classification. The summation of the priority scores $LC^i(Q)$ of set of mis-classified TMCs Y will be lower under best size of Q as the number of mis-classifications will also be lower. The error function for each value of Q will be summation of priority scores calculated using TH_{fnl} for the set of mis-classified TMCs as shown in the Eqn. 10. The error function for different values of Q can be seen in Figure 3. From the result, we can say that the optimal size of Q is in range of 10-20.

$$E = \arg\min_{|Q|} \Big(\sum LC^i(Q) \quad \forall i \in Y \Big) \tag{10}$$

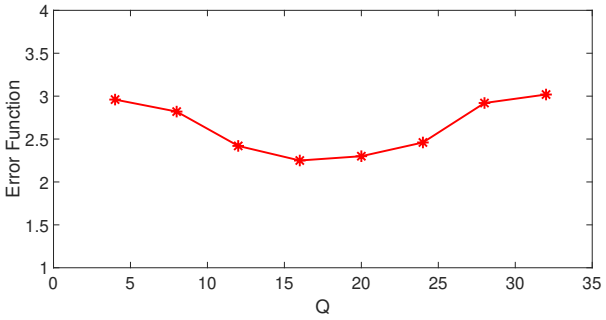


Figure 3: Error rate under different values of Q.

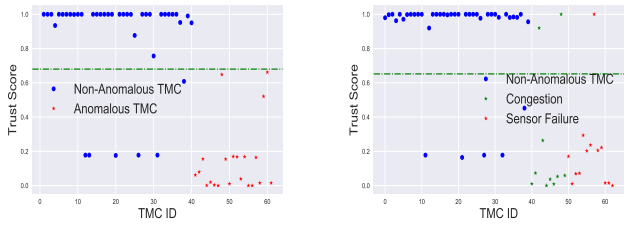
V. Experimental Results

Description of Datasets: We have used a real-world vehicular dataset from Nashville, Tennessee to validate the proposed solution. The dataset has vehicular data recorded in real-time over a period of 4 months (January to April) with 1271 Traffic Message Channels (TMC) [10]. We used the first two months (January and February) for training the model. March data is used for cross-validation and April data is used as the test set. The dataset contains the ground truth for accidents and congestion. The results from this section are considered from 60 TMCs belonging to a 10 different RSU clusters.

A. Trust Score Classification of TMCs

The trust scoring model is applied to the test set of the Nashville dataset. The active learning parameters we derived from the training and cross-validation will be used for classification. The test data contains TMCs reporting wrong information under both additive and deductive anomalies.

A higher trust score implies the TMC is under a normal behaviour (free flow) while the lower trust score is the result of either congestion or sensor failures. Intuitively, a congestion will always be a deductive anomaly but the sensor failure can result in either additive or deductive anomaly. The trust model generates a score for all the TMCs in the current time frame, we then used the TH_{fnl} value from active learning for the classification. The test data includes TMCs with simulated sensor failures including stuck value, calibration and omission anomalies. Stuck value is simulated using true value right before the start of anomaly. For calibration anomaly, a constant random number is added or subtracted for each TMC reading. Figure. 4(a) shows the performance of trust scoring model in detecting the additive anomalies caused by the sensor malfunction. Figure 4(b) depicts the performance of the model under different deductive anomalies including congestion and sensor failures. The sensor failures can be filtered using existing non-recurrent congestion detection mechanisms [17].

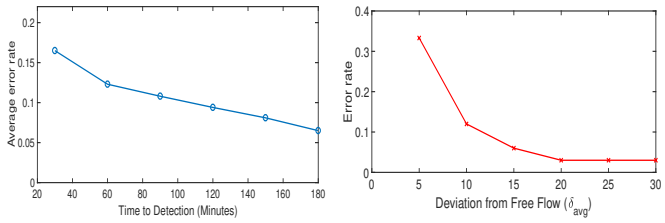


(a) (b)

Figure 4: Classification of Anomaly: (a) Additive (b) Deductive

B. *Performance Analysis*

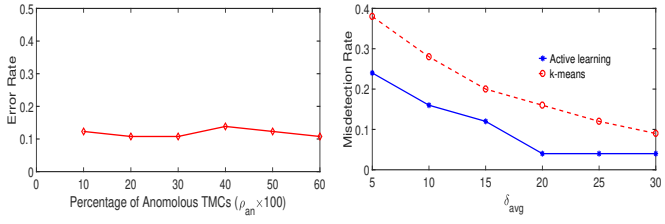
The time to detection of anomalies is a critical factor in vehicular networks given the real-time nature of the applications. The accidents and congestion need be detected quickly to warn the other vehicles to avoid the congested routes. The performance must be good at lower detection time for detecting the anomalies. Fig. 5(a) shows the performance of the model with the detection time ranging from 30 minutes upto 3 hours. The result shows the proposed model is able to detect the congestion and accidents with an accuracy of over 85% (the error rate is $< 15\%$) in only 30 minutes.



(a) (b)

Figure 5: (a) Time to detection (b) Margin of failure.

The model works well under different levels of congestion/failure. This can be seen in Figure. 5(b) where the performance is good even when deviation in speed is less than 10 mph. We have simulated the anomalies for different ρ_{an} to see the impact on error rate and Figure 6(a) indicates that the performance is not affected by the number of TMCs with the anomalies (ρ_{an}). The labeling cost of active learning will only be a small fraction compared to supervised classification models where all N TMCs required to be labeled. In comparison with unsupervised classification models, the Figure. 6(b) shows the active learning achieves lower mis-detection rate than using k-means at different δ_{avg}.



(a) (b)

Figure 6: Performance (a) ρ_{an} (b) Active Learning vs K-means.

VI. Conclusion

In this work, we have presented an anomaly detection model for the IoT sensors in smart transportation. The anomaly could be any abnormal traffic incident or due to sensor malfunction. We used the folded Gaussian trust scoring model to generate the trust score for each TMC depending on its measurements. Then, we applied an active learning approach to classify the TMCs with anomalous behavior. This also helps to detect any traffic incidents in near real-time as the proposed model is able to detect the anomalies within 30 minutes with good accuracy. In future we will extend the model to distinguish between different types of anomaly. This would help the network to take the required safety measures immediately.

Acknowledgment This research was supported by National Science Foundation, USA grants SATC-2030611, SATC-2030624, OAC-2017289

References

[1] M. Obaidat, M. Khodjaeva, J. Holst, and M. B. Zid, "Security and privacy challenges in vehicular ad hoc networks," in *Connected Vehicles in the Internet of Things*. Springer, 2020, pp. 223–251.

[2] H. Zhang and J. Li, "Modeling and dynamical topology properties of vanet based on complex networks theory," *Aip Advances*, vol. 5, no. 1, p. 017150, 2015.

[3] M. Wilbur, A. Dubey, B. Leão, and S. Bhattacharjee, "A decentralized approach for real time anomaly detection in transportation networks," in *2019 IEEE SMARTCOMP*. IEEE, 2019, pp. 274–282.

[4] J. Liu, D. Ma, A. Weimerskirch, and H. Zhu, "A functional co-design towards safe and secure vehicle platooning," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, 2017, pp. 81–90.

[5] E. U. Warriach, M. Aiello, and K. Tei, "A machine learning approach for identifying and classifying faults in wireless sensor network," in *2012 IEEE 15th ICCSE*. IEEE, 2012, pp. 618–625.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[7] N. Fan and C. Q. Wu, "On trust models for communication security in vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 90, p. 101740, 2019.

[8] M. Sun, M. Li, and R. Gerdes, "A data trust framework for vanets enabling false data detection and secure vehicle tracking," in *2017 IEEE CNS*. IEEE, 2017, pp. 1–9.

[9] S. Bhattacharjee, A. Thakur, and S. K. Das, "Towards fast and semi-supervised identification of smart meters launching data falsification attacks," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 173–185.

[10] "hereapi," https://developer.here.com/.

[11] G. P. Hancke, G. P. Hancke Jr *et al.*, "The role of advanced sensing in smart cities," *Sensors*, vol. 13, no. 1, pp. 393–425, 2013.

[12] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, "Smart cities of the future," *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.

[13] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE IoT journal*, vol. 1, no. 1, pp. 22–32, 2014.

[14] L. Santos, C. Rabadao, and R. Gonçalves, "Intrusion detection systems in internet of things: A literature review," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2018, pp. 1–7.

[15] F. Sun, A. Dubey, and J. White, "Dxnat—deep neural networks for explaining non-recurring traffic congestion," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 2141–2150.

[16] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[17] B. Anbaroğlu, T. Cheng, and B. Heydecker, "Non-recurrent traffic congestion detection on heterogeneous urban road networks," *Transportmetrica A: Transport Science*, vol. 11, no. 9, pp. 754–771, 2015.