A Decentralized Approach For Real Time Anomaly Detection In Transportation Networks

Michael P. Wilbur[1], Abhishek Dubey[2], Bruno P. Leão[3], Shameek Bhattacharjee[4]

***Abstract*—Internet of Things (IoT), edge/fog computing, and the cloud are fueling rapid development in smart connected cities. Given the increasing rate of urbanization, the advancement of these technologies is a critical component of mitigating demand on already constrained transportation resources. Smart transportation systems are most effectively implemented as a decentralized network, in which traffic sensors send data to small low-powered devices called Roadside Units (RSUs). These RSUs host various computation and networking services. Data driven applications such as optimal routing require precise real-time data, however, data-driven approaches are susceptible to data integrity attacks. Therefore we propose a multi-tiered anomaly detection framework which utilizes spare processing capabilities of the distributed RSU network in combination with the cloud for fast, real-time detection. In this paper we present a novel real time anomaly detection framework. Additionally, we focus on implementation of our framework in smart-city transportation systems by providing a constrained clustering algorithm for RSU placement throughout the network. Extensive experimental validation using traffic data from Nashville, TN demonstrates that the proposed methods significantly reduce computation requirements while maintaining similar performance to current state of the art anomaly detection methods.**

***Index Terms*—Smart Cities, Transportation, Anomaly Detection, Decentralized**

I. Introduction

Emerging trends and challenges: Internet of Things (IoT), edge/fog computing, and the cloud are fueling rapid development in smart connected cities. Given the increasing rate of urbanization, the advancement of these technologies is a critical component of mitigating demand on already constrained transportation resources. Recent research on smart transportation systems has focused on optimal route planning for congestion reduction, which has shown huge potential impact on maximizing existing transportation resources [1]. The costs of optimizing route planning are relatively low compared to large scale infrastructure upgrades, making this an attractive option for city planners and transportation experts.

Approaches to optimal route planning are typically data-driven [2], [3], [4]. The scale and real-time nature of these systems require shared computing architectures to handle the high velocity and volume of data originating from small sensors placed throughout the network. One solution is edge/fog computing. In this case, services are moved to road-side units (RSUs), which are low-powered edge devices [5] situated between the sensor level and the cloud. Each RSU hosts various computation services for a collection of sensors, and communicates with the cloud. Implementing a network of RSUs moves computation to the edge of the network, creating a decentralized data processing system.

Data-driven approaches are susceptible to data integrity attacks. The dynamic nature of real-time routing systems means that the effects of such an attack have immediate impact and substantial cascading consequences [6]. Additionally, the distributed and shared nature of the underlying architecture provides multiple points of entry, making data integrity attacks even more likely. Given the potential human and economic impacts of such an attack, the trustworthiness of data in smart transportation networks is of critical importance. While there is substantial research regarding anomaly detection in transportation networks [7], these approaches are often computationally costly and do not adapt well to the real-time nature of distributed smart transportation data networks. Despite the critical importance of data integrity in such systems, research in this area remains underdeveloped.

Current state of the art statistical detection methods typically rely on measures of central tendency such as median and mean or their variants. While this approach works for *deductive* attacks and *additive* attacks, in which sensor readings are decreased or increased respectively, it fails for *camouflage* attacks in which sensor readings are increased at some sensors and decreased at other sensors. Camouflage attacks are of particular importance when working with data-integrity attacks in transportation networks, as such an attack would aim to divert traffic and resources to specific regions or roads and thus, maximize the effects of an attack.

Therefore we aim to improve data integrity in decentralized smart transportation systems by proposing a novel real-time anomaly detection algorithm for deductive and camouflage data integrity attacks. Our approach maintains similar accuracy to traditional methods, while addressing two critical components of scaling anomaly detection to decentralized systems. First, it reduces the computational costs associated with computationally expensive traditional anomaly detection by avoiding continuous computation on all sensors in real time. This is accomplished by continuously monitoring anomalies at the RSU level using a statistical means approach for aggregate anomaly detection and reserving the more computationally costly sensor level detection for cases in which anomalies are found at the RSU level. Second, our approach is designed

[1]Michael P. Wilbur and [2]Abhishek Dubey are with the Department of Electrical Engineering and Computer Science and the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37240

[3]Bruno P. Leão is with Business Analytics & Monitoring at Siemens Corporate Technology, Princeton, NJ

[4]Shameek Bhattacharjee is with the Department of Computer Science at Western Michigan University, Kalamazoo, MI 40998

978-1-7281-1689-1/19/$31.00 ©2019 IEEE
DOI 10.1109/SMARTCOMP.2019.00063

so that the anomaly detection process itself is distributed, mirroring the natural architecture of modern decentralized smart networks and allowing seamless integration with such systems.

We also provide a constrained hierarchical clustering algorithm for RSU placement in an existing transportation system fitted with traffic sensors. As shown later, this approach improves zone level detection while also maximizing spare processing capacity at the RSU level.

Contributions: This paper presents a decentralized anomaly detection approach and architecture for distributed smart transportation systems. Our focus is on *orchestrated data-integrity attacks*, in which an organized attacker falsifies data in a systematic attack process. This paper's main contributions are as follows:

- Anomaly detection is framed as a decentralized computational system allowing for real-time processing and scalability.
- An algorithm is provided for RSU placement which maximizes processing capacity of the RSU network and optimizes central tendency anomaly detection methods.
- We present a novel real time anomaly detection algorithm which reduces the computational costs associated with traditional anomaly detection methods while maintaining similar accuracy.

Outline: We start by defining the problem and model assumptions in Section II. Related work is covered in III. The System Model is covered in Section IV, while the Sensing Architecture is covered in Section V. RSU placement is outlined in Section VI and the anomaly detection framework is proposed in Section VII. Finally, simulations and results are provided in Section VIII.

II. Problem Statement

Our primary concern is orchestrated data integrity attacks in smart, decentralized transportation systems.

A. Problem Overview

The goals of our system are the following:

- *Real-time* identification of orchestrated data-integrity attacks.
- *Decentralized implementation*. The system should integrate easily with modern smart-city infrastructure and be optimized for common hardware limitations in such systems.
- *Deductive and Camouflage Attacks* - extend traditional statistical means anomaly detection to camouflage attacks in which mean and median are unchanged.
- Reduce computation requirements compared to traditional anomaly detection methods.

B. Assumptions

To achieve the above goals, we make the following assumptions.

1) Sensor Model: We assume that the city has sensors capable of transmitting traffic speed data wirelessly to an RSU.

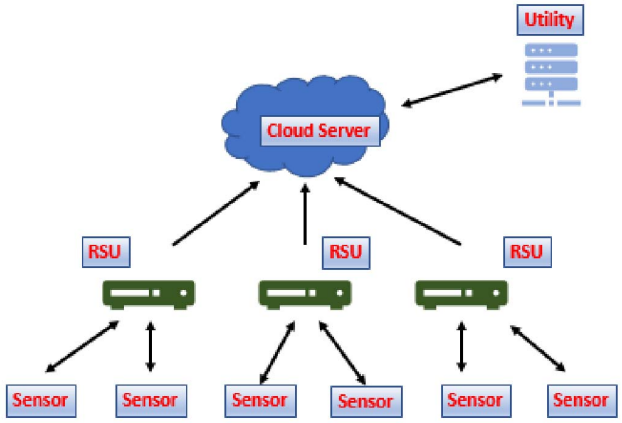

Fig. 1. System Architecture

In our investigation, speed data is collected by the sensor and sent to its associated RSU.

2) Road-side Units: RSUs are low-powered fog nodes [5] placed throughout the transportation network which are capable of collecting and transmitting data from a collection of sensors to a centrally located cloud-based routing system.

3) Centralized Cloud: A centralized cloud network is available to provide additional processing capabilities for sensor level anomaly detection.

4) Attack Model: The attacker is capable of compromising a subset of sensors or RSUs by manipulating their outputs. These attacks occur at the sensor level. As the focus of this paper is orchestrated data-integrity attacks, sensor or RSU faults from physical failures is outside the scope of this paper.

C. Our Approach

The architecture for our system is detailed in Figure 1, and consists of three fundamental components: the Sensor level, RSU level and Cloud. Our anomaly detection framework thus consists of two components, zone level detection and sensor level detection. Zone level detection is run at the RSU level, while the more computationally expensive sensor level detection runs at the cloud. Framing detection in this way maximizes existing hardware resources while reducing computation requirements compared to traditional detection approaches.

A major focus of this paper is on the integration and implementation of the anomaly detection framework in decentralized smart transportation networks. As RSUs are fog nodes, a fundamental question is how to deploy these devices throughout the network. We identify three critical considerations to answering this question. First, RSUs should be located as close to possible to the sensors streaming to it in order to minimize network latency. Second, as RSUs are low-powered devices, the maximum number of sensors mapped to a single RSU is to be constrained. Lastly, we look to group sensors together as to maximize the efficiency of our anomaly detection approach.

III. Related Work

Smart city research has advanced rapidly in recent years. A large focus of this research has focused on implementation of sensor systems for transportation, communication and infrastructure monitoring [8], [9], [10], [11], [12]. In general, anomaly detection is focused on finding deviations in single (point) or sequence (collective) values from normal expected behavior. Traditional anomaly detection is based on classification, statistical, state based, clustering or information theory [7]. Classification methods are usually based on Support Vector Machines (SVM), Bayesian Models, Gaussian Processes or Neural Networks [13]. These methods require large scale, detailed and accurate models of system behavior. Additionally, supervised classification models require careful consideration regarding user data privacy. This is of particular concern when dealing with transportation systems and the specific movement of users over time. State based methods use Kalman Filtering [14] to estimate normal behavior. These methods require making realistic assumptions on data distributions, a challenging task. Additionally hardware considerations must be accounted for [15].

Our primary concerns regarding the anomaly detection problem are accuracy, computational requirements and easy distribution over a decentralized network. For this reason, our zone level detection uses a statistical approach. Related statistical approaches include auto-regressive, exponential or cumulative weighted moving averages (ARMA, EWMA, CWMA) and Cumulative Sum Control Chart (CUSUM) of data as metrics under normal operating behavior. These approaches are light weight, and do not necessarily require anomalous data. Our work presents a hybrid approach which uses a statistical mean ratio that has proven effective in detecting data-integrity attacks in power grid networks [16] and Gaussian Processes for sensor level detection [17].

Hierarchical anomaly detection has shown to be useful in monitoring large scale distributed web architectures [18]. The advantage of hierarchical anomaly detection is that the detection computation can be balanced between low-powered edge devices and central computation clusters. One approach is to keep a central model of expected data behavior to compare with current data [19]. In this case, when anomalous patterns are found in the system the second, more computationally expensive, procedure of identifying anomalous nodes within the subsystem is performed [20], [21].

RSU placement has been studied in relation to maximizing connectivity for smart cities using intersection-priority [22], minimizing event reporting times along highways [23] and maximizing information flow in urban areas [24]. Approaching RSU placement through the context of anomaly detection efficiency is a new topic.

IV. System Model

A. Data Overview

To simulate the framework provided in Figure 1, historical data is collected from the HERE API [25] for use as real-time sensor data for Nashville, TN. Two months of data was extracted from February 12, 2018 to April 12, 2018 for use as historical training and reference data. Additionally, two weeks of data from April 16, 2018 to April 27, 2018 was extracted for testing and simulation. Only weekdays (Monday-Friday) are considered.

The HERE data is composed of time stamped speed recordings, identified by its Traffic Message Channel identification (TMC ID), [26]. Each TMC represents a segment of road in which the speed was recorded. In our framework each TMC ID acts as a sensor which provides speeds for optimal routing. There are 9,979 TMCs, and therefore sensors, in our data set.

B. Data Integrity Attack Overview

Traditional anomaly detection in transportation systems focus on detecting faulty sensors [13] [17], whether from hardware failure or software issues in the collection of data. In this model, anomaly detection is run in the cloud for each sensor in isolation. Data-integrity attacks on the other hand are orchestrated from a collection of sensors simultaneously to maximize the effect of the attack on the global transportation system.

The shared nature of computing resources in smart connected cities provide multiple entry points for attackers, making attacks likely events. Additionally, the dynamic real-time nature of such systems means that well designed attacks will have substantial cascading effects throughout the system. In this sense, focused localized attacks on a collection of sensors will propagate throughout the network quickly.

While traditional anomaly detection operate at the sensor level, the identification of orchestrated attacks requires aggregate detection across groups of sensors. In this context, organized data integrity attacks spanning multiple sensors within a selected region can have cascading effects throughout the transportation system.

Our focus is primarily on two types of data integrity attacks. In the first type of attack a selected percentage of sensors have their speed values reduced and is referred to as a deductive attack. These attacks aim to diverge traffic away from attacked sensors by convincing the routing system that certain roads have more congestion than in reality.

The second type of attack is camouflage attacks, in which an organized attacker balances additive and deductive attacks to evade detection and exert certain behaviors on the system. Camouflage attacks are of particular concern in transportation routing systems as an attacker can deviate network behavior at a fine granular level to maximize impact of the attack. One scenario would be an attack aimed at gathering vehicles along a specific road segment or crowding drivers in a highly dense area. Identifying attacks of this nature is of critical importance to first responders and the defense industry, yet as this approach would leave mean and median unchanged, camouflage attacks evade traditional central tendency approaches.

C. Simulated Deductive and Camouflage Attacks

To simulate deductive attacks and camouflage attacks, we use the historical standard deviation of a sensor's speed, represented by σ_s, as a basis for altering speed value d at attacked

sensor s. Therefore d_s^a represents the speed value at sensor s when attacked while d_s is the actual speed recorded at sensor s when not attacked. The severity of the attack is governed by δ. Equation 1 represents the process for altering speeds from a deductive attack at a single sensor while Equation 2 represents the process for altering speeds from an additive attack at a single sensor.

$$d_s^a = d_s - \delta * \sigma_s \tag{1}$$

$$d_s^a = d_s + \delta * \sigma_s \tag{2}$$

Each RSU r is responsible for a subset of sensors $S^r \subset S$ where S^r is the subset of sensors at RSU r and S represents all the sensors in the network. Therefore, if we look to simulate a deductive attack at RSU r during time window k affecting p percentage of sensors, then p percentage of sensors are randomly selected for the attack.

Conversely, to simulate a camouflage attack during time window k, then p percentage of sensors at that RSU are selected for attack and each of the attacked sensors is randomly assigned to have its speed readings altered by a deductive attack from Equation 1 or an additive attack from Equation 2.

V. Sensing Architecture

In this section, we present a decentralized system architecture for efficient sensing over a large city in real time. The system is comprised of three central components as shown in Figure 1.

A. Road Sensor System - Sensor Level

Traffic information is maintained by sensors distributed throughout the network edges. The sensor units are responsible for capturing current speed values at each road. Together, the sensor network provides real-time monitoring of the transportation network. In the context of our data, each TMC ID [26] represents a sensor streaming real-time vehicle speed information.

B. Roadside Unit System - RSU Level

Roadside Units (RSUs) are small, low powered devices with wireless capabilities [5]. RSUs have two main responsibilities. First, the RSU level is responsible for communicating data from the sensors to the central cloud. Second, spare processing capacity is used for zone level anomaly detection described in Section VII-A. A depiction of the interaction between the sensor level and RSU level is shown in Figure 2.

C. Utility System and Cloud Service

The cloud service is a broad term incorporating the utility system, routing services and long term data storage. For this work we are primarily concerned with the utility system, which is a collection of high powered computation nodes residing in the cloud. The role of the utility system is providing processing for sensor level detection.

Fig. 2. Data Collection Framework - RSU-Sensor Interaction

VI. RSU Deployment - Clustering Procedure

The way in which RSU devices are deployed affects resource utilization and network efficiency. Therefore in this section we provide a constrained hierarchical clustering algorithm for RSU deployment.

As each RSU is responsible for a subset of sensors, ultimately the goal of the algorithm is to match each sensor s_i with an RSU. Through a mapping process, each RSU r will be responsible for the data collected from a subset of sensors $S^r \subset S$ where S^r is a collection of sensors mapped to RSU r.

Since zone level detection outlined in Section VII-A is optimized when sensors with similar traffic patterns are grouped together (see Section VIII-C), feature sets are generated for each cluster using training speed data from the HERE API. For cluster c consisting of sensors S^c, the speed data from these sensors is broken into 30 minute time windows from 7:00AM to 9:00PM, resulting in 28 features total. By taking the mean speed at each time window k, the feature set for cluster c is represented by $F^c = \{f^c(k_1), ..., f^c(k_{28})\}$. Clusters are grouped together by similarity. We therefore use euclidean distance to measure the similarity between two clusters.

Algorithm 1 RSU Clustering

Input: m, η
Initialize: $C \leftarrow S$
while $len(C) > m$ **do**
  $l_{min} = \infty$
  for $i = 0$ **to** $len(C)$ **do**
    $c_j \leftarrow nearest(c_i, C)$
    if (len(S^{c_i})+len(S^{c_j})) $\leq \eta$ **then**
      $l_{(i,j)} \leftarrow euclideanDist(F^{c_i}, F^{c_j})$
      if $l_{(i,j)} < l_{min}$ **then**
        $l_{min} \leftarrow l_{(i,j)}$, $c_v \leftarrow c_i$, $c_w \leftarrow c_j$
      end if
    end if
  end for
  $c_{new} \leftarrow merge(c_v, c_w)$
  add c_{new} to C
  remove c_v and c_w from C
end while

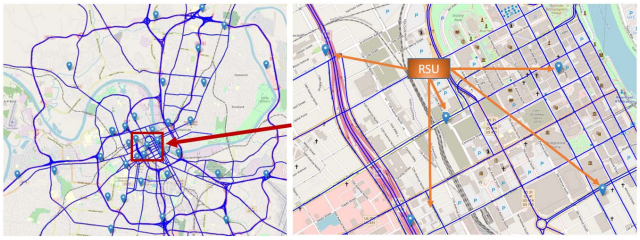

Fig. 3. Cluster RSU - full layout and downtown Nashville. The clustering approach results in multiple RSUs in the highly travelled downtown area, allowing for resources to be deployed according to demands of the sensor network.

The clustering procedure is detailed in Algorithm 1. Algorithm 1 relies on three helper functions:

- $nearest(c_i, C)$: returns the cluster whose centroid is geo-spatially closest to the centroid of cluster c_i, according to haversine distance.
- $euclideanDist(F^{c_i}, F^{c_j})$: returns the euclidean distance between the feature sets of clusters c_i and c_j.
- $merge(c_v, c_w)$: returns a new cluster. The feature set of the new cluster is recalculated using the combined set of sensors in the new cluster.

Line one specifies the input parameters where m is the target number of clusters and η is the maximum number of sensors in a cluster. In the initialization step, C represents the set of all clusters. C is initially set such that each cluster consists of a single sensor.

The clustering procedure starts at line (3) and continues until the number of clusters equals m. As we loop through each cluster c_i, the geographically nearest cluster c_j is identified. If the η constraint is satisfied and the euclidean distance between F_i and F_j is less than l_{min} then we reassign l_{min} to $l_{(i,j)}$ and update c_v and c_w accordingly. After each cluster is iterated through, (c_v, c_w) are merged into a single cluster c_{new} which is added to C and c_v, c_w are subsequently removed.

The visual representation of the cluster RSU network is provided in Figure 3. For comparison, a grid RSU layout where the geo-spatial boundary of the sensor network is divided into a square grid with an RSU located at the center of each grid was generated as shown in Figure 4.

Comparing the two layouts, the cluster RSU layout does a better job concentrating RSUs in areas where there are a high number of sensors. Additionally, by only merging spatially adjacent clusters, the subset of sensors at each RSU maintains a connected sub-graph of road edges.

The effect of η is illustrated by the sensor distributions in Figures 5 and 6. By limiting the maximum number of sensors in each RSU, the processing and networking demands placed on each RSU can be controlled. Conversely, the grid layout includes two RSUs that taken together, are responsible for approximately 30% of all the sensors in the network. This imbalance in sensor distribution creates high stress on a few RSUs while under-utilizing the resources at the remaining RSUs.

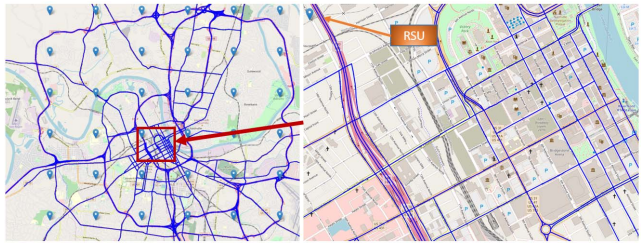

Fig. 4. Grid RSU - full layout and downtown Nashville. The grid layout results in only one RSU in the highly travelled downtown area.

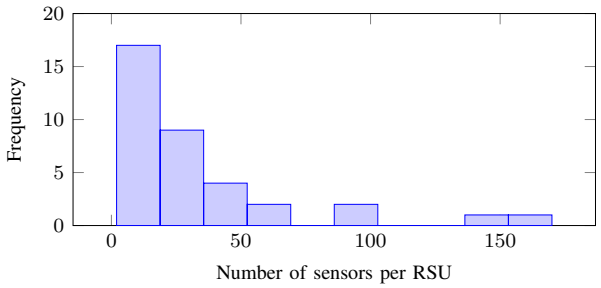

Fig. 5. Grid RSU layout histogram - sensors per RSU distribution. This layout places high stress on a small number of RSUs while under-utilizing the full processing capabilities of the network.

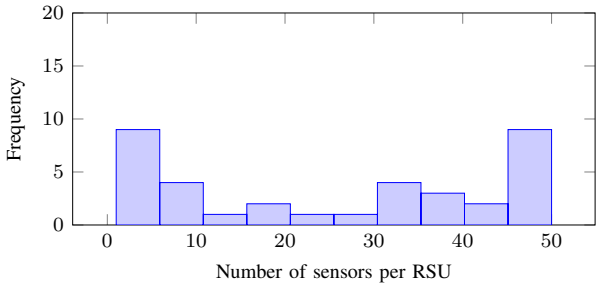

Fig. 6. Cluster RSU layout histogram - sensors per RSU distribution. Number of sensors in an RSU does not exceed $\eta = 50$. Constraining the number of sensors at an RSU places an upper bound on processing demand and ensures processing requirements do not exceed the capacity of RSU hardware.

VII. Anomaly Detection

This section describes our novel two-tiered anomaly detection approach in which zone level detection is continuously run at the RSU network and sensor level detection is used to identify sensors compromised by data integrity attacks. Sensor level detection is only performed on a set of sensors when an attack is first identified at the zone level.

A. Zone Level Detection

The zone level detection provides a mechanism for identifying data integrity attacks at the RSU level. Zone level detection is processed at the RSUs.

Each sensor continuously transmits time-stamped speed data to its RSU. Since each RSU r is responsible for a subset of sensors, the RSU collects the data from its set of sensors in

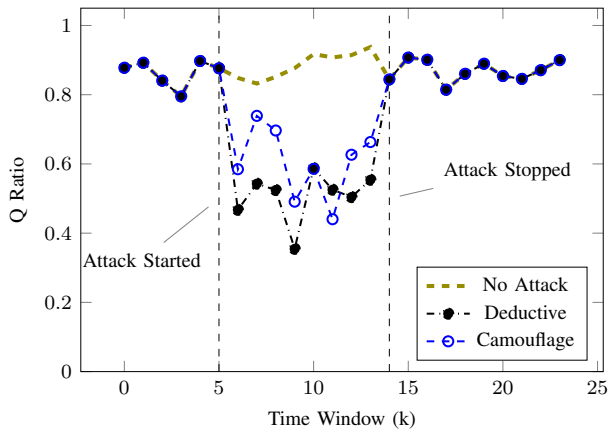

Fig. 7. $Q^r(k)$ under deductive and camouflage attacks at a single RSU. Time Window (k) set to 30 minute intervals, δ=2.5 and p=35%.

the last time window k. At each time window k, the harmonic mean $HM^r(k)$ and arithmetic mean $AM^r(k)$ are calculated per Equations 3 and 4 respectively. The statistical metric used for anomaly detection is the ratio of $HM^r(k)$ to $AM^r(k)$, as shown in Equation 5.

$$HM^r(k) = \frac{S}{\sum_{s=1}^{S} \frac{1}{d_s}} \tag{3}$$

$$AM^r(k) = \frac{\sum_{s=1}^{S} d_s}{S} \tag{4}$$

$$Q^r(k) = \frac{HM^r(k)}{AM^r(k)} \tag{5}$$

While traditional central tendency detection methods based on arithmetic mean or median are effective in detecting additive or deductive attacks, camouflage attacks go undetected since arithmetic mean and median remain the same. As shown in Figure 7, where speed readings for 35% of sensors at a selected RSU were subjected to a δ attack of 2.5, Q^r responds to camouflage attacks as well as deductive attacks.

For detection, $Q^r(k)$ is compared to the historical average and standard deviation of $Q^r(k)$ at time window k as shown in Equations 6 and 7. ϵ^r is a threshold that is unique to each RSU. An investigation for determining ϵ^r is provided in Section VIII-A

$$Q^r(k) < Q^r_{ave}(k) - \epsilon^r * Q^r_{std}(k) \tag{6}$$

$$Q^r(k) > Q^r_{ave}(k) + \epsilon^r * Q^r_{std}(k) \tag{7}$$

B. Sensor Level Anomaly Detection

For many smart transportation applications, such as optimal routing systems, we must identify which sensors are attacked to mitigate the effects of data integrity attacks in real time. Therefore sensor level detection is required.

For sensor level detection we use Gaussian Processes to get the expected speed and standard deviation at a given sensor using the 15 sensors closest to that sensor. This approach assumes a high correlation between speed readings at nearby sensors [17]. We use CUSUM for detection, however as sensor level detection is not continuous in our two-tiered anomaly detection approach the process is restricted to two windows.

As a kernel function, the commonly used RBF (squared exponential) kernel is used. A study of detection accuracy and computation time between continuous sensor level detection compared to two-tiered anomaly detection is provided in Section VIII.

VIII. Simulations and Results

A. Determination of ϵ For Zone Level Detection

The effectiveness of zone level detection is highly dependent on ϵ. For each RSU we simulated 100 deductive and 100 camouflage attacks with δ held constant at 2.5 in which 35% of the sensors at an RSU are attacked. For each attack, a random time window in the testing set between 7:00AM and 9:00PM was attacked and zone detection was performed. To obtain false positive and true negative results, zone detection was also run at the same time window without the presence of a data integrity attack. The process was repeated for ϵ^r values ranging from 0 to 10 and recall (TPR) and false positive rate (FPR) were recorded at each simulation step.

The cost of false positives at the zone level in two-tiered anomaly detection is only in terms of the increased computation time required to run sensor level detection. Since the cost of false positives is low, the value of ϵ^r is set such that the number of false positives is approximately 20% at each RSU. Therefore while the exact value of ϵ^r is unique at each RSU, we can expect the resulting false positive rate to be roughly 20%.

Figure 8 provides a graphical representation of this process for an example RSU, where recall was 99% and 94% for deductive and camouflage attacks respectively when FPR was 20%. As discussed in the following section, recall at the zone level remains relatively consistent across the RSU network following this procedure.

B. Zone Level Detection - Investigation of δ and p

Here we investigate the bounds for which our zone level detection is viable. There are two primary considerations in quantifying the severity of a data integrity attack. First, the severity of the attack on each affected sensor is represented by δ (see Equations 1 and 2). Second, the percentage of sensors affected by the attack (p) represents the breadth of an attack at each RSU.

Two simulations are configured. First, p was held constant at 35% and δ was varied from 0 to 3.5. For each δ value, 100 deductive attacks and 100 camouflage attacks were again simulated at each RSU in the network. However for this simulation, true and false positives and negatives at each RSU were aggregated together at each value of δ, resulting in a single recall value for the entire network at every δ. The results

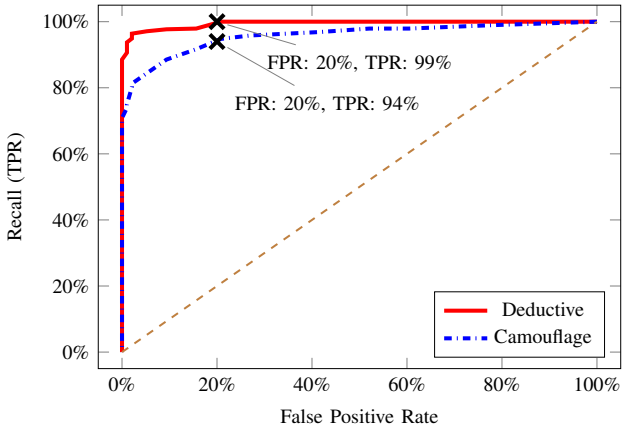

Fig. 8. Epsilon true positive rate (TPR) vs false positive rate (FPR) for a single RSU, deductive and camouflage attacks. These curves were generated for each RSU and the value of ϵ^r was selected such that FPR was 20%. Attack parameters: $\delta = 2.5, p = 35\%$

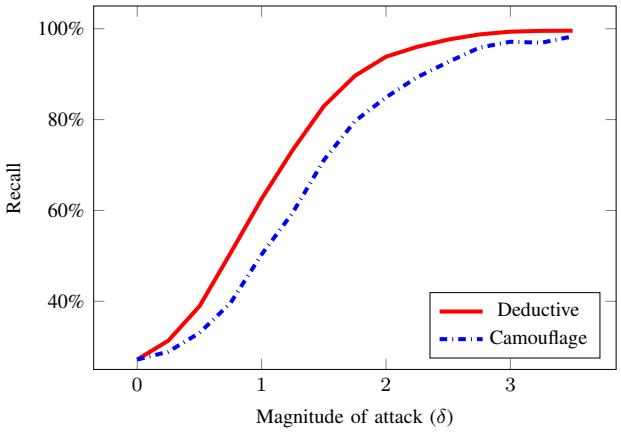

Fig. 9. Recall (TPR) aggregated across all RSUs with five or more sensors vs magnitude of attack (δ). ϵ^r used for detection and the percentage of sensors attacked p is held constant at 35%. Recall for the network is greater than 90% when δ is greater than 2.25 for both deductive and camouflage attacks.

of this simulation are provided in Figure 9, and show that for both deductive and camouflage simulations the recall is greater than 90% when δ is greater than 2.25.

For the second simulation the same procedure was followed except this time δ was held constant at 2.5, while p was varied from 0% to 60%. As shown in Figure 10, zone level detection retains 90% accuracy for attacks affecting as low as 25% of the sensors at an RSU.

C. Zone Level Detection Comparison - Grid vs Cluster RSU Deployment

In Section VI we discussed the advantages of constrained hierarchical clustering for RSU placement in terms of maximizing hardware resources. Here we investigate the benefits of this approach in terms of anomaly detection.

The same zone level attack simulation as detailed in Section VIII-A was applied to the cluster RSU and grid RSU networks respectively, with $\delta = 2.5$ and $p = 35\%$. For detection, ϵ^r, generated from Section VIII-A, is used. To find ϵ^r for each RSU in the grid network, the process in Section VIII-A was repeated for the grid network. Recall statistics for each RSU is provided in Figure 11 while precision is shown in Figure 12.

Both networks are capable of running zone level detection, as average recall was over 90% for both RSU configurations. This implies that our zone level detection algorithm is an adequate solution regardless of RSU layout. However, recall is higher for the cluster RSU configuration showing that clustering groups of sensors by traffic pattern similarity has a positive effect on zone level anomaly detection.

D. Two-Tiered Anomaly Detection vs Sensor Only Detection

In Section VII two-tiered anomaly detection was outlined. We now move on from zone level detection and investigate two-tiered anomaly detection compared to continuous sensor

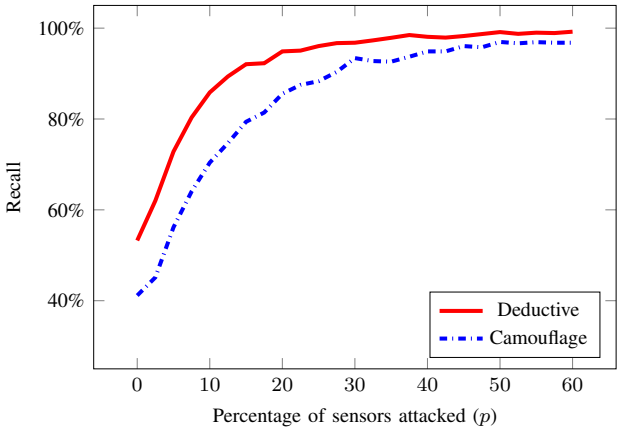

Fig. 10. Recall (TPR) aggregated across all RSUs with five or more sensors vs percentage of sensors attacked at each RSU (p). Full network simulation - each unique ϵ^r used for detection and δ is held constant at 2. Recall for the network is greater than 90% for attacks affecting 25% or more sensors for both deductive and camouflage attacks.

only detection with Gaussian Processes. The simulation procedure remains the same as outlined in Section VIII-C, however now we find true positives, true negatives, false positives and false negatives at the *sensor level*.

Recall and precision are provided in Figures 13 and 14 respectively. Note that while true and false positives and negatives were calculated at the sensor level, recall and precision as shown in Figures 13 and 14 are aggregated at the zone level. This procedure allows us to visualize the interaction between zone level detection in the previous section and two-tiered detection provided here, as well as directly compare computation time per RSU as shown in Figure 15.

We find that precision and recall are similar between two-tiered and sensor only detection. The major difference between these approaches is in computation time, where two-tiered detection is 35% less than sensor only detection and the

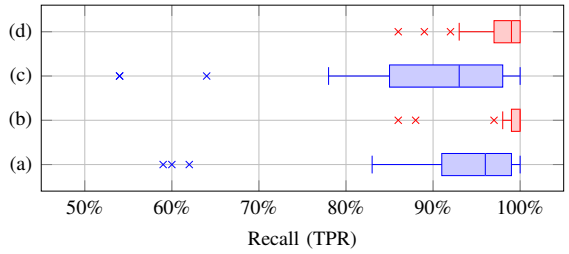

Fig. 11. Zone level recall of (a) grid RSU layout - deductive attack simulation, (b) Cluster RSU layout - deductive attack simulation, (c) grid RSU layout - camouflage attack simulation, (d) cluster RSU layout - camouflage attack simulation. Each data point represents recall at a single RSU in the network. Only RSUs with more than 5 sensors considered. Attack parameters: $\delta = 2.5, p = 35\%$

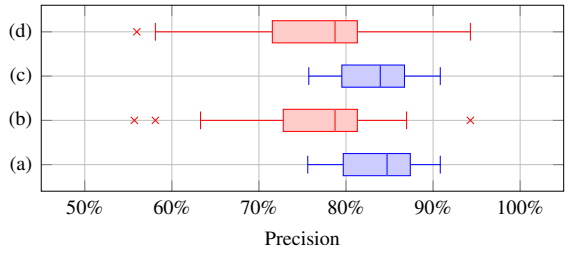

Fig. 12. Zone level precision of (a) grid RSU layout - deductive attack simulation, (b) Cluster RSU layout - deductive attack simulation, (c) grid RSU layout - camouflage attack simulation, (d) cluster RSU layout - camouflage attack simulation. Each data point represents precision at a single RSU in the network. Only RSUs with more than 5 sensors considered. Attack parameters: $\delta = 2.5, p = 35\%$

computation time of zone level detection was negligible in relation to the other two approaches.

An important observation is that our simulation procedure effectively attacked 50% of the time, a much higher percentage than can be expected in an actual deployment scenario. As two-tiered detection only requires sensor level detection when an attack is detected at the zone level and the computation time from zone level detection is negligible, it can be assumed that a 35% reduction in computation time between two-tiered and sensor only detection is a conservative estimate.

IX. Conclusion and Future Work

In this paper we presented a novel two-tiered anomaly detection framework that maintains similar accuracy to current state of the art systems with a significant reduction in processing requirements. Additionally we covered the integration of our anomaly detection framework in decentralized smart transportation systems and provided a constrained hierarchical clustering algorithm for RSU deployment.

Our current work focuses on deductive and camouflage attacks. We would like to extend this to a variety of potential attacks. Therefore, future work will include extending this work to additive attacks as well as strategic attacker events in which the attacker has a comprehensive understanding of transportation system behavior. Additive attacks have worked in other aggregate anomaly detection cases [16]. These attacks

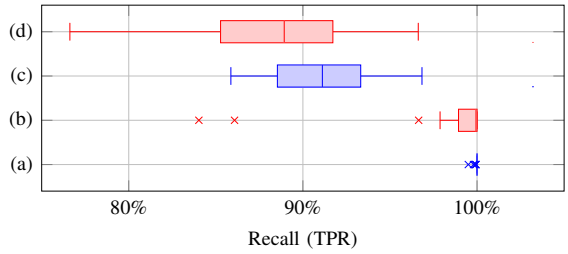

Fig. 13. Sensor level recall of (a) sensor only (GP) - deductive attack simulation, (b) two-tiered detection - deductive attack simulation, (c) sensor only (GP) - camouflage attack simulation, (d) two-tiered detection - camouflage attack simulation. Each data point represents aggregate recall of sensor level detection at a single RSU in the network. Only RSUs with more than 5 sensors considered. Attack parameters: $\delta = 2.5, p = 35\%$

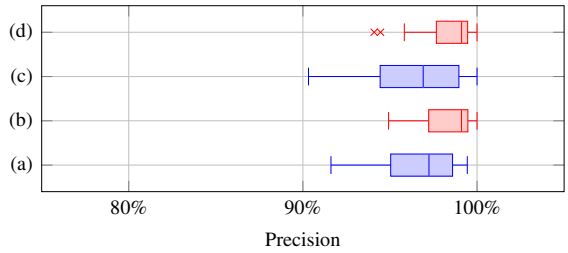

Fig. 14. Sensor level precision of (a) sensor only (GP) - deductive attack simulation, (b) two-tiered detection - deductive attack simulation, (c) sensor only (GP) - camouflage attack simulation, (d) two-tiered detection - camouflage attack simulation. Each data point represents aggregate precision of sensor level detection at a single RSU in the network. Only RSUs with more than 5 sensors considered. Attack parameters: $\delta = 2.5, p = 35\%$

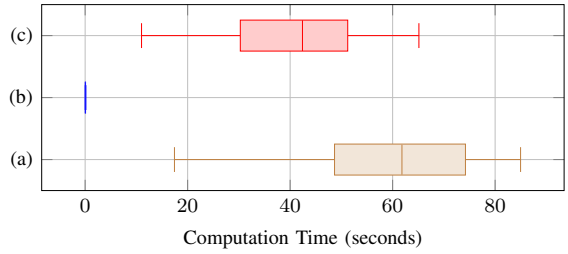

Fig. 15. **Computation time** (seconds) of (a) sensor level detection (GPs), (b) zone level detection, (c) two-tiered detection

cause an increase in the Q value, which is capped at one. By taking the inverse of our metric in relation to a jam factor, which is also available as one of the data streams from the HERE API, we can extend this work to such attacks.

Additionally, we would like to investigate the cascading effects of data-integrity attacks on routing systems. We also plan to use additional information, such as weather or planned events, to predict anomalies ahead of time.

Acknowledgment

This work is sponsored in Siemens, CT and National Science Foundation under award numbers 1818901 and 1647015.

REFERENCES

[1] C. Samal, L. Zheng, F. Sun, L. J. Ratliff, and A. Dubey, "Towards a socially optimal multi-modal routing platform," *arXiv preprint arXiv:1802.10140*, 2018.

[2] F. Sun, C. Samal, J. White, and A. Dubey, "Unsupervised mechanisms for optimizing on-time performance of fixed schedule transit vehicles," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–8.

[3] F. Sun, Y. Pan, J. White, and A. Dubey, "Real-time and predictive analytics for smart public transportation decision support system," in *Smart Computing (SMARTCOMP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–8.

[4] V. K. Shah, S. Bhattacharjee, S. Silvestri, and S. K. Das, "Designing sustainable smart connected communities using dynamic spectrum access via band selection," in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 2017, p. 12.

[5] F. Perry, K. Raboy, E. Leslie, Z. Huang, D. Van Duren *et al.*, "Dedicated short-range communications roadside unit specifications." United States. Dept. of Transportation, Tech. Rep., 2017.

[6] M. B. Sinai, N. Partush, S. Yadid, and E. Yahav, "Exploiting social navigation," *arXiv preprint arXiv:1410.0151*, 2014.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[8] 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, 2012.

[9] 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.

[10] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. A. Pardo, and H. J. Scholl, "Understanding smart cities: An integrative framework," in *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 2012, pp. 2289–2297.

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

[12] S. Eisele, I. Mardari, A. Dubey, and G. Karsai, "Riaps: resilient information architecture platform for decentralized smart systems," in *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE, 2017, pp. 125–132.

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

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

[15] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. S. Shen, "Energy-theft detection issues for advanced metering infrastructure in smart grid," *Tsinghua Science and Technology*, vol. 19, no. 2, pp. 105–120, 2014.

[16] 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*. ACM, 2018, pp. 173–185.

[17] A. Ghafouri, A. Laszka, A. Dubey, and X. Koutsoukos, "Optimal detection of faulty traffic sensors used in route planning," in *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*. ACM, 2017, pp. 1–6.

[18] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: a statistical anomaly approach," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 76–82, 2002.

[19] V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou, and B. Maglaris, "Hierarchical anomaly detection in distributed large-scale sensor networks," in *Computers and Communications, 2006. ISCC'06. Proceedings. 11th IEEE Symposium on*. IEEE, 2006, pp. 761–767.

[20] A. G. Prieto, D. Gillblad, R. Steinert, and A. Miron, "Toward decentralized probabilistic management," *IEEE Communications Magazine*, vol. 49, no. 7, 2011.

[21] G. Biswas, H. Khorasgani, G. Stanje, A. Dubey, S. Deb, and S. Ghoshal, "An application of data driven anomaly identification to spacecraft telemetry data," in *Prognostics and Health Management Conference*, 2016.

[22] J. Chi, Y. Jo, H. Park, and S. Park, "Intersection-priority based optimal rsu allocation for vanet," in *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*. IEEE, 2013, pp. 350–355.

[23] B. Aslam and C. C. Zou, "Optimal roadside units placement along highways," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*. IEEE, 2011, pp. 814–815.

[24] B. Aslam, F. Amjad, and C. C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," in *Computers and Communications (ISCC), 2012 IEEE Symposium on*. IEEE, 2012, pp. 000 423–000 429.

[25] (2018) Here api. [Online]. Available: https://developer.here.com/

[26] (2018) Traffic message channel. [Online]. Available: https://wiki.openstreetmap.org/wiki/TMC