Fall 2021

Security against data falsification attacks in smart city applications

Venkata Praveen Kumar Madhavarapu

Department: Computer Science

SECURITY AGAINST DATA FALSIFICATION ATTACKS IN SMART CITY APPLICATIONS

by

VENKATA PRAVEEN KUMAR MADHAVARAPU

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2021

Approved by:

Sajal K. Das, Advisor
Shameek Bhattacharjee
Venkata Sriram Siddhardh Nadendla
Tony T. Luo
Nan Cen

ABSTRACT

Smart city applications like smart grid, smart transportation, healthcare deal with very important data collected from IoT devices. False reporting of data consumption from device failures or by organized adversaries may have drastic consequences on the quality of operations. To deal with this, we propose a coarse grained and a fine grained anomaly based security event detection technique that uses indicators such as deviation and directional change in the time series of the proposed anomaly detection metrics to detect different attacks. We also built a trust scoring metric to filter out the malicious devices. Another challenging problem is injection of stealthy data falsification. To counter this, we propose a novel information-theory inspired data driven device anomaly classification framework to identify compromised devices launching low margins of stealthy data falsification attacks. The modifications such as expected self-similarity with weighted abundance shifts across various temporal scales, and diversity order are appropriately embedded in resulting diversity index score to classify the devices launching different attacks with high sensitivity compared to the existing works. Active learning, a semi-supervised classification approach is used to cluster the malicious and benign sensors depending on the score.

Adversarial machine learning (AML) is a technique that fools the machine learning models with the malicious input. The resulting performance of the existing machine learning models will drop when the adversary employs AML. Common types of AML techniques are evasion attacks and poisoning attacks. For this purpose, we proposed a Generative Adversarial Network (GAN) based solution to detect different kinds of evasion and poisoning attacks. Our proposed solutions are validated with the help of real-world smart metering datasets from Texas and Ireland, and smart transportation data from Nashville.

ACKNOWLEDGMENTS

I express my gratitude to my advisor, Dr. Sajal K. Das, and co-advisor, Dr. Shameek Bhattacharjee for their advice, support and guidance throughout my doctoral studies at Missouri University of Science and Technology. I am thankful to my parents, Venkateswara Rao and Dhana Lakshmi as they supported me throughout this journey. I must also mention my colleague, collaborator and close friend Prithwiraj Roy who helped me immensely.

I am very grateful to Dr. Venkata Sriram Siddhardh Nadendla, Dr. Tony Luo and Dr. Nan Cen for being my PhD committee members and giving their insights that helped me significantly improve the contents and presentation of my dissertation. I am also grateful to Dr. Yanjie Fu for being my committee member during my PhD qualifier.

I am really grateful to my masters advisor Dr. Bidyut Gupta who encouraged me to pursue my PhD career and helped me through my transition from masters to PhD.

Finally, I would like to acknowledge the National Science Foundation for supporting this work with NSF grants CNS-1545050, CNS-1818942, and SaTC-2030624.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

A smart city is a framework, predominantly composed of Information and Communication Technologies (ICT), to develop, deploy, and promote sustainable development practices to address growing urbanization challenges. A big part of this ICT framework is essentially an intelligent network of connected objects and machines that transmit data using wireless technology and the cloud. Citizens engage with smart city ecosystems in various ways using smartphones and mobile devices and connected cars and homes. Pairing devices and data with a city's physical infrastructure and services can cut costs and improve sustainability. Communities can improve energy distribution, streamline trash collection, decrease traffic congestion, and even improve air quality with help from the IoT.

There are different smart city applications including smart energy, smart healthcare, smart transportation, smart agriculture, smart infrastructure [3]. This is shown in Figure 1.1. In this work we will be predominantly concentrating on smart grid and smart transportation.

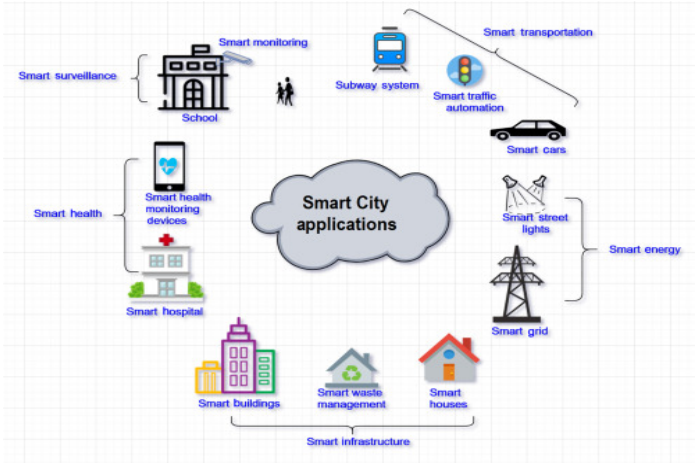


Figure 1.1. Smart City Applications

1.1. SMART GRID

Smart grid is one of the important part of the smart city. This work focuses on the possible threats in smart grid and proposes solution for different types of data falsification attacks. In this section, we will see what is a smart grid and its components along with the advantages of the system.

1.1.1. Smart Grid Architecture. The electrical power delivery system has often been considered the greatest and most complex network ever built. It consists of wires, cables, towers, transformers, monitoring devices, and circuit breakers, all connected together. The transfer of power from generation in the grid to utilization in homes and industries will be as shown in Figure 1.2.

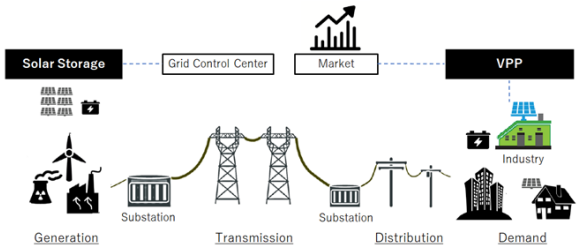


Figure 1.2. Electric Power System.

Historically, the electric power grid operators and planners had limited information for the system status and behavior of the grid. The only available information was measurements from decentralized SCADA (supervisory control and data acquisition) systems, mostly recorded at several-second intervals, and they did not include the physical state variables of the a.c network like the complex voltages at every node, and time-shifted voltage information. Thus the primary focus of the system was designed for the most extreme conditions, specifically, peak loads and faults – and then try to ensure that the grid operated within that expected range. Despite the good design, operation, and maintenance efforts, over 90% of customers' electric outages occur due to problems on the distribution system rather than from transmission or generation level problems. Moreover, with the growth of distributed energy resources (example: rooftop photo-voltaic cells), two-way electric-

ity flows and new customer devices such as electric vehicles necessitate better situational awareness and insight into distribution system conditions and performance to make the grid more robust, more efficient, more distributed, re-configurable, more interactive, with faster protection and control.

To meet these requirements, smart grid integrates modern advanced sensor technology, measurement technology, communication technology, information technology, computing technology, and control technology into it, where information and electricity flow bi-directionally and the smart grid can: (1) Enable active participation by customers; (2) Accommodate all generation and storage options; (3) Enable new products, services, and markets; (4) Optimize asset utilization and operate efficiently; (5) Anticipate and respond to system disturbances. The architecture of Smart Grid can be seen in Figure 1.3

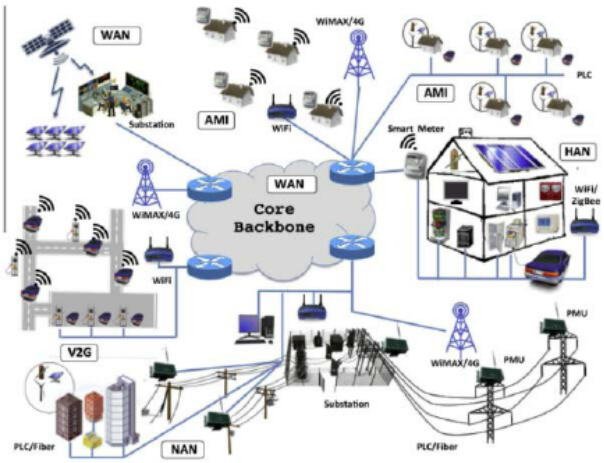


Figure 1.3. Smart Grid Architecture.

1.1.2. Advanced Metering Infrastructure. Advanced Metering Infrastructure (AMI) is one of the basic units of the smart grid technology. AMI collects data on loads and customer's power consumption [4], from Smart Meters installed on the customer site (see Figure 1.4). Such data plays a pivotal role in several critical tasks such as automated billing, demand response, load forecast and management [4, 5, 6].

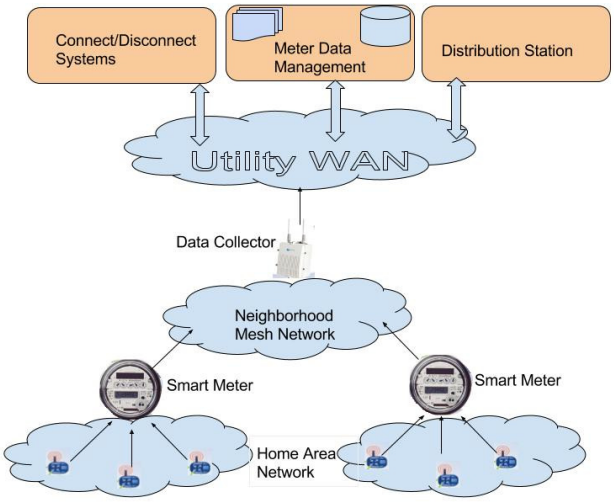


Figure 1.4. Architecture of AMI [1]

Apart from automated billing, strategic tasks are expected to be performed by future smart grids, based on the AMI power consumption data. For example, AMI will have implications on tasks such as daily and critical peak shifts [7, 8]. When the consumption increases beyond a critical limit, emergency 'peaker plants' are currently used by most utilities for additional power generation to meet the demand. However, such peaker plants are extremely carbon as well as cost intensive. In the modern grid, the utility will also have the option for automated demand response where utilities pay customers to shut certain appliances temporarily (peak shifting) to obviate the need for additional generation [9, 10]. In general, an accurate short or long term data on loads and consumption will aid in accurate demand response, load forecast and planned generation in the future smart grid [11]. Therefore, the integrity of the AMI data is of utmost importance.

Defense against falsification of power consumption data from AMIs, has largely focused on *electricity theft* [12, 13, 14, 15], where individual customers are primary adversaries who report lower than actual usage for lesser bills. Since isolated smart meters belonging to rogue customers reduce the value of power consumption, we term such an adversarial attack as a *Deductive* mode of data falsification. However, it has been widely acknowledged that given the cyber and interconnected nature of AMI, it could potentially

be the target of organized adversaries such as cyber criminals [16], utility insiders [17], or business competitors [18]. Organized adversaries can compromise *several smart meters* and then *spoof* false power consumption data [13] from smart meters. Organized adversaries are more equipped to crack/leak cryptographic secrets, have a higher attack budget, and possess the ability to simultaneously attack other elements of the grid (e.g., audit logs, transformers meters) in order to avoid easy consistency checks on false data.

1.2. SMART TRANSPORTATION

Vehicular Ad hoc Network (VANET) is the connection of group of vehicles that can communicate with each other and with the infrastructure domain through internet and radio channels. VANETs are a subgroup of MANETs where communicating nodes are mainly vehicles and roadside infrastructures. At present, VANETs have many implementations across different aspects like smart transport systems, driving assistance, public security, roadside facility locator, toll collection, road traffic control, freeway internet connection and increasing security and efficacy of freeway systems. Through the use of Dedicated Short-Range Communication (DSRC) [19], VANETs support Intelligent Transportation System (ITS) [20]. Wireless Access in Vehicular Environment (WAVE) [21] is one of the standards to implement VANET. Figure 1.5 illustrates the basic topology of VANET.

Two types of communication technologies are implemented for VANET. One is Vehicle to Vehicle (V2V) and another is Vehicle to Infrastructure (V2I). This is shown in Figure 1.6 Vehicles consist of GPS, processors, sensors and antennas which are known as On Board Unit (OBUs) to correspond with other vehicles. Vehicles also communicate with infrastructures at the roadside at a static distance from each other known as Road Side Units (RSUs). RSUs can be mobile and they use wired or wireless medium to communicate with each other and the Internet. Vehicles can be connected to Internet through V2I since RSUs are connected to the Internet. Real time and emergency messages can be transmitted using V2V communications to avoid accidents and traffic congestions. Figure 1.5 Vehicular Ad

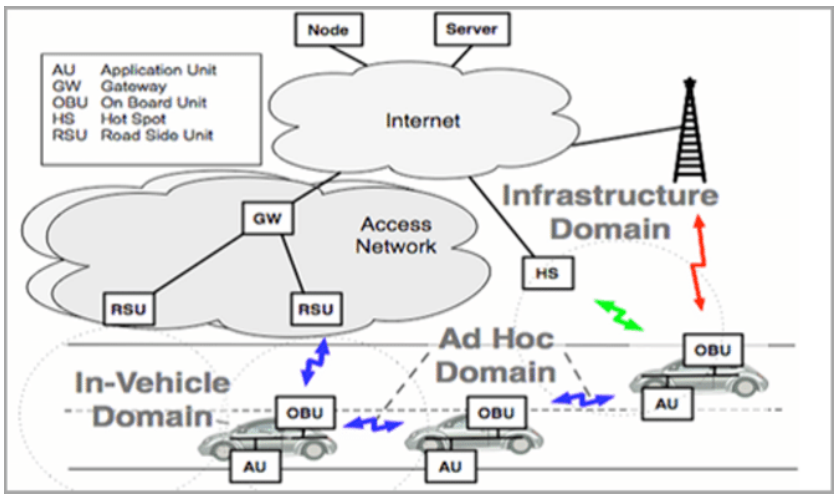


Figure 1.5. Vehicular Network

Hoc Network (VANET). VANETs are required to implement security measures, for instance, secrecy, reliability and approval to offer protection against invaders and mischievous nodes since VANETs transmit emergency, life critical real-time information. Wormhole attack [22, 23], Purposeful attack [24], Illusion attack [25], Denial of Service (DoS) [26], Sybil attack [27, 28, 29] are some of the security attacks which can hamper the privacy of the person driving the vehicle as well as the vehicle. Eventually, these attacks may cause death of human lives by reducing traffic safety. Hence, many researchers are extensively working on the security of VANETs. The primary reason of providing security in VANET is necessary so that the original identity of the drivers cannot be disclosed at any time in VANET since malicious nodes can launch attacks using this information as false identity.

During V2I communication safety and privacy is very important since drivers and vehicles have to disclose their identity to communicate with RSUs. Vehicles need to ensure the authenticity of the received information before reacting to the received information. V2I is also responsible for recording the speeds of vehicles through Traffic Message Channels (TMCs). The data from a group of TMCs will be aggregated at the RSUs. The aggregated data is used to make the traffic decisions which will be shared with the vehicles, So, the

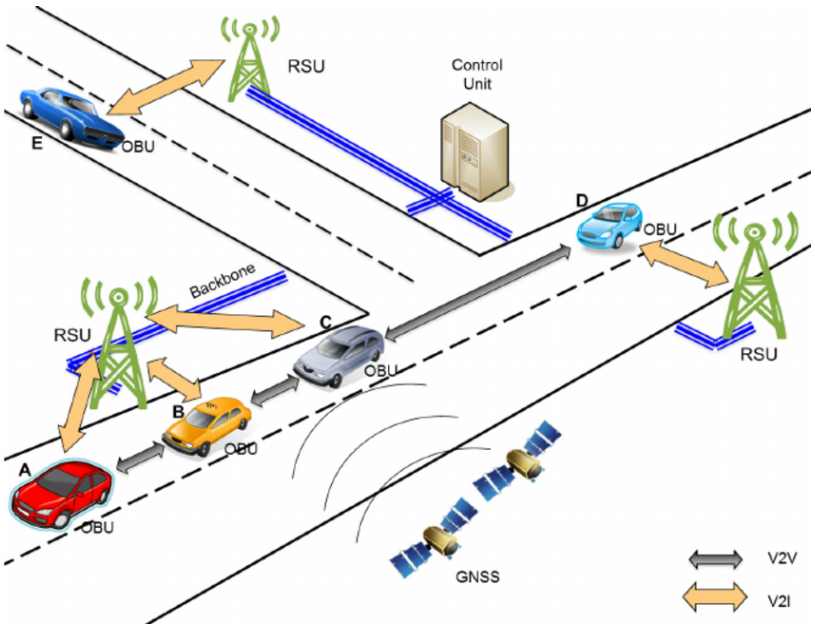


Figure 1.6. V2V and V2I Communications [2]

collected data needs to be accurate to share the true information. There are different possible security threats that can lead to data falsification. Different security issues in smart city applications will be discussed in later sections.

2. SECURITY THREATS IN SMART GRID AND SMART TRANSPORTATION

There are different possible security threats in smart city. We concentrate on two main things, data falsification and adversarial learning.

2.1. DATA FALSIFICATION

We assume an organized adversary that orchestrates data falsification attacks from multiple sensors (smart meters or TMCs) via cyber or physical exploit [30]. Smart meters receive power consumption from various appliances via the Home Area Network (HAN) and sends it to the utility side via the Neighbourhood Area Network. Either the (i) input to the smart meter, (ii) the power consumption data at rest inside the smart meter, (iii) or data in flight may be falsified. An example of falsifying power consumption data at rest is the Puerto-Rico Grid Attack of 2012, where hundreds of smart meter's optical ports were manipulated using laser probes by utility insiders [16, 31], causing the smart meters to record lower than actual power consumption. Similarly, load altering attacks reported in [32], have shown the possibility to change the inputs from appliance loads to the smart meter. Similarly, the data in flight from multiple smart meters to the NaN gateway may be falsified by a traditional man-in-the middle attack. Finally, another possibility is an organized adversary that controls a set of smart meters like a Botnet, collect data from intercepted smart meters, and inject advanced data falsification strategies, that we discuss under the stealthy attack strategies. We discuss the threats for smart meters which can also be applicable to other smart city applications.

Our approach is agnostic of the exploit used to falsify the data. Of course, depending on the exploit the attack scales, strengths, and strategies will vary. Our intention is to capture various kinds of data falsification attack realizations instead of a specific one, since exploits tend to evolve over time and just because an attack has not been realized before, does not mean they will not be experienced in future. We capture this generic data falsification

attack landscape by parameterizing the attack strategy space; taking into account the full range attack scales, strengths, strategy combinations in this section. The following features characterize our threat model:

A) Attack Scale: The fraction of compromised meters, $\rho_{mal} = \frac{M}{N}$, is the *attack scale*, where M is the number of unique smart meters compromised by an organized adversary in a given network. Traditional use of Kullback-Leibler Divergence (KLD) model with statistical aggregates work well, if $\rho_{mal}\%$ [33, 34] are smaller. However, resilience against higher ρ_{mal} has been reported only when associated margins of false data per meter is too high (which facilitates easier detection). However, if the attack budget is high, or a creative adversary finds a cheaper exploit to compromise a meter, or the network size is smaller, then the attack budget constraint does not automatically imply a lower fraction of compromised meters [33]. This is because, in reality, the value of M depends also on the creativity of its exploit, and the micro-grid size N. Given large values of ρ_{mal} are possible in the real world, we take into account a wide variation of ρ_{mal} between 0.10 to 0.90.

B)Average Margin of Attack Strength: Average margin of false data is the average extent of falsification introduced per meter. We observed that in most previous works, the average margin of false data is not parameterized as a variable except in two recent works [33, 35], which report that these methods completely fail to detect meters when their average margin of false data is $\delta_{avg} < 400$. This happens because the standard deviation of data streams are high (430W-480W in AMI applications) due to randomness of human activity, making it difficult for previous methods to achieve success.

C) Attack Types: We consider three different attack types. The adversary seeks to falsify original data points $P^i_{act}(t)$ representing actual energy consumption at time t by some factor δ_t , where $\delta_t \in [\delta_{min}, \delta_{max}]$ and the long term average value of δ_t is δ_{avg} (avg. margin of attack strength).

(i) Additive Attacks: Here the smart meters seek to increase the data from its original values, such that $P^i(t) = P^i_{act}(t) + \delta_t$. Motivation of such attacks are discussed in [32].

(ii) Deductive Attacks: Smart meters seek to decrease the data from its original values, such that $P^i(t) = P^i_{act}(t) - \delta_t$; this is equivalent to electric theft and the most commonly seen attack type [16].

(iii) Alternating Switching: In such an attack, every compromised meter alternates between launching additive and deductive attacks with the same margin of false data at different times of the day to take advantage of dynamic pricing/demand response of electricity. When the prices are high (due to higher demand), it launches a deductive attack, while compensating with an equal margin additive attack when the pricing is low (due to lower demand), causing the mean consumption trends from individual compromised meters practically unchanged. This is device level equivalent to a camouflage attack reported in [30] from two sets of meters in the *same time*, thus blinding a micro-grid level anomaly detector. However, our variation of camouflage attack is launched from the same end point meter to camouflage the end device (meter) level detectors.

D) Stealthy Attack Distribution Strategies: Now we focus on 'how' false data is introduced in the smart meters data streams. Apart from a non-stealthy random bias, we analyze our solution against four stealthy strategies, viz. (i) the data order aware, (ii) incremental ramp, (iii) KLD minimization (iv) persistent strategies. AMI applications are not real time systems; they can tolerate some delay. Therefore, if there is some timing delay due to coordination for the stealthy strategies, it is still practical. We assume that a reasonably organized attacker will have an idea of the data distributions and mechanisms used by usual anomaly detectors, and craft the following strategies accordingly:

(i) Data Order Aware Strategy: It is a stealthy falsification strategy that minimizes the chance of detection against mechanisms utilizing proximity (e.g., Euclidean L_2 distance) between the reported and original data distribution, while keeping the same δ_{avg}. Additionally, this strategy makes sure that the maximum and minimum values in the original and falsified distribution are not different, to prevent obvious statistical outliers.

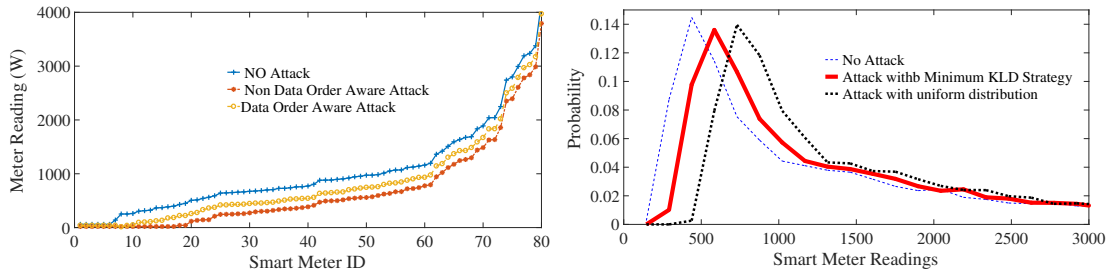
The following strategy is implemented in the following manner: At any time slot t, the adversary sorts the actual recorded data vector from its compromised set of devices such that $P_t^{(1)}(act) \leq, \cdots, P_t^{(m)}(act), \leq P_t^{(M)}(act)$; as well as its corresponding bias vector $\delta_t^1(min) \leq, \cdots, \leq \delta_t^M(max)$. Under an additive attack, the minimum actual data is changed with the highest $\delta_t(max)$, while the maximum observed data is modified with lowest $\delta_t(min)$, and so on like an inverse matching, such that $P_t^{(1)}(act)+\delta_t(max), \cdots, P_t^{(M)}(act)+\delta_t(min)$, subject to the fact that it does not violate bounds on the historical distribution. For a deductive attack, the maximum recorded data is modified by matching with the maximum bias $\delta_t(max)$, while the lowest actual recorded data is altered with the lowest $\delta_t(min)$. For alternating switching attack, the additive and deductive attacks alternate with the strategy mentioned above.

In Figure 2.1(a), the blue line corresponds to the non-attacked value of compromised meters. The yellow and red lines correspond to a realization of falsified data under a data order aware and non-data order aware strategy *with same* δ_{avg}*=200W and* $\rho_{mal} = 40\%$ for 'deductive' attacks from Texas dataset. The same revenue impact is achieved with both strategies, but chances of detection (using proximity/distance/similarity) are smaller in data order aware strategy. The width of the interval of $\delta_t \in [\delta_{min}, \delta_{max}]$ is known as the *aperture of attack*. The aperture is varied as necessary to minimize the euclidean and KLD.

(ii) Incremental/ Ramp / Boil-frog Strategy: This strategy involves a very gradual increase in of δ_t bias over time, until intended δ_{avg} is reached. This attack strategy is termed as boil-frog in AI security and ramp attack in cyber physical system (CPS) security. The strategy causes all temporal metrics to record minimal changes that evolve over time to bypass detection

(iii) KLD Minimizing Strategy: The falsified data is injected in a manner which minimizes the KLD, while preserving the target δ_{avg}. Figure 2.1(b) shows an illustration for a single meter where the adversary crafts a distribution (bold red line) that minimizes the KLD; thus being closer to the actual data distribution (blue line) than to a uniform random bias attack (gray line), even when the $\delta_{avg} = 200$ for both attack strategies.

(iv) Persistent Strategies: These attacks are launched by an adversary that knows our defense model and tries evasion attacks. We show performance under such evasion attacks, by showing the extent to which undetectable strategy space is reduced, and break even time of adversary.



(a) (b)

Figure 2.1. Attack Strategy (a) Data Order Aware (b) KLD Minimizing

2.2. ADVERSARIAL MACHINE LEARNING

Adversarial Machine Learning (AML) is a technique of fooling the machine learning model by providing malicious input. This malicious input is called Adversarial example. This can be understood from the Figure 2.3 where the machine learning model that works well under normal conditions, will result in an undesired outcome when we provide an adversarial example as input. A real world application of adversarial example is shown in Figure 2.2. The machine learning model that can classify the picture as panda will fail to classify correctly when we introduce some noise intelligently to fool the system. In Figure 2.2, the nearly same panda image is classified as panda before the noise and gibbon after

noise by the same classification model. ϵ is the noise level factor. There are some ways to deal with these attacks. One way is to generate possible adversarial examples and retrain the model using the examples.

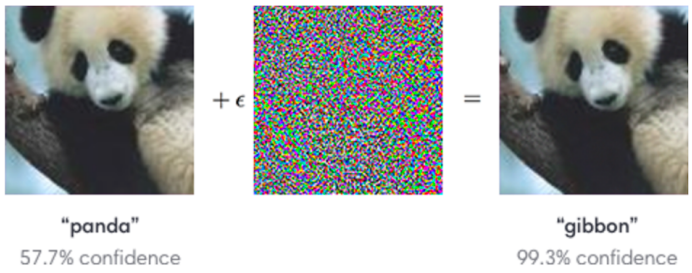


Figure 2.2. Generation of Adversarial Example

AML can be implemented using two main strategies, Evasion attacks and Poisoning attacks. Evasion attacks are quite similar to the example in Figure 2.3. Consider a spam email that is obfuscated to escape the spam filtering. This is an example for evasion attack. The evasion attacks happen during the testing phase of the machine learning system. Poisoning attacks takes place on the training data. The poisoning attacks contaminate the training set which will be used to re-train the model. Intrusion Detection Systems (IDS) which are re-trained can be subjected to poisoning attacks disrupting the whole system.

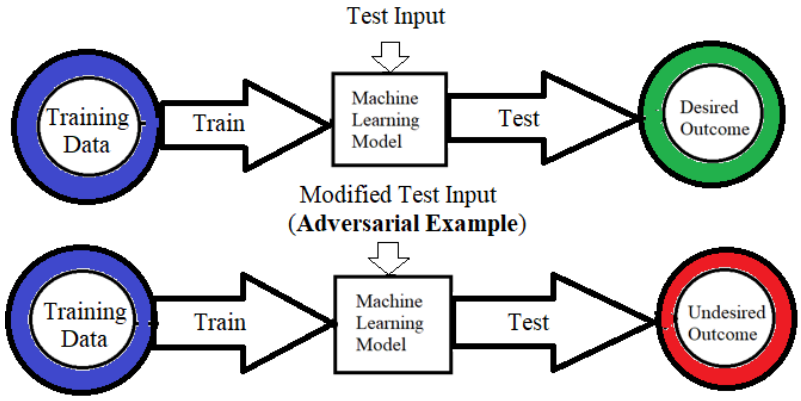


Figure 2.3. Adversarial Example

2.2.1. Evasion Attacks. Evasion attacks are adversarial attacks that are introduced during the testing phase. As shown in Figure 2.4, the machine learning model with good performance can result in undesired outcomes when the testing data is intelligently modified using adversarial examples. This technique was successfully in image recognition to result in a completely different classification output by just modifying few pixel's intensities that is imperceptible to the human eye. In case of ML based security models, for detecting false data injection, the evasion techniques can be used by the adversary to falsify the data points and still escape the detection.

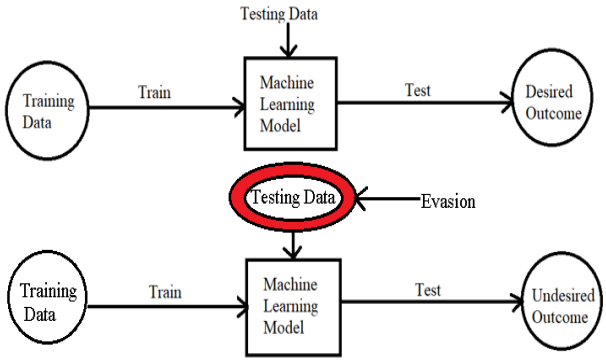


Figure 2.4. Evasion attack on machine learning model

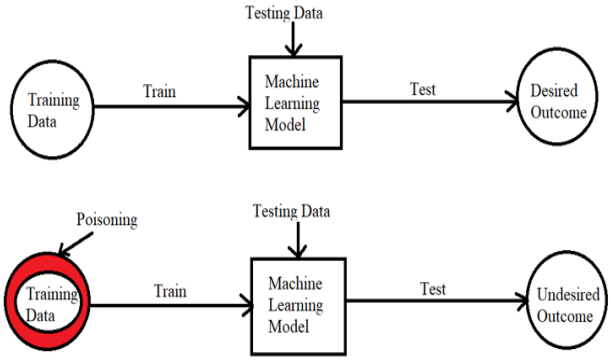


Figure 2.5. Poisoning attack on machine learning model

2.2.2. Poisoning Attacks. These attacks are rare compared to evasion attacks. As shown in Figure 2.4, it happens during the training process. We are aware that adversaries will have full knowledge of our defense mechanism when published. Our base assumption is that the defender has a significant portion of training set that is not attacked. As for this

work, we have only focused on evasion attacks that occur in the testing phase and ignored possibility of poisoning attacks. This is because reducing the undetectable attack strategy space itself is quite a challenging problem. Active poisoning attacks will be part of our future work.

Finally, since CPS application security is still a new field, datasets containing real attacks are unavailable. Therefore, we have parameterized the entire attack strategy space in terms of attack strength, scale, strategies, and types, and reported the failure bounds. This should alleviate concerns over non-availability of real attacks.

3. LITERATURE REVIEW

3.1. EXISTING SMART GRID SECURITY DEFENCE MECHANISMS

Existing approaches for detecting smart meters launching data falsification attack can be broadly divided into three categories, (i) classical machine learning, (ii) information-theoretic, (iii) consensus-based statistical approaches.

The classical machine learning methods use SVMs [12], decision/regression trees (DRT) [36], and neural networks [37, 38]. In [12], the problem was investigated using SVM with a radial basis network when $\delta_{avg} = 450W$, but the percentage of compromised meters assumed was just 1%. In the DRT method [36], the approach does not parameterize the attack strategy space fully, and the attack strength and scales are unclear. Surveyed by[39], the false alarm rates reported by most neural network based methods are much higher even as they do not generalize for an unbounded attack strategy spaces and low profile attacks.

Information-theoretic approaches proposed in [33, 40] use Kullback-Leibler divergence to classify compromised meters, and is a competing approach to our model. Hence, we will show how our attacks perform under these defenses and our solution's detection performance compare with these approaches.

Consensus-based approaches used classical statistics [15, 41], time series [14], robust statistics and density-based learning [35] to identify such smart meters. We chose to compare with [35], since it outperforms the others, and parameterizes the attack strategy space with attack scales and strengths. State estimation based methods are not used since they depend on putting extra monitoring hardware in the higher layers of a smart grid. Note that [30] applies to stealthy attacks at a micro-grid level and but not at the meter level, thus it does not feature in our comparisons.

Gaussian trust model, KL divergence based trust model are some of the existing models to detect the data falsification in smart grid. The approaches are discussed in the following sections.

3.1.1. Folded Gaussian Classifier. The folded Gaussian model uses a trust scoring model based on the distance of the reading from the mean of the data. Then a discrete rating will be assigned depending on the distance. This will be finally used to calculate a score for each meter to judge if the smart meter's readings are falsified or not.

3.1.2. Kullback-Leibler-divergence Trust Model. The work proposed in [33] uses the Kullback-Leibler Divergence between the probability distribution of binary rating levels that use the first standard deviation only. Rather than just having our probability distribution, it adds in approximating distribution. With KL divergence we can calculate exactly how much information is lost when we approximate one distribution (P) with another (Q) over probability space χ, shown in Eqn. 3.1.

$$D_{KL}(P||Q) = \sum_{x \epsilon \chi} P(x) \times log\Big(\frac{P(x)}{Q(x)}\Big) \tag{3.1}$$

This can be used to calculate and compare the probability distribution of the binary rating levels compared to the attack free training stage. The model was designed in a such a way that higher value of divergence implies a big change in the usage trend and will be marked as falsifying data.

Our analysis of existing works in smart meter data falsification found that most methods fail to classify correctly when the attack margins are below 400W, regardless of the attack types and strategies (elaborated in the threat model section). Additionally, the possibility of physical attacks causing data falsification (optical laser attacks [16] and acoustic transduction attacks [42, 43]) render cryptography based and network traffic based intrusion detection methods on IoT devices inadequate. Cybersecurity practices such as static analysis, and signed software updates do not protect against a sensor recording false data since the physical attacks influence the output of sensor hardware that is trusted

by software/firmware [42, 44]. Furthermore, several studies [45, 46] have noted that embedded/hardware/in-situ security of smart meters that provide some protection against physical attacks, is not cost effective due to the large scale nature of meter deployment and variety in commodity hardware. Therefore, providing a device level data driven behavioral anomaly scoring technique is necessary not only as an extra level of security, but as a principle approach for trusting the data from a distributed set of IoT devices (e.g. smart meters), which motivates our approach.

However, non-typical but benign conditions can cause changes in data due to weather conditions, and seasons; and low margin data falsification attacks can easily hide behind such randomness. Our anomaly based detection approach should distinguish between such events and changes that are caused due to attacks. Finally, the method has to generalize across various attacks and datasets.

3.2. ADVERSARIAL MACHINE LEARNING IN SMART GRID

The usage of adversarial examples started in the image processing and recognition. [47] shows the impact of adversarial examples in image processing and proposed a solution on how to handle those threats. The adversarial attacks can be classified as black-box attacks and white-box attacks. In case of black-box attacks, the adversary has the knowledge of the machine learning model used in the system where as in white-box attacks, the adversary don't have access to such information. [48] explains some solutions on dealing with the black-box attacks in image processing. [49] follows the approach to deal with the adversarial examples.

In this work, we will show the impact of AML in smart grid security systems. Some works like [14, 50, 51] have been done employing AML in the security of smart grid. [52] shows how to handle data poisoning by eliminating the outliers. The possibility of evasion attacks is more than poisoning attacks in the smart grid environment. This is because, the security models are not often retrained which is required for injecting the poisoning attacks.

To our knowledge, none of the existing works provide the analysis of the impacts of evasion attacks or the solutions to deal with such attacks in smart grid. Protection against evasion attacks will make the security model more robust.

3.3. ANOMALY DETECTION IN SMART TRANSPORTATION

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 [53, 54, 55]. 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 [56].

Traditional anomaly detection schemes are based on classification, statistical inference, state-based analysis, and clustering [57]. Classification based detection schemes usually rely on Support Vector Machines (SVM), Bayesian Models, Gaussian Processes or Neural Networks [58]. 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 [59] require realistic assumptions on the data distribution to estimate normal behavior which is a challenging task. In [60], 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 infrastructure management.

3.4. DATASETS AND DESCRIPTION

We have used two real AMI datasets to validate the proposed solutions. The first dataset is Ireland Social Sciences Data Archives [61] containing 5000 meters from six regions in the city of Dublin, Ireland, collected between 2009-2010. Three out of these six regions, have more than 1000 smart meters. The rationale for choosing this dataset is to investigate the scalability of our framework for large micro-grids. The second dataset is Pecan Street Project [62] containing hourly power consumption data from 215 houses from a Solar village in Texas, USA, collected between 2014-2016. Hence, we have chosen two datasets that are inherently different in terms of their geography, climate, randomness, and extreme difference in sizes.

For smart transportation, we have used dataset from Nashville, Tennessee which consists of vehicular data recorded in real-time over a period of 4 months (January to April 2019) with 1271 Traffic Message Channels (TMCs) [63]. The dataset contains the ground truth for accidents and congestion.

4. DETECTION OF DATA FALSIFICATION ATTACKS IN SMART GRID

The introduction of Smart Grid led to a lot of advantages like reduction in peak power production, demand pricing, real time billing. This also results in cyber attacks where the adversary will try to manipulate the data in a way to gain profit. It is proven that organized attacks are also possible on the smart grid. This section proposes a solution for detection such organised attacks in smart grid.

4.1. CONTRIBUTIONS

We have discussed different possible security threats for smart grid in the introduction. There are some existing security models to deal with such attacks. The existing security models are also discussed in the introduction. The existing works are not able to detect different possible attacks like additive, deductive and alternate switching using the same security model. The existing anomaly detection methods are not capable of filtering out the malicious meters in case of anomaly detection.

This work is the first effort to establish trustworthiness in AMI against multiple attack types and faults with coarse and fine grained attack strategies. Secondly, our focus is on orchestrated data falsification attacks devised by organized adversaries rather than just rogue customers. Our method works well for even higher fractions of compromised meters, unlike most statistics based methods due to the embedding of real time attack responses into the trust model. To demonstrate detection sensitivity in terms of margin of false data, we assume the full attack strategy space and show that detection rates are high across a wider threat landscape. Additionally, our method's time to detection of compromised meters is quick even under opportunistic attack strategies that are sporadic over time domain, via attack time probability ratio embedding. Our proposed method is light weight and gives better performance compared to the classical bad data detection mechanisms which use

expensive multi-class SVM and neural network based training models. We also discuss about the limitations of our proposed framework *under adversary's knowledge of our defense mechanism*, which motivates the direction in which further research should be conducted.

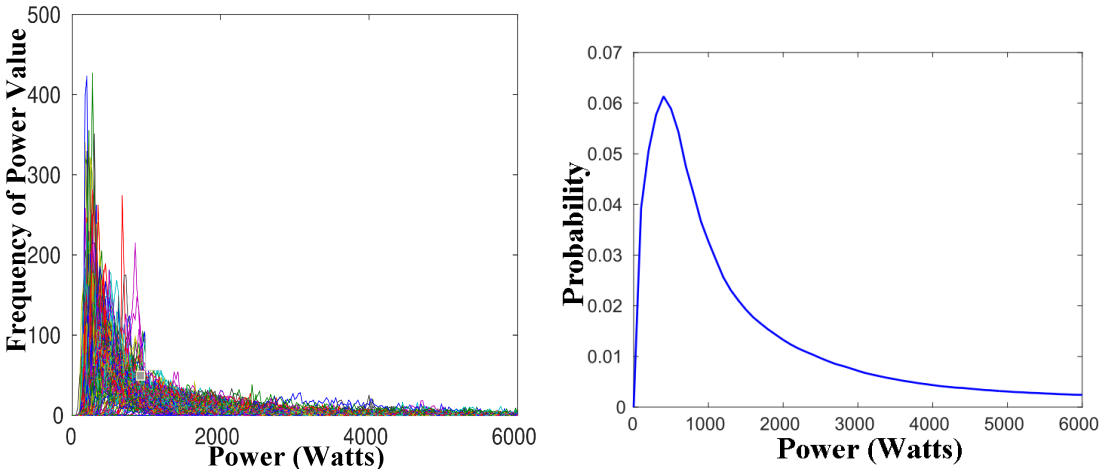
Section 2.2 describes the system and threat models while Section 2.3 discusses the proposed framework with theoretical analysis. Section 2.4 includes a special embedding method required to counter opportunistic attacks. Section 2.5 shows the experimental results for real-world data.

4.2. SYSTEM AND THREAT MODELS

In this section, we discuss the network architecture of the AMI, characterize the distribution of two real datasets, and propose the threat model for organized data falsification in AMI.

4.2.1. Architecture. We consider a collection of N smart meters reporting power consumption data to a Neighborhood Area Network (NaN) Gateway (acts as an edge computing node) periodically and independently. The i-th smart meter, records an actual power consumption data, say $P^i_{act}(t)$ at the end of each time slot t (t is slotted *hourly*). The reported power consumption $P^i_{rep}(t)$ is equal to $P^i_{act}(t)$, if i is not compromised. However, $P^i_{rep}(t) \neq P^i_{act}(t)$, if i is compromised by an adversary. We model $P^i_{act}(t)$ as the realizations of a random variable P^i, that denotes power consumption from the i-th meter. The NaN gateway piggybacks data from each smart meter and sends it to the utility via a Wide Area Network (WaN) Gateway that collects data from multiple such NaN gateways. Occasionally, there is another network hierarchy known as the Field Area Network (FAN) gateway which connects NaN and WaN and may host edge computing services. Both FaN and WaN may host the security monitoring mechanisms. Deploying security mechanism at the FaN is a decentralized implementation, while deployments at the WaN is a centralized implementation. Our framework works regardless of the implementation. The current

evaluation proposed mechanism assumes a decentralized implementation given the size of the microgrid datasets. Moreover, [64] has observed the benefits of decentralized security implementations over centralized ones.



(a) (b)

Figure 4.1. Power Consumption Distribution: (a) All Houses (b) Mixture

4.2.2. Data Set Characterization and Transformations. To characterize the distribution of the random variable P^i from the i-th smart meter, we conducted preliminary investigations on real power consumption data sets with 800 [62] (Texas Dataset) and 5000 meters [61] (Irish Dataset) collected on an hourly basis. The Texas dataset contains data across the years 2014, 2015 and 2016. Throughout the paper, data from 2014 and 2015 are used as the historical training set, while 2016 serves as a testing set. The Irish dataset contains approximately 535 days of data from years 2009-2010, that we use to prove the generality of our results.

Each home consists of one smart meter in the datasets. We observed that for each meter, the power consumption can be approximated as a log normal distribution. We also observed that all such log normal distributions are *clustered close* to each other; that is, the variance between them is not arbitrarily large. Figure 4.1(a) summarizes the results from all the houses in the Texas dataset. Thanks to this observation, we can approximate the

aggregate of the individual log normals using a mixture distribution, which is also lognormal as evident from Figure 4.1(b). Let P_{mix} denote the approximate lognormal mixture of all P^i.

Next we transform all P^i using a Box-Cox transformation technique to obtain an approximate *normally distributed* r.v. denoted as $\hat{P}^i$. Let $\hat{P}_{mix}$ denote the mixture of all the $\hat{P}^i$. Results of $\hat{P_{mix}}$, for different months is depicted in Figure 4.2(a). The box-cox transformation serves a dual purpose. First, it maps the data points to a lower portion real axis. Some interesting statistical properties of proposed Pythagorean Mean based invariants are more prominent in this lower-dimensional real axis which increases the relative sensitivity of Harmonic Mean to Arithmetic mean differences and their ratios (used for detecting anomaly) under false data injections. Below, we describe the box-cox transformation technique and how we apply it in our context.

4.2.2.1. Box-cox transformation. The transformation of non-normal data into approximate normal distribution can be achieved using the following method. Given any set of data-points $\boldsymbol{D} = \{D^{(1)}, \cdots, D^{(k)}, \cdots, D^{(n)}\}$, where n denotes the total number of data points in $\boldsymbol{D}$, the box cox transformation of $\boldsymbol{D}$ is given by $\hat{\boldsymbol{d}} = \{d^{(1)}(\lambda), \cdots, d^{(k)}(\lambda), \cdots, d^{(n)}(\lambda)\}$:

$$d^{(k)}(\lambda) = \begin{cases} \frac{(D^{(k)})^{\lambda}-1}{\lambda} & \text{if} \quad \lambda \neq 0; \\ ln(D^{(k)}) & \text{if} \quad \lambda = 0 \end{cases} \tag{4.1}$$

where λ is an appropriate transformation parameter chosen from a possible set $\lambda^* \subseteq \mathcal{R}$, such that

$$\lambda = _{\lambda \in \mathcal{R}} \; f(\boldsymbol{D}, \lambda^*)$$

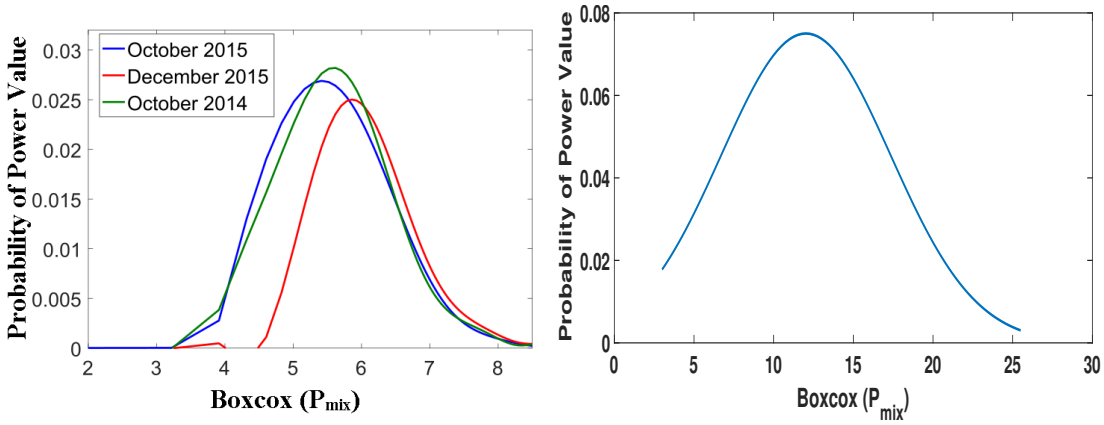
where $f(\boldsymbol{D}, \lambda^*)$ is the logarithm of the likelihood function such that $\bar{d}(\lambda) = \frac{\sum_{i=1}^{n} d^{(k)}(\lambda)}{n}$ is the arithmetic mean of the transformed data.

$$f(D, \lambda) = -\frac{n}{2} ln \left[\sum_{i=1}^{n} \frac{[d^{(k)}(\lambda) - \bar{d}(\lambda)]^2}{n} \right] + (\lambda - 1) \sum_{i=1}^{n} ln(d^{(k)}) \tag{4.2}$$

4.2.2.2. Applying transformation to the datasets. The data from each smart meter i (analogous to D) is transformed onto the box cox transformed scale by using the above procedure. Thereafter, we build the time series of the whole dataset in the box cox transformed scale as:

$$\hat{p}(t) = \{\hat{p}^1(t), \cdots, \hat{p}^i(t), \cdots, \hat{p}^N(t)\}$$

where $\hat{p}(t)$ denotes the reported time series over all smart meters $i \in \{1, N\}$ at each time slot. The appropriate λ is learned from the historical training set (2014, 2015), and the same is applied to the testing set (2016) and Irish Dataset (2010). To prove the generality of this method, we repeated the experiments for the Irish data set [12], and reported similar results which are included in the preliminary version of our work [33]. The distribution for Irish dataset after box-cox transformation After the transformation, 67% and 68% of data points fall within the first standard deviation for the Texas and Irish data sets respectively. However, the resultant distributions as a whole are not symmetric about the mean and 64% and 69% of the data are lesser than the mean and 36% and 31% of the data are greater than of the mean. This asymmetry is another factor that affects the observations in the anomaly detection phase.



(a) (b)

Figure 4.2. After BoxCox: (a) Monthly Texas (b) Yearly Irish

The proposed framework has three phases: (a) Anomaly-based Security Event Detection (b) Attack Context Embedded Trust Scoring Model. The anomaly detection phase indicates the nature of the security event in terms of the information such as the presence, type, strategy, and strength of the concerned data falsification attack. Such information extracted from the security event detection aids in the calculation of certain attack response metrics such as an unbiased robust mean, a median absolute deviation, and an attack probability time ratio by the attack context generation phase. Such attack response metrics are supplied to the trust scoring model phase that calculates a linearly separable score for each meter and uses it to identify the compromised meters launching data falsification attacks. The embedding of the attack context based response metrics improve the accuracy of compromised meter classification and the classification convergence times regardless of the attack types, margins, and strategies inflicted.

The anomaly detection phase is further divided into two parts: (i) coarse grained anomaly detection for attacks for all strategies except on-off and omission strategies; (ii) fine grained anomaly detection for on-off and omission strategies. Note that, the coarse and fine grained anomaly detectors run simultaneously in the framework since any attack strategy is possible in reality. While both anomaly detection variants help to calculate the robust mean and median absolute deviation, the attack probability time ratio is relevant only for the fine grained anomaly detector. The trust model is further divided into three parts: (a) estimating parameters of true proximity distributions (b) estimating parameters of observed proximity distribution with appropriate attack context embedding (c) The Kullback-Leibler Divergence calculation.

4.3. ANOMALY DETECTION MODEL

In this section, we propose the coarse grained and fine grained anomaly based security event detection scheme. The proposed event detection scheme leverage the properties of how different data falsification types change the Pythagorean Means (such as Harmonic,

Geometric, Arithmetic Means) of an attacked time series. We propose an invariant for both coarse and fine grained anomaly detection schemes, that is stable under no attacks. The evidence of invariant stability is proved through two real datasets gathered from 200 from a solar village in Texas, USA [62], and 5000 smart meters in Dublin, Ireland. Then, we show how these invariants exhibit visibly evident changes under various attacks, which forms the premise for inferring the presence of attack, type of data falsification, and the strategy used by the adversary that collectively reconstructs the security event. Based on the nature of the security event, an attack context is generated (in the form of robust mean, median absolute deviation, attack probability time ratio). The attack context information is forwarded to the trust based scoring model which enables accurate identification of the compromised smart meters.

First, we propose the detection metric (or invariant). Second, we explain the reasoning behind the design of the proposed invariant. Third, we establish the normal range of the under no attacks. Fourth, we propose the detection criterion to detect the occurrence of an orchestrated attack that needs a consensus (location and scale) correction. Fifth, we show how the attack type could be determined given the incidence of attack. Finally, we show how the knowledge of the incidence of attack and its corresponding type is used to estimate an approximate robust mean and median absolute deviation (collectively called as robust consensus measures). Information on the robust consensus measures is supplied to the entropy based trust model for improved classification that maximizes detection sensitivity for a wide range of δ_{avg} and ρ_{mal} values while minimizing the incidence of false alarms.

4.3.1. Pythagorean Means. The various Pythagorean means (arithmetic, geometric and harmonic means) in a particular time slot t is given by:

$$AM(t) = \frac{\sum_{i=1}^{N} \hat{p}^i(t)}{N} \quad , \qquad GM(t) = \Big(\prod_{i=1}^{N} \hat{p}^i(t) \Big)^{\frac{1}{N}} \quad , \qquad HM(t) = \frac{N}{\sum_{i=1}^{N} \frac{1}{\hat{p}^i(t)}}$$

The average of all these hourly means $AM(t)$, $GM(t)$ and $HM(t)$ over a particular day ($t \in [1, 24]$) is represented by $\overline{AM}(T)$, $\overline{GM}(T)$, and $\overline{HM}(T)$ respectively where $T \in [1, 365]$. For example, $\overline{AM(T)} = \sum_{t=1}^{24} AM(t)/24)$ and so on. Due to the well known Pythagorean mean inequality, $\overline{HM}(T) \leq \overline{GM}(T) \leq \overline{AM}(T)$ holds.

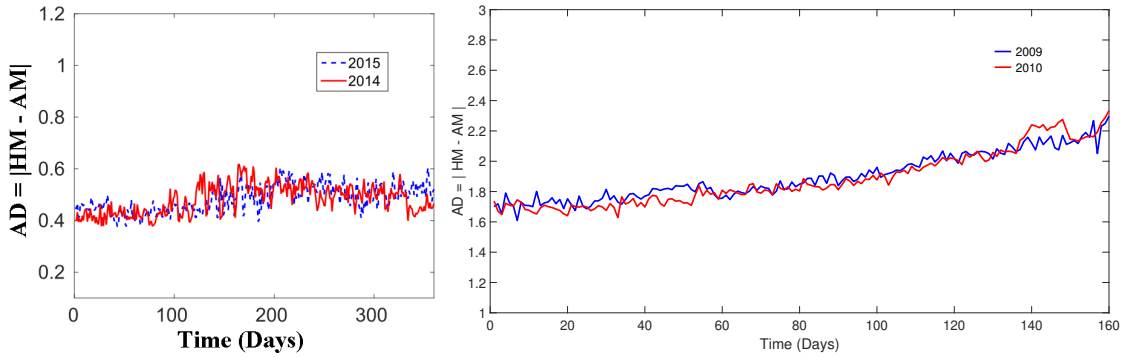
4.3.2. Proposed Coarse Grained Invariant ($AD(T)$). From our statistical studies over two big datasets, we discovered that the time series of the absolute difference between average daily harmonic and arithmetic mean power consumption is an effective invariant across datasets. Theoretical reasoning behind the stability of the harmonic mean and arithmetic combination has been extensively discussed and presented in our previous work [1]. Formally, the coarse grained invariant is quantified by $AD(T)$ and is defined as:

$$AD(T) = \left| \quad \overline{AM}(T) - \overline{HM}(T) \quad \right| \tag{4.3}$$

Eqn. 4.3, is designed as an anomaly detection metric for two main advantages: First, the time series of $AD(T)$ is a highly stable invariant of the aggregate power consumption, compared to other parametric and non-parametric measures that are functions of the instantaneous or historical arithmetic mean power consumption as proved in our previous work [1]. Furthermore, our previous work in the context of smart transportation systems [65, 66] showed that this observation of stationarity in harmonic and arithmetic mean generalizes across application domains under careful spatial and temporal considerations. High invariance over time or a given context is one of the desired properties of anomaly detectors [67].

Figure 4.3(a) shows the instantaneous values of $AD(T)$ for two different years (2014 and 2015). It can be verified that under no attacks, the average value of $AD(T)$ is about 0.49 and the values are relatively stable over time across both years. Similarly, Figure 4.3(b), shows the time series of $AD(T)$ for the portion of the Irish dataset that has a historical overlap between two years 2009-2010. The $AD(T)$ of the Irish dataset is stable over history, since AD(T) on the T-th day is one year is not arbitrarily different from the AD(T) of

the corresponding T-th day in the previous year. Both Figures 4.3(a) and 4.3(b) is in complete contrast to the Figure 4.4 that shows the average arithmetic mean $\overline{AM}(T)$ for the Texas dataset, can be seen as neither stable over time or over history. As it is well known that anomaly detection metrics ideally need high invariance under normal operations, we, therefore, conclude that $AD(T)$ a better invariant compared to any derivative of the popular arithmetic mean and standard deviation. Additionally, since the values are not arbitrarily different, the variance in the $AD(T)$ samples is also lesser compared to the variance in $\overline{AM}(T)$ samples.



(a) (b)

Figure 4.3. Time Series of proposed $AD(T)$: (a) Texas Dataset (b) Irish Dataset

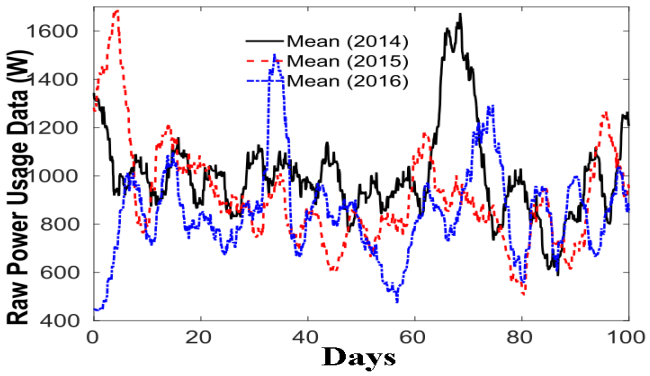
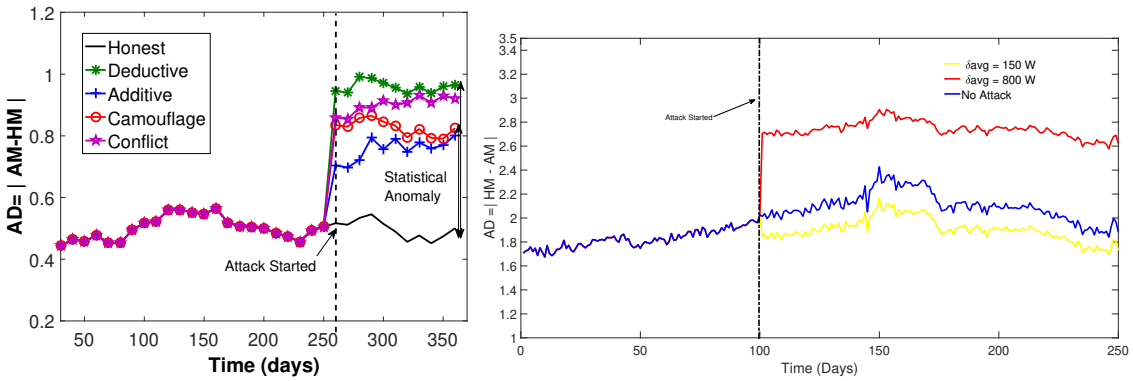


Figure 4.4. Unstable $AM(T)$ for Texas Dataset

4.3.3. Summary of Security Properties of Proposed $AD(T)$. The second advantage is that harmonic, geometric and arithmetic mean possesses certain special mathematical properties that produce unique changes in the time series of $AD(T)$, whenever data falsification occurs from a subset of data sources, that otherwise produced a stationary $AD(T)$.



(a) (b)

Figure 4.5. $AD(T)$ deviation under attacks (a) Texas Dataset (b) Irish Dataset

While harmonic mean and geometric mean is a strictly Schur-Concave function [68], the arithmetic mean is both Schur Concave and a Schur Convex function of its arguments (the numbers involved in the calculation of the means). Such a difference in the strictness of Schur-Concavity property produces six unique novel properties in the context of data falsification that we had identified. The direction of deviation depends on the skewness in the datasets, but the fact there will be deviation is generic and independent of the datasets. These six properties are divided into two-sub groups based on the direction of change in the $AD(T)$. The direction of change in $AD(T)$ is dependent on whether the δ_{avg} is greater or lesser than a certain threshold Γ, given a particular ρ_{mal}, attack type, and the skewness in the data distribution. The theoretical and experimental proof of the properties has been established in our earlier work [1]. For the sake of completeness, we now provide a summary of these properties in harmonic and arithmetic means, that cause the deviation in $AD(T)$ under attacks.

A) Case 1: For all attacks with $\delta_{avg} > \Gamma$, the following hold true:

Property 1: Under additive attacks, the harmonic mean grows slower compared to the arithmetic mean, thus $AD(T)$ will increase.

Property 2: Under deductive attacks, the harmonic mean decays faster compared to the arithmetic mean decay rate, thus $AD(T)$ will increase.

Property 3: Given the same δ_{avg} and the same set of arguments, the decay in harmonic mean is larger for deductive attacks compared to growth in harmonic mean for additive attacks. Therefore, in a camouflage attack with the same δ_{avg}, the resultant harmonic mean will be lesser than the original harmonic mean, while the arithmetic mean will not change. Thus $AD(T)$ will increase.

It is easy to conclude that all the above three properties will cause the $AD(T)$ *to increase after attacks* compared to before attacks because the gap between $\overline{HM}$ and $\overline{AM}$ widens, so does its absolute value represented by $AD(T)$. This is experimentally verified in Figure 4.5(a), where attack injected after the 250-th day shows a sharp increase in the $AD(T)$ for various attack types.

B) Case 2: For all attacks with $\delta_{avg} < \Gamma$, the above three properties are reversed and the following hold true:

Property 4: Additive attacks will show larger growth in harmonic mean compared to arithmetic mean growth, thus $AD(T)$ will experience a decrease.

Property 5: Deductive attacks will show smaller decay compared to arithmetic mean decay, thus $AD(T)$ will experience a decrease.

Property 6: $AD(T)$ will decrease if actual number of data points attacked with additive are smaller than the actual mean. This is typically true for power consumption datasets that are right skewed, hence the mean is shifted towards the right tail of the distribution. If such data is attacked, on average more number of datapoints being modified will be smaller than the actual arithmetic mean.

It is easy to conclude that properties 4-6 will cause the $AD(T)$ *to decrease after attacks* compared to before attacks because the gap between $\overline{HM}$ and $\overline{AM}$ narrows, so does its absolute value represented by $AD(T)$. This is experimentally verified in Figure 4.5(b), where attack injected after the 100-th day shows a decrease in the $AD(T)$.

Approximation of crossover Γ: For directional switching of the proposed $AD(T)$, the approximate bounds on the average value of Γ and its details have been published in our earlier work [1]. The closed form of Γ is not possible due to non-existence of the closed form. However, approximation of the lower and upper bounds are given by the following The approximate (average case) lower bounds are: $\Gamma^{-}(rlow) = \Gamma^{+}(llow) =$

$$\Gamma_{low} = \frac{\sigma}{M} + \frac{\sigma}{\sqrt{M}}\sqrt{\frac{N-M}{N-1}} + \sigma \tag{4.4}$$

where + and − superscripts denote additive and deductive manipulation and l and r denote whether the bias points are on the left or right of the actual mean. The approximate upper bounds are: $\Gamma^{+}(lhigh) = \Gamma^{-}(rhigh) =$

$$\Gamma_{high} = max(\sigma^2, \frac{2\sigma}{M} + \frac{\sigma}{\sqrt{M}}\sqrt{\frac{N-M}{N-1}} + 2\sigma) \tag{4.5}$$

4.3.4. Identifying Normal Range of $AD(T)$. Let the standard deviation of the $AD(T)$ samples in the training set be denoted as $\sigma_{AD(T)}$. Given that the $AD(T)$ metric is stable over history as evident from earlier results, the normal range can be a residual margin around the historical values. The margins can be a parameterized by a scalar factor $\gamma \in (0, 3]$ of the standard deviation of the $AD(T)$ samples, such that the upper threshold for $AD(T)$ at the T-th window in the testing set i given by:

$$AD^{test}_{max}(T) = AD_{hist}(T) + \gamma\sigma_{AD(T)}$$

and the corresponding lower threshold is:

$$AD^{test}_{min}(T) = AD_{hist}(T) - \gamma\sigma_{AD(T)}$$

Please note that, it is possible that smaller δ_{avg} (stealthy) or smaller ρ_{mal} values (isolated or small scale adversaries) will not create enough deviation for the $AD(T)$ to fall outside the $AD^{norm} \in [AD^{test}_{min}, AD^{test}_{max}]]$ range. However, such smaller attacks will also not drastically affect the consensus measures (mean and standard deviation). As mentioned earlier, the one

of the purpose of the anomaly detection phase in our framework is to provide an unbiased instantaneous mean and median absolute deviation to the trust model across either a high ρ_{mal} or δ_{avg} values. Therefore, successful detection of incidence and type of attack, is only required when attacks are strong enough to influence the consensus significantly. To this end, the simple definition of $AD^{norm} \in [AD^{test}_{min}, AD^{test}_{max}]]$ is sufficient. If attacks are not detected, however, at the same time they do not affect the consensus in a significant way. In such cases, the trust scoring model proposed later will be still successful in detecting the compromised meters regardless.

4.3.5. Coarse Grained Detection for Organized Data Falsification. From Figure 4.5(a), it is easy to conclude that for all attack types, the AD^{obs} is larger than the AD^{norm} learned from the training phase. The AD^{norm} act as a safe margin for the invariant, and anything outside of it is inferred as an orchestrated attack that needs a location and scale correction as a response. As long as the attack continues the $AD(T)$ remains higher than the normal values.

$$AD^{obs}(T) : \begin{cases} \in AD^{norm} & \text{No Organized Falsification ;} \\ > AD^{norm} & \text{Organized Falsification Occurred;} \end{cases} \tag{4.6}$$

4.3.6. Determining the Type of Data Falsification Attack. From the above, we conclude that an authentic change in the observed distribution may cause the mean consumption to increase or decrease but AD^{obs} remains the same as compared to the historical range of values $AD^{norm} = [AD^{min}, AD^{max}]$. An additive attack causes both HM and AM of consumption to increase but also causes AD^{obs} to increase compared to its normal range. This way a legitimate versus a malicious increase in the data can be distinguished. A deductive attack causes the HM and AM of mean consumption to decrease and causes AD^{obs} to increase from the historical range. Similarly, camouflage and conflict attacks do not have much change in the AM of the consumption but triggers a large increase in the $AD^{obs}(T)$.

In this way, it is possible to infer which type of data falsification has been launched. A summary of the above discussion to determine the presence and type of attack is given in Table 4.1.

Table 4.1. Concluding Security Events

AD	AM	HM	GM	Conclusion
Increased	Increased	Increased	Increased	Additive
Increased	Decreased	Decreased	Decreased	Deductive
Increased	Same	Decreased	Decreased	Camouflage
Decreased	Increased	Increased	Increased	Additive Low
Increased	Any	Any	Any	Conflict
Same	Don't Care	Don't Care	Don't Care	No Attack

4.4. ATTACK CONTEXT RESPONSE METRICS

Given that an attack has been inferred that bias the instantaneous (hourly) consensus measures, we need a consensus correction scheme. The knowledge of the attack type could be leveraged to *unbias* the consensus measures. This is because, the manner and extent to which different instantaneous means such as, $HM(t)$, $GM(t)$ and $AM(t)$ and corresponding standard deviations get biased by different attack types, is unique (from Property 1,2,3 and their corollary). Alternatively, one may be tempted to use the historical values of mean and standard deviation on the corresponding hours of the T-th day in the previous years. However, as already shown in Figure 4.4, the mean values on the same days on successive years vary greatly and hence historical values are not reliable. Therefore, it is required that for a successful statistical detection, a robust mean (location consensus) and a robust measure of dispersion is calculated.

4.4.1. Estimation of Robust Mean as a Response. For the calculation of robust mean, we need to reconstruct the actual mean from the observed mean using knowledge of how each attack type changes these means. Additionally, the extent of change triggered in the $AD(T)$ metric also depends on δ_{avg} and/or ρ_{mal}. Hence, an adjusted robust mean helps to estimate an approximate value closer to the original mean. Note that, the highest possible δ_{avg} is lesser in deductive attacks than additive ones because the feasible margin of deductive false data is bounded by zero. As the margins of false data or compromised

fraction increases, the observed consensus gets more and more biased. To prevent this, we ought to have a consensus correction step. Otherwise statistics based trust models will not be able to identify the compromised meters.

From the statistical observations, we see that the $HM(t)$ is more proximate to the actual AM than the observed $AM(t)$, under the effect of additive attacks, due to a slower increase in HM as opposed to AM. However, the $HM(t)$ itself is not a robust mean consensus, if either δ_{avg} or ρ_{mal} is large. Therefore, we propose to use $\mu_R(t) = HM(t) - AD(t)$ as the estimated robust mean aggregate under additive attacks, which is closer to the original instantaneous arithmetic mean. Therefore, we deduct the $AD(t)$ since it is the index of the extra deviation caused by the attacks. As an example, in Table 4.3, for additive attack $HM - AD = 7.92 - 0.76 = 7.16$, which is very close the actual AM value of 7.053.

In contrast, for deductive attacks, due to $HM \leq GM \leq AM$ property, the $HM(t)$ is even lesser than the already biased $AM(t)$. But, $GM(t) + AD(t)$ is more robust than AM for deductive attacks, and results show that it is a good approximation to the actual mean. From the example in Table 4.3, it can be verified that for deductive attack, the robust mean $\mu_R = 6.29 + 0.79 = 7.08$ is closer to the actual mean 7.05. For camouflage attacks, AM is the most robust and hence μ_R is set as the AM. For conflict attacks, GM is an intermediate robust choice as it shows relative stability to both partially positive and negative outliers. The recommended mean correction for each attack type is tabulated in Table 4.2.

Table 4.2. Robust Mean Responses

Security Incident	**Choice of $\mu_R(t)$**
Additive	HM-AD
Deductive	GM+AD
Camouflage	AM
Conflict	GM
No Attack	AM

Table 4.3. Attacks on Various Means

Parameter	**Actual**	**Add**	**Deduct**	**Camo**	**Conf**
AM	7.053	8.68	6.67	7.04	7.26
GM	6.860	8.35	6.29	6.65	6.89
HM	6.680	7.92	5.88	6.02	6.11
AD	0.373	0.76	0.79	1.02	1.15

4.4.2. Estimating a Median Absolute Deviation as a Response. If the presence of an attack is discovered from the anomaly detector, then we know that the instantaneous standard deviation of the observed data is biased. The $\sigma(t)$ in the testing set under attacks

will increase regardless of the type of data falsification attack (except for low additive attacks). Therefore, a directional correction of the standard deviation is not possible like $\mu_R(t)$ based on the attack types. While standard deviation is a very popular measure of dispersion (scale parameter) to build proximity distributions, we argue the use of a less common statistical measure of dispersion known as *'Median Absolute Deviation' (MAD)*, which is defined as follows:

For power consumption data at any time t, $\hat{p}(t) = \{p^1(t), \cdots, p^i(t), \cdots, p^N(t)\}$, the data's median is defined as $\tilde{p}(t) = Median(\hat{p}(t))$. The median absolute deviation is defined as: $MAD(t) = Median(|p^i(t) - \tilde{p}(t)|)$.

The MAD is a much more robust measure of dispersion (or more robust scale parameter) compared to the traditional standard deviation because MAD is more robust and remains less affected due to outliers (reducing false alarms under no attacks) and extreme values (under stronger margin attacks) compared to standard deviation. This is because measures such as standard deviation are derived from variance which uses squares of the difference between those outlying datapoints and the true mean. Squares produce very high values when datapoints are greater than 1, thus causing an unwarranted increase in the standard deviation. This is the cause of increased missed detection under attacks and increased false alarms under no attacks. Therefore, we depart from the traditional use standard deviation for characterizing the probability distribution of the proximity of individual smart meters data with the consensus.

The measured $MAD(t)$ of the historical time slots, before the inference of orchestrated attack is therefore embedded as the robust measure of dispersion or the robust scale parameter in the trust model in the event of an attack indication from the anomaly detector. As shown later, the mean correction, robust scale parameter as median absolute deviation and attack probability time ratio embedding facilitates quick detection, this approximation works well.

Both robust mean and median absolute deviation bias correction improves results significantly compared to the preliminary version of this work in [33]. The failure points for higher δ_{avg} values completely disappear. While, the above adjustment of mean location parameter and median absolute deviation may not always be perfectly close to the actual mean and median deviation, our results show that classification performance is much better under these approximate bias corrections rather than just using the exact harmonic mean and standard deviation as the location and scale parameters as done in our preliminary work [33].

4.5. TRUST SCORING MODEL

We pursue a light weight learning approach for identifying compromised smart meters that launch data falsification. The prior historical data set is considered as the authentic distribution of power consumption. From the historical data set, a *true proximity distribution* denoted as $\boldsymbol{X}^i$ for each smart meter is generated based on its reported consumption's proximity to the arithmetic mean of the authentic data set. Since the authentic historical data set is attack-free, the measure of consensus is arithmetic mean (AM), denoted by $\mu(t)$ and the standard deviation is $\sigma(t)$.

In the observed data set under test, we define $\mu_R(t)$ and $MAD_R(t)$ as the robust mean and median absolute deviation of the observed distribution based on the inferred security incident. In the testing set, a *current proximity distribution*, denoted by $\boldsymbol{Y}^i$, for each smart meter i is calculated based on the proximity of its reported consumption data $p^i_{rep}(t)$ to $\mu_R(t)$. In the absence of a detected security incident, the robust mean and median absolute deviation equals $\mu(t)$ and $\sigma(t)$ (like in the historical set). However, when an attack is present, the $\mu_R(t)$ is set according to model based on the inflicted attack type and strategy. This way the attack context is embedded via the appropriate robust mean as a response to the detected attack context. Similarly, the $MAD_R(t)$ is set to the historical median absolute deviation if there is an indication of an attack.

If the true distribution is very different from the current distribution, it is an indication that this meter's data is *unusually* different and this difference in the probability space is measured as *Kullback-Leibler divergence* (also called KL Distance) which measures the *relative entropy* between the two distributions. The higher the divergence between the two distributions, the more the indication of anomalous behavior. The trust of a meter is calculated at the end of the frame F (in days). The total number of observations over the time frame is given by TS. For the relative entropy trust model, we had time frames of length $F = 10$ days and $F = 30$ days. Therefore, the number of time slots monitored in the frame of observation is $TS = F * 24$.

4.5.1. True and Current Proximity Distributions as Meter Evidence. We introduce a binary random variable $X^i = \{0, 1\}$ for each meter i, for $i = 1, \ldots, N$, which acts as a historical reference distribution. If the historical data reported $\hat{p}^i_{rep}(t)$ at time t from meter i falls within one standard deviation of $\mu(t)$, then $X^i = 1$, else 0. Formally,

$$X^i(t) = \begin{cases} 1 & \text{if } \hat{p}^i_{rep}(t) \in \{\mu(t) \pm \sigma(t)\}; \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

where $X^i(t)$ follows a Bernoulli distribution with parameter r, that is the probability of $X^i = 1$ is r, and the probability of $X^i = 0$ is $1 - r$.

Suppose, $S(X)$ be the variable that denotes the number of successes, that is $S(X^i) = \sum_{t=1}^{TS} X^i(t)$. Let $S(X) = k$ be the observed value of the variable for any meter i, such that number of success in the true distribution is $S(X^i) = \sum_{t=1}^{TS} X^i(t) = k$.

Similarly, we have a binary random variable Y^i for the current distribution of each smart meter, such that the probability of $Y = 1$ is q and the probability of $Y = 0$ is $1 - q$. In this case, the number of successes is denoted by a variable $R(Y^i) = \sum_{t=1}^{TS} Y^i(t)$. Let $R(Y) = j$ denote the number of successes for any such meter i such that number of successes in the current distribution is $R(Y^i) = \sum_{t=1}^{TS} Y^i(t) = j$. If an attack has been detected through the anomaly detection phase, then the robust mean $\mu_R(t)$ and the robust standard deviation

$\sigma_R(t)$ is calculated, and the Y^i is calculated based on them. In this way attack context is embedded such that Y^i remains unbiased from the effects of orchestrated attacks. However, in the absence of any detected attacks, $\mu_R(t) = \mu(t)$. Formally, the current proximity distribution is given by:

$$Y^i(t) = \begin{cases} 1 & \text{if } p^i_{rep}(t) \in \{\mu_R(t) \pm MAD_R(t)\}; \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Intuitively, in absence of attacks, the distribution of Y should be very close to X. On the contrary, the two distributions should show a difference when an attack is present.

4.5.2. Estimating Parameters of True and Current Proximity Distributions. Next, we need to estimate the parameters r and q for corresponding distributions X^i and Y^i. An obvious estimate is the minimum variance unbiased estimate (frequentist), which is the sum of all successes divided by the total number of observations TS. However, this approach may cause $r = 0, q = 0$, or $r = 1, q = 1$, for which the relative entropy (see Eqn. 4.15) is undefined. Moreover, frequentist probability unbiased estimator makes sense only if there is a large set of observations. However, since our trust model works on a shorter horizon of time (typically on a few days or monthly basis), such approaches are improper. Hence, we need to accommodate a Bayesian approach for estimation of r and q, so it is theoretically sound and mathematically tractable. Since the following is true for all meter's i, we drop the suffix i from the notational simplicity.

First, we estimate the parameter of r. We prove that the estimated probability $r = \frac{k+1}{TS+2}$, where k is the realization of the total number of successes observed. Thus $S(X) = k$ follows a binomial distribution with parameter r.

Hence, the probability of observing exactly k successes out TS times, given the probability of success of each trial was r, is given by,

$$P(S(X) = k|r) = \binom{TS}{k} r^k (1-r)^{TS-k} \tag{4.9}$$

The Bayesian posterior estimate of r, based on prior TS observations by Bayes theorem, is given as:

$$P(X(TS+1) = 1|S(X) = k) = \frac{P(X(TS+1)), S(X) = k)}{P(S(X) = k)} \tag{4.10}$$

The denominator is the marginal probability of $P(S(X) = k)$ marginalized over all possible outcomes of r. Hence,

$$P(S(X)) = \int_0^1 \binom{TS}{k} r^k (1-r)^{TS-k} f(r) dr \tag{4.11}$$

Assuming conditional independence between $S(X)$, r and $X_i(t+1)$ of the prior and likelihood can be solved as:

$$\begin{aligned} P(X(TS+1)), S(X) = k) \implies \\ = \int_0^1 P(X(TS+1) = 1|r) P(S(X) = k|r) dr \end{aligned} \tag{4.12}$$

Since there is no prior information on r, we assume a non-informative prior such that $f(r) = 1$, for Eqn. (4.11) and Eqn. (4.12). Plugging in Eqn. (4.11) and Eqn. (4.12) into Eqn. (4.10), it can be shown that:

$$P(X(TS+1) = 1|S(X) = k) = \frac{k+1}{TS+2} = r \tag{4.13}$$

Similarly,

$$q = \frac{j+1}{TS+2} \tag{4.14}$$

It can be verified that $r, q \neq 0, 1$. Hence, the logarithms of distributions X^i and Y^i for the i-th smart meter, (described in terms of probability parameters $r^{(i)} = \frac{k^{(i)}+1}{TS+2}$ and $q^{(i)} = \frac{j^{(i)}+1}{TS+2}$), in Eqn (4.15) is always defined and exist even as $k^{(i)} = 0$ or $j^{(i)} = 0$.

4.5.3. Kullback-Leibler Divergence based Scoring and Classification. We adopt the *Kullback Leibler divergence* to measure the difference between the historical distribution X^i and the observed distribution Y^i for a smart meter. Note that X^i and Y^i are not consumption patterns but a trend on proximity to the middle quartile. Subsequently, the KL distance is transformed into a trust value between 0 and 1 by passing it through an inverse square root function that produces linearly separable trust values between compromised and honest meters via a single threshold.

The KL distance between two distributions X and Y for a smart meter i, is given by:

$$D_i(X^i||Y^i) = (1 - r^{(i)}) \times ln\Big(\frac{1 - r^{(i)}}{1 - q^{(i)}}\Big) + r^{(i)} \times ln\Big(\frac{r^{(i)}}{q^{(i)}}\Big) \tag{4.15}$$

The $D_i(X^i||Y^i)$ is a positive real value that indicates the divergence between the observed and the historical proximity distribution. Hence, the smaller the value of $D_i(X^i||Y^i)$ the better it is in terms of being trustworthy and the larger it becomes the less trustworthy it becomes since a larger divergence indicates a mismatch between the true and observed proximity distributions. Given this, the final trust value Q^i of a smart meter i, is given by:

$$Q^i = \frac{1}{1 + \sqrt{D_i(X^i||Y^i)}} \qquad 0 \leq Q^i \leq 1 \tag{4.16}$$

The rationale of Eqn. 4.16, is a scaling function that scales the lowest value in $D_i(X^i||Y^i)$ a trust score that is closest to 1 while the highest value in $D_i(X^i||Y^i)$ gets the exponentially lower trust score with increasing $D_i(X^i||Y^i)$. The exponential nature ensures a risk aversion towards progressively increasing distance in the probability space.

4.5.4. Limitation of Coarse Grained Anomaly Detection based Trust Model. Since the coarse grained anomaly detection has an observation granularity of 24 hours, it is not suitable for detection of opportunistic omission and on-off strategies that are discon-

tinuous and sparsely distributed over the time domain. Therefore, an anomaly monitoring metric with a daily time granularity such as $AD(T)$ will not be sensitive and fail to provide the early indication of the attack's presence that is necessary to embed in the attack context.

Apart from failing to identify the incidence, type, and robust consensus, there will be another hurdle for the subsequent pipelined trust model. Since in most of the time slots, there are no attacks from the meters, the evidence against each meter will have reduced sensitivity when observed over a time frame. This is because the probabilities (modeled by evidence) in information theoretic measures (such as KL Divergence) are steady state long term measures. When observed over the time frame, the detection of meters will be delayed due to a lesser change in evidence counts. However, if the trust model is made aware of the *incidence of such non-continuous strategies* and the *approximate start and stop times of such attacks*, the evidence against meters collected on those specific slots may be weighted as more important (while others as less important). This would facilitate quicker classification of such meters while running the trust scoring model through information theoretic measures. This is achieved by calculating the fraction of the time frame that a meter was under such attack strategies (*defined later as attack probability time ratio*). This motivates the need for a fine grained anomaly detection phase that runs in parallel with coarse grained anomaly detection metric and associated attack context embedding.

4.6. FORMAL SECURITY ANALYSIS

We do the theoretical analysis in terms of attack parameters to formally specify the impact of attacks on the effectiveness of the defense method. Specifically, we assess the security level of our mechanism by taking into account what an intelligent adversary might do to bypass the invariant based anomaly detection and the compromised meter detection trust model. Here, we also show closed form theoretical expressions of our observations that will help generalize our framework.

4.6.1. Theoretical Analysis of Deviation in $AD(T)$ Under Attacks. As a part of the theoretical security analysis of the anomaly detection phase, we provide the closed form approximate estimated deviation in the anomaly detection metric $AD(T)$. This can be estimated by calculating the expected harmonic mean and arithmetic mean, given an attack type, ρ_{mal} and δ_{avg}. Below we provide an estimation of the harmonic mean followed by the arithmetic mean. Finally, we show how closely the theoretical result from the closed form expression matches the experimental result to prove accuracy of analysis. We also show that change in $AD(T)$ observed experimentally also matches the theoretical analysis. Because our detector uses the values in a box-cox transformation domain, we have carefully estimated it for real data values and found their box cox equivalents on the transformed scale.

$$Nor(AM^{ba}(t)) = \frac{\sum_{i=1}^{N} P^{i}_{rep}(t)}{N}; \quad Nor(AM^{ba}(T) = \frac{\sum_{t=1}^{24} Nor(AM_{ba}(t))}{24} \tag{4.17}$$

Similarly, $Nor(HM_{ba}(T))$ and $Nor(GM_{ba}(T))$ can be calculated. For brevity, we drop the T from the following analysis. Since the closed form expression of the harmonic mean does not exist, we first estimate the new geometric mean $Nor(GM^{esaa})$ after the attack. Then, we harness the following Pythagorean equation that calculates the estimated harmonic mean from the estimated geometric and arithmetic means:

$$Nor(HM^{esaa}) \approx \frac{(Nor(GM^{esaa}))^2}{Nor(AM^{esaa})} \tag{4.18}$$

where $Nor(HM^{esaa})$ and $Nor(AM^{esaa})$ denote the estimated HM and AM values after an attack.

1) Estimation of the Geometric Mean after attack: Let $Nor(GM^{ba})$ denote the geometric mean of a power consumption data before attack in the original data domain and is defined by:

$$Nor(GM^{ba}) = \left(\prod_{i=1}^{N} P^{i}_{rep} \right)^{\frac{1}{N}} = \sqrt[N]{(P^1 \times P^2 \cdots P^M \times P^{M+1} \cdots \times P^N)} \tag{4.19}$$

Similarly, let the estimated geometric mean after additive attack from $\rho_{mal} = M/N$ meter and δ_{avg} in the original data domain be denoted as $Nor(GM^{esaa})$ such that:

$$Nor(GM^{esaa}) = \sqrt[N]{(P^1 + \delta_{avg}) \times (P^2 + \delta_{avg}) \cdots \times (P^M + \delta_{avg}) \times (P^{M+1}) \times \cdots \times (P^N)} \tag{4.20}$$

Now we need to convert each $P^i + \delta_{avg}$ term into a multiplier of P^i. Let the ratio between the δ_{avg} and the actual data from the $i-th$ meter before attack P^i be given by a dummy variable:

$$\alpha^i = \frac{\delta_{avg}}{P^i} \tag{4.21}$$

Since P^i is a completely random physical quantity, we will need to characterize the α variable as a property that is shared across datapoints under an attack.

From the studies, we know that for the power consumption distribution, most of the data points are within the first standard deviations from the mean (say $\overline{P}$). For the Irish and Texas dataset, more percentage of datapoints (70%) are lesser than the mean $\overline{P}$ compared to percentage of data points greater than the mean (30%) on average. While the 30% values are lesser and greater than the mean cancel the effect of each other on the estimation of P^i, the remaining fraction of samples represents an imbalance factor (say $\nabla = 0.40$).

$$\alpha = \frac{\delta_{avg}}{\overline{P} - \nabla\sigma}$$

$$Nor(GM^{esaa}) \approx \sqrt[N]{(P^1 + \alpha.P^1) \times (P^2 + \alpha.P^2) \cdots \times (P^M + \alpha.P^M) \times (P^{M+1}) \times \cdots \times (P^N)}$$

$$Nor(GM^{esaa}) \approx \sqrt[N]{(1+\alpha)P^1 \times (1+\alpha)P^2 \cdots \times (1+\alpha)P^M \times (P^{M+1}) \times \cdots \times (P^N)}$$

$$Nor(GM^{esaa}) \approx \sqrt[N]{(1+\alpha)^M \times P^1 \times P^2 \cdots \times P^M \times (P^{M+1}) \times \cdots \times (P^N)}$$

$$Nor(GM^{esaa}) \approx \sqrt[N]{(1+\alpha)^{\rho_{mal}*N} \times P^1 \times P^2 \cdots \times P^{\rho_{mal}*n} \times (P^{\rho_{mal}*n+1}) \times \cdots \times (P^N)}$$

$$Nor(GM^{esaa}) \approx \sqrt[N]{(1+\alpha)^{\rho_{mal}*N} \times P^1 \times P^2 \cdots \times P^{\rho_{mal}*n} \times (P^{\rho_{mal}*N+1}) \times \cdots \times (P^N)}$$

$$Nor(GM^{esaa}) \approx (1+\alpha)^{\rho_{mal}} \sqrt[N]{P^1 \times P^2 \cdots \times P^M \times (P^{M+1}) \times \cdots \times (P^N)}$$

From Eqn. 4.19, the current result of $Nor(GM^{esaa})$ from above can be reduced to the following:

$$Nor(GM^{esaa}) \approx (1+\alpha)^{\rho_{mal}} Nor(GM^{ba})$$

$$Nor(GM^{esaa}) \approx \left(1 + \frac{\delta_{avg}}{\overline{P} - \nabla\sigma}\right)^{\rho_{mal}} Nor(GM^{ba}) \tag{4.22}$$

Plugging in the real values of $\sigma, \nabla, \rho_{mal}, \delta_{avg}$ and $Nor(GM^{ba})$, we obtain the estimated theoretical geometric mean after the attack as $Nor(GM^{esaa}) = 410$, while the actual measured geometric mean after the attack was recorded as $Nor(GM^{expaa}) = 390$. This indicates that this is a reasonably close approximation. Now the next step is to calculate $Nor(AM^{esaa})$ to plug it in Eqn. 4.18 for estimation of the new harmonic mean $Nor(HM^{esaa})$.

2) Estimation of AM after attack: Let the $Nor(AM^{esaa})$ denote the arithmetic mean attack after attack. For the following estimation, assume the attack to be additive. Similarly, this method could be used to estimate other attack types. Given the $\rho_{mal} = M/N$ is the fraction of compromised meters and δ_{avg} is the average falsification margin per meter, then the estimated attacked arithmetic mean is under additive attack is:

$$Nor(AM^{esaa}) = Nor(AM^{ba}) + \left(\rho_{mal} * \delta_{avg}\right) \tag{4.23}$$

3) Estimation of HM after attack: The Eqn. 4.22 and Eqn. 6.4 can be plugged in the following:

$$Nor(HM^{esaa}) \approx \frac{(Nor(GM^{esaa}))^2}{Nor(AM^{esaa})} \tag{4.24}$$

4) Estimation of Box-Cox Equivalents of Means: Let the $Box(Nor(AM^{ba}(T), \lambda)$ denote the box cox equivalent of the mean before attack in the normal scale such that:

$$Box(Nor(AM^{ba}(T), \lambda) = \frac{(Nor(AM^{ba}(T))^{\lambda} - 1}{\lambda} \tag{4.25}$$

Similarly, $Box(Nor(HM^{ba}(T), \lambda)$, and $Box(Nor(GM^{ba}(T), \lambda)$ are corresponding box-cox equivalent values of harmonic and arithmetic means before the attack. Similarly, the box-cox equivalent values of them after attack $Box(Nor(AM^{esaa}), \lambda)$, $Box(Nor(GM^{esaa}), \lambda)$, $Box(Nor(HM^{esaa}), \lambda)$, can be easily estimated.

5) Final Estimation of $AD(T)$ after attack: Note that the box-cox equivalent of the arithmetic mean gives a slightly different answer compared to the arithmetic mean of data in a power transformation scale (the experimental result). Let the difference be $\kappa = |Box(Nor(AM^{esaa}), \lambda) - Box(Nor(AM^{ba}), \lambda)|$. The estimated arithmetic, geometric, and harmonic means calculated over box-cox transformed arguments (what our method actually implements), after the additive attack is given by the following:

$$\overline{AM}^{esaa} = \overline{AM}^{ba} + \kappa; \qquad \overline{GM}^{esaa} = Box(Nor(GM^{esaa}); \qquad \overline{HM}^{esaa} = Box(Nor(HM^{esaa}) \tag{4.26}$$

For the estimation of arithmetic mean, the estimation of change (κ) will result in a closer approximation compared to direct box-cox calculation for a given ρ_{mal} and δ_{avg}. Let be the value of the $AD(T)$ metric after the attack be $AD^{esaa}(T) = |\overline{HM}^{esaa} - \overline{AM}^{esaa}|$. Thus, The expected deviation in the $AD(T)$ metric after an attack of ρ_{mal} and δ_{avg} for additive attacks is given by:

$$E(\Delta AD(T)) = |\overline{HM}^{ba} - \overline{AM}^{ba}| - |\overline{HM}^{esaa} - \overline{AM}^{esaa}| \tag{4.27}$$

The theoretical deviation in the $AD(T)$ metric for a $\rho_{mal} = 0.40$ and $\delta_{avg} = 800$W is 0.553. For the same attack the experimental result shows the change of $AD(T)$ to be 0.712. This indicates a reasonable approximation as well as the positive magnitude of change. Additionally, the theoretical value shows a increase in the $AD(T)$ which is also seen in the experimental result.

Table 4.4. Estimation Accuracy of Invariants with Irish Dataset

Parameter	Experimental	Theoretical
$\overline{AM}^{esaa}$	14.5245	14.257
$\overline{HM}^{esaa}$	11.8113	11.703
$AD^{esaa}(T)$	2.713	2.554
$E(\Delta AD(T))$	+0.712	+0.553

4.6.2. Optimal Evasion δ_{avg} Against Anomaly Detection Invariants. For an optimal evasion of our anomaly detection step, the adversary would want to use the maximum δ_{avg}, which creates a deviation in the invariants, that is just within the designed safe margin. In practice, since the adversary does not know the current $AD(T)$ value (since he cannot possibly control 100%) of the meters, he relies on the historical $AD(T)$, which can be possibly known the adversary through a database hack. Therefore, the adversary would ensure that given its attack type, and the fraction of compromised meters, the δ_{avg}, should be such that the following condition satisfies:

$$|AD^{esaa}(T) - AD_{hist}(T)| < 0.75 * \sigma_{AD(T)} \tag{4.28}$$

Specifically, expanding the Eqn. 4.26, we get the theoretical expected change in the statistical invariants as a function of the ρ_{mal} and δ_{avg} (the two key variables apart from the attack type that changes the in-variants). Thus, the estimated optimal evasion δ_{avg} can be found by the adversary solving the following optimization problem:

$$\delta_{avg}^{evasion} = \arg\max_{\delta_{avg}} f(\delta_{avg}) \tag{4.29}$$

$$\text{s.t.} \qquad f(\delta_{avg}) < 0.75 \times \sigma_{AD(T)}$$

$$\text{where } f(\delta_{avg}) = |AD^{esaa}(T) - AD_{hist}(T)| \quad = \quad |(|\overline{HM}^{esaa} - \overline{AM}^{esaa}|) - AD_{hist}(T)|$$

Note that estimated means $\overline{HM}^{esaa}$ and $\overline{AM}^{esaa}$ are given by the following as a function of the attack:

$$\overline{HM}^{esaa} = Box\left(\frac{\left(\left(1 + \frac{\delta_{avg}}{\overline{P} - \nabla\sigma}\right)^{\rho_{mal}} Nor(GM^{ba})\right)^2}{Nor(AM^{ba}) + \left(\rho_{mal} * \delta_{avg}\right)}\right) \tag{4.30}$$

$$\overline{AM}^{esaa} = (\overline{AM}^{ba} + |Box\Big(Nor(AM^{ba}) + (\rho_{mal} * \delta_{avg})\Big) - Box\Big(Nor(AM^{ba})\Big)|) \tag{4.31}$$

We can see that the above equations are a function of the ρ_{mal} and δ_{avg}, which formally analyses the effect of any attack on the statistical invariants. We have proven the approximation accuracy of our expression in Table 4.5 by showing how theoretical values approximate to experimental observations.

Table 4.5. Evasion δ_{avg}: Experiment vs Theory

ρ_{mal}	Exp. $\delta_{avg}^{evasion}$	Theo. $\delta_{avg}^{evasion}$
20	400	380
30	370	360
40	350	330
50	330	320
60	320	300

Table 4.6. Inferred MAD at Invariant Evasion Points

ρ_{mal}	Theo. Evasion δ_{avg}	$MAD^{evasion}(t)$	Current Mean
20	380	347	652
30	360	356	684
40	330	352	708
50	320	357	736
60	300	361	756

4.6.3. Formal Estimation of Robust Mean under Attacks. For robust mean closed form derivation, we just plug in the values of $Nor(AM^{esaa})$, $Nor(GM^{esaa}$, $Nor(HM^{esaa})$ or their box-cox transformed equivalents, (expressions derived previously) and plug into the Table 4.2 to find the theoretical value as shown below:
The $Box^{-1}(x)$ is defined as: $Box^{-1}(x) = (x * \lambda + 1)^{1/\lambda}$ where x is the value in box cox scale being remapped and λ is the box cox transformation parameter. The $\overline{AM}^{esaa}$ under camouflage is the same as the observed arithmetic mean, since it balances out the mean by virtue of its attack type.

$$\mu_R^{Additive}(t) = Box^{-1}(\overline{HM}^{esaa} - AD(T)), \quad \mu_R^{Deductive}(t) = Box^{-1}(\overline{GM}^{esaa} + AD(T)) \tag{4.32}$$

$$\mu_R^{Camouflage}(t) = Box^{-1}(\overline{AM}^{esaa}), \qquad \mu_R^{Conflict}(t) = Box^{-1}(\overline{GM}^{esaa}) \tag{4.33}$$

where $\overline{GM}^{esaa} = Box\left(\left(1 + \frac{\delta_{avg}}{\overline{P} - \nabla\sigma}\right)^{\rho_{mal}} Nor(GM^{ba})\right)$, and $\overline{HM}^{esaa} = Box\left(\frac{Nor(GM^{esaa})^2}{Nor(AM^{esaa})}\right)$

4.6.4. Condition for Successfully Evading of Meter Detection. Note that, we already proved that as ρ_{mal} increases, our invariant criterion forces the δ_{avg} to be smaller. Hence, the attacker cannot unilaterally increase one attack parameter to arbitrarily change

the median absolute deviation. Therefore, at the theoretical evasion δ_{avg}, we first present, the current median absolute deviation (under attacks) by varying from the ρ_{mal} from 20% to 60%, as listed in Table 4.6.

The trust score depends on the divergence between proximity distributions X_i and Y_i. The adversary has to bypass the invariant based anomaly detection to ensure that the mean and median absolute deviation correction does not take place. Furthermore, the adversary has to make sure that the majority of it's compromised meter readings are within the observed (biased) mean and the median absolute deviation (MAD) range. However, on average we say that to bypassing meter detection reliably the following condition needs to be satisfied for a given ρ_{mal}.

$$\delta_{avg}^{bypass} \leq min(\delta_{avg}^{evasion}, MAD^{evasion}(t))) \tag{4.34}$$

Let us look at a specific example from Table 4.6. For $\rho_{mal} = 40\%$, the $\delta_{avg}^{evasion}$ is 330 and the MAD at that evasion δ_{avg} based attack is 352. The min(330,352) is 330, which is the theoretical value to bypass the trust model. In our experiments, for $\delta_{avg} > 330$, the missed detection rate is lower than 10%, however at when $\delta_{avg} < 330$, it starts missing meters and missed detection becomes about 30%. This is also repeated in the Texas dataset in experimental results, where below 330, the missed detection becomes between 30%-40% proving correctness.

4.7. SPECIAL CASE STUDY ON FINE GRAINED ANOMALY BASED TRUST MODEL

Now we propose the customized version of our trust model that can *run in parallel* for effective identification under on-off or omission attack strategies. It is important to note that the fine grained anomaly based detection will produce different responses than

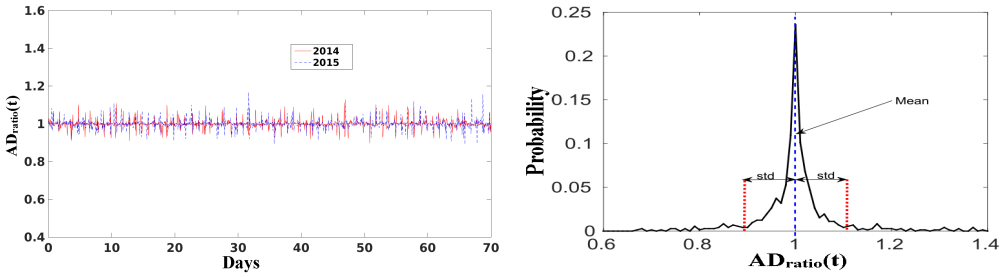
the coarse grained one, and therefore will invoke an augmented and modified version of the proposed trust model with novel embeddings of responses produced by the fine grained anomaly based security event detector.

4.7.1. Fine Grained Anomaly based Security Event Detection. In this subsection, we will introduce the invariant (metric) for fine grained anomaly detection, justify the choice of invariant, establish a detection criterion for fine grained attacks, determine attack type, strategy, start and stop times, and calculate the attack probability time ratio.

4.7.1.1. Proposed invariant. We propose a more fine-grained detection metric denoted by $AD_{ratio}(t)$ that is computed hourly, in contrast to $AD(T)$ that is computed daily. The $AD_{ratio}(t)$ is the ratio of the absolute difference between 'hourly' arithmetic and harmonic means between the previous $t-1$ and current time slot t. At any time slot t, the metric is defined as:

$$AD_{ratio}(t) = \frac{AD(t-1)}{AD(t)} \tag{4.35}$$

where $AD(t) = |HM(t) - AM(t)|$. The time series of the proposed metric AD_{ratio} for the Texas Dataset is shown in Figure 4.6(a).



(a) (b)

Figure 4.6. Texas Data (a) Time Series of $AD_{ratio}(t)$ (b) Distribution of $AD_{ratio}(t)$

4.7.1.2. Identifying normal range of $AD_{ratio}(t)$. Figure 4.6(b) shows the distribution of the proposed $AD_{ratio}(t)$ for the historical training dataset (2014 and 2015). It can be seen that the distribution of $AD_{ratio}(t)$ has a mean value of 0.998 with a standard devi-

ation of 0.1. Very few sample $AD_{ratio}(t)$ values lie beyond the second standard deviation. Let $AD^{norm}_{ratio} \in [AD^{min}_{ratio}, AD^{max}_{ratio}]$ denote the normal range of this fine grained $AD_{ratio}(t)$ metric.

4.7.1.3. Investigating effect of various attacks on $AD_{ratio}(t)$. For deductive attacks, we had mentioned that the decay rate of Harmonic Mean is larger compared to the decay in Arithmetic mean given the dataset. Therefore,

$$HM(t) - HM(t-1) > AM(t) - AM(t-1)$$

Solving the above, we get,

$$\frac{HM(t-1) - AM(t-1)}{HM(t) - AM(t)} < 1$$

$$\implies AD(t-1)/AD(t) < 1 \implies AD_{ratio}(t) < 1.$$

From the above, it is clear that a deductive or omission (which is a virtual deductive attack) attack when initiated, will cause a *sharp drop* in the proposed $AD_{ratio}(t)$ metric. When the attack stops, there will be a *sharp rise* in the AD_{ratio} metric, since the harmonic mean has to increase more than the arithmetic mean to restore the original ratio that is very stable and $AD_{ratio}(t) \rightarrow 1$. Therefore, the difference between $HM(t) - AM(t)$, will be much lesser compared to $HM(t-1) - AM(t-1)$. Since the denominator decreases when the attack stops, the $AD_{ratio}(t)$, experiences a sharp rise. Experimental verification of this is provided in Figure 4.7.

Similarly, for additive attacks, harmonic means have a slower growth rate compared to the arithmetic mean. Therefore,

$$HM(t) - HM(t-1) < AM(t) - AM(t-1)$$

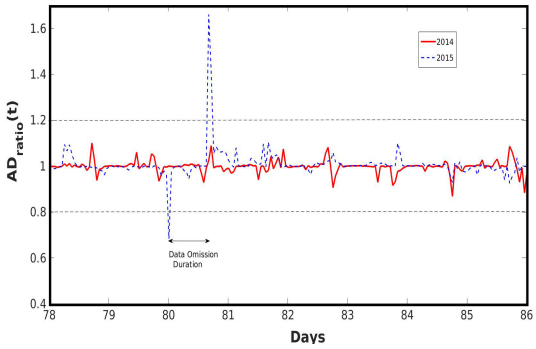


Figure 4.7. Omission Attack Example

Table 4.7. Concluding Fine Grained Security Events

$\mathbf{AD_{ratio}(2c-1)}$	**|FGAT|**	$\mathbf{t^{oe}}$	**Conclusion**
$> AD^{max}_{ratio}$	High	Constant	Additive ON-OFF
$< AD^{min}_{ratio}$	High	Constant	Ded/Camo ON-OFF
$< AD^{min}_{ratio}$	High	Varying	Omission Attack
$< AD^{min}_{ratio}$	Sparse	Don't Care	Omission Failure

Solving the above, we get

$$\frac{HM(t-1) - AM(t-1)}{HM(t) - AM(t)} > 1$$

$$\implies AD(t-1)/AD(t) > 1 \implies AD_{ratio}(t) > 1$$

From the above, it is clear that for additive attacks the $AD_{ratio}(t)$ must increase when attacks start, while for deductive and camouflage attacks the $AD_{ratio}(t)$ must decrease.

4.7.1.4. Detecting incidence of fine grained attacks. The following equation is similar to the coarse grained logic for confirming presence of opportunistic fine grained attacks.

$$AD_{ratio}(t) : \begin{cases} \in \{AD^{norm}_{ratio}(t)\} & \text{No Attack;} \\ \notin \{AD^{norm}_{ratio}(t)\} & \text{Fine Grained Attack} \end{cases} \tag{4.36}$$

4.7.1.5. Determining fine grained attack types and strategies. To reconstruct the security events under fine grained attack strategies, we first need to record the sequence of time slots where the event $AD_{ratio}(t) \notin \{AD^{norm}_{ratio}(t)\}$ occurred over the observed time duration, into a vector $FGAT = \{t(1), t(2), \cdots, t(c), \cdots, t(C)\}$, where $c \in \mathbb{N}$ is the set of first C natural numbers. The odd and even entries of the set FGAT are represented by $t(2c-1)$ and $t(2c)$ respectively and $|FGAT|$ is the cardinality of this set over the time frame under observation. Additionally, let the time difference between the pairs of odd entries and even entries be $t^{oe} = |t(2c-1) - t(2c)|$.

There are three important facets to monitor. First, the set of t^{oe} values help distinguish between deductive ON-OFF and omission attacks having similar signatures. Second, the cardinality of $|FGAT|$ is important to *distinguish between the possibility of omission attack versus omission failures*. Third, whether $AD_{ratio}(t)$ corresponding to the odd entries$t = 2c-1$ in FGAT are greater than AD^{max}_{ratio} or smaller than AD^{min}_{ratio}, help differentiate between additive, deductive, and camouflage data falsification types.

If the t^{oe} is constant for all odd values of c, then there is an on-off attack. Given that t^{oe} is constant, if $AD_{ratio}(2c-1) > AD^{max}_{ratio}$, it is an additive on-off attack, while an $AD_{ratio}(2c-1) < AD^{min}_{ratio}(t)$, it is an deductive on-off attack. Therefore, odd entries $AD_{ratio}(2c-1)$ helps to distinguish between additive, deductive or camouflage attacks. Since attacks are launched and stopped at periodic intervals, the $|FGAT|$ will not be singleton or sparse.

If the set of $t^{oe} = |t(2c-1) - t(2c)|$ consists of variable values, the $|FGAT|$ is not singleton or sparse, and $AD_{ratio}(2c-1) < AD^{min}_{ratio}(t)$, it is an omission attack (deliberate). On the other hand, if $t(2c-1) - t(2c)$ consists of variable values, $|FGAT|$ is singleton or sparse, the $AD_{ratio}(2c-1) < AD^{min}_{ratio}(t)$ is a omission failure due to non-adversarial reasons.

The missing data from a subset of houses at any time slot t is perceived as a deductive attack where actual power consumption values are replaced by null values which are lesser than actual data. This causes the harmonic mean to decay at a rate greater than compared to the decay in the arithmetic mean. Therefore, the difference between arithmetic mean and harmonic mean at time slot t increases compared to the previous time slot $(t-1)$ with no data omission. Therefore, the $AD_{ratio}(t)$ value between time slots t and $t-1$ experiences a sharp decrease. As long as the degree of omission stays same the AD_{ratio} is restored to normal value. When omission stops there will be another drastic change, where the harmonic mean will grow faster than the AM, such that the $AD(t)$ decreases compared to the $AD_{ratio}(t-1)$ calculated with missing data. Hence, there is a sharp drop in the proposed AD_{ratio}. This can be verified from Figure 4.7.

4.7.2. Estimation of Attack Probability Time Ratio as a Response. Apart from the robust consensus measures, which are required for fine grained attack strategies, we also need another additional response that needs to be embedded into the subsequent trust modeling step. This response is known as the *attack probability time ratio*.

The attack probability time ratio P_{attack} is an indicator of the fraction of time slots that the system was under attack over an observed time frame. For example, for an on-off attack having an ON period of 6 hours of attack in a day, $P_{attack} = 1/4$. Therefore, the fraction of time slots with no attack is $(1 - P_{attack})$, will be automatically considered as successes even when this meter is launching data falsification attacks. Therefore, in the probability space, these meters will not be further apart when there are on-off attacks versus no attacks. Hence, the time to detection of such meters will be significantly larger. To

reduce this, we need to keep track of the P_{attack}, and embed this information in the trust model. Such P_{attack} can be estimated from our designed FGAT vector, by the

$$P_{attack} = \frac{\sum_{c=1}^{C} |t(2c) - t(2c-1)|}{TS}$$

4.7.3. Trust Scoring Model with Attack Probability Time Ratio Embedding. Since on-off and omission strategies are discontinuous over time, the number of failures will not be as high compared to the case of continuous attacks in an observed time frame. This will produce $q^{(i)}$ values of compromised meters which are still high and therefore proximate to the parameter r in the true distribution. Hence, the time to detection convergence of meters with missing data (omission) or discontinuous falsification of data (on-off) will be time consuming, due to lack of evident separation in the probability space, which leads to classification errors as well.

Since the fine grained anomaly detector gives an early indication on the time slots when such on and off attack happened (from FGAT vector), a lesser weight can be given to the number of successes observed by weighing it with the fraction of duration the system is not under attack ((i.e., $1 - P_{attack}$)) in the observed frame F. In this manner, the time to detection of these meters could be improved. Under these opportunistic attack strategies, which are captured in the fine grained anomaly detector, the Eqn. 4.14 in the trust model is modified by weight to the number of successes j. This weight is $(1-P_{attack})$, which prevents the value of q to be very high even when the number of OFF periods is large compared to the ON period of attacks over the observed time frame containing TS windows. Hence, due to the attack context awareness, the observed distribution q under evidence of on-off and omission attacks (from the fine grained detector) for each meter is modified as:

$$q^{(i)} = \frac{(1 - P_{attack})j^{(i)} + 1}{TS + 2} \tag{4.37}$$

Eqn. 4.37, can be explained by the following: Note that the q is the probability that $Y^i(t) = 1$, meaning the meter $i's$ reading is falling within the robust mean and median absolute deviation. However, in an on-off attack, there are off periods, where this compromised meter's data is likely to achieve a value of 1. Hence, the probability of q over a given time frame TS is not remarkably different from r. Since the probability of q is specified by the number of successes j, a discounting factor of $1 - P_{attack}$ is required, since these $1 - P_{attack}$ time was not under attacks was a part of the OFF period. that be counted on as the FGAT vector shows evidence of orchestrated data falsification on selective ON periods (e.g., when prices are high/demand is high, etc.,).

The value of q is lesser compared to a value that contributes the entire observed j towards the probability of success. This ensures a larger difference between q and r in the probability space, which facilitates quick classification that is apparent even when the attacker acts honestly in majority of the time slots. The modification by Eqn. 4.37 is termed as *attack probability ratio time embedding* that customizes the trust model for better and quick classification of the compromised meters.

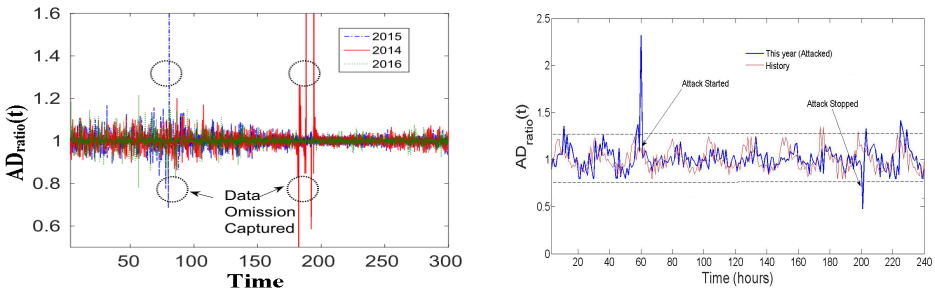
The relative entropy based trust model detect compromised meters only if the δ_{avg} is greater than the median absolute deviation of the datasets. From the Eqn. 22 and Eqn., 23, it is clear that if the δ_{avg} is lesser than the MAD, in most time instances, the Y_i of the attacked meters will be within that deviation and therefore be labeled as one instead of zero more frequently. Thus, there will not be a significant change in the probability of $p(Y_i = 1) = q$ in the attacked set. Therefore the deviation between X_i and Y_i in the probability space, will not be evident to produce a divergence that could clearly classify the malicious meters from the honest ones. Our studies from real datasets indicate that the MAD ranges between $290W - 350W$. Therefore, in our approach the missed detection errors increase $\delta_{avg} < 300$. However, the error rates are better than existing works across datasets as shown in the comparison in Section 5.6.

Intuitively, one solution to this limitation is to introduce multinomial evidence labels for each meter instead of binary labels (0,1), and then calculate the distances between the distributions in the probability space with a similar entropy measure. However, our experience showed that this is not enough to improve classification accuracy. This motivates the need for an alternative approach, that complements the relative entropy approach, when $\delta_{avg} < MAD$.

4.8. EXPERIMENTAL RESULTS

We utilized two big datasets for the performance evaluation of our proposed method. The first dataset is an hourly power consumption dataset from PeCan Street Project [62], containing 200 and 800 houses from a solar village near Austin, Texas for years 2014, 2015, 2016. The 2014 and 2015 dataset is used for learning (training), while 2016 is used a testing set. Two 90 day periods representing two seasons in 2016 were used as a scenario under attacks to generate the malicious dataset. The malicious data sets were generated from the real data samples that were fed with our threat model with various ρ_{mal} and δ_{avg}. The second dataset is a power consumption dataset from 5000 houses from six micro-grid regions in Dublin, Ireland [61], which was utilized to prove the scalability and generality of our proposed approach. The datasets are publicly accessible.

The experimental section is divided into four parts: (i) First, we show some results related to the fine grained anomaly detection; (ii) Second, we show supervised classification results for 200 houses for all attack types over various δ_{avg} value (iii) Third, we show unsupervised classification (using K-means) for 200 houses. (iv) Fourth, we show a performance evaluation in terms of classification error rates for both 800 houses and 5000 houses using unsupervised classification, to prove that error rates scale well for larger micro-grids and works across different combinations of ρ_{mal} and δ_{avg} for various datasets, (v) we show real time nature of detection of smart meters, (vi) a few comparisons of our performance with existing works.



(a) (b)

Figure 4.8. AD Value under (a) Data Omission (b) Additive On-Off Attack

4.8.1. Fine Grained Anomaly Detection Forensics. Here we show some results on how the fine grained anomaly detection metric can detect opportunistic strategies such as Omission and On-Off.

1) Data Omission Strategy: Figure 4.8(a), shows a result on the uncleaned real dataset with missing data. We do not know whether this was due to an attack or a network failure. Nonetheless, this is analogous to data omission, and our proposed fine grained anomaly detection metric $AD_{ratio}(t)$, can capture such events. Since the metric $FGAT$ contains only two entries for the whole year, it is evident that this is particular data omission is likely an isolated failure, rather than an attack. A magnified version was shown previously in Figure 4.7, to prove that the $AD_{ratio}(t)$ first decreases (when omission starts) and then increases (when omission stops).

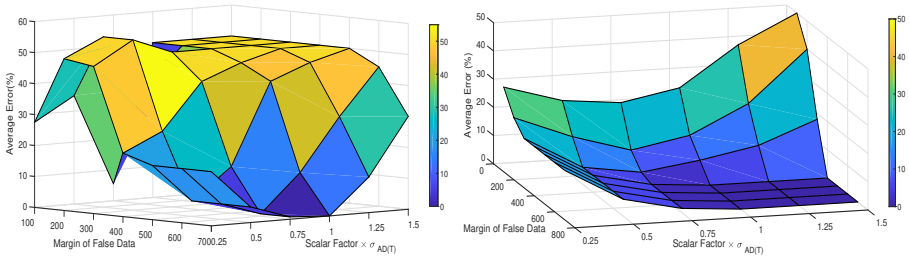
2) On-Off Strategy: We study a small timeline of say 10 days, and start additive attacks (ON) and then stop it (OFF), it is possible to detect the ON period of attacks with the proposed $AD_{ratio}(t)$ metric. As an example, Figure 4.8(b), shows an additive attack with δ_{avg} = 600, which was launched from the 60-th hour to the 200-th hour of this time-line. Note that, in additive attacks the harmonic mean grows at a much slower rate compared to the growth in arithmetic mean (given a sufficiently high δ_{avg}). Hence, at the 60-th slot the difference between the arithmetic mean and harmonic mean is larger than the previous time slots. There the ratio $AD_{ratio}(t)$, shows a sharp increase.

4.8.2. Effectiveness of the Anomaly based Attack Context Generation. The effectiveness of the anomaly detection step is directly related to the embedding of attack context in the proposed trust model which in turn preserves the classification accuracy, lowers false alarm rates and, improves time to accurate classification of the compromised meters. Therefore, the effectiveness of the anomaly detector is demonstrated through the minimization of classification error rates (defined as the average of missed detection and false alarm rates).

The effectiveness of the anomaly detector is also directly dependent on the value of threshold ($\pm\gamma\sigma_{AD(T)}$) around the historical $AD(T)$ value. Recall, that γ is the scalar factor that parameterizes the threshold variation. Therefore, to demonstrate the effectiveness of anomaly detector we show the error rates (average of missed detection and false alarms) as a function of the varying margins of false data and variable candidate thresholds in the anomaly detector. Through this, we also demonstrate the optimal threshold range that the anomaly detectors should use to minimize the error rate in classification.

1) Effectiveness of Error Rate Minimization: We report a $0.75\sigma_{AD(T)}$ as a threshold that produces minimal error rates across extreme values of ρ_{mal} and over all trained values δ_{avg}. This study is done because the defender has no control on the actual ρ_{mal} and δ_{avg} values that will manifest. Figure 4.9(a) clearly shows that a global minima for classification error rate exists for a threshold of $0.75\sigma_{AD(T)}$, which produces minimal error rates regardless of δ_{avg} among all candidate thresholds for the Irish dataset for $\rho_{mal} = 15\%$ under additive attacks. Figure 4.9(b) shows that the minimal error rate is achieved for the same $0.75\sigma_{AD(T)}$ across all δ_{avg} for different $\rho_{mal} = 50\%$ under a deductive attack.

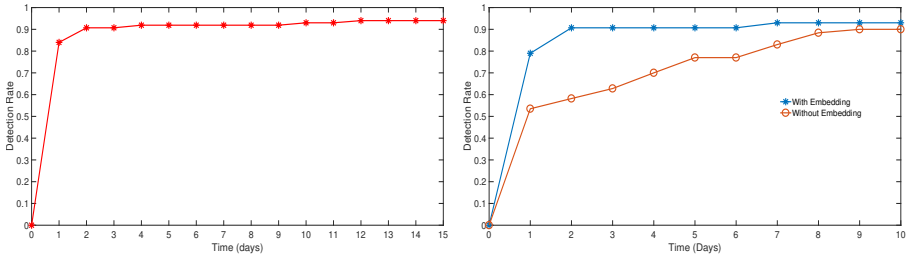
2) Effectiveness of Time to Detection (TTD): Figure 4.10(a) is a CDF that is a testimony of the convergence times to the detection rate for an additive attack with $\rho_{mal} = 20\%$ and $\delta_{avg} = 600$ and a data-order aware strategy. The classification of compromised meters is not only accurate but also happens in a very quick time. The steady state detection rate as observed from Figures 4.10(a) is achieved within 2 days. Additionally, Figure 4.10(b),



(a) (b)

Figure 4.9. Error Rate Minimization (a) Low ρ_{mal} = 15%; (b) High ρ_{mal} = 50%

shows the effectiveness of the probability of attack time ratio embedding (as a result of the fine grained anomaly detector) into the trust model, and proves that it improves the time to detection of compromised meters significantly. The Figure 4.10(b), shows the comparison between the CDF of detections with and without embedding under an on-off strategy with an on-to-off ratio of $1:3$. We can observe that the circled line corresponding to detection rate without the P_{attack} embedding approaches its steady state after atleast 10 days compared to the blue line with the probability of attack time ratio embedding that approaches the steady state detection rate of 90% within just 2 days.



(a) (b)

Figure 4.10. Performance (a) TTD of Compromised Meters (b) Comparative Effectiveness of P_{attack} embedding

4.8.3. Supervised Classification. In this case, the threshold is obtained from a small set of training meters from the training dataset which is then applied to the testing set with the full set of meters in test set. Later, we show how our proposed approach performs in an unsupervised mode as well.

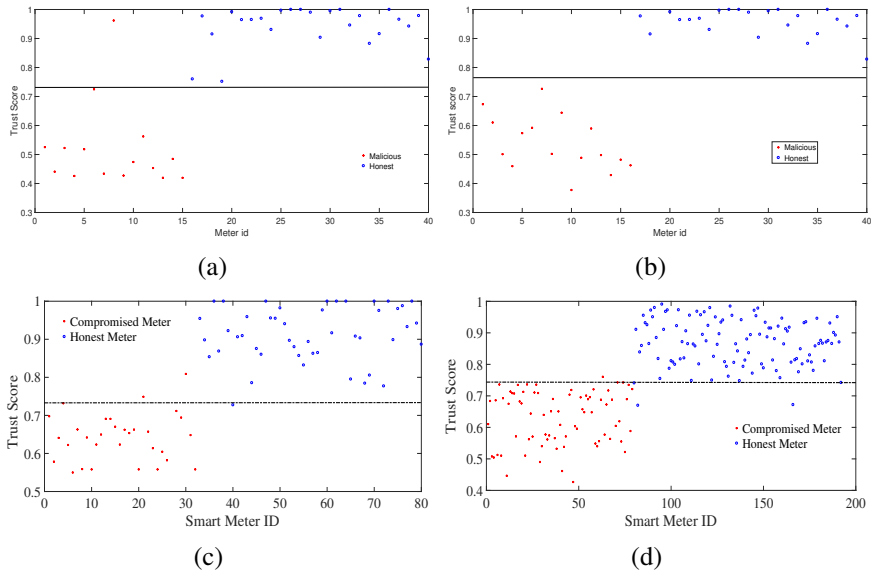
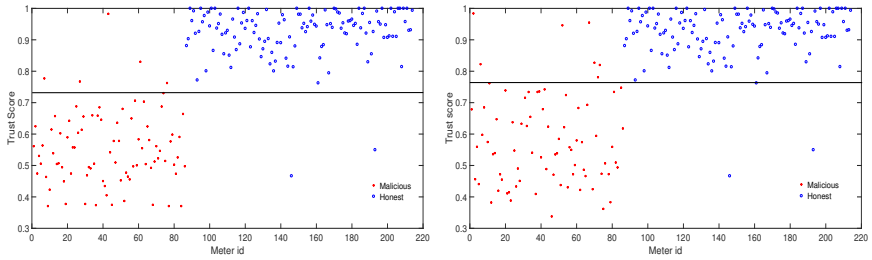


Figure 4.11. Training Set: (a) Additive; (b) Deductive (c) Effect of Meter Sizes (d) Effect of Different Season

4.8.3.1. Training set. First, we use a training data set from 40 houses and use power consumption reported in 2014 for a month. In each training case, we labeled 40% meters as compromised ($\rho_{mal} = 0.4$) and alter their reported values with $\delta_{avg} = 500W$ and then plotted, the corresponding trust values. We chose intermediate values of ρ_{mal} and δ_{avg} to prevent overfitting or underfitting. We use the trust scores of these labels, to calculate a threshold that can linearly separate between compromised and non-compromised nodes. We use a decision tree based classifier called CART (Classification and Regression Trees) to find the supervised thresholds. The results of training for additive and deductive attacks are shown in Figures 4.11(a), 4.11(b). Then we studied, the effect of meter training size by repeating this with 80 meters (See Figure 4.11(c)) as well as the effect of the training time period (seasonal change) on all meters (See Figure 4.11(d)) to test the sensitivity of training for supervised classification. The conclusion is that all thresholds are close.

4.8.3.2. Classification with testing set. For testing illustration, we use 2016 dataset from Texas and the attack launching period is one month. We set $\rho_{mal} = 0.4$ and $\delta_{avg} = 600W$. *More results over completely different combinations of* ρ_{mal} *and* δ_{avg} *are presented*

later to prove the robustness performance. Results for additive and deductive attacks shown in Figures 4.12(a) and 4.12(b), exhibit a clear separation between honest and compromised nodes with a false alarm rate of 1.5% in both the cases. The missed detection rate is 5% and 8% for additive and deductive attacks, respectively.

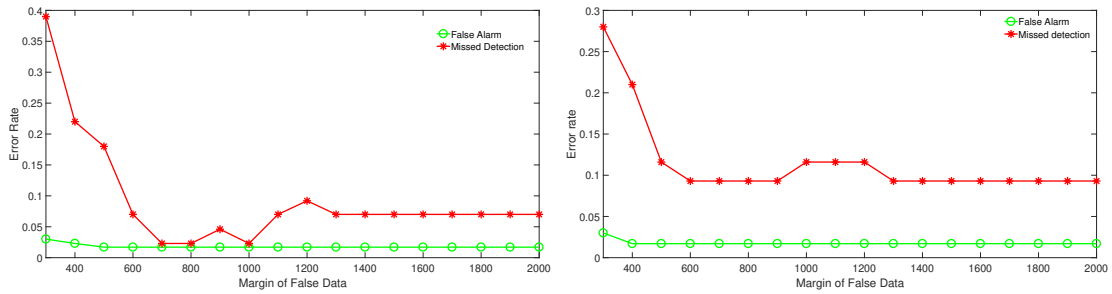


(a) (b)

Figure 4.12. Testing Sets: (a) Additive; (b) Deductive

4.8.4. Classification Performance Evaluation. Figure 4.13(a), shows the classification error rates for a larger dataset of 800 houses in terms of missed detections and false alarms under additive attack for the unsupervised classification approach over all possible values of δ_{avg}, given a $\rho_{mal} = 0.50$. From this, we can conclude that the relative entropy approach works well for most values of δ_{avg} even when 50% of the nodes are compromised. Particularly, the missed detection is higher than false alarms, which means detection rate is more of a concern for additive attacks particularly, when $\delta_{avg} < 400$. We report 22% missed detection and 2% false alarm at $\delta_{avg} = 400$. At $\delta_{avg} = 300$, the missed detection rate increases to 39%. Therefore, we experimentally verify that is methodology is not well suited for the margin of false data lesser than the median absolute deviation of the dataset.

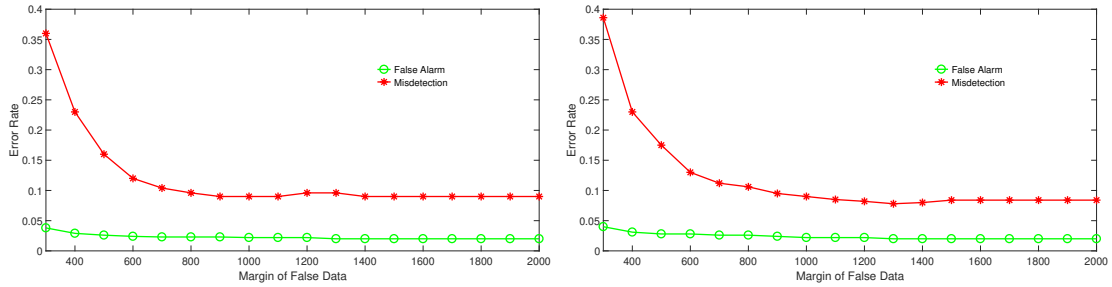
Figure 4.13(b) shows the classification error rates in terms of missed detections and false alarms for unsupervised classification approach over all possible values of δ_{avg}, given $\rho_{mal} = 0.50$ under a deductive attack for 800 houses. This indicates the robustness of our solution across all margins of false data under deductive attacks. The missed detection rate does not have an upper evasion point compared to our preliminary work [33] and other information theoretic approaches.



(a) (b)

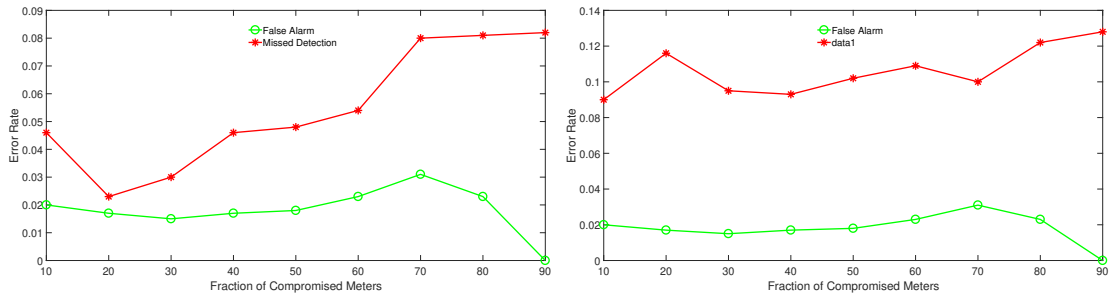
Figure 4.13. Error Sensitivity Analysis over δ_{avg} (Texas): (a) Additive (b) Deductive

Figures 4.14(a) and 4.14(b), shows the classification error rates in terms of missed detections and false alarms for the unsupervised classification approach over all possible values of δ_{avg}, given a $\rho_{mal} = 0.20$ under a camouflage and conflict attack.



(a) (b)

Figure 4.14. Error Sensitivity Analysis over δ_{avg} (Texas): (a) Camouflage (b) Conflict



(a) (b)

Figure 4.15. Error Sensitivity Analysis over ρ_{mal} (Texas): (a) Additive (b) Deductive

Figure 4.15(a) confirms that the error rate is within 10% for all possible fractions of compromised nodes as high as 90%, for the additive attack. This indicates the robustness of our solution to higher fractions of compromised nodes for additive attacks. Additionally, Figure 4.15(b), indicates the robustness of our solution to various margins of false data under deductive attacks. The missed detection rate does not have an upper evasion point in terms of ρ_{mal}.

4.8.5. Comparisons with Existing Work and Scalability of Error Rates. Figure 4.16 shows that the false alarm rate for the Irish dataset across 5000 houses is less than 2%. Additionally, the missed detection rate is below 20% for any $\delta_{avg} \geq 350$W. Second, the Figure 4.16, compares our performance for deductive attacks with existing works in terms of missed detection (MD) and false alarm (FA) rates, that use techniques such as One class SVM [12], multi-class SVM [12], F-Deta (Information Theory based) [40], folded Gaussian trust [35]. The proposed approach's performance in terms of FA and MD is shown in solid lines with season wide cross validation. From the Figure 4.16, it is evident that across various margins of false data, our FA and MD rates are lowest compared to the other approaches. Additionally, across the same chosen δ_{avg}, our work remains resilient under high fractions of compromised meters compared to previous works.

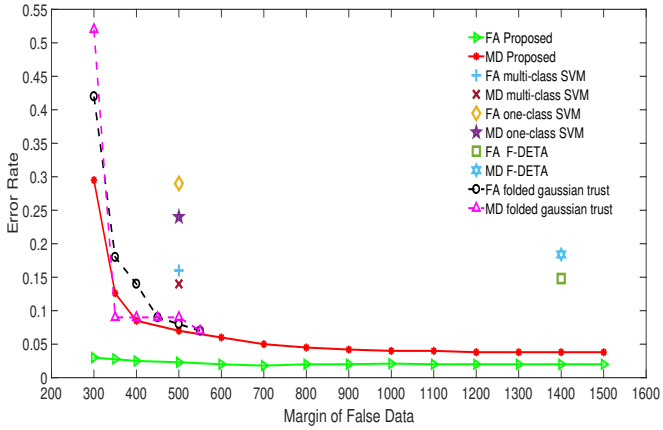


Figure 4.16. Error Rate Comparison with Existing Works: Irish Dataset

Table 4.8, also quantifies the advantages and benefits of our framework in comparison to some of the recent works in this area, in terms of 'other aspects' that are not directly comparable with previous works. These aspects include ranges of studied margin of false data δ_{avg} and ρ_{mal}, detection rate convergence times, applicability to multiple attack types, and both coarse and fine grained opportunistic attack strategies. While our framework applies to all attack types, other works focus on deductive attacks except our previous work. Therefore, the numbers for our framework in Table 4.8 are for deductive attacks only for a fair comparison. However, our work is much broader compared to existing works since it addresses an umbrella of various threats simultaneously. Some entries in the table marked NA when a concerned parameter that is not reported explicitly. Moreover, our work shows error sensitivity performance over both datasets.

Our framework has a much better performance over a wide attack strategy space with ρ_{mal} ranging from 1% to 90% and δ_{avg} ranging from $300W$-$2000W$ compared to the existing works that assume a narrower or fixed attack strategy space in terms of ρ_{mal} and δ_{avg}. Works such as [40] reasonable missed detection rates, but assume a very high δ_{avg} of above 1000W which facilitates easier classification. The false alarm rate at only select δ_{avg} is provided and the detection time is not clear. At this assumption, our missed detection rate is less than 6% and false alarm rates are 8% for a larger dataset of 800 meters. The work in [12] has a small ρ_{mal} of 0.72%, but at their assumed $\delta_{avg} = 400W$, our MD and FA rates are better for both additive and deductive attacks across lower and higher ρ_{mal} values while needing the same number observations per day. Our work can also perform classification in an unsupervised mode compared to the supervised approach with a high training time as reported in [12]. The upper evasion limit of high δ_{avg} and ρ_{mal} vanishes, compared to our preliminary work [33], due to the robust mean and median absolute deviation correction and convergence times are preserved under omission and on-off attacks. Our recent work [69] also showed that harmonic and arithmetic mean calculations are compatible with fully

homomorphic encryption schemes enabling privacy preserving security computations in AMI. Therefore, our security method unlike others will be compatible with AMI privacy requirements [70].

Table 4.8. Comparison with Existing Work

Parameter	Proposed	CPBETD [12]	ARMA [14]	Prior [33]	F-Deta [40]
False Alarm	1.5%-4%	29%	33%	11%	NA
Missed Detection	30%-0%	24%	28%	8%	10%-36%
δ_{avg}	300-3000W	400W	NA	700-800W	1000W-2000W
ρ_{mal}	1% – 90%	1%	NA	≤ 40%	55%
Attack Type	All	Deductive	Deductive	All	Deductive
Detection Time	2-3 days	77days	30 days	30 days	NA
Opportunistic strategies	Yes	Yes	No	No	No

4.9. INFERENCES

We proposed coarse and fine grained anomaly based security event detection technique that serves as an early indicator of the presence of organized data falsification attack, infers the attack type, and strategy inflicted, which helps to reconstruct an attack context that includes a response metrics such as robust mean, standard deviation, attack probability time ratio, which depend on what kind of threat has been inflicted. Based on this attack context, the relative entropy trust model adapts itself dynamically in runtime, to produce linearly separable trust scores that can identify the compromised meters injecting false data with higher accuracy and in near real time. In all, we showed that our framework applies regardless of the high fraction of compromised nodes, and across various margins of false data in an unsupervised classification mode as well with very low time to detection of compromised meters.

5. DETECTION OF STEALTHY SMART GRID ATTACKS

Lower margins of attack strength are stealthier and hence harder to detect. Additionally, the parameter that quantifies the total percentage of such compromised smart meters in a micro-grid is termed as *'attack scale'*. Orchestrated and coordinated attacks often have larger attack scales compared to the isolated attacks. Moreover, a smart adversary can find a cheaper exploit to compromise smart meters, thereby allowing even a lower margin attack to have a significant impact on the utility when compared to adversary's total attack cost. Orchestrated attacks are usually launched by organized and stealthy adversaries (business competitors, organized cyber criminals), who will expect to lower the margins of data falsification per meter such that meters are not easily caught, by hiding behind the randomness of smart meter data. Rival nation states may also be motivated to launch organized attacks, since meter data dictate the generation and distribution of electricity to critical infrastructures.

5.1. CONTRIBUTIONS

We propose a novel information-theoretic anomaly scoring framework, called *Modified Diversity Index Scoring*, that captures smart meters launching additive, deductive, and alternating switching attack types across a wide range of very low to very high margins of attack strengths and attack scales, while also lowering false alarms and missed detection, compared to existing approaches, for various stealthy attack strategies.

Specifically, we first establish an analogy between the intelligent data falsification attacks in smart meters and the monitoring of ecological balance of species distributions in a geographical region. Next, we show that information-theoretic approaches, such as Renyi and Tsallis Entropies (popular in ecology), Shannon's Entropy and Kullback-Leibler Divergence common in computer security; are not sufficient to address this problem. Thereafter, by studying the effects of various attack types on the probability of relative

abundance of each discretized space of the random variable of power consumption, we identify the need for modifications to the existing information theoretic measures. To this end, we introduce modifications to the concept of Renyi Entropy and Hill's Diversity Entropy by embedding a notion of a weighted expected self-similarity mapping of a smart meter IoT device across multiple temporal scales. Next, we embed an appropriate order of the entropy and a weighted relative abundance vector to capture subtle drifts in the horizontal, vertical and incline directions in the probability space, thereby resulting in a diversity index score. The higher the diversity index score, the more likely is the meter launching data falsification attacks. Thereafter, we offer a supervised approach to the learn the parameters of our proposed model that maximizes the separation of diversity index scores between the set of labeled compromised and honest meters, accompanied by cross-validation.

We validate the proposed framework with multiple *full year* real datasets, demonstrating its generalization across a wide range of attack strengths, scales, types, and strategies, across seasons. Experimental results show that our method exhibits lower false alarms and missed detection even when the average attack strengths *per meter* lower than $400W$ (which causes evasion in previous defenses) for both Texas Dataset (200 meters) and Irish Dataset (1300 meters). Specifically, we show that model generalizes to successfully detect deductive, alternating switching attacks and strategies that were not used to train the model. A comparison with existing works exhibits improved performance in terms of reduced undetectable attack strategy space when the attacker has knowledge of our method. We also provide a tradeoff between impact of missed attacks versus cost of base rate false alarms (when there no attacks in the test set).

5.2. DIVERSITY INDEX BASED TRUST SCORE

5.2.1. Forming Species Self Similarity Matrix. We build a square matrix $\boldsymbol{D}$ of RxR dimensions known as the *species self similarity matrix*, where only diagonal entries are non-zero, and quantifies the effective level of similarity (or difference) of the relative

abundance of a species with itself between the current time window (where the diversity index is being calculated) compared to past windows. The past may be previous years's history or a shorter term history of a set of previous consecutive time windows. For smart city application context, we use consecutive time windows, given the observation that shifting trends in data, can diametrically change the self similarity of species without presence of attacks over yearly time horizons.

To build **D**, the simplest approach would be an absolute difference between the relative abundance of each species category between the current and the previous time window. Mathematically, let matrix $\mathbf{p}(f-1)$ denote the species abundance in previous time window $f-1$ for the i-th meter and the same at the current time window f is denoted by $\mathbf{p}^*(f)$. Then, the most simple self similarity matrix could be $\mathbf{S}(\mathbf{f}) = |p(f-1) - p^*(f)|$, where:

$$\boldsymbol{p}(f-1) = \begin{bmatrix} p_1 & 0 & \dots & 0 \\ 0 & p_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & p_R \end{bmatrix}_{R\times R} \qquad \boldsymbol{p}^*(f) = \begin{bmatrix} p_1^* & 0 & \dots & 0 \\ 0 & p_2^* & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & p_R^* \end{bmatrix}_{R\times R}$$

However, we found two problems with this approach. (1) this will fail to detect incremental ramp or boil-frog attack strategies, that cause very small vertical changes over time. Hence, we need to look over a longer time horizon for 'sustained' vertical changes. (2) there could be false alarms, since some of the meters may show a higher change in the legitimate difference of relative abundance in species without attacks in given pairs of windows. Without any transformation, it creates a higher change in the eventual trust score under benign changes.

This gives an intuition that once an idea on the bounds of legitimate vertical changes is learned, changes beyond that can be over-weighed, while changes below those bounds can be discounted. These two aspects are embedded in the following way: Let the difference

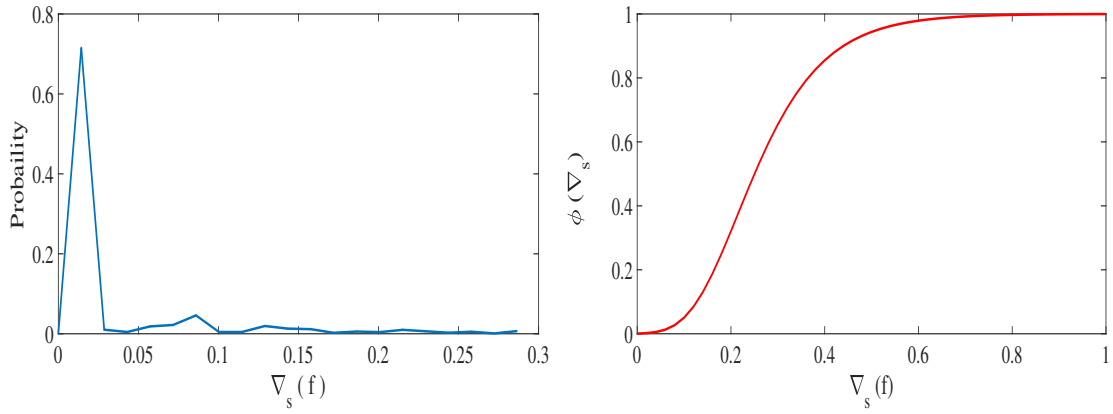


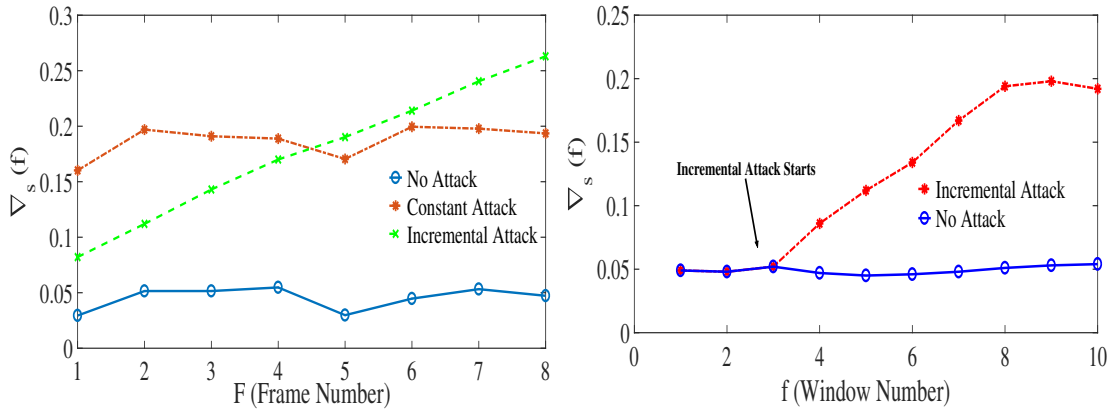
Figure 5.1. Texas Dataset (a) Benign Sample $\nabla_s(f)$ (b) The ϕ transformation function

between relative abundance vector between any two consecutive windows be denoted by $\epsilon_s(k) = p_s(k-1) - p_s(k)$, a shorter term similarity. Then we keep a long term memory of ϵ_s for each species represented by:

$$\nabla_s(f) = \sum_{k=f-F}^{f} \epsilon_s(k) \tag{5.1}$$

such that $\nabla_s(f)$ keeps the cumulative sum of the differences observed between pairs of time windows for a sliding frame containing F previous windows. When there are no attacks, $\nabla_s(f)$ has no increasing trend (see Figure 5.2(a)) and the values are typically very small (See Figure 5.1(a)). Infact, across an appropriate frame length (F), the ∇_s flattens out (blue lines in Figures 5.2(a) and 5.2(b)). In contrast, for incremental attacks, there is a small monotonic increasing trend in ∇_s (green and red lines in Figures 5.2(a) and 5.2(b) respectively). For all other strategies, the average ∇_s is larger, under attacks.

The species self-similarity matrix is given by $\boldsymbol{D}(f)$ such that each diagonal element is computed through a function of the form $(\phi(\nabla_s(f))$, such that the diagonal elements in $\boldsymbol{D}(f)$ is a mapping that takes the ∇_s across the frame within each species as the input and mathematically written as:



(a) (b)

Figure 5.2. Frame (a) Varying Length (b) Frame Tracking under Incremental Ramp Strategy

$$\boldsymbol{D}(f) = \begin{bmatrix} |\phi(\nabla_1) & 0 & \dots & 0 \\ 0 & \phi(\nabla_2) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \phi(\nabla_R) \end{bmatrix}_{R \times R}$$

where each entry

$$\phi(\nabla_s) = \frac{1}{(1 + A_b e^{-B_b(\nabla_s)})^{1/\nu}}; \quad \phi(\nabla_s) \in [0, 1] \tag{5.2}$$

is a generalized sigmoidal function which inputs the vertical change over a frame of length F at a time window f. The ϕ transformation produces the necessary weighing that reduces the false alarm rate while not sacrificing missed detection. Here the B_b is a growth rate parameter controlling the value of ∇_s for which the $\phi(.)$ function reaches its max value, while ν is a displacement parameter that controls the value of ∇_s, where the $\phi(.)$ function enters the exponential growth phase. The A_b is a parameter that decides the initial y-intercept, when ∇_s is zero. Figure 5.1(b), shows the ϕ function.

5.2.2. Expectation of Temporal Self-Similarity. Now we quantify the overall average change in the similarity of the i-th meter. Let $\boldsymbol{p}$ denote the probability abundance vector of species calculated over a time window in near history (ideally just before attack starts) and the $\boldsymbol{D}$ is a probabilistic measure related to that p (given the design of $\boldsymbol{D}$), so that we get something similar to a second-order expectation where the random variable is itself a probability vector $\boldsymbol{p}$. Mathematically, we do the following operation:

$$\Big[E(D)\Big]_{Rx1} = \Big[\boldsymbol{D}\Big]_{R\times R}\Big[\boldsymbol{p}\Big]_{Rx1} \tag{5.3}$$

where $E(D)$ is an Rx1 matrix where each element represents the expectation (average) change in the self similarity (in terms of probability of species abundances) of the corresponding species between over this time frame. Each element of the $E(D)$ is of the form $(\phi(\nabla_s) * p_s)$, which gives an idea on the index of vertical change within each species s between two time frames. Let us call $\boldsymbol{p}$ as the reference probability vector.

Now it could be tricky to get the correct reference vector belonging to a frame just before attacks, especially if incremental attack strategy inflicted. However, $\nabla_s(f)$ also allows us to pinpoint this by backtracking the $\nabla_s(f)$ variation (See Figure 5.2(b)) and the p is built from before the window just before the change-point of $\nabla_s(f)$.

5.2.3. Diversity Order Embedding. From theoretical intuition, one requirement was to magnify changes in intermediately rarer species, which we accomplish here. We add the order into the expectation of similarity in a similar way that appears as a power in the Hill's Diversity Index, in order to achieve the embedding of non-uniform vertical change such that we get the following:

$$[M]_{Rx1} = [E(D)]^q \tag{5.4}$$

$$\boldsymbol{M} = \begin{bmatrix} \Big(\phi(\nabla_1).p_1\Big)^q \\ \vdots \\ \Big(\phi(\nabla_s).p_s\Big)^q \\ \vdots \\ \Big(\phi(\nabla_R).p_R\Big)^q \end{bmatrix}_{R\times 1} \tag{5.5}$$

5.2.4. Magnifying Quantity of Species with Changes. From theoretical intuition, we need to finally ensure that very small incremental changes but happening in many unique IDs of rare species, has importance in the resultant functional form of modified diversity index we are striving to achieve. We put more emphasis on the shifts in rarer species as a weight to each of the species in the $R \times 1$ matrix, $[E(D)]^q$. Note that we need a scalar value for the diversity index trust score and the quantity weight matrix needs to be a $1 \times R$ matrix for the scalar to exist. Hence, we seek to design a weight vector that is $1 \times R$ dimension.

$$[W]_{1xR} = \Big[J\Big]_{1xR} - \Big[\boldsymbol{p^T}\Big]_{1xR} \tag{5.6}$$

where $[J] = [1 \ldots 1]_{1xR}$ is a matrix containing all 1's for R columns, where the intuition is that the one minus a rare value will be a high value and too many of these occurrences, will push the resulting scalar to a higher portion in the number line. Hence, the result weight factor is given by:

$$\boldsymbol{W} = \Big[(1-p_1) \quad \ldots (1-p_s) \quad \ldots \quad (1-p_R)\Big]_{1\times R} \tag{5.7}$$

5.2.5. Final Modified Diversity Index Trust Score. The diversity index based trust score of the q-th order for a smart meter i is given by the multiplication of $\boldsymbol{W}$ and $\boldsymbol{M}$. The reason they are multiplied is to achieve the functional form of Hill's index, as we will see next.

$$TR^i(q) = W \times M = \Big(([J] - [\boldsymbol{p^T}]) \times \Big[\boldsymbol{D}\,\boldsymbol{p}\Big]^q\Big) \tag{5.8}$$

Verify how Eqn. 5.8 confirms to the original functional form of mathematical abstraction of a diversity index score for identifying data falsification. By plugging in Eqn. 5.5 and Eqn. 5.7 in Eqn. 5.8, we get a scalar value due to matrix multiplications of dimensions 1XR and Rx1, which gives:

$$\mathrm{TR}^i(q) = (1-p_1).(\phi(\nabla_1))^q.p_1^q \quad + \cdots + (1-p_R).(\phi(\nabla_R))^q.p_R^q$$

If we assume $(1-p_s) = x_s$ and $(\phi(\nabla_s))^q = y_s$, then the above reduces to the desired abstraction of the mathematical functional form of Hill's index.

Hence, to conclude the modified diversity index score of a meter i that can detect compromised meters is:

$$rD^i = \left(([J] - [\boldsymbol{p^T}]) \times \left[\boldsymbol{D\,p}\right]^q\right) \tag{5.9}$$

where $rD^i > 0$, if $q > 0$. The whole exponent factor of $\frac{1}{1-q}$ in the original functional form is ignored since it does not provide any added classification advantage as far tracking changes. Another important point is the nature of change in diversity score after the attack is launched, and its effect on the final distribution of rD^i values of compromised versus honest meters. Due to the nature of Eqn. 5.9, where changes in each species are added up, the meters launching data falsification will experience an increase in the diversity scores after the attack. In contrast, the non-compromised meters will exhibit a lower diversity score than the compromised meters. We will verify this in the experimental results section.

5.3. PARAMETER LEARNING AND THRESHOLD

Now that we have the architecture of our base model, we need to provide a generalizable way of learning various parameter values given any dataset. Our approach towards this is a supervised one, where we divide the training set into two parts: first, without any attacks; the second containing attacks from a subset of meters we choose and program them

to simulate a limited set of attacks. Our method learns parameters according to a target objective function that maximizes the difference between the diversity index scores of the honest and malicious classes in the training set. Later on, we use cross-validation set to find a threshold and then apply it on a testing set for performance evaluation.

5.3.1. Training Set Details. We use the full year of 2014 as the training set for Texas dataset. The attack starts after the end of 6-th month. The malicious class labels contain the following attack features: An additive attack with δ_{avg} = 100W, ρ_{mal} = 30%, with an incremental ramp strategy that increases by 20W every 15 days. The idea is that if it detects for the smallest and slowest moving attack, it will be able to detect anything stronger. Other parts of the threat model are not used for training, since we need to verify that our method is *generalizable* to detect 'mutated' and 'unknown' attack realizations that it was not trained on.

5.3.2. Decision Variables. The controllable decision variables are namely A_b, B_b, ν, sw, q and F which are candidates for optimization. Among these, parameters strongly related to the dataset are B_b and ν, others are weakly related to the dataset. Note that the δ_{avg} and ρ_{mal} are uncontrollable decision variables which are beyond defenders knowledge. However, it is known that if we observe a linear separability between diversity index scores of a compromised and honest set of devices, for a lower δ_{avg}, it will automatically hold for higher δ_{avg} values by virtue of our scoring design. Therefore, during learning, we train with only select candidates of δ_{avg} that are below the desired lower bound of sensitivity δ_{avg}^{dlb}. For tractability of search space, we partition the candidate species widths and candidate δ_{avg} into discrete partitions with upper and lower bounds δ_1 and δ_P.

5.3.3. Objective (Error) Function. The objective function (or the error/loss function) should maximize the separation between compromised and honest devices, in terms of the distribution of their diversity index scores. Hence, we used the squared difference

of average of diversity index scores between the compromised and honest sets in the training set. Intuitively, that combination of parameters/decision variables that maximizes this objective function is the optimal parameter set.

$$e = max\Big(\frac{\sum(rD^h)}{N-M} - \frac{\sum(rD^m)}{M}\Big)^2 \tag{5.10}$$

$$\text{s.t.} \qquad A_b > 0; \qquad 0 < B_b < 1; \qquad 0 < \nu < 1$$

$$\text{s.t.} \qquad 0 < q < \infty; \qquad w_1 < sw \leq \delta^{dlb}_{avg}; \qquad 1 \leq F < F_{max}$$

It might seem that there too many variables to optimize. However, in reality, the search space of sw, q, A_b turns out to be bounded and small, once we apply the following pruning logic and design considerations: The candidate species width sw is upper bounded by the desired lower bound sensitivity of attacks δ^{dlb}_{avg}, which is small, making the sw range limited. Furthermore, given the role of the Renyi diversity order, we can prune the search space of diversity order to $q \in (0, 1]$.

The optimization can be solved using a grid search; or an efficient method like gradient descent which scales well when there are many parameters with a wide search space. For gradient descent to work, the error function needs to be transformed into a convex function. Our objective function is a concave function with a global maxima. Such functions can be converted into a convex function using the negative logarithm of the original objective function, and then apply gradient descent. However, accuracy depends on the smoothness of the convex function. In our implementation, the number of parameters is limited, and has a smaller search space either by design or through pruning. Hence, we solved our optimization, using a grid coordinate search method.

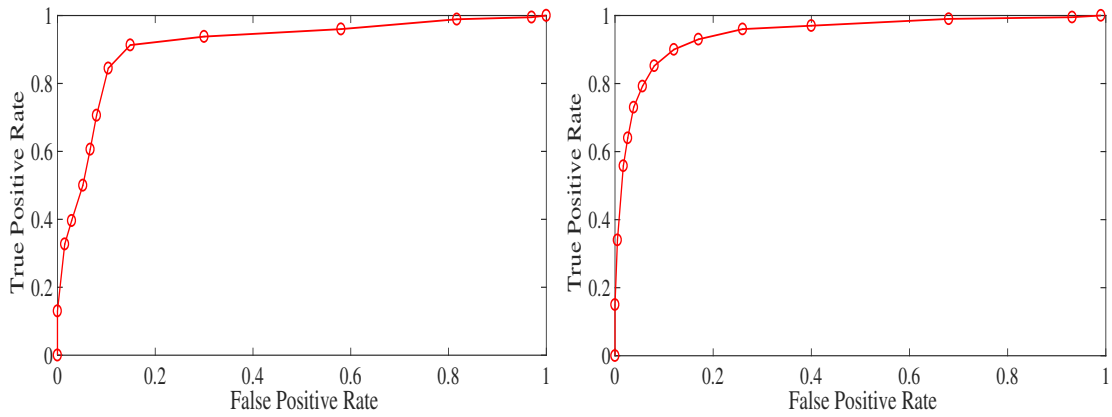
For Texas data, we found the following (near) optimal parameter values: $\nu = 0.05, B_b = 0.1, q = 0.55, sw = 100, A_b = 0.3$. To cross-check for parameter values for a different dataset, we repeated this process for over the Irish dataset. The first 7 months of

the dataset were used as training set, and attack labels were introduced after the end of the 3rd month, using the same attack features as the Texas dataset. We solved the parameters separately and found $\nu = 0.03, B_b = 0.12, q = 0.5, sw = 100, A_b = 0.3, F = 8$ and window length is 15 days. We can observe that ν and B_b are slightly different (due to dataset specifics), while other parameters are closer due to their relationship with attack model and underlying theory.

5.3.4. Threshold Selection. Cross-validation ascertains whether the optimal values generalize well or not to maximize the linear separation of scores, and also learn a classification threshold that generalizes during the testing set. We use a Receiver Operating Characteristics (ROC) curve to get the full spectrum of possibilities of false alarm (FA) to true positive (TP) rates. From this, based on the defender's desirable maximum tolerable false alarm rate, the corresponding threshold giving that FA rate is chosen, and then applied to the testing set for security performance evaluation.

Cross-validation Dataset: For Texas Dataset, we used 2015, partitioned into 12 partitions for cross-validation. For Irish dataset, we used 6 partitions, starting from the 8-th month of 2009. We average the parameter outputs to provide more accurate estimate of model prediction performance. For Texas dataset, we got: $\nu = 0.04, B_b = 0.08, q = 0.55, sw = 100, A_b = 0.29$, while For Irish dataset, we got $\nu = 0.03, B_b = 0.1, q = 0.5, sw = 100, A_b = 0.31$. We used these values to retrofit in the model and generated the diversity index scores of both classes. Then thresholds are varied according to desired false alarm rate.

ROC Curve: Figures 5.3(a) and 5.3(b), shows the ROC curve under a $\delta_{avg} = 100$ from cross-validation, with an AUC of 0.89 and 0.93 respectively. In general, the ROC curves for various δ_{avg} can be plotted. A utility can use his desired maximum allowable false alarm rate and find the corresponding threshold using this ROC.



(a) (b)

Figure 5.3. ROC in Cross-validation: (a) Texas (b) Irish

5.4. EXPERIMENTAL RESULTS

This section includes cross-validation and testing set results of both Texas and Irish datasets for the smart metering application. The experimental result section is divided into the following subsections: (i) Attack Implementation on Test set description; (ii) Performance results (iii) Cost Benefit Analysis (iv) Comparison with other works

5.4.1. Attack Implementation on Testing Set. For each attack type, and strategy (discussed in the threat model) we did the following: For the Texas dataset, the 2016 year's data (having a duration of a year), we had five attack start points interspersed approximately by two months to cover the entire testset duration. Similarly, for the Irish dataset, the final five months of the 2010 data were used as a test set, with two attack start points interspersed in a two-month duration. This is done to show that regardless of the start point of attack, the reported missed detection is unbiased. Hence, five (or two) versions of the attacked testing set are obtained for each attack type for Texas (and Irish) datasets respectively.

In each version, we had six different sets of compromised meters per attack scale value (to remove compromised meter selection bias), making a total of 30 (or 12) versions. Each such version is attacked with the indicated several different δ_{avg} (from the compromised ones of course), and then fed to the diversity index model. Then, the final result on missed detection and false alarms is reported by combining the results from all these versions.

For reporting baseline false alarm rate (where there are no attacks throughout the year or test duration), we counted the false alarms accordingly. Additionally, note that we have parameterized the space of attack strengths and scales covering all possible values. There is no availability of real attack dataset in this area, but our implementation included the gold standards for performance evaluation covering any gaps that might otherwise exist. Note that deductive, alternating switching attacks attack types, KLD minimizing strategies were not used for training. We put these in test set only to understand whether the method generalizes to previously unseen attacks.

5.4.2. Performance Results. Instead of ROC curve, we show (i) missed detection (MD) rates across a wide δ_{avg} range, for different thresholds based on user's tolerable FA rate; (ii) the *base rate FA*, which is false alarm rate, when there are no attacks throughout the test set; because most companies have a concern on lowering FA rates (because the prior probability of an actual attack is low). The ROC curve from cross-validation, is used to pick four corresponding thresholds that gave 2%,5%,8% and 10% FA rate; which are then applied to the test set.

5.4.2.1. Generalizing against untrained attacks. We first show performance under previously unseen attack types (deductive and alternating switching) across varying δ_{avg} values and the new ρ_{mal} = 40%; threats which did not feature in the training phase, using a data order aware strategy.

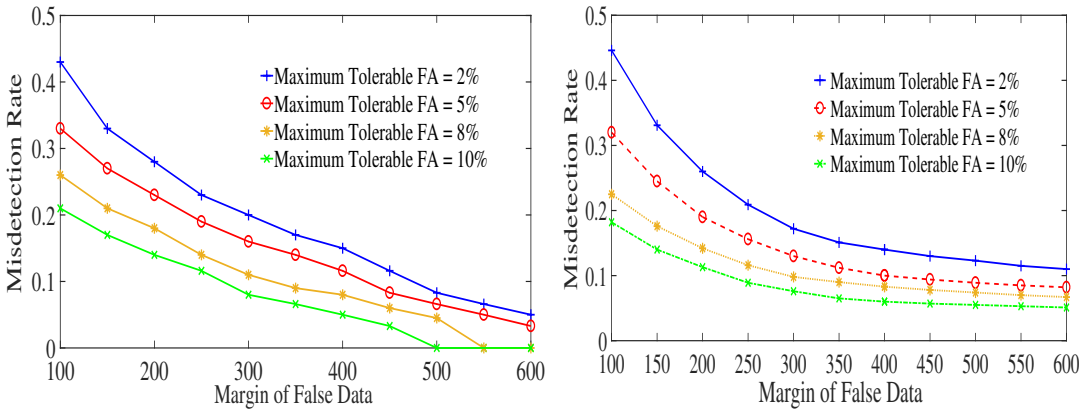


Figure 5.4. Deductive Attacks MD rates (a) Texas Data (b) Irish Data

Figures 5.4(a) and 5.4(b) show the MD rates across various δ_{avg} against 'deductive attacks' under the Texas and Irish datasets respectively. Each line corresponds to a performance given by different thresholds corresponding to that particular tolerable base FA rate. Similarly, Figures 5.5(a) and 5.5(b) show the MD rate for alternating switching attacks for Irish and Texas datasets respectively.

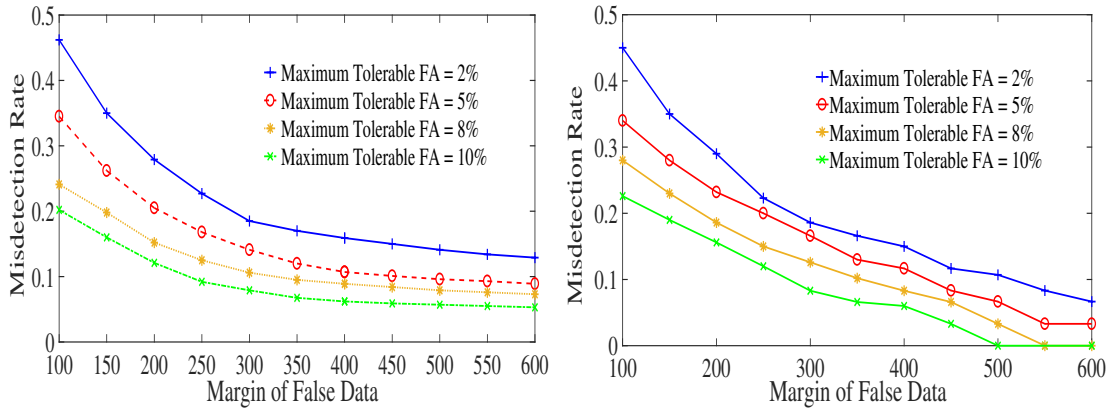


Figure 5.5. Alternating Switching MD rates (a) Irish Data (b) Texas Data

Performance against untrained KLD minimization strategy Figure5.6(a) at tolerable FA rate of 10%, is shown for the 3 attack types. The performance is slightly worse compared to the data order aware strategy. The increase in mis-detection rate on average for the KLD minimizing strategy across all attack types and δ_{avg} values, is 7.3% keeping the same FA rate. The Figure 7.6(b) shows that our method scales well and is invariant to changing ρ_{mal}.

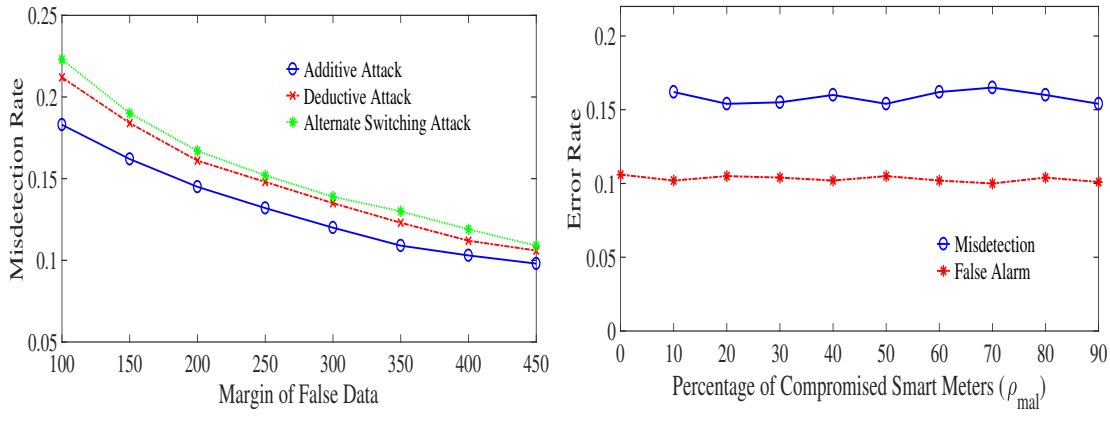


Figure 5.6. Performance: (a) KLD Minimizing Strategy (b) Invariance to Attack Scales

5.4.2.2. False alarm performance. A concern on anomaly based scoring frameworks are false alarms and their costs. A summary of *base rate false alarm performance in the testset* is included in Table 5.1. The Texas dataset has more shifting trends (due to renewable penetration), thus it has more base rate FA than Irish dataset.

Table 5.1. Base Rate False Alarm Percentages in test set

Tolerable FA Threshold	Irish Test Set FA	Texas Test Set FA
2%	2.11%	2.60%
5%	5.33%	6.25%
8%	8.86%	9.37%
10%	10.58%	10.93%

5.4.3. Cost Benefit Usability of our Performance. Here we analyze the *costs of MD and FA rates* from the perspective of real life usability. Once inferred as attacked, an audit trail is done by utilities on each device for confirmation. According to [71], audit inspections are billed for a median cost of CA = \$141 per device, while [72] reported the average time to inspect each meter device is 55-65 minutes. Audits are an annual affair in many companies and our test set is also for one year. There are two options for audit for a utility: (1) a utility wide audit (expensive), (2) an audit on those devices detected as positive (less costly). Let different utilities have different tolerable false alarm that vary between 2% to 10%. There is a loss due to audit on false alarms but a gain for detecting compromised meters successfully. We consider here only the monetary value per Kilo Watts hour (KWH) of electricity that is falsified. The effective profit/loss per year can be calculated as:

$$NProfit = \frac{\delta_{avg} \times \eta \times E \times 365}{1000} \times (M - md) \tag{5.11}$$

where M is the number of meters compromised, md is number of missed detections, η is the number of reports/day, E = \$0.12 per KWH is average cost of electricity in USA (could be as high as \$0.38 in some states). On the other hand, the cost of false alarms per year is: $L = CA * fa$ and $NetBenefit = NProfit - L$ where CA is the cost of audit/meter, and fa is the number of false alarms.

In Table 5.2, we provide the practical implication of our performance under user tolerable false alarm rates of 2% and 10%, with ρ_{mal} = 40%, in terms of monetary benefit. Given the numbers, a 2% tolerable FA is more profitable for Irish data, while 10% tolerable FA is more profitable for Texas data, for the same δ_{avg}. Since the difference in losses is not drastic, our recommendation for utilities is to choose 10% tolerable rate, since it will give much lower MD when attack actually occurs. Since the Irish data has a large micro-grid, the benefit is large, underscoring that the benefit is scalable.

Table 5.2. Profit/Loss Per Year with our Framework

Tolerable FA Threshold	δ_{avg}	$NetBenefit$: **Irish**	$NetBenefit$: **Texas**
2%	100	+ 21,219.12	+ 4,868.88
10%	100	+ 16,922.34	+ 5,597.16
2%	400	+ 141,686.64	+ 30,413.04
10%	400	+ 138,441.06	+ 32,087.47

5.4.4. Comparison with Previous Research. We compare our performance with 3 categories of existing methods: (i) classical ML, (ii) information-theoretic, (iii) statistical learning. Classical ML uses SVMs [12], decision/regression trees (DRT) [36]. The [12] outperforms [36], hence we compare our work with [12]. For information theoretic approaches [33, 40], we chose to compare with [33] (though mainly it showed the Texas data results) since it reports for various δ_{avg} unlike [40]. Statistical learning based method [35] outperform [14, 15, 41] and hence is chosen for comparison. The Figure 5.7, shows a comparison of our method with existing works under our threat model (assuming deductive attacks over Irish dataset since its common to all previous works). We can observe that the MD rate of our method (blue- solid line) is much lower compared to other works especially for lower δ_{avg}, with a threshold corresponding to 10% acceptable FA. This is fair comparison since Refs [12, 33, 35] have FA rates $\geq$ 10%, even for attacks with $\delta_{avg} > 350$W.

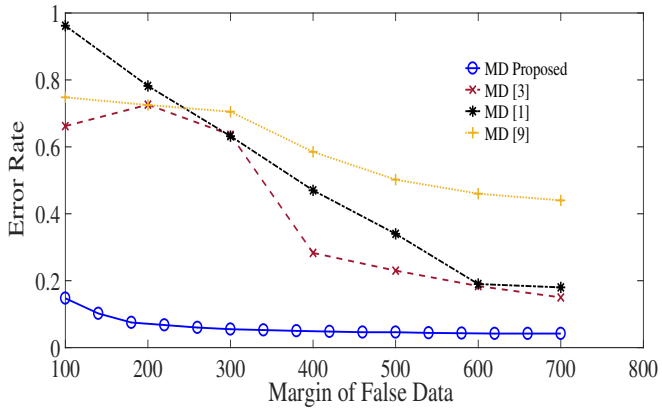


Figure 5.7. Performance Comparison with Existing Research

5.5. ESTIMATION OF DIVERSITY INDEX

Diversity Index is a score based on the difference in probabilities of historical (or prior to attack) and current data across the species. It is dependent on the horizontal and vertical change of species probability. The main attributes that result in the change of those probabilities is Species Width (SW) and margin of false data (δ_{avg}). As we are using a default species width (100), the margin of false data will be the major factor that determines the diversity index of a smart meter. Now, we will try to derive a mathematical relation between the Diversity Index score and δ_{avg}.

We have considered some assumptions for these estimation.

1. Attack is continuous across the frame.
2. The difference between dmin and dmax is low (100 or below).

From the derivation of diversity index, the first step is the calculation of RXR diagonal matrix D(f). The diagonal values are the difference of probabilities of respective species from historical (p_i) and current data (p_i^*) where $i \in [1, R]$.

The next steps to calculate the diversity index are based on the species probabilities before attack. So, there is no impact of margin of false data on diversity index score after the calculation of D(f). This means that we need to estimate the RXR matrix D(F). We

already have p_i, $i \in [1, R]$ which is based on existing data. This brings us to the species probabilities after the attack p_i^*, $i \in [1, R]$. Now we need to estimate the values of p_i^*, $i \in [1, R]$ for a given δ_{avg}.

It is observed that species concentration change with introduction of attack. If we can estimate the change of concentration of species, we can also estimate the diversity index score. Assuming uniform distribution of δ_t value between δ_{min} and δ_{max}, we will have the following δ_{avg}.

$$\delta_{avg} = \frac{\delta_{max} - \delta_{min}}{2} \tag{5.12}$$

Now, having the knowledge of the species width (SW), we can estimation the average shift in the concentration of the species. The average shift is represented by SH and can be calculated as shown in Eqn. 5.13.

$$SH = \frac{\delta_{avg}}{SW} \tag{5.13}$$

We need to extract the integer SH_i and decimal part SH_d of the shift (SH) to calculate the estimated shift in the number of readings across different species. The two values will be calculated from values from the shift value SH as shown in Eqn. 5.14.

$$SH_i = \text{floor}(SH) \qquad SH_d = SH - \text{floor}(SH) \tag{5.14}$$

Once we have SH_i and SH_d, we can estimate the probabilities of species p_i^* after the injection of attack based on the values of p_i is shown in Eqn. 5.5.

$$\mathrm{p}_i^* = \begin{cases} 0 & \text{if } i <= SH_i; \\ (1 - SH_d) \times p_{(i-SH_i)} + SH_d \times p_{(i-SH_i+1)} & \text{if } i > SH_i \end{cases}$$

5.6. INFERENCES

We offered a novel information-theoretic anomaly scoring technique that showed successful detection of smart meters launching data falsification with very low to high attack strengths and attack scales are possible, using AMI as proof of concept. The proposed method's accuracy generalizes well across two different datasets, with completely different years of data collection, countries, sizes of micro-grids. The conclusion is that the method is a way of inferring security status in terms of data integrity where inherent variances are higher than impactful attack strengths. Additionally, we conclude that for a cognizant attacker, the undetectable strategy space in smart energy AMI is reduced from what was achieved by previous works, without a drastic increase in false alarms.

6. DETECTION OF EVASION ATTACKS IN SMART GRID

6.1. BACKGROUND

In previous sections, we have proposed attack detection models for different types of attacks in smart grid. The knowledge of the model to the attacker can lead to several problems. One problem is that the adversary can launch attacks that will not be detected by the proposed defence models. This process is called Adversarial machine Learning(AML).

6.2. CONTRIBUTIONS

We propose a Generative Adversarial network (GAN) based solution to detect and eliminate evasion attacks in smart grid. This helps to avoid the usage of training data that could be anomalous either due to external intrusion or internal data management error. By filtering out the possibly anomalous training data, we make sure the Machine Learning (ML) model is more robust and devoid of evasion attacks.

This work mainly focuses on providing a solution to deal with evasion attacks and validate the solution with the existing smart grid security models [33, 35]. First, we discuss about different types of evasion attacks and how they impact the performance of the attack detection ML models. The impact of evasion will be shown using the Gaussian Trust model [35], and [33] which is based on Kullback-Leibler Divergence by comparing the performance of the model with and without the evasion attack. Next, we propose a GAN based solution on how to deal with the evasion attacks. A GAN has two important parts, generator and discriminator. The generator keeps generating evasion data samples which will be used to train the discriminator. This iterative process will lead to the improvement of discriminator in detecting the presence of evasion attacks. The resulting discriminator

can be used as a filter for the security model to avoid evasive data. Finally, We validate the proposed solution with multiple real datasets, demonstrating its generalization across very low to high attack strengths, scales, types, and strategies.

Section 6.3 introduces different types of evasion attacks and shows their impact on detection models for data falsification in smart grid. Section 6.4 describe the Diversity Index model system that will be used to validate the proposed solution along with the datasets used. Section 6.5 presents the GAN based solution to detect these poisoning attacks. Section 6.6 describes experimental results for two security models for both Texas and Irish datasets.

6.3. IMPACT OF EVASION ATTACKS

In this section, we elaborate on two types of evasion attacks called standard evasion attacks and smart evasion attacks. The purpose of evasion attack is to escape the detection model either by being stealthy or creating adversarial examples that meets the target of the attack. These changes will be made to be very small where the performance of the security model drops in case of standard evasion attacks where as the attack margin will be high and changes will be made with the knowledge of the model in case of smart evasion attacks. The standard evasion attacks can be capable of escaping the detection but the adversary cannot achieve the desired margin of false data that can guarantee profit. The smart evasion attacks when done appropriately can escape the detection model while guaranteeing the targeted margin of false data.

6.3.1. Random Evasion Attacks. In the standard evasion attacks, the adversary will manipulate the test data by introducing noise in the training data randomly. When the machine learning model uses this evasion data, its performance will become inconsistent. The standard poisoning of the data will be done by selecting the standard readings from the training set and modifying the true values. The modification will be made in a way that the value stays in the minimum and maximum readings from the selected data. Consider

D_{min} and D_{max} will be the minimum and maximum meter readings respectively. Let D_i and D'_i be the true and evasion readings respectively. The poisoning is done over a selected data points. If more data points are modified, the poisoning could be easily detectable by observing the data distribution.

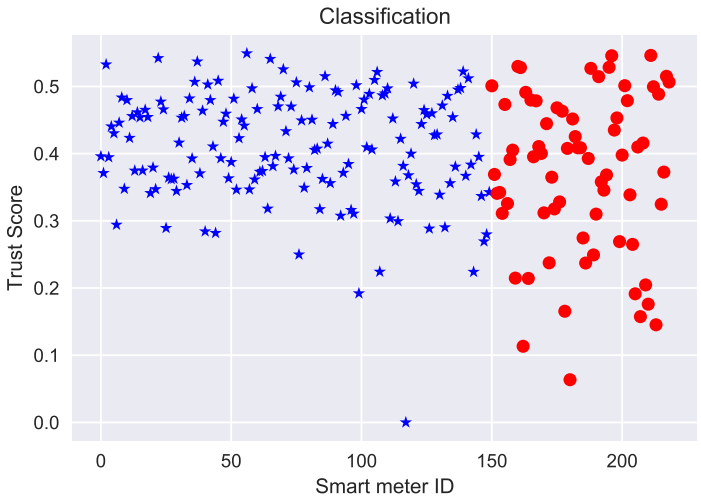


Figure 6.1. Standard Evasion Attack at δ_{avg} = 300

Once the standard poisoning is introduced into the training set, the machine learning model will not work desirably. One possible solution to handle the standard poisoning attacks is to introduce the robust statistics (like Median Absolute Deviation, Trimmed Mean) into the machine learning model. The purpose of robust statistics is that they will work with wide range of probability distributions and also resilient to data outliers. The elimination of outliers helps to eliminate some of the noise and thereby reduces the impact of poisoning. This solution is particularly helpful when the attack is black-box. It is very challenging to poison the data effectively in balck-box attacks. But, in case of white-box attacks, the data could be poisoned intelligently to impact the performance of the ML model. Using the robust statistics alone cannot prevent the impact of poisoning in white-box attacks. we refer to these attacks as smart poisoning attacks.

6.3.2. Smart Evasion Attacks. Smart evasion attacks does not go for lower margins of false data to escape detection. For an attacker to introduce smart evasion attacks, he should have the knowledge of the machine learning model. These are classified as white-box attacks as explained in the introduction. In such cases, the attacker can carefully make changes to the data in a way to escape the detection. This enables the attacker to achieve higher margins of false data which could increase the profit of the attack.

The best way to deal with smart evasion attacks is to detect them and avoiding them. In this paper, we are using Generative Adversarial Networks (GANs) as the tool to detect the smart evasion attacks. A GAN mainly consists of two things, Generator and Discriminator. The generator creates different possible evasion samples or adversarial examples. These evasive data samples will be used for training the discriminator along with the true samples that we have from the historical data. Once the discriminator is well trained, it should be able to classify whether the given data is an evasion sample or not. The working of a GAN can be seen in the Figure 6.2.

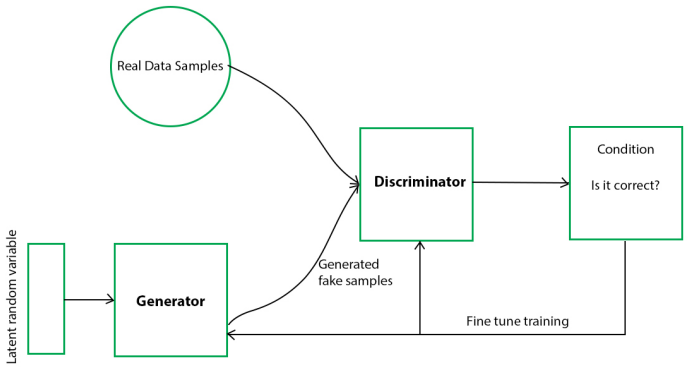


Figure 6.2. Generative Adversarial Network

6.4. GENERATOR MODEL

Generative models concern with how data is generated given a classification model that produces an output. We observed that the [35] uses a discriminative approach. First, we will discuss how we can generate the evasion data samples or adversarial examples using the generative approach.

6.4.1. Overview of Solution. From the folded gaussian model, we know the trust score of a smart meter will be higher when as more and more smart meter readings fall in the first standard deviation. So, with this knowledge of the adversary, the attack can be injected in a way to keep certain changed readings closer to the temporal mean, which boosts the trust score and potentially lead to misclassification, while preserving the δ_{avg} impact constraint. Interestingly, there is a design similarity of this approach with DBSCAN and KL distance trust, although the actual mathematics of each approach is very different. The similarity is in the discretization of the data into levels and using the probability density of each level to combine it into a clustering approach. This is what our adversarial method seeks to harness and gives the power of transferability. While it may not completely evade the classifier, increase in the trust score will degrade the meter detection success.

6.4.2. Generating Evasion Data using Generator. In this step, generation of the evasion data is shown from the true data samples. The input $\boldsymbol{P}$ is the true electricity readings of a smart meters across T time slots in a frame. The output of generator Q, will be the evasion sample with same size which the adversary needs to generate to escape detection.

$$\boldsymbol{P} = \left[P_1, ..P_t, ..., P_T\right]_{1\times T} \quad \boldsymbol{Q} = \left[P_1^{'}, ..P_t^{'}, ..., P_T^{'}\right]_{1\times T} \tag{6.1}$$

The generative model's design will depends on the architecture of the defense model. A close look into the folded Gaussian trust model reveals that the smart meter is classified honest, when it has a relatively high trust score. So, the generative model needs to create falsified data per smart meter $\boldsymbol{Q}$ such that it results in a higher trust score even in the presence of an attack.

To accomplish the above, the generator has to select appropriate instantaneous δ_t (Eqn.6.2) perturbations over a time frame that results in highest trust score possible, by still preserving the strategic target δ^f_{avg} of the adversary, required to inflict the targeted damage.

$$\boldsymbol{F} = \Big[\delta_1, \delta_2, ..., \delta_t, ..., \delta_T\Big]_{1\times T} \tag{6.2}$$

$$P^{'}_t = P_t \pm \delta_t \qquad \delta^f_{avg} = \frac{\sum_{t=1}^{T} \delta_t}{T} \tag{6.3}$$

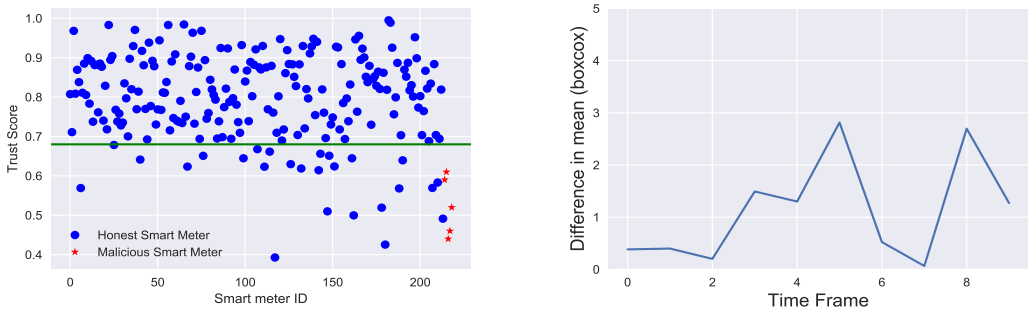
The working logic of the scoring model shows that when a data point is within first standard deviation on either side of the mean (rating level 4), it contributes to a higher trust because the weight of such an observation is proportional to the probability density of observing the level 4, which is the highest in the benign dataset. The rating levels 3,2,1 which indicate increasing distance of the data points from the mean, contributes less to the trust score, due to the same proportionality feature, because the probability densities in the benign dataset for levels 3,2,1 are much lower.

This gives an intuition that, if the data points stay closer to the mean even after false data injection, it should lead to higher trust score. To do this, the adversary needs to find the best values for X, Y, Z and A from Table 7.1 which will be the number of readings in each discrete rating level in the time frame under the crafted evasion attack. The discrete levels depend on the mean and standard deviation. This requires the adversary to estimate the values of mean, standard deviation after the attack. The threshold for classification decides the malicious smart meters. So, it also needs to be estimated to evade detection.

Estimating Safe Threshold (TH): The smart meters with trust score higher than the threshold will be classified as honest. In the [35], the threshold was generated through a k-means which depends on the final distribution of the trust scores and attack incidence flag generated by the anomaly detector.

Since, the adversary may not be sure of the threshold for classification, the threshold needs to be estimated to escape the detection. This threshold can be estimated by observing the trust scores produced by inputting honest smart meter readings from historical data.

The threshold is selected using k-means and it will be highest in case of less malicious meters. As the number of malicious meters increase, it creates more lower trust scores in the final input to the k-means, and therefore, the final threshold starts to decrease. So, the safe threshold will be selected by applying the Gaussian trust classification using very low value of M. This is shown in Figure 6.3(a).



(a) (b)

Figure 6.3. Texas Data (a) Safe Threshold (TH) (b) Difference in mean

Estimating Mean (μ): The exact trust score estimation for Gaussian trust model needs the knowledge of mean of current data. This is unknown to the adversary as the mean value is based on current time slot and over all smart meters. Let μ_t denote the arithmetic mean at time t after attack. Since μ_t is unknown to the adversary, it needs to be estimated using the knowledge of the data before the attack. The difference in mean over two consecutive time frames is very low and can be seen in boxcox scale from figure 6.3(b).The value of mean from the previous time frame before the attack at the exact time slot t will be μ_{t-T}. Given, $\rho_{mal} = M/N$ is the fraction of compromised meters and δ^f_{avg} is the targeted margin of false data, the estimated arithmetic mean for an additive attack is shown in Eqn. 6.4.

$$\mu_t = \mu_{t-T} + \left(\rho_{mal} * \delta^f_{avg}\right) \tag{6.4}$$

Estimating Standard Deviation (std): Estimation of the exact standard deviation after attack at run time is impossible to know before the attack, given a knowledge of a subset of meters ρ_{mal} and the δ_{avg}.

However, studying the dataset we found that the standard deviation is cyclostationary. So will take the standard deviation after attack, σ_t same as the standard deviation from the previous time frame at same time slot, σ_{t-T}. This leads to some data points in higher discrete level to come down to lower discrete levels after the attack leading to higher than estimated trust score.

Estimating δ_{avg} per Each Discrete Level: Once we estimate the mean and standard deviation given the δ^f_{avg}, we have to estimate the average possible margin of false data at each discrete level. P_{t-T} will be the smart meter reading at time t from the time frame before the detection of attack. The maximum margin of false data for P_{t-T} in each discrete level for an additive attack is shown in Eq. (6.5). This will be calculated for all T readings in the time frame.

$$\begin{aligned} \delta^X_h &= \mu_{t-T} + \sigma_{t-T} - P_{t-T} \qquad & \delta^Y_h &= \mu_{t-T} + 2\sigma_{t-T} - P_{t-T} \\ \delta^Z_h &= \mu_{t-T} + 3\sigma_{t-T} - P_{t-T} \qquad & \delta^A_h &> \mu_{t-T} + 3\sigma_{t-T} - P_{t-T} \end{aligned} \tag{6.5}$$

Using the historical data, the margin of false data per each reading in a discrete level is calculated over the same time frame from the previous year. Considering the number of readings in each discrete level over history as $X_{hist}, Y_{hist}, Z_{hist}, A_{hist}$ we can calculate the average margin of false data in each discrete rating level using Eq. (6.6).

$$\begin{aligned} \delta^X_{avg} &= \frac{\sum_{h=1}^{X_{hist}} \delta^X_h}{X_{hist}} \qquad \delta^Y_{avg} = \frac{\sum_{h=1}^{Y_{hist}} \delta^Y_h}{Y_{hist}} \\ \delta^Z_{avg} &= \frac{\sum_{h=1}^{Z_{hist}} \delta^Z_h}{Z_{hist}} \qquad \delta^A_{avg} = \frac{\sum_{h=1}^{A_{hist}} \delta^A_h}{A_{hist}} \end{aligned} \tag{6.6}$$

Finding Optimal Parameters for Evasion: The trust score can be reformulated as Eqn. 6.7 by combining the Eqs. (7.4), (7.5), and (7.6), where $W(l) = w \times l$. The value of w for each discrete level is extracted using historical data. To create an optimal evasion attack, the trust score (TR) should be just above the threshold (TH) separating the honest and malicious smart meters. At the same time, the readings should meet the targeted margin of false data. To generate the evasion data, we have to estimate the number of values in each discrete rating level that can guarantee evasion and δ^f_{avg}. For this, we have to solve the optimization problem in Eqn. 6.8 to find the best values for X, Y, Z and A.

$$TR = \frac{1}{(K)^\eta}(X\ W(4) + Y\ W(3) + Z\ W(2) + A\ W(1))^\eta \tag{6.7}$$

$$\begin{aligned} \min \quad & (TR - TH) \\ \text{s.t.} \quad & \frac{X\delta^X_{avg} + Y\delta^Y_{avg} + Z\delta^Z_{avg} + A\delta^A_{avg}}{T} = \delta^f_{avg}, \\ & TR \geq TH, \\ & X + Y + Z + A = T, \\ & X, Y, Z, A > 0 \end{aligned} \tag{6.8}$$

The second constraint shows that best possible value for the trust score is nearly equal to the threshold. The third constraint allows to reduce the problem from 4 unknown variables to three unknown variables by replacing A with $T - X - Y - Z$.

The optimization problem has 3 unknown variables and can be solved using linear programming as all the constraints are linear. We used the simplex method to solve the formulated optimization. Upon solving the optimization problem defined above, we get the

values of X,Y,Z and A. Now in this step we will generate the evasion data Q. Using the estimated values μ_t, σ_t and known true reading P_t, the δ_t values will be calculated similar to Eqn. 6.5 for all T time windows. Then, we finally create the evasion data using Eqn. 6.3 from the δ_t values.

The whole process is shown considering an additive attack. For deductive, camouflage and alternate switching attacks, the only difference will be in the estimation of mean and Eqn.6.5. The rest of the process will be the same.

6.5. DISCRIMINATOR MODEL

The purpose of discriminator is to detect the evasive data which could pass though the seurity model. Discriminator uses a neural network that takes the test data as input and gives a result between 0 and 1. If the output is closer to 1 means that the data might be modified using evasive strategy. The input of the neural network is a set of $N \times T$ data points of N smart meters across T time slots. The discriminator checks each smart meter for detecting the presence of evasive attack. For this purpose, we need to extract features from the T data points of each meter into a much smaller set which retains the important characteristics of the data that enables to detect the evasion attacks.

First, we will compress the size of the input which is N data points into R data points ($R << N$). T decision making regarding the poisoning of the data. There are many ways to compress the size of set of values like down-sampling but these methods will only take one points from set of points. Here we are trying to extract in a way that each output point is dependent on the set of all the points it got extracted from and at the same time holds the characteristics to detect the poisoning of the data. There are again different possibilities like mean, min and max for compression. The compression we are taking is in terms of smart meter readings. We will take all the N meter readings and arrange them in a sorted order and divide them into R bins each of size SW where $SW = N/R$. Now, we need to extract a feature from each of the R bins that defines all the meter readings in each bin. The diversity

index model, says that probability of a meter reading in a given range can help detect data falsification but for that we need to select an optimal value of SW. This is because if the value of SW is too small or too large, it will result in over-fitting or under-fitting. To find the optimal SW, we have defined an objective function that gives the difference of diversity index scores across compromised and honest smart meters. Taking all the variables as constant for the calculation of diversity index score by just varying the value of SW, we have calculated the objective function. The optimal value of SW is where the value of e is maximum. The Figure 6.4 shows the optimal SW value for the texas data.

$$e = \left(\frac{\sum(rD^h)}{N-M} - \frac{\sum(rD^m)}{M}\right)^2 \tag{6.9}$$

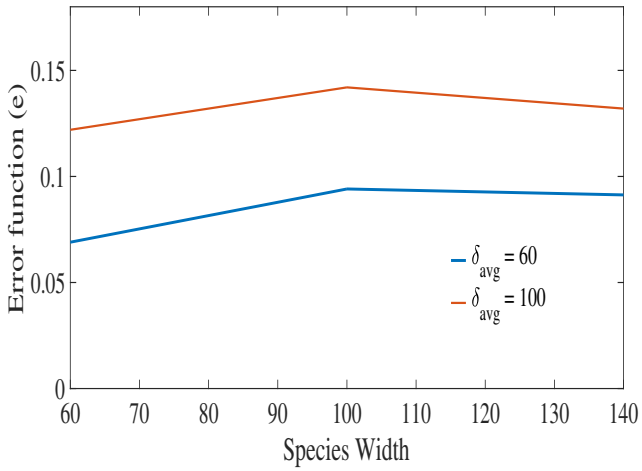


Figure 6.4. Optimal size of SW

Once we got the optimal SW, we can get the vector X of length R from the input of N smart meter readings. Mathematically, we use the Bayesian interpretation of posterior relative abundance to calculate each value of X_i. This is shown in Eqn. 6.10.

$$X_i = \frac{C_i + 1}{N + R} \tag{6.10}$$

where, C_i = number of readings in $[(i-1) * SW, i * SW)$

Once, we have the extracted feature vector $[X_1, X_2, ..., X_i, ..., X_R]$, it is provided as input to the fully connected neural network. This will result in a binary classification based on single output value y from the neural network. The neural network is shown in Figure 6.5. If the value of y is closer to 1, it shows poisoning and if it is closer to 0, it shows no poisoning. The neural network will be trained using the existing data and evasion samples generated using the generator. The weights of the neural network will be updated using back-propagation. For this we are using a loss function as shown in Eqn. 6.11. We are using the sigmoid activation function shown in Eqn. 6.12 for building the discriminator neural network. Once, this neural network goes through a lot of training samples, the weights of the neurons will get updated to detect the evasion attacks.

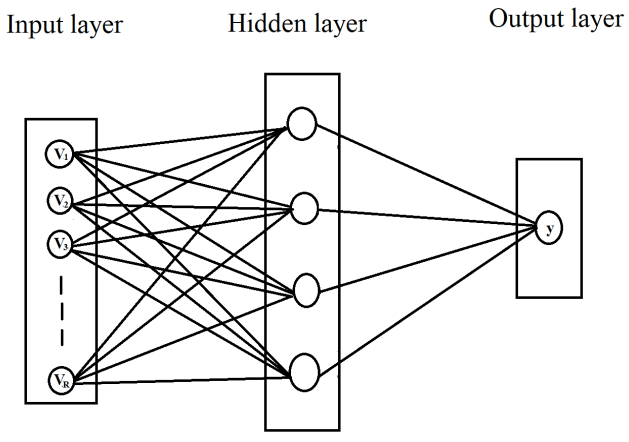


Figure 6.5. Discriminator Neural Network

$$\phi(z) = \frac{1}{1 + e^{-x}} \tag{6.11}$$

$$\text{Discriminator loss} = r * (1 - y) + (1 - r) * y$$

$$\text{where, } r = \begin{cases} 1, & \text{for poisoned data} \\ 0, & \text{otherwise} \end{cases} \tag{6.12}$$

$$\phi(z) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{6.13}$$

6.6. EXPERIMENTAL RESULTS

We have defined the procedure for the development of generator and discriminator in the previous section. Now, we will implement the generator and discriminator and analyze the performance of the system. First, we will show the results of the generator.

The purpose of generator is to create an attack that should be able to get through the security model. Now, we will show the performance of the generator by comparing the performance of the security model under attack with and without evasion strategy.

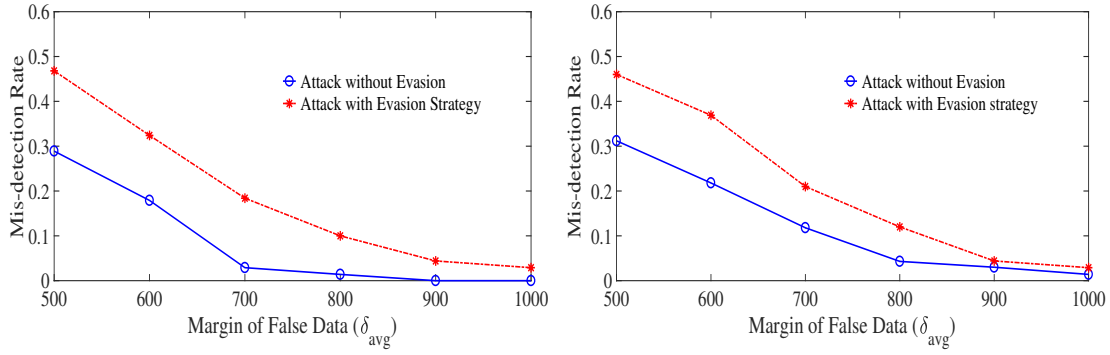
(a) (b)

Figure 6.6. Performance: (a) No Evasion (b) Evasion

The impact of evasion on a real dataset can be seen from Figure 6.6(b). The performance of classification is worse compared to Figure 6.6(a). This is because the data is generated in a way to evade the detection method and it is evident from the difference in performance.

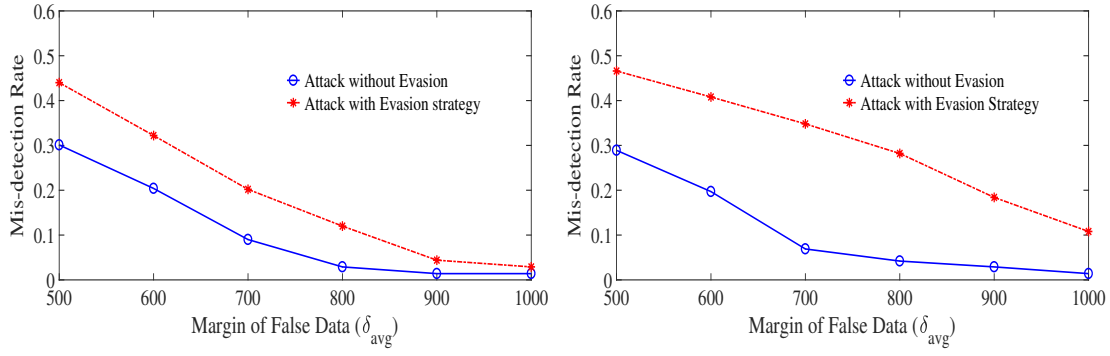
Now we investigate whether the success observed in Figure 6.6(b) generalizes across any margin of false data (δ_{avg}) and attack type. To assess this, we invoke our generative evasion strategy with different input values of δ_{avg}, and generate correspond evasion data. We repeat this for each attack type: additive, deductive, camouflage, and alternating switching.

Missed detection rates for additive and deductive attack across δ_{avg} with a ρ_{mal} of 0.3 is shown in Figures 6.7(a) and 6.7(a) respectively. Similarly, Figures 6.8(a) and 6.8(b) show the results for camouflage and alternate switching attack types respectively, under the same attack features. From observing the Figures 6.7(a), 6.7(a), 6.8(a), 6.8(b), the red lines (which correspond to the performance under our evasion strategy) are always showing a higher missed detection rate, than the blue line (attacks without our evasion strategy).



(a) (b)

Figure 6.7. Performance (a) Additive attack (b) Deductive attack



(a) (b)

Figure 6.8. Performance (a) Camouflage attack (b) Alternate Switching attack

The neural network defined in the discriminator part of the previous section has been implemented and trained using year 2015 of Texas data and evasion samples generated using the generator. Figure 6.9 shows the classification performance of discriminator in

detecting the smart meters that use evasion strategy. The result is shown for δ_{avg} of 500. The result shows that we are able to detect 96% of smart meters with just 1.5% False alarm rate. This is much higher than the trust model's 40% detection rate at δ_{avg} of 500.

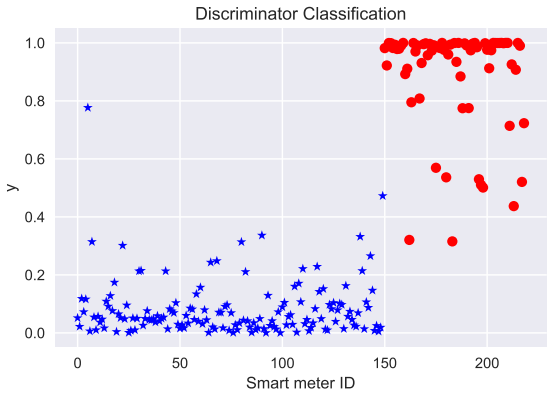


Figure 6.9. Performance of Discriminator for Texas data

6.7. INFERENCES

We have presented different types of evasion attacks in smart grid. Then, we have proposed solutions to handle those attacks. We have used Generative Adversarial Networks (GANs) based method to create the evasion attack samples using generator and detect the evasion tactics by training the discriminator with the generated adversarial examples. Finally, we have demonstrated the importance of the proposed solution by showing the performance of a machine learning model with evasion strategy. The final experimental results show that the discriminator can be used as a filter to detect the evasion attacks.

7. ACTIVE LEARNING BASED DETECTION OF SENSOR FAILURE AND CONGESTION IN REAL-TIME VEHICULAR NETWORKS

Smart transportation is an essential clog in the wheel that runs current and future smart cities. It uses 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. Figure 7.1 illustrates the basic architecture of road infrastructure in a smart transportation network [60].

The Traffic Message Channel (TMC) sensors are deployed on various road segments to capture the ambient speeds as vehicles pass by. Regardless of the type of IoT sensing end-point, multiple such sensors forward information to a Road Side Unit (RSU). The aggregated data from the RSU is used to analyze the state of the traffic in real-time and to provide improved traffic management decisions to the vehicles. Numerous RSUs are deployed to cover the smart city area. Apart from vehicle speeds, the RSUs also receive other information (e.g. time, location) and transmit them to an edge/fog computing module that implements the decision services (e.g., traffic information, selection of the fastest route). It also supports the smart transportation network with services such as 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 to make accurate decisions.

There are several scenarios that can produce erroneous data from TMCs. In case of incorrect reporting of vehicle information such as speed, location can result in incorrect interpretation of the traffic situation, which might lead to severe traffic jams. For exam-

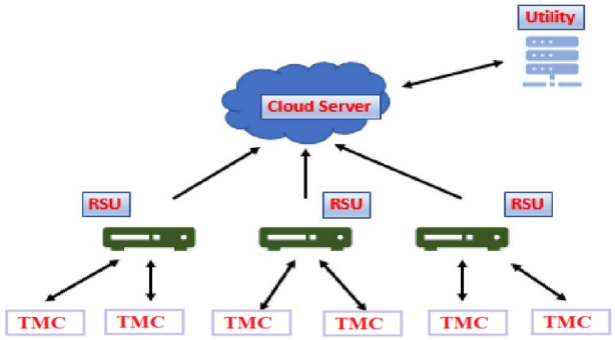


Figure 7.1. Architecture of a Road Infrastructure

ple, stuck value anomalies [73] where the sensor gets stuck at a particular sample value. Calibration errors in these sensors can also cause the reported data to be above or below the actual reading. Another common reason for faulty data is low battery. Similarly, some sensor errors can stop data collection altogether. Environmental disasters can also hinder the ability to supply accurate data from a large fraction of such IoT sensing devices. For a large community scale IoT infrastructure, we need a scalable and lightweight anomaly detection technique that can quickly detect these malfunctioning IoT sensors, such that the maintenance personnel can be dispatched to fix/replace them.

While several theories of anomaly detection [57] and consensus based trust scoring models [60] 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, [35] proposed a novel semi-supervised framework for identifying compromised IoT devices sending falsified data. However, it uses unsupervised classification and did not have a fixed threshold. Such methods are highly sensitive to outliers. Similarly, the supervised machine learning approaches such as decision trees, Support Vector Machine (SVM) require labeling of the complete training set. This puts a tremendous burden on the infrastructure and large scale com-

putations also increase the carbon footprint. Ideally, for community scale IoT, we need a device level anomaly detection classifier that does not need to learn and train over the whole training data from the full network.

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 trust scoring model is based on the [35] which is built for smart meters. The second part is the classification of anomalous TMCs. For this, we proposed an active learning 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 [63]. We show that the model is able to detect the traffic incidents and TMCs that are malfunctioning with an accuracy of more than 90%.

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. Active learning also reduces the cost of labeling needed for training the model compared to supervised clustering algorithms.

The rest of the paper is organized as follows. Section 7.1 defines the anomalies and their impacts. Section 7.2 presents the threat model and the trust scoring mechanism to detect the anomalies. Section 7.3 offers the experimental results.

7.1. 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.

The deviation from free flowing traffic may be due to an accident or congestion owing to malfunction of speed sensing. We consider these as anomalies. In this work, we propose a model to detect such anomalies at the TMC level in real-time to take necessary actions. Let's consider M TMCs record an anomaly of the total N TMCs. We define $M/N = \rho_{mal} \in [0, 1)$. For example, $\rho_{mal} = 0.05$, means 5% of the total number of TMCs have readings that deviate from the free-flow either due to heavy traffic, accident, or sensor failure. A sensor failure can result in following situations:

Stuck Value Anomaly: In case of sensor failure, the reporting value gets stuck at the value in which the sensor was last correctly working. This results in reporting of the same value which is not the true value.

Calibration Anomaly: If condensation builds up on the sensor, 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.

Depending on the average speed, the anomalies could be classified as deductive or additive based on the deviation from the free-flow. For example, for the deductive anomaly of a TMC, the actual speed of information S_t^i from the i-th TMC at time t will be lower than the free-flow situation. These anomalies are possible under congestion due to heavy traffic, accident, or sensor failure. 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.

7.2. PROPOSED APPROACH

The detection model consists two main steps. The first step is the scoring model. The second step is classification. In the scoring model, a trust score will be calculated depending on the vehicular readings of each TMC. In the second step, we use active learning model to classify the benign/non-anomalous TMCs from anomalous ones. Our method is divided into the following 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. The sub-modules 2-4 deal with the classification based on active learning.

7.2.1. Trust Scoring Model. The trust scoring model will be used to identify the 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 window T ($<$ 2 hours). The trust scoring model starts with the 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 window. 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 7.1, 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

Table 7.1. 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

further from the μ_H. Over the time window T, all the discrete ratings over time frame T for each TMC i is collected to form a rating vector sequence $\vec{r^i_{sort}}$ sorted in descending order of discrete ratings.

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 matters. It doesn't impact if the reading is greater or lesser than the mean. Intuitively, in case of any accident/congestion, TMCs will have more lower ratings leading to lesser weights and ultimately lower trust score. If the deviation is larger, it assigns a non-linearly decreasing density value based on the shape of the Gaussian distribution. We represent this as w_t. This is calculated from x_t and cw_t

$$x_t = 1 + \frac{(K-1)t}{(T-1)} \qquad \forall \quad t \in T \tag{7.1}$$

$$cw_t^i = \frac{1}{\sigma_{dr}^i \sqrt{2\pi}} e^{-\frac{(x_t - \mu_{BR})^2}{2(\sigma_{dr}^i)^2}} \tag{7.2}$$

$$w_t^i = \frac{cw_t^i}{\sum_{t=0}^{T-1} cw_t^i} \tag{7.3}$$

All the density values are combined to form a weight vector $\vec{W^i}$ for each TMC i as in Eqn. 7.4. 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 $\vec{r^i_{sort}}$ as shown in Eqn. 7.5.

$$\vec{W^i} = [w^i_1, w^i_2, ..., w^i_t, ...] \qquad \forall \quad t \in T \tag{7.4}$$

$$R^i = \vec{r^i_{sort}} \cdot \vec{W^i} \tag{7.5}$$

As the ratings will be positive regardless 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 Eq. 7.6. The value of K depends on the number of rating levels (4, in our case).

$$TR^i = \frac{1}{(K)^\eta}(R^i)^\eta \tag{7.6}$$

7.2.2. 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, 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 Figure 7.2). 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 Figure 7.2) Q_a. The verification is allowed by a ground truth data set available from Nashville Police and Emergency Response Units [63].

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 Figure 7.2 making $|Q| = 20$ labels. For the rest of the TMCs (denoted by green marker in Figure 7.2), 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.

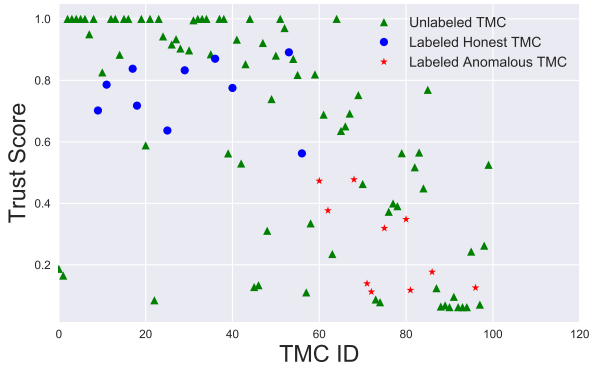


Figure 7.2. Initial Manual Labeling of few TMCs.

Now, we use the set Q, to calculate the 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 indicate that they are linearly separable.

7.2.3. 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 to have the probability of each TMC i belonging to anomalous class (P_a^i) and benign class (P_b^i). The probabilities of each class is based on the trust score (TR^i) of the TMC i and the $TH(j)$ is shown in Eqn. 7.7 and Eqn. 7.8. When a data point is equal to the threshold, the probability that data point belongs to either class is 0.5. As the data point gets farther from the threshold, the probability that data points belongs 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. 7.9. 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.

$$P_a^i(j) = \begin{cases} \frac{1}{2} - \frac{TR^i - TH(j)}{2\times(1-TH(j))}, & \text{If } TR^i > TH(j) \\ \frac{1}{2} + \frac{TH(j) - TR^i}{2\times TH(j)}, & \text{Otherwise} \end{cases} \tag{7.7}$$

$$P_b^i(j) = 1 - P_a^i(j) \tag{7.8}$$

$$LC^i(j) = 1 - \max(P_a^i(j), P_b^i(j)) \tag{7.9}$$

7.2.4. 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. 7.9.

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: $TH(j) = \text{SVM}(Z(j-1))$
9: $TH_{fnl} = TH(j)$

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 Q size is too small, it can result in under-fitting and a bigger size of set Q can result in over-fitting. So, we need to find 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. It is shown in Eqn. 7.10. The error function for different values of Q can be seen in Figure 7.3. From the result, we can say that the optimal size of Q is in range of 10-20.

$$E = \arg\min_{|Q|} \left(\sum LC^i(Q) \quad \forall i \in Y \right) \tag{7.10}$$

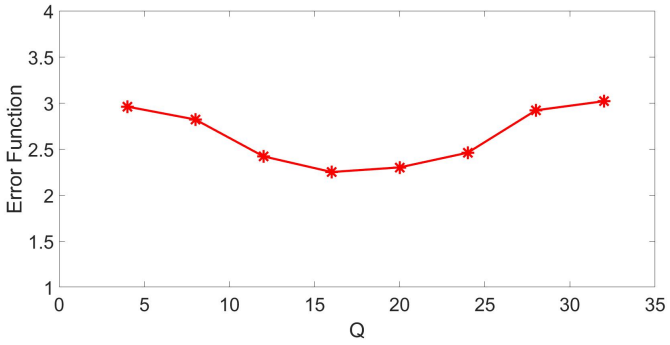


Figure 7.3. Error rate under different values of Q.

7.3. EXPERIMENTAL RESULTS

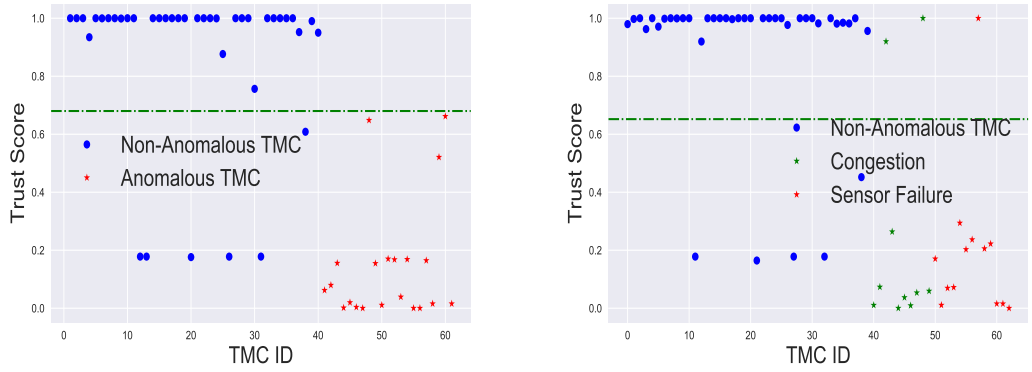
Description of Datasets: We have used vehicular dataset from Nashville, Tennessee to validate the proposed solution. Out of the 4 months of available data, We used the first two months (January, February) for training the model. March data is used for cross-validation and April data is used as the test set. The results from this section are considered from 60 TMCs belonging to a 10 different RSU clusters.

7.3.1. Trust Score Classification of TMCs. The trust scoring model is applied to the test set of the Nashville dataset. The active learning parameters we got 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. The ground truth for the accidents and congestion are also available to test the detection accuracy.

A higher trust score implies the TMC is under a normal behaviour. The lower trust score will be a result of either congestion or sensor failure. Congestion will always be a deductive anomaly. The sensor failure can result in either additive or deductive anomaly. The trust model generates a score for all the TMCs depending on the readings. We then

used the TH value from active learning for the classification. The Figure 7.4(b) shows the performance of trust scoring model in detecting the additive anomalies caused by the sensor malfunction.

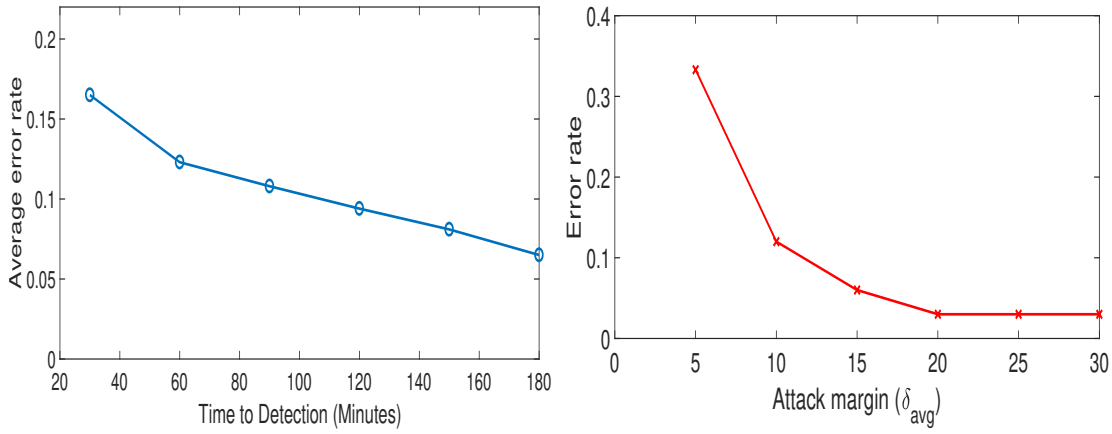
The two main possibilities for congestion are accident and heavy traffic. The congestion can be differentiated from the deductive anomaly due to sensor failures by analyzing the speed information of the vehicles. The congestion will result in speeds with close proximity from different vehicles in the selected time frame.So, the standard deviation will be lower in case of congestion. For a sensor failure, this cannot be guaranteed. Figure 7.4 shows the performance of the model under different deductive anomalies. The result also shows the cause of anomaly for the TMCs that registered lower trust scores whether it is congestion or sensor malfunction.



(a) (b)

Figure 7.4. Classification of Anomaly: (a) Additive (b) Deductive

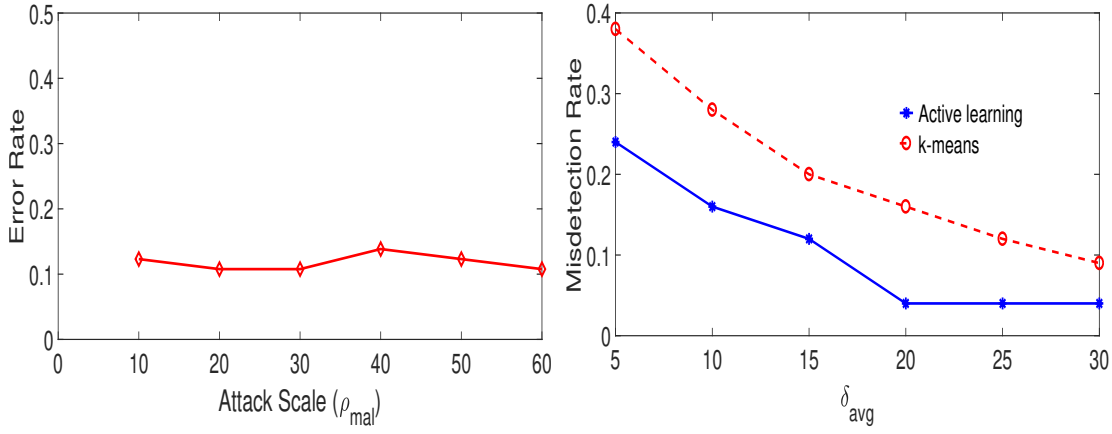
7.3.2. Performance Analysis. The time to detection of anomalies is a crucial factor in vehicular networks. The accidents and congestion should be detected immediately to warn the other vehicles to avoid the congested routes. The performance must be good at lower detection time for detecting the anomalies. Figure 7.5(b) shows the performance of the model with the detection time ranging from 5 minutes upto 1 hour. The result shows the proposed model is able to detect the congestion and accidents with an accuracy of over 85% in only 5 minutes.



(a) (b)

Figure 7.5. Performance (a) Time to detection (b) Type of failure

The active learning provides advantage of reduced labelling cost compared to supervised classification models. In comparison with unsupervised classification models, the performance should be better. The figure 7.6 shows the classification performance using both k-means and active learning.



(a) (b)

Figure 7.6. Classification performance (a) ρ_{mal} (b)Active Learning vs k-means.

7.4. INFERENCES

In this work, we have presented an anomaly detection model for the TMCs. The anomaly could be any abnormal traffic incident or due to TMC malfunction. We used the folded Gaussian trust scoring model to generate the trust score for each TMC depending on its measurements. We applied an active learning approach to identify the TMCs with anomalous behavior. This helps to classify any traffic incidents in near real-time as the proposed model is able to detect the anomalies within 10 minutes with good accuracy. In future we will extend the model to distinguish the reason for congestion. This would help the network to take the required safety measures accordingly.

8. ONGOING RESEARCH WORK

8.1. DETECTION OF POISONING ATTACKS IN SMART GRID

Adversarial machine learning (AML) is a technique that fools the machine learning models with the malicious input. The resulting performance of the machine learning models will not be good when the adversary employs AML. One of the common types of AML techniques is poisoning attacks. In poisoning attacks, the adversary will manipulate the training data on which the machine learning models rely on. The changes will be done in a way that leads to the bad performance of the machine learning model. In this paper, we showed two types of poisoning attacks called random poisoning attacks and smart poisoning attacks. We showed the impact of poisoning attacks on the machine learning model we have implemented to detect the false data in smart grid. Then, we have proposed a Generative Adversarial Network (GAN) based solution to detect different kinds of poisoning attacks. Our proposed solution is validated with the help of two real smart metering datasets from Texas and Ireland.

8.2. ANOMALY DETECTION IN DETECTION IN AUTOMATIC GENERATION CONTROL (AGC)

Automatic Generation Control (AGC) is a critical control function of the power grid. It controls the amount of power generation and maintains the balance between power generation and load distribution, which keeps frequency at the scheduled value (i.e. 60 Hz in the U.S.). AGC periodically receives information about the power system's frequency and tie-line power flow between neighboring balancing areas. Using the current models proposed for smart grid, we want to implement the anomaly detection models for AGC.

9. CONCLUSION AND FUTURE DIRECTIONS

In this dissertation we studied the security issues in smart grid and proposed trust scoring model to detect different kinds of attacks in smart grid. The proposed solution is not good under stealthy attacks in range of 100W. For this, offered a novel information-theoretic anomaly scoring technique that showed successful detection of smart meters launching data falsification with very low to high attack strengths and attack scales are possible, using AMI as proof of concept. The proposed method's accuracy generalizes well across two different datasets, with completely different years of data collection, countries, sizes of micro-grids. The conclusion is that the method is a way of inferring security status in terms of data integrity where inherent variances are higher than impactful attack strengths. Additionally, we conclude that for a cognizant attacker, the undetectable strategy space in smart energy AMI is reduced from what was achieved by previous works, without a drastic increase in false alarms. We have proposed some models to deal with evasion attacks. The detection of sensor failures and congestion in real-time transportation networks has been proposed and tested using real-world data from Nashville.

As part of future work, we will study how to strengthen the model under training data poisoning attacks and give theoretical estimations of expectation of change in diversity index score as a function of various attack parameters, and check on whether retraining over the untrained attacks improves missed detection performance. We also want to check if the proposed diversity index model generalizes for other parts of smart city like autonomous vehicles.

APPENDIX

PUBLICATIONS

1. JOURNAL PAPERS

- S. Bhattacharjee, P.K. Madhavarapu and S.K. Das. "Attack Context Embedded Data Driven Trust Diagnostics in Smart Metering Infrastructure" ACM Transactions on Privacy and Security (TOPS) 2020.

2. PEER-REVIEWED CONFERENCE PAPERS

- S. Bhattacharjee, P.K. Madhavarapu and S.K. Das. "Diversity Index based Scoring Framework for Identifying Smart Meters Launching Stealthy Data Falsification Attacks" ACM ASIA Conference on Computer and Communications Security 2021.

3. PAPERS UNDER REVIEW

- P.K. Madhavarapu, S. Bhattacharjee, S.K. Das. "Generative Adversarial Network Based Solution for Detecting Evasion Attacks in Smart Grid"
- P.K. Madhavarapu, S. Bhattacharjee, S.K. Das. "Active Learning Augmented Folded Gaussian Model for Anomaly Detection in Smart Transportation"

4. PAPERS IN PREPARATION

- S. Bhattacharjee, P.K. Madhavarapu, S.K. Das. "A Unified Framework for Fast Identification of Data Falsification Attacks in Smart Living IoT"
- S. Bhattacharjee, P.K. Madhavarapu, S.K. Das. "Diversity Index Model Analysis and Comparison with Neural Networks"

REFERENCES

[1] Shameek Bhattacharjee and Sajal K Das. Detection and forensics against stealthy data falsification in smart metering infrastructure. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[2] Julio A Sanguesa, Javier Barrachina, Manuel Fogue, Piedad Garrido, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Sensing traffic density combining v2v and v2i wireless communications. *Sensors*, 15(12):31794–31810, 2015.

[3] Danda B Rawat and Kayhan Zrar Ghafoor. *Smart cities cybersecurity and privacy*. Elsevier, 2018.

[4] Ramyar Rashed Mohassel, Alan Fung, Farah Mohammadi, and Kaamran Raahemifar. A survey on advanced metering infrastructure. *International Journal of Electrical Power & Energy Systems*, 63:473–484, 2014.

[5] Atieh R Khamesi, Eura Shin, and Simone Silvestri. Machine learning in the wild: The case of user-centered learning in cyber physical systems. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 275–281. IEEE, 2020.

[6] Valeria Dolce, Courtney Jackson, Simone Silvestri, Denise Baker, and Alessandra De Paola. Social-behavioral aware optimization of energy consumption in smart homes. In *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 163–172. IEEE, 2018.

[7] Peak shift. https://www.smartgrid.gov/files/The_Smart_Grid_Promise_DemandSide_Management_201003.pdf.

[8] Stefano Ciavarella, Jhi-Young Joo, and Simone Silvestri. Managing contingencies in smart grids via the internet of things. *IEEE Transactions on Smart Grid*, 7(4):2134–2141, 2016.

[9] Peaker vs demand. https://energy-solution.com/2015/01/29/enabling-automated-demand-response-pge-dras/.

[10] Atieh R Khamesi, Simone Silvestri, Denise A Baker, and Alessandra De Paola. Perceived-value-driven optimization of energy consumption in smart homes. *ACM Transactions on Internet of Things*, 1(2):1–26, 2020.

[11] Vincenzo Agate, Atieh R Khamesi, Simone Silvestri, and Salvatore Gaglio. Enabling peer-to-peer user-preference-aware energy sharing through reinforcement learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.

[12] Paria Jokar, Nasim Arianpoo, and Victor CM Leung. Electricity theft detection in ami using customers' consumption patterns. *IEEE Transactions on Smart Grid*, 7(1):216–226, 2015.

[13] Stephen McLaughlin, Dmitry Podkuiko, and Patrick McDaniel. Energy theft in the advanced metering infrastructure. In *International Workshop on Critical Information Infrastructures Security*, pages 176–187. Springer, 2009.

[14] Daisuke Mashima and Alvaro A Cárdenas. Evaluating electricity theft detectors in smart grid networks. In *International Workshop on Recent Advances in Intrusion Detection*, pages 210–229. Springer, 2012.

[15] Wei Yu, David Griffith, Linqiang Ge, Sulabh Bhattarai, and Nada Golmie. An integrated detection system against false data injection attacks in the smart grid. *Security and Communication Networks*, 8(2):91–109, 2015.

[16] Attack. https://www.maximintegrated.com/content/dam/files/design/technical-documents/white-papers/smart-grid-security-recent-history-demonstrates.pdf.

[17] Puerto rico. http://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/.

[18] Ted Koppel. *Lights out: a cyberattack, a nation unprepared, surviving the aftermath*. Broadway Books, 2015.

[19] Christian Cseh. Architecture of the dedicated short-range communications (dsrc) protocol. In *VTC'98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*, volume 3, pages 2095–2099. IEEE, 1998.

[20] Subir Biswas, Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *IEEE communications magazine*, 44(1):74–82, 2006.

[21] Daniel Jiang and Luca Delgrossi. Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008-IEEE Vehicular Technology Conference*, pages 2036–2040. IEEE, 2008.

[22] Nai-Wei Lo and Hsiao-Chien Tsai. Illusion attack on vanet applications-a message plausibility problem. In *2007 IEEE Globecom Workshops*, pages 1–8. IEEE, 2007.

[23] Seyed Mohammad Safi, Ali Movaghar, and Misagh Mohammadizadeh. A novel approach for avoiding wormhole attacks in vanet. In *2009 Second International Workshop on Computer Science and Engineering*, volume 2, pages 160–165. IEEE, 2009.

[24] SS Manvi, MS Kakkasageri, and DG Adiga. Message authentication in vehicular ad hoc networks: Ecdsa based approach. In *2009 International Conference on Future Computer and Communication*, pages 16–20. IEEE, 2009.

[25] Jinyuan Sun and Yuguang Fang. A defense technique against misbehavior in vanets based on threshold authentication. In *MILCOM 2008-2008 IEEE Military Communications Conference*, pages 1–7. IEEE, 2008.

[26] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21, 2005.

[27] Y-C Hu, Adrian Perrig, and David B Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 3, pages 1976–1986. IEEE, 2003.

[28] Gilles Guette and Bertrand Ducourthial. On the sybil attack detection in vanet. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6. IEEE, 2007.

[29] Chenxi Zhang, Xiaodong Lin, Rongxing Lu, and P-H Ho. Raise: An efficient rsu-aided message authentication scheme in vehicular communication networks. In *2008 IEEE international conference on communications*, pages 1451–1457. IEEE, 2008.

[30] Shameek Bhattacharjee and Sajal K Das. Detection and forensics against stealthy data falsification in smart metering infrastructure. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[31] Security. https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/.

[32] Amir-Hamed Mohsenian-Rad and Alberto Leon-Garcia. Distributed internet-based load altering attacks against smart power grids. *IEEE Transactions on Smart Grid*, 2(4):667–674, 2011.

[33] Shameek Bhattacharjee, Aditya Thakur, Simone Silvestri, and Sajal K Das. Statistical security incident forensics against data falsification in smart grid advanced metering infrastructure. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pages 35–45, 2017.

[34] Ankit Singh Rawat, Priyank Anand, Hao Chen, and Pramod K Varshney. Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks. *IEEE Transactions on Signal Processing*, 59(2):774–786, 2010.

[35] Shameek Bhattacharjee, Aditya Thakur, and Sajal 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*, pages 173–185, 2018.

[36] Anish Jindal, Amit Dua, Kuljeet Kaur, Mukesh Singh, Neeraj Kumar, and Sukumar Mishra. Decision tree and svm-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics*, 12(3):1005–1016, 2016.

[37] Kush Khanna, Bijaya Ketan Panigrahi, and Anupam Joshi. Ai-based approach to identify compromised meters in data integrity attacks on smart grid. *IET Generation, Transmission & Distribution*, 12(5):1052–1066, 2017.

[38] Breno C Costa, Bruno LA Alberto, André M Portela, W Maduro, and Esdras O Eler. Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process. *International Journal of Artificial Intelligence & Applications*, 4(6):17, 2013.

[39] Rong Jiang, Rongxing Lu, Ye Wang, Jun Luo, Changxiang Shen, and Xuemin Shen. Energy-theft detection issues for advanced metering infrastructure in smart grid. *Tsinghua Science and Technology*, 19(2):105–120, 2014.

[40] Varun Badrinath Krishna, Kiryung Lee, Gabriel A Weaver, Ravishankar K Iyer, and William H Sanders. F-deta: A framework for detecting electricity theft attacks in smart grids. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 407–418. IEEE, 2016.

[41] Eduardo Werley S Angelos, Osvaldo R Saavedra, Omar A Carmona Cortés, and André Nunes de Souza. Detection and identification of abnormalities in customer consumptions in power distribution systems. *IEEE Transactions on Power Delivery*, 26(4):2436–2442, 2011.

[42] Kevin Fu and Wenyuan Xu. Risks of trusting the physics of sensors. *Communications of the ACM*, 61(2):20–23, 2018.

[43] Denis Foo Kune, John Backes, Shane S Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. Ghost talk: Mitigating emi signal injection attacks against analog sensors. In *2013 IEEE Symposium on Security and Privacy*, pages 145–159. IEEE, 2013.

[44] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *2017 IEEE European symposium on security and privacy (EuroS&P)*, pages 3–18. IEEE, 2017.

[45] High defense cost. https://www.epri.com/#/pages/product/000000000001026553/.

[46] Money trumps security. https://www.cnet.com/news/money-trumps-security-in-smart-meter-rollouts-experts-say/.

[47] Deqiang Li, Ramesh Baral, Tao Li, Han Wang, Qianmu Li, and Shouhuai Xu. Hashtran-dnn: A framework for enhancing robustness of deep neural networks against adversarial malware samples. *arXiv preprint arXiv:1809.06498*, 2018.

[48] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017.

[49] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7), 2018.

[50] Shuva Paul and Zhen Ni. Study of learning of power grid defense strategy in adversarial stage game. In *2019 IEEE International Conference on Electro Information Technology (EIT)*, pages 292–297. IEEE, 2019.

[51] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. Generative adversarial network for synthetic time series data generation in smart grids. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2018.

[52] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

[53] Gerhard P Hancke, Gerhard P Hancke Jr, et al. The role of advanced sensing in smart cities. *Sensors*, 13(1):393–425, 2013.

[54] Michael Batty, Kay W Axhausen, Fosca Giannotti, Alexei Pozdnoukhov, Armando Bazzani, Monica Wachowicz, Georgios Ouzounis, and Yuval Portugali. Smart cities of the future. *The European Physical Journal Special Topics*, 214(1):481–518, 2012.

[55] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[56] Leonel Santos, Carlos Rabadao, and Ramiro Gonçalves. Intrusion detection systems in internet of things: A literature review. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–7. IEEE, 2018.

[57] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[58] Fangzhou Sun, Abhishek Dubey, and Jules White. Dxnat—deep neural networks for explaining non-recurring traffic congestion. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2141–2150. IEEE, 2017.

[59] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[60] Michael Wilbur, Abhishek Dubey, Bruno Leão, and Shameek Bhattacharjee. A decentralized approach for real time anomaly detection in transportation networks. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 274–282. IEEE, 2019.

[61] Irish dataset. http://www.ucd.ie/issda/data/.

[62] MS Windows NT kernel description. http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2010-09-30.

[63] hereapi. https://developer.here.com/.

[64] Alvaro A Cárdenas, Robin Berthier, Rakesh B Bobba, Jun Ho Huh, Jorjeta G Jetcheva, David Grochocki, and William H Sanders. A framework for evaluating intrusion detection architectures in advanced metering infrastructures. *IEEE Transactions on Smart Grid*, 5(2):906–915, 2014.

[65] Michael Wilbur, Abhishek Dubey, Bruno Leão, and Shameek Bhattacharjee. A decentralized approach for real time anomaly detection in transportation networks. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 274–282. IEEE, 2019.

[66] Jose Paolo Talusan, Francis Tiausas, Keiichi Yasumoto, Michael Wilbur, Geoffrey Pettet, Abhishek Dubey, and Shameek Bhattacharjee. Smart transportation delay and resiliency testbed based on information flow of things middleware. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 13–18. IEEE, 2019.

[67] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[68] Weifeng Xia and Yuming Chu. The schur convexity of gini mean values in the sense of harmonic mean. *Acta Mathematica Scientia*, 31(3):1103–1112, 2011.

[69] Yu Ishimaki, Shameek Bhattacharjee, Hayato Yamana, and Sajal K Das. Towards privacy-preserving anomaly-based attack detection against data falsification in smart grid.

[70] Meter privacy. https://skyvisionsolutions.files.wordpress.com/2014/08/utility-smart-meters-invade-privacy-22-aug-2014.pdf.

[71] Audit trail. https://www.fairwarning.com/blog/power-audit-trail-data-security/.

[72] Inspection time. https://www.ovoenergy.com/help/smart-meter-installation#how-long-will-i-have-to-wait-until-i-get-my-smart-meter.

[73] Ehsan Ullah Warriach, Marco Aiello, and Kenji Tei. A machine learning approach for identifying and classifying faults in wireless sensor network. In *2012 IEEE 15th International Conference on Computational Science and Engineering*, pages 618–625. IEEE, 2012.

VITA

Venkata Praveen Kumar Madhavarapu was born in Vijayawada, Andhra Pradesh, India. He graduated with a Bachelor of Technology (B.Tech.) in Computer Science Engineering in 2015 from VR Siddhardha Engineering College, Vijayawada, India. He received his Masters degree in Computer Science from Southern Illinois University, Carbondale, USA in Fall 2016. He joined Missouri University of Science and Technology, Rolla, USA as Ph.D. scholar in Computer Science in Fall 2017 under Dr. Sajal K. Das. During this time, he served as graduate teaching assistant in three graduate and undergraduate courses. In December 2021, he received his Ph.D. in Computer Science from Missouri University of Science and Technology.