GridWatch: A Smart Network for Smart Grid

Masoud Hemmatpour
Dept. Computer Science
Arctic University of Norway

Changgang Zheng
Dept. Engineering Science
University of Oxford

Noa Zilberman
Dept. Engineering Science
University of Oxford

Phuong Hoai Ha
Dept. Computer Science
Arctic University of Norway

***Abstract*—The adoption of decentralized energy market models facilitates the exchange of surplus power among local nodes in peer-to-peer settings. However, decentralized energy transactions within untrusted and non-transparent energy markets in modern Smart Grids expose vulnerabilities and are susceptible to attacks. One such attack is the False Data Injection Attack, where malicious entities intentionally inject misleading information into the system. To address this threat, this paper proposes GridWatch, an effective real-time in-network intelligent framework to detect false data injection attacks. Gridwatch operates in a hybrid model. It deploys inference model in the programmable network devices and also on the server to detect false data injection attacks. GridWatch was evaluated using a real-world dataset from Austin, Texas, and can detect false data injection attacks with 94.8% accuracy. GridWatch on average performs 4 billions transactions per second in less than 1.8 microsecond latency.**

***Index Terms*—Smart Grid, security, False Data Injection Attack (FDIA), in-network machine learning, peer-to-peer (P2P) trading**

I. Introduction

Traditional power grids generally carry power from a few central generators to a large number of users or customers. A Smart Grid represents a transformative evolution in the way electricity is generated, transmitted, distributed, and consumed. Unlike traditional power grids, a Smart Grid leverages cutting-edge technologies to enhance efficiency, reliability, and sustainability across the entire energy ecosystem [14]. Widespread adoption of small-scale renewable energy sources is vital in modern power systems in order to achieve energy sustainability and reduce the environmental impact of electricity generation [14]. From these distributed energy resources emerged a new form of energy trading mechanism known as Peer-to-Peer (P2P) energy trading [12]. As experts are working to address new challenges associated with the P2P energy trading mechanism, a critical focus is on the security of P2P systems and identifying potential attack vectors. A False Data Injection Attack (FDIA) in a P2P energy system means the malicious injection of inaccurate or deceptive information into the system's data streams. This type of attack can have serious consequences on the integrity, reliability, and security of the P2P energy trading process [12].

Already significant effort was dedicated to identifying and mitigating attacks in the electric power grid [3]. However, leveraging ML methods in P2P systems is a novel strategy [12]. The current dominant trend is offline data analysis, as computational costs are a major obstacle for using real-time systems [5]. The latency of the state-of-the-art solution [12] is about 0.043 second, and can be higher when communication delays are accounted for. The current research landscape lacks solutions capable of detecting FDIA in real-time and within realistic scenarios [3]. One of the main challenges is to detect the attack as early as possible in real architecture. This paper addresses this gap by introducing a state-of-the-art architecture, called GridWatch, capable of real-time FDIA detection and offering actionable responses to mitigate attacks' impact. GridWatch uses emerging Programmable Network Devices (PNDs) for the analysis of incoming traffic and the identification of potential attacks.

In this paper, we assert that the network is a strategic location for detecting FDIAs and promptly reacting to them in real-time. We introduce the first in-network processing solution for real-time detection of FDIAs in Smart Grid. Additionally, while previous approaches have utilized ML methods to identify FDIAs [13], to the best of our knowledge, this paper is the first in-network machine learning solution capable of real-time FDIAs detection, achieving high accuracy and low latency [19].

We propose an in-network machine learning approach to detect attack and perform smart action to mitigate attack impact. In-network computing is a new technology that has emerged over the last few years. It refers to the execution of programs typically running on end-hosts within PNDs [19]. PNDs possess the capability to execute customized programs at the line rate speed, enabling the timely identification of potential malicious traffic. In-network machine learning involves executing the entire process (or at least inference) directly within the network infrastructure [19]. This work builds upon in-network machine learning techniques to detect malicious injection of false data in Smart Grid data stream.

The main contributions of this research can be summarized as follows:

- Proposing an in-network solution to detect False Data Injection Attack (FDIA) in P2P energy trading system.
- Developing real-time in-network mitigation techniques against detected FDIA.
- Implementing and evaluating the solution using a real-world dataset and providing a detailed performance analysis of the approach.

This paper is organized as follows: Section II presents an overview of peer-to-peer distributed energy trading. False data injection attacks are described in Section III. Section IV describes in-network computing and the relevant requirements for this study. Section V outlines the attack examined in this paper. Section VI describes the system design of GridWatch. In

Section VII, we present the evaluation of GridWatch. Section VIII discusses the proposed solution. Finally, in Section IX we draw conclusions.

II. Peer-to-peer Distributed Energy Trading

To maximize the advantages of renewable energy generation, energy trading among households within a residential microgrid has emerged as an efficient strategy [2]. The variability of renewable generation technologies may result in an excess of energy in some households and a deficiency in others. Consequently, residences with surplus energy can engage in selling the excess to homes facing deficits, a process known as local energy trading. However, the success of local energy trading is contingent upon the accessibility of energy consumption/generation data and the reliability of energy trading signals [10]. This practice, known as Peer-to-Peer (P2P) trading, represents a decentralized approach for consumers to directly exchange electricity [17]. The trading model that is explored in this work is adapted from [18]. Fig. 1 illustrates the conceptual model of the market under consideration. The model assumes the presence of prosumers, which includes both prosumers and pure consumers, along with suppliers and a community coordinator. The suppliers are presumed to possess their own energy generation capabilities, intended to supply energy to community households when local solar generation from prosumers is insufficient to meet demand.

The temporal dimension is divided into time slots. Prosumers, including pure consumers, make decisions regarding the quantity of energy to buy or sell for each time slot, considering their individual solar power generation and load consumption. Subsequently, they submit their energy transaction requests to the community coordinator.

The coordinator plays a pivotal role in the market dynamics by computing internal pricing for energy transactions among prosumers. This pricing is based on the net load, which is the difference between the total energy supply and total demand within the community during the specified time slots.

In addition, suppliers forward their energy and price bids to the coordinator. The coordinator engages in negotiations with suppliers regarding the external energy price and conducts trades on behalf of all prosumers and consumers. From a market perspective, the community coordinator can be managed by a centralized infrastructure like the cloud or be co-managed by a decentralised autonomous organisation.

III. False Data Injection Attacks

With the transition from conventional power systems to smart grids, a complex network of interconnected sensors consistently collects data crucial for maintaining the secure and dependable operation of the grids. The complexity of large scale decentralized energy markets is highly vulnerable to several security and privacy concerns which makes it an appealing attack target for malicious adversaries [7].

FDIA is a type of attack where an adversary intentionally introduces malicious or incorrect data into a system to compromise its integrity, reliability, or the decisions made based on that data. These attacks can have serious consequences, especially in Smart Grids where accurate and reliable data is crucial for the proper functioning of the system [12]. The concept of FDIAs was coined by Liu *et al.* [9] and became one of the stealthiest and devastating attacks on power systems [3].

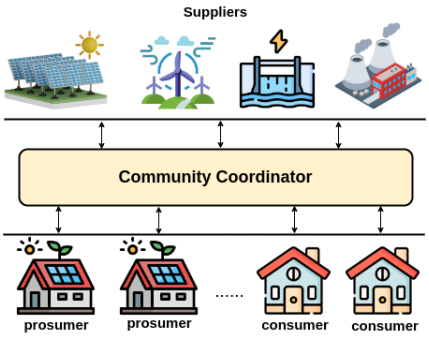


Fig. 1: P2P market model.

Over the last few years, a considerable body of research has emerged addressing threats related to FDIAs in power systems. For example, in a local energy market, trusted third parties may act as an attacker and cause loss of benefits for the legitimate participants, whereas harnessing benefits for itself. It was demonstrated [4] that in the presence of an attacker, the profits/savings at the legitimate participants have significantly decreased, up to 94% at certain hours.

An adversary is able to manipulate the measurements taken at several meters in a power system, and it can sometimes change the state estimate at the control center in a way that will never be detected by classical bad data detectors. Kosut *et al.* [6] developed a heuristic method to find bad adversarial attacks. Liu *et al.* [8] viewed the false data detection problem as a matrix separation problem and proposed two methods, the nuclear norm minimization and low rank matrix factorization to solve this problem.

Wang *et al.* [15] investigated the detection of FDIAs in AC state estimation where attackers cannot obtain the accurate transmission line admittances and the accurate estimated system state variables. They proposed a detection method using secure phasor measurement unit data.

Mohammadi *et al.* [12] explored two threat scenarios based on a novel FDIA model in a local P2P energy trading system. In these scenarios, an attacker gains free energy by manipulating prosumers' consumption and demand. They proposed an instance-based machine learning (ML) classifier on the server for detecting FDIAs.

IV. In-network computing

Traditional data networks only transport data from one node to another, with no computation within the network. However, recent Programmable Network devices (PNDs) allow to run customized computation on the network devices themselves [19]. PNDs come in various forms, such as SmartNIC and Programmable Switch (referred to as Smart Switch), and serve different purposes within a network, shortly described in this section.

SmartNIC is an evolution of standard Network Interface Cards (NIC) [19] with advanced processing capabilities, both programmable and using dedicated accelerators. This was attained by adding processing units to a NIC, such as ARM CPU cores or FPGA modules.

Smart Switch architectures permit the data plane functionality to be fully re-configurable [1]. Functionality of a switch can be logically split into control plane and data plane [19]. The control plane is in charge of establishing packet processing policies, such as where to forward a packet. The data plane is responsible for executing the packet processing at very high speed. Data plane functionality of a Smart Switch can be implemented in an Application specific Integrated Circuit (ASIC), an Field-programmable Gate Array (FPGA), or a Network Processor (NP).

In-network computing refers to the offloading of programs or computation tasks to network devices such as SmartNICs and Smart Switches [19]. In-network computing takes advantage of network devices' high processing speeds and low overheads in physical space, energy, and cost, as they are already part of network infrastructure. Realization of in-network computing allows networks to become part of available computing resources. It provides better integration of communication and computing resources when diverse application requirements need to be addressed. However, implementing ML within the network is challenging. Not only network devices are resource constrained, but their architecture doesn't lend itself easily to machine learning scale and complexity.

There are two common forms of ML in the network [19]: Network-Assisted ML, which uses network devices primarily for model training acceleration and feature collection, while the inference takes place on the end host. In-Network ML refers to complete ML processes, either training or inference, done entirely within the network. There are very few in-network machine learning frameworks that support more than one ML model. Planter is an automated in-network ML framework that support a large number of ML models on different platforms [21]. IIsy extends Planter's capabilities by introducing a hybrid methodology for ensemble models. This approach involves deploying a small model on a switch and a larger model on the backend. The location of decision making dictated by inference's confidence, where the high-confidence decisions will be made inside the network, while the low-confidence decisions need to be forwarded to the backend for further process.

V. Attack

This research seeks to leverage existing attack methodologies and proposes a protection framework that can be practically implemented in real-world scenarios. Previous research of FDIAs suffers from a lack of realistic experimentation and insufficient corroboration of FDI attack evaluations [3]. There were few works on P2P trading that rely on real-world datasets. One of them, used in this work, studied how energy could be gained for free through FDIA in local P2P energy trading scenarios [12].

In the potential threat scenario, a malicious energy seller, resembling a supplier, can initiate an attack by taking the role of a prosumer. An alternative motivation could be the pursuit of profit by suppliers, achievable through the reduction of incentives for individuals to adopt the role of an energy-selling prosumer [12].

Fig. 2 illustrates the components of the attack based on the P2P energy model described in section II. A solar-powered smart home equipped with Home Energy Management (HEM) regulates overall energy consumption. Smart plugs transmit their data to the HEM unit via a Zigbee network. The HEM unit shares household consumption data with other households (i.e., prosumers) and the coordinator.

The attacker attempts to manipulate households' consumption in two ways: an insider attack directed at the physical HEM host, and an outsider attack targeting the HEM through the network.

In the outlined threat scenario, the compromised HEM unit disseminates false consumption data to the coordinator, resulting in inaccurate demand information within the neighborhood. Following the HEM system's compromise, the attacker gains control over the flow of demand information in the communication links between HEM and the coordinator, allowing to falsify demands reported by the prosumers' HEM systems.

The coordinator calculates both internal and external prices based on the falsified demands, and prosumers update their demands based on the false prices. At this point, the quantity of energy supplied may deviate from the actual total demand of prosumers, resulting in a potential grid imbalance. When suppliers furnish more energy than the genuine demand, the surplus energy remains unconsumed by the prosumers. This happens because the prosumers under attack are unaware that their recent demand has been manipulated by the attacker. Consequently, an opportunity emerges for the attacker to exploit this excess energy without cost, potentially achieved by installing a battery on the grid side of their home's smart meter. This strategic placement renders the smart meter incapable of accurately recording the energy consumed by the battery storage, thus enabling the attacker to utilize the energy without detection or recording. We have access to a real-world dataset stemming from instances of this particular attack, utilized in this work [11].

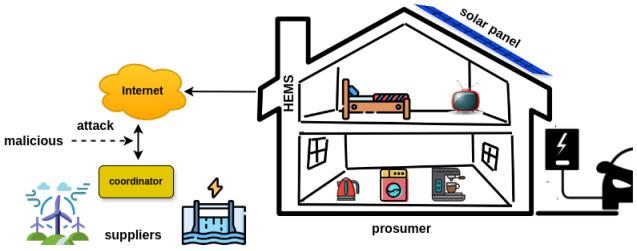


Fig. 2: Malicious user in the system.

VI. GridWatch

In this work, we introduce a novel architecture for real-time detection of FDIAs. Our solution leverages in-network

machine learning techniques to identify anomalies. Fig. 3 illustrates our proposed architecture, referred to as GridWatch. This architecture can be implemented on any smart switch positioned between the Home Energy Management (HEM) system and the community coordinator.

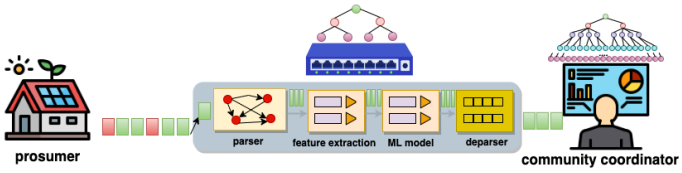


Fig. 3: GridWatch strategically positioned between the prosumer and the community coordinator.

GridWatch, implemented on a smart switch, enables the detection of malicious prosumers while data is being transmitted from the HEMs to the community coordinator. The architecture used by GridWatch builds upon the Protocol Independent Switch Architecture (PISA) of the smart switch. PISA features a multi-stage pipeline for processing packets, where each stage can be programmed separately. Each stage adds nanosecond-scale latency, but as the pipeline moves data rather than instructions, the switch can process multiple packets simultaneously and can execute instructions in parallel.

PISA typically comprises three main components. First, the *Parser* is programmed to specify the header fields to be recognized and matched by subsequent stages. Second, a sequence of *Match-Action* units, constructed using a combination of Static Random Access Memory (SRAM) and Ternary Content Addressable Memory (TCAM), is programmed to match and take action based on identified header fields. These units are used for the implementation of in-network machine learning models. Last, the *Deparser* is responsible for re-serializing the packet metadata into the packet before its transmission into the network. This structured PISA architecture empowers GridWatch to efficiently process and analyze incoming data packets, allowing a timely detection of anomalies, particularly those indicative of malicious prosumer activity.

P4 is a domain-specific language that is target agnostic, supporting PISA-based switch-ASIC [1]. P4 and ASICs enable the implementation of machine learning models within smart switches, with support for line rate execution. However, integrating algorithms, including machine learning models, into programmable network devices presents several common challenges. These challenges include constraints such as limited memory, small number of processing stages, no floating point support, and others [19]. There are few P4 based in-network ML frameworks that support multiple ML models such as **Planter** [21] and **IIsy** [20]. Planter, designed as a rapid prototyping in-network machine learning framework and implemented in Python, facilitates the automatic generation of P4 code tailored for deployment on smart switches [21]. IIsy adopts a hybrid methodology, executing a compact model on the switch while running a more extensive model on the backend, based on the confidence level of the compact model [20]. IIsy aims to maximize the performance of ML prediction, while still achieving high throughput and low latency within the switch ingress.

In GridWatch, the initial step parses incoming packets and extracts relevant features from the parsed data. Subsequently, packets are classified using the extracted features and a pre-trained ML model, implemented using Planter on Match-Action units. If the traffic is deemed non-suspicious and non-malicious, it is allowed to proceed uninterrupted towards its destination. If the incoming traffic from the prosumer is deemed suspicious, it will be marked as malicious and the coordinator will receive a notification.

In this study, six different features are used to pre-train the model, including household's identification number, energy generation, shiftable loads, base loads, energy consumption, and household's energy buying or selling request as described the attack scenario [12].

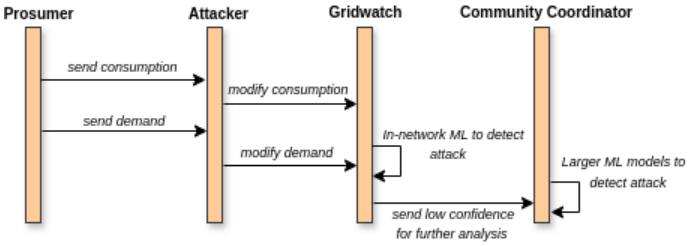


Fig. 4: Overview of attack detection process by GridWatch.

As Fig. 4 illustrates an attacker influencing the system by falsifying the consumption and demand of a prosumer. Gridwatch architecture is implemented in two parts. A small model is implemented in the network devices (i.e., smart switch) and a large model is implemented on the community coordinator server. First, the small model looks for the attack, and assigns a certain confidence level to classified traffic. If the classification of a packet has a confidence level below a configured threshold, the switch passes the packet to the server for further analysis by a large model. We leverage IIsy [20] to support this hybrid deployment.

VII. Evaluation

GridWatch performance is evaluated using an attack dataset based on a real-world data [12]. The dataset contains the data on the 1st day of August 2018, from 7:00 to 19:00, when solar panels can generate energy. The dataset has six main features; user (prosumer/consumer) ID, energy generation, shiftable loads, base loads, energy consumption (shiftable loads+base loads), and household's demand (it is equal to the difference between the generation and the consumption). It has two classes of events, namely attack (FDI) event and normal event. The attack data was generated by modifying the shiftable load and/or the base load of prosumer, and by updating the total consumption based on the modified shiftable/base loads. The methodology for generating attacks was outlined in [12]. GridWatch is trained to differentiate attack events from normal ones with varying numbers of features, ranging from 3 to 6.

Accuracy and F1 score (a more nuanced index, a combination of precision and recall, especially for imbalanced datasets) metrics have been adopted to evaluate the performance of the detection methods. The output of a detection model is divided into True Positive (TP): indicating a correct positive classification, True Negative (TN): a correct negative classification, False Positive (FP): an incorrect positive classification, and False Negative (FN): an incorrect negative classification.

Accuracy is the ratio of the number of correct predictions to the number of total predictions. F1 score = $\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$, where Precision is the number of true positive predictions divided by the number of true positive and false positive predictions. Recall is the number of true positive predictions divided by the number of true positive and false negative predictions [13]. FPR = $\frac{\text{FP}}{\text{FP+TN}}$ and TPR = $\frac{\text{TP}}{\text{TP+FN}}$. The Area Under the Curve (AUC) is the under plot of the TPR against the FPR.

GridWatch utilizes both Planter[1] and IIsy[2]. Table I shows the accuracy of the baseline, pure in-network ML-based GridWatch, and hybrid GridWatch. The baseline is Mohammadi *et al.* [12], who discovered the attack, and the accuracy is as they reported. The evaluation used the set of features 'ID', 'Generation', 'Demand', 'Sload', 'Bload', 'Consumption', same as in baseline. As the Table I shows, for comparable models, Random Forest (RF) achieves the highest accuracy, reaching 92.82%, surpassing the baseline accuracy of 91.22%. The AUC score of FPR and TPR for RF on all cases is above 0.9, and 3.8% FPR for 6Fs. All the results obtained through GridWatch achieve higher accuracy compared to the baseline. Increasing the number of features set does not necessarily improve accuracy beyond a certain level.

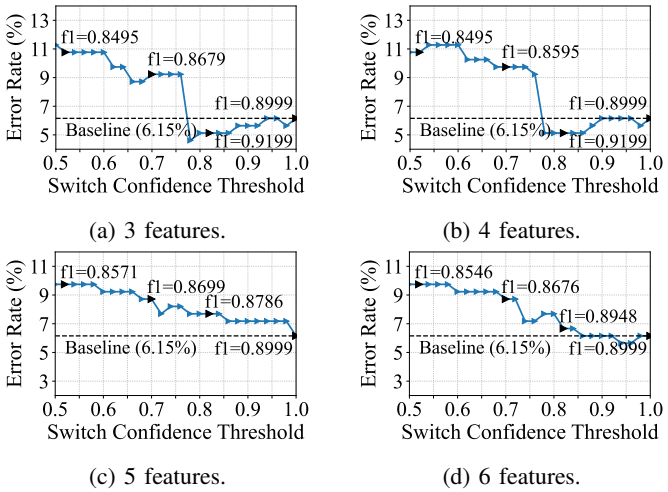
Models	3Fs	4Fs	5Fs	6Fs
GridWatch - Naïve Bayes	0.789	0.769	0.846	0.851
GridWatch - K-means	0.769	0.769	0.769	0.769
GridWatch - Decision Tree	0.902	0.907	0.907	0.897
GridWatch - Random Forest	0.897	0.907	0.912	0.928
GridWatch - **Hybrid** (conf=0.8)	0.948	0.948	0.923	0.923
Baseline - modified K-means [12]				0.912

TABLE I: Accuracy of different models, as a function of number of features (Fs).

Fig. 5 illustrates the inference accuracy of the GridWatch, as a function of the confidence threshold for classification within the switch. The baseline result shows the misclassification rate of a server-based large RF model with 200 trees and 10000 maximum leaf nodes, which is 6.15% and an F1 score of 0.899. With 3 and 4 features, with a confidence threshold of ≈ 0.8, the switch handles about 87% of the traffic, achieving a system F1 score of 0.919.

As depicted in Fig. 6 (a), as the confidence threshold rises above 0.9, there is a decrease in the proportion of traffic classified by the switch. Additionally, as Fig. 6 (c) shows overall throughput diminishes after reaching the 0.9 threshold, accompanied by an increase in average latency in Fig. 6 (d).

[1]https://github.com/In-Network-Machine-Learning/Planter
[2]https://github.com/In-Network-Machine-Learning/IIsy



(a) 3 features. (b) 4 features.

(c) 5 features. (d) 6 features.

Fig. 5: GridWatch performance with different features.

However, as in Fig. 6 (b), switch accuracy already reaches the same level as the server at a confidence threshold of 0.85.

VIII. Discussion

Since attacking prosumers in peer-to-peer distributed energy trading can have significant economic or physical impacts on power systems [3], [13], it is critical to detect and mitigate such attacks promptly, increasing the security and reliability of the power system.

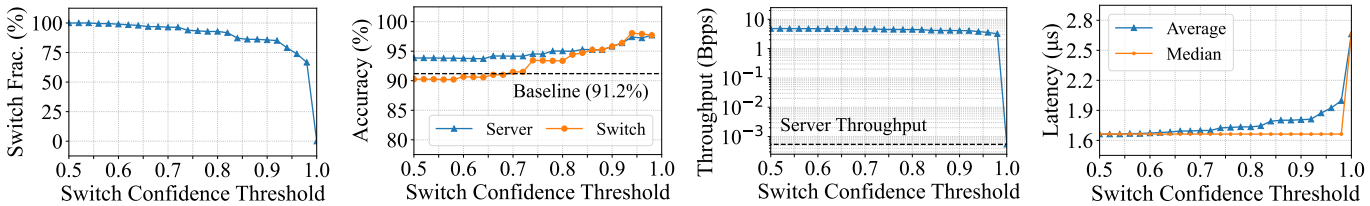
Prior research has demonstrated the capability of machine learning algorithms to identify FDIAs in power systems [13]. However, employing ML models as a defense tactic within P2P trading systems is a novel approach [12].

Our GridWatch solution stands out as it requires no modifications on the prosumer's end, including their HEM system. Furthermore, it does not require to have a full knowledge of the distributed network. Instead, the defensive strategy is integrated directly into the communication network, enabling real-time functionality and updates. Furthermore, this solution is power efficient [20]. Below, GridWatch is discussed in terms of updating the model over time, latency, and target deployment.

Model Update. Data is known to skew over time. Retrained In-network ML models can be updated during runtime on network devices [16], [20] within tens of milliseconds. However, more complex changes like changing the type of ML models [21] require recompilation and momentary stopping of traffic.

System latency. The average ($1.8\mu s$) and median latency of GridWatch are 3 orders of magnitude better than the traditional solutions ($0.043s$) running on a coordinator, with a confidence threshold up to 0.9. Still, low-confidence samples (typically less than 20%) need to be forwarded to the backend, therefore experiencing the same latency as traditional systems.

Deployment Target. The inference performance of GridWatch is a property of the target deployment platform. Both inference and system performance will change on different platforms. For example, FPGA is less resources limited and



(a) Switch Fraction. (b) Accuracy. (c) Throughput. (d) Latency.

Fig. 6: GridWatch Insights and System Performance.

can support larger models while having a lower throughout and larger latency, while on a switch-ASIC, the model size is more limited but the system performance is high.

IX. Conclusion

This work proposes that an optimal location to detect attacks is within the network. Emerging programmable network devices enable the implementation of machine learning models within the network infrastructure. The advantages of in-network ML model over traditional ML models include low latency in attack detection and no needed alterations by prosumers. Furthermore, in-network computing shows promise in terms of power efficiency.

We introduced GridWatch, an in-network ML solution designed to detect attacks in P2P systems. The system was evaluated using a verified attack scenario and real-world dataset. GridWatch uses a hybrid deployment model, leveraging both server and network devices to detect attacks With 94% accuracy rate and minimal latency.

X. Acknowledgment

The work was supported by European Commission under MISO (grant 101086541) and SMARTEDGE (grant 101092908, UKRI 10056403) projects, Research Council of Norway under eX3 project (grant 270053) and Sigma2 (grant NN9342K), and VMware. For the purpose of Open Access, the author has applied a CC-BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission.

References

[1] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95, 2014.

[2] Per Goncalves Da Silva, Dejan Ilić, and Stamatis Karnouskos. The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading. *IEEE Transactions on Smart Grid*, 5(1):402–410, 2013.

[3] Muhammad Akbar Husnoo, Adnan Anwar, Nasser Hosseinzadeh, Shama Naz Islam, Abdun Naser Mahmood, and Robin Doss. False data injection threats in active distribution systems: A comprehensive survey. *Future Generation Computer Systems*, 140:344–364, 2023.

[4] Shama N Islam, Md Apel Mahmud, and Aman Maung Than Oo. Impact of optimal false data injection attacks on local energy trading in a residential microgrid. *Ict Express*, 4(1):30–34, 2018.

[5] Charalambos Konstantinou and Michail Maniatakos. A data-based detection method against false data injection attacks. *IEEE Design & Test*, 37(5):67–74, 2019.

[6] Oliver Kosut, Liyan Jia, Robert J Thomas, and Lang Tong. Limiting false data attacks on power system state estimation. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2010.

[7] Zhetao Li, Jiawen Kang, Rong Yu, Dongdong Ye, Qingyong Deng, and Yan Zhang. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE transactions on industrial informatics*, 14(8):3690–3700, 2017.

[8] Lanchao Liu, Mohammad Esmalifalak, Qifeng Ding, Valentine A Emesih, and Zhu Han. Detecting false data injection attacks on power grid by sparse optimization. *IEEE Transactions on Smart Grid*, 5(2):612–621, 2014.

[9] Yao Liu, Peng Ning, and Michael K Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):1–33, 2011.

[10] Ritwik Majumder, Gargi Bag, and Ki-Hyung Kim. Power sharing and control in distributed generation with wireless sensor networks. *IEEE Transactions on Smart Grid*, 3(2):618–634, 2012.

[11] Sara Mohammadi, Frank Eliassen, Yan Zhang, and Hans-Arno Jacobsen. Attack dataset austin texas 2018. https://ieee-dataport.org/documents/attackdatasetaustintexas2018, 2021.

[12] Sara Mohammadi, Frank Eliassen, Yan Zhang, and Hans-Arno Jacobsen. Detecting false data injection attacks in peer to peer energy trading using machine learning. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3417–3431, 2021.

[13] Ali Sayghe, Yaodan Hu, Ioannis Zografopoulos, XiaoRui Liu, Raj Gautam Dutta, Yier Jin, and Charalambos Konstantinou. Survey of machine learning methods for detecting false data injection attacks in power systems. *IET Smart Grid*, 3(5):581–595, 2020.

[14] Maria Lorena Tuballa and Michael Lochinvar Abundo. A review of the development of smart grid technologies. *Renewable and Sustainable Energy Reviews*, 59:710–725, 2016.

[15] Zhenhua Wang, Haibo He, Zhiqiang Wan, and Yan Sun. Detection of false data injection attacks in ac state estimation using phasor measurements. *IEEE Transactions on Smart Grid*, 2020.

[16] Mingyuan Zang, Changgang Zheng, Lars Dittmann, and Noa Zilberman. Towards Continuous Threat Defense: In-Network Traffic Analysis for IoT Gateways. *IEEE Internet of Things Journal*, 2023.

[17] Chenghua Zhang, Jianzhong Wu, Yue Zhou, Meng Cheng, and Chao Long. Peer-to-peer energy trading in a microgrid. *Applied energy*, 220:1–12, 2018.

[18] Min Zhang, Frank Eliassen, Amir Taherkordi, Hans-Arno Jacobsen, Hwei-Ming Chung, and Yan Zhang. Energy trading with demand response in a community-based p2p energy market. In *2019 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm)*, pages 1–6. IEEE, 2019.

[19] Changgang Zheng, Xinpeng Hong, Damu Ding, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. In-network machine learning using programmable network devices: A survey. *IEEE Communications Surveys & Tutorials*, 2023.

[20] Changgang Zheng, Zhaoqi Xiong, Thanh T Bui, Siim Kaupmees, Riyad Bensoussane, Antoine Bernabeu, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. IIsy: Hybrid in-network classification using programmable switches. *IEEE/ACM Transactions on Networking*, 2024.

[21] Changgang Zheng, Mingyuan Zang, Xinpeng Hong, Liam Perreault, Riyad Bensoussane, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. Planter: Rapid Prototyping of In-Network Machine Learning Inference. In *ACM SIGCOMM Computer Communication Review*. 2024.