2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 979-8-3503-1855-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SmartGridComm60555.2024.10738074

Leveraging Compact Data Accumulator to Enable In-Network Anomaly Detection in Programmable Switches for Power Grids

Zheng Hu*, Hui Lin†, Yanfeng Qu*, Dong Jin*
*Electrical Engineering and Computer Science, University of Arkansas, Fayetteville, AR, USA
{zhenghu, yqu, dongjin}@uark.edu
†Electrical, Computer and Biomedical Engineering, The University of Rhode Island, Kingston, RI, USA
huilin@uri.edu

***Abstract*—The escalating cyber-attacks targeting power infrastructure underscore the critical importance of smart grid security. However, existing solutions often struggle with the challenge of balancing security and performance overhead, leading to suboptimal protection or increased operational latency. To address this, we propose an intrusion detection system (IDS) designed to operate within P4-based programmable network devices, enabling real-time identification of critical attacks like distributed denial-of-service (DDoS) and false data injection (FDI). Central to our approach is a novel data structure optimized for time series data, capturing key information such as packet timing and data payload distribution. Leveraging decision trees, a robust machine learning technique, enables effective anomaly detection and prediction. Additionally, we integrate data compression techniques to reduce device memory usage while maintaining detection accuracy. Our evaluation results demonstrate minimal overhead in packet processing speed with 1 to 20 nanoseconds differences per packet, and enhanced data storage efficiency with compression ratios reaching up to 60.9%. Despite these optimizations, there is only a slight decrease in detection accuracy, such as a 2.81% drop in detecting false data injection attack (FDIA).**

***Index Terms*—Data Compression, Distributed Network Protocol (DNP3), Network Security, P4, Programmable Network, Supervisory Control and Data Acquisition (SCADA) System, Smart Grid,**

I. Introduction

Smart grids integrate advanced computing and communication technologies to optimize energy processes, yet this increased connectivity also amplifies vulnerabilities to cyber threats. As a result, securing smart grids is imperative to ensure reliable energy delivery, protect critical infrastructure, and uphold consumer privacy and safety.

Current cybersecurity solutions always involve a trade-off between security and performance, stalling the feasibility of deploying them in delay-sensitive critical infrastructures with demanding service availability. On the one hand, wide-area monitoring systems collect a large amount of data, attempting to increase the grid visibility at a much finer granularity. Leveraging these data, many studies dedicate research efforts to various security mechanisms, such as intrusion detection systems, encryption algorithms, and authentication protocols [1]–[3], increasing the accuracy of detecting various anomalies. On the other hand, control operations in power grids require a demanding communication latency to ensure the detection and mitigation of real-time events. The increasing amount of data in security analysis can introduce significant latency and computational burden, thereby hindering real-time operations and responsiveness of smart grid.

The obstacle caused by the security-performance trade-off is not unique to power grids. To address this obstacle in general-purpose computing environments like data centers, many studies design *in-network* analysis and security solutions by retrofitting conventional probabilistic data structures (e.g., Bloom Filter and Sketches [4], [5]) based on Programming Protocol-independent Packet Processors (P4) architecture [6]–[11], which introduces data plane programmability in network devices. These probabilistic data structures profile the occurrence of network-level events and enable threshold-based detection. However, power grids and other industrial control systems (ICS) heavily rely on time series analysis to reveal anomalies in the trajectory of a control process, leading to the design of their domain-specific data structures focusing on compressing the storage space of timed meter measurements [12]. Inspired by successful in-network security solutions in general-purpose computing environments, we raise a critical question: *Can we retrofit proprietary data structures for power grid applications in P4-enabled network devices, achieving a breakthrough of security-performance trade-off?*

This paper aims to affirmatively answer this question by integrating data compression methods, widely used in power grid applications, for the first time within programmable data planes to bolster smart grid data security. In addition to reducing storage space, we design security solutions directly utilizing compressed data, leveraging line-rate hardware pipelines in programmable switches. Our design is based on an important observation: normal physical states in power or control systems typically deviate around a stability point. This deviation widens when disruptions occur. We utilize data compression methods to filter out baseline information

while profiling sensitive changes, which are critical to revealing potential disruptions. Consequently, we can make a significant breakthrough in the security-performance trade-off, i.e., achieve accurate intrusion detection and reduce storage overhead simultaneously with negligible detection latency.

Our contributions in this paper are summarized as follows:

- We design a data structure called the Meter Data Accumulator (MDA) tailored for handling time series data in smart grids, capturing key information such as packet timing and payload distribution.
- We integrate data compression techniques within the MDA to optimize resource utilization and reduce storage overhead without compromising accuracy or performance.
- We develop an efficient intrusion detection system based on MDA to operate entirely within programmable network devices, enabling real-time identification of critical attacks like Distributed Denial of Service (DDoS) and False Data Injection Attacks.
- We maintain high packet processing performance with minimal overhead (1 to 20 nanoseconds per packet) and achieve a compression ratio of up to 60.9%, significantly reducing memory requirements.

The remainder of this paper is organized as follows: Section II presents the background of P4 and data structure for compressed measurement and related security solutions. Section III describes our proposed design, including the MDA with data compression, and anomaly detection using decision trees. Section IV presents the experimental results and analysis. Finally, Section V concludes the paper with future works.

II. Background and Related Work

A. Background

Programmable Network Data Planes Enabled by P4. Current Internet companies are undergoing a foundational change in their network infrastructures, transitioning from network programmability enabled by a *centralized control plane* to a high-speed hardware platform equipped with *decentralized data planes*. Field Programmable Gate Arrays (FPGA) and standardized programming interfaces such as P4 are the major forces driving such transition, permitting network owners to run customized packet processing functions [13]. By adopting this advanced data plane programmability, many network infrastructures, including ICS networks, can possess the following advantages: (i) customized information extraction across the full network stack (i.e., from the data link layer to the application layer); (ii) line-rate packet-level analysis based on unique application-specific integrated circuits (ASICs); (iii) intrinsic integration of network monitoring and traffic engineering in the same hardware pipeline; and (iv) transparent deployment requiring minimal changes to the existing infrastructures.

Data Structure to Compress Smart Meter Measurements. To facilitate data storage and upload, a protocol stack known as Device Language Message Specification initiated the effort to standardize the configuration and format of smart meter measurements. The protocol stack includes methods to reduce the amount of exchanged and stored data, e.g., replacing the repetitive data value with a NULL byte or encoding value changes. As utility providers begin to collect data more frequently, the effectiveness of these methods downgrades significantly. Consequently, current studies have enhanced these data compression methods with Generalized Deduplication [12]. This new algorithm decouples a data value into basis and deviation, increasing the data compression rate by storing the basis value once while tracking deviation values at runtime. This data compression is lossless, requiring restoration before using the data for power grid applications, such as state estimation.

B. Related Work

In the evolving landscape of cyber-physical defense solutions, the dichotomy between CPU-based and hardware-accelerated or P4-based methodologies has significantly influenced data processing and management strategies. CPU-based systems can be broadly categorized into *model-based* and *data-driven* approaches. The model-based approach employs predefined theoretical algorithms to address a variety of computational tasks, focusing on optimizing even compressed storage. Our previous research developed an optimization-based network management scheme using software-defined networking (SDN) to quickly restore connectivity and observability in phasor measurement unit (PMU) networks, ensuring efficient control and monitoring of power systems [14]. Additionally, we explored optimization-based rule compression techniques to further reduce the number of rules required, enhancing the efficiency and scalability of the system [15]. Despite its efficiency, it often faces challenges with real-time applications due to delays and generally overlooks the intricacies of network and application layers, potentially compromising responsiveness and adaptability in dynamic environments.

Conversely, data-driven methods excel in handling real-time data through sophisticated statistical and machine learning techniques, making them adept at recognizing patterns and anomalies swiftly. Siniosoglou et al. proposed a system that employs a unique combination of Autoencoder and Generative Adversarial Network (GAN) architectures, designed to detect operational anomalies and classify various types with high accuracy [16]. Hannan et al. propose a framework that combines Vector AutoRegression (VAR) for event detection and machine learning for event classification, aimed at enhancing electric grid reliability by analyzing PMU data [17]. However, these methods typically do not address throughput at the line rate, a critical element in maintaining network integrity and performance.

P4-based in-network processing stands out by enabling dynamic network task management. It programs devices to respond instantly to traffic flows, making it exceptionally suitable for scenarios such as DDoS attack mitigation [9]–[11]. These studies showcase P4's effectiveness in high-performance and adaptive security mechanisms but often

TABLE I: Contributions Compared to Existing Security Solutions in the Related Work

Solution	Line-rate Detection (✓/✗/◐)		Compressed Storage	Lossless Storage
	Network Layer	Application Layer		
Model-based [14], [15]	✗	✗	✓	✓
Data Driven-based [16], [17]	✗	✗	✓	✓
P4-based (Network Focus) [8], [9], [11], [18], [19]	✓	◐	✓	✗
Our Approach (MDA, ICS Focus)	✓	✓	✓	✓

focus mainly on flow information, neglecting application-layer information.

Addressing existing limitations, our solution integrates network and application layer data analysis by examining both headers and payloads. This integrated architecture employs P4-based processing for real-time responsiveness while implementing data compression methods for efficient memory utilization. This collaborative approach ensures swift, precise, and proactive cybersecurity measures within power grids.

III. System Design

We present a design leveraging line-rate programmable data planes for high-performance intrusion detection against attacks compromising network availability and control operations in power grids (Figure 1). Today's programmable network data planes are generally enabled by P4 architecture primitives, which differ from general CPU architectures. Consequently, the core of our method includes the design of a new data structure, which we refer to as *Meter Data Accumulator* (MDA), enabling the compact storage of analog meter data and effective intrusion detection in grid operations. On top of MDA, we fine-tune intrusion detection to operate entirely within programmable network devices. This enables the identification of critical attacks in real-time as network packets traverse their pipelines, as illustrated by the *in-network intrusion detection* component in Figure 1. Combining both MDA and intrusion detection components, we achieve in-network security solutions that can correlate knowledge from both network and application layers, a feature critical to identifying anomalies in power grids.

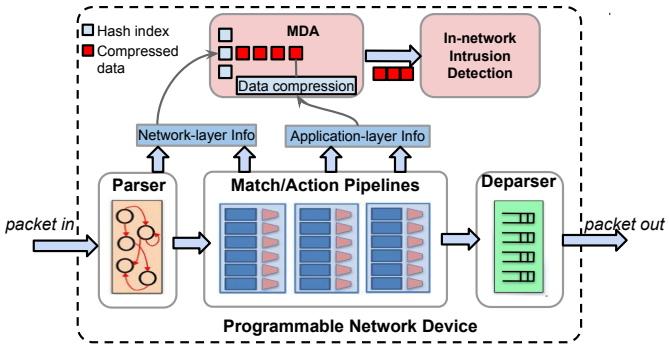

Fig. 1: Design of leveraging MDA to enable P4-based in-network anomaly detection.

A. Meter Data Accumulator (MDA)

As shown in Figure 2, MDA is our newly designed data structure that enables (i) compact storage of a series of meter data directly in programmable data planes and (ii) accurate intrusion detection to reveal anomalies related to computer networks and power grid control applications. To achieve this objective, we employ a two-dimensional array with m columns and d rows. MDA groups measurements from the same sensor (i.e., presenting a specific physical property) into the same row. Therefore, we can store up to m measurements for each sensor in a specific period.

Indexing Data Source. To distinguish measurements from different sensors, we use the hash of the 5-tuple network information, i.e., source IP, source port, destination IP, destination port, and transport layer protocol, as an index to select a row in MDA. General-purpose network environments, e.g., cloud data centers, commonly use the 5-tuple hash to identify network flows because it can map variable-length network attributes into a fixed-width index, enabling efficient storage and rapid access to the stored information.

However, the hashing algorithm suffers from the possibility of collisions, where different 5-tuple combinations may map to the same index. For example, in our implementation, we used the CRC32 hashing algorithm, which maps 5-tuple information into a 32-bit hash value. Accordingly, $2^{32/2} = 65,536$ network sessions can result in approximately 50% probability collision. While having $65,536$ network sessions in a data center is common, power grid networks would introduce a small chance of collision due to three reasons: (i) a meter or a sensor rarely creates more than one network session to deliver measurements periodically; (ii) network devices are deployed in distributed power substations, which normally contains not more than a few thousands of sensors, a number too small

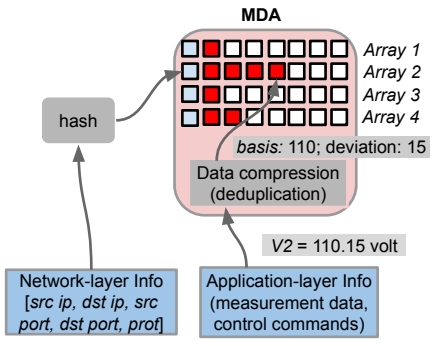

Fig. 2: Structure of meter data accumulator (MDA).

to cause high probability collision; and (iii) power grid communication networks often support hierarchical infrastructure, further reducing the number of devices that a single network device may need to connect.

Compressing Data Values. In power grids, measurement data are encoded in floating points to preserve computational accuracy. Directly storing those values in MDA can be very storage-consuming. Therefore, we exploit and retrofit the Generalized Deduplication (GD) [20] procedure, which is commonly used to compress analog meter data defined by various network protocols based on hardware primitives available in P4 architecture. Hence, we can achieve compact storage directly within network devices without losing data accuracy. GD operates by splitting a data value extracted from incoming packet payload into two portions, i.e., a "basis" and a "deviation." The "basis" portion undergoes deduplication and will be stored only once in the array entry in MDA, effectively removing repetition and reducing data transmission loads. The "deviation" portion is stored directly for each observed data, maintaining the accuracy of the measurement. Implementing GD requires fundamental arithmetic operations, which can be fully implemented by P4 programming language and executable in the Tofino platform.

To optimize MDA's performance, we need to adjust its design parameters, e.g., the bit-length of the hash value or the number of stored measurements for each sensor, according to the requirements of power grid applications and the configurations of network devices (i.e., the size of memory and registers). For example, during benign operating hours, we can store a measurement for a significant period to save storage space. While power grids experience dramatic changes (e.g., due to attacks or accidents), we can store data at a higher frequency to accurately analyze the grid behavior. Automatically tuning MDA's parameters can be encoded in the control planes connecting to programmable network devices according to observed grid states, which we will leave as future work.

B. In-network Intrusion Detection

Because MDA captures the essential information from meter data, we can achieve accurate intrusion detection on anomalies found in both network and application layers by fine-tuning existing machine-learning techniques based on MDA's design parameters. To demonstrate this advantage, we employ decision trees as an example, a powerful machine learning technique that can learn patterns and rules from labeled data and make predictions on new, unseen data. Its recursive structure requires a small number of computational methods, making it an attractive solution to be implemented within network switches [21].

1) Data Preprocessing and Feature Selection: We use different decision trees to process network layer information, e.g., the number of packets observed per time unit, to detect denial of service attacks, and application-layer information, e.g., meter data values, to detect false data injection attacks. While data preprocessing can reduce the training overhead and improve the performance of decision trees, it is interesting that MDA has already performed preprocessing in its data compression procedure.

For example, in Figure 3, we compare two decision trees, all attempting to detect the anomaly of voltage magnitude deviating from a safety margin (i.e., $[109.5, 110.5]$). The left tree, relying on the original voltage magnitude, can include many branches leading to the decision related to the 110-volt baseline value. In the right tree, however, we can directly work around the new decision variables (i.e., 0) by dividing the voltage magnitude into the basis and deviation portions. We ignore unnecessary decisions related to basis values applied to each sampled data set. We can use a smaller tree to identify fine-grained changes in relevant features, such as the frequency of measurement values falling into specific ranges and the temporal distribution of measurements.

More advanced feature selection techniques, like information gain or principle component analysis, are not sensitive to baseline values and thus can be directly applied to the deviation portion without sacrificing analysis accuracy. For this impact, we will leave it to future work.

2) Training the Decision Tree Model: We can easily append each included array with a label to make MDA ready to train a machine-learning model, introducing negligible storage overhead. Unlike neural networks, decision trees involve a comparatively small set of parameters. Consequently, it becomes feasible to directly perform the training within network switches by using data stored in MDA. Specifically, to train decision trees for anomaly detection, the algorithm selects the feature and threshold that best separates the normal and anomalous instances using splitting criteria, such as Gini impurity or information gain, at each tree node.

3) Real-time Anomaly Detection: The prediction process in a decision tree involves iterating through each node and making decisions based on the trained parameters. Because the decision trees are inherently implemented with network pipelines based on P4's architecture hardware, we can directly determine whether an incoming packet includes malicious data, which are stored accordingly. As a result, we can make real-time anomaly detection and attack prevention in the same network pipeline, allowing system operators to understand the

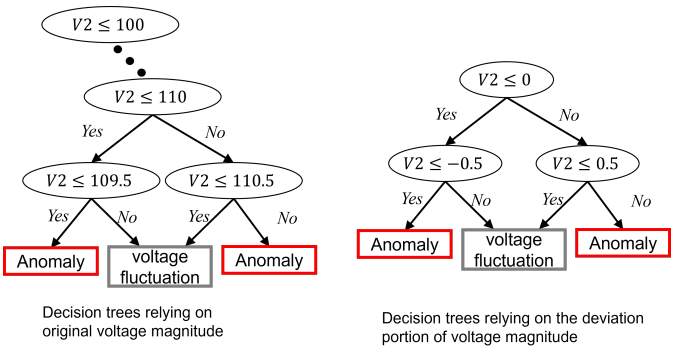

Fig. 3: Example decision tree applying to the original and the deviation portion of a voltage magnitude.

attack logic and preventing physical damage from happening simultaneously.

IV. Evaluation

We implemented an intrusion detection system on P4 switches, employing MDA for compact storage of analog meter data. Through a comprehensive evaluation, we assessed the system's performance overhead, memory saving, and detection accuracy against common security threats, including DDoS and FDIA attacks.

A. Experiment Setup

Testbed. We implemented our system on a hardware testbed consisting of an Aurora 610 network switch with P4 programmable pipelines, connected to two servers running 64-bit Linux OS (Ubuntu 20.04). Each server boasts dual 64-core processors, 1 TB of RAM, and a 12-TB hard disk drive capable of sustaining a data transfer rate of 248 MB/s. Additionally, the servers are equipped with NVIDIA Quadro P400 GPUs and network interface cards.

Evaluation Metrics. To evaluate overhead, we measure the per-packet processing time as the difference between the *ingress_global_timestamp* and *egress_global_timestamp* measured on the P4 switch for each packet. For memory storage savings, we use the compression ratio: $\left(\frac{\text{compressed data size}}{\text{original data size}}\right) \times 100\%$. For attack detection performance, we use (1) Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$, the percentage of correctly detected attacks; (2) Precision $= \frac{TP}{TP+FP}$, the proportion of true positive predictions among all positive predictions; (3) Recall $= \frac{TP}{TP+FN}$ (also known as sensitivity), the proportion of true positive predictions among all actual positive instances; and (4) F1 Score $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$, the harmonic mean of precision and recall. TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

B. Performance Overhead

We evaluate the overhead of our P4-based attack detection system with data compression on our hardware testbed. Each experimental run involved transmitting 3,000 Distributed Network Protocol 3 (DNP3) packets. To gauge the system's robustness, we varied the payload size of the DNP3 packet from 8 bytes to 250 bytes, aligning with the protocol's maximum data payload size. It is worth noting that the maximum length of a link layer frame is 292 bytes when considering CRC and headers.

Figure 4 compares the average per-packet processing time in two scenarios: simple forwarding and integration with our attack detection module. The x-axis represents packet payload sizes, ranging from 8 to 250 bytes, while the y-axis denotes processing time in microseconds (µs). Firstly, the P4 switch data plane exhibits rapid packet processing capabilities, operating at hardware line rates within microseconds, regardless of the enabled detection module. Secondly, processing time scales linearly with packet payload size for both scenarios. Thirdly, results show a marginal increase in processing time, averaging approximately 0.01%, when the security module is activated. Specifically, enabling the detection module results in a minimal increase in processing time. As illustrated by the green dotted line in Figure 4, the time difference between simple forwarding and the detection module ranges from 1 to 20 nanoseconds.

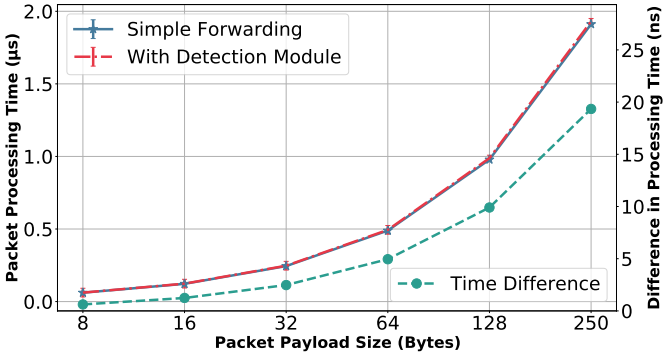

Fig. 4: Comparison of packet processing time with and without the attack detection module.

Although the maximum data payload size for DNP3 is 250 bytes, we extended the payload size to 64k to assess the effectiveness of our data compression scheme, applicable to other existing or future protocols. Despite the increase in payload size, the overhead remains relatively marginal. For instance, only a discernible 7.1% difference is observed for a packet size of 10,000 bytes. This minimal overhead underscores the module's effectiveness in preserving processing speed akin to simple forwarding while offering security functions on the network data plane.

C. Data Compression

Table II summarizes the data compression rates for the eight datasets analyzed in our study. Integrating data compression techniques into the P4 programmable switch enables substantial data storage enhancements, with compression ratio ranging from 32.1% to 60.9%. This robust performance allows for increased data retention within the same storage space, thereby reducing the need for frequent offloading.

D. Attack Detection Accuracy

1) Distributed Denial of Service (DDoS) Attack: To evaluate the effectiveness of our proposed P4-based attack detection system incorporating data compression, we first present a case study focusing on DDoS attack detection. Our experimental setup involved a DNP3-based network with a P4 switch at the center of a star-shaped topology, comprising one DNP3 data aggregator and five DNP3 field devices. Four devices

TABLE II: Data Compression Ratios

Dataset	Compression Ratio (%)	Dataset	Compression Ratio (%)
#1	60.90	#5	32.13
#2	56.63	#6	43.80
#3	45.34	#7	52.91
#4	59.63	#8	40.47

sent normal measurements (21-byte packets at one packet per second) to the aggregator. One device was compromised, generating high-volume traffic of up to 10 Mbps for an event buffer flooding attack. The data aggregator had a limited buffer size capable of storing 1000 packets. On the P4 switch, we developed a decision tree model using specific features outlined in Table III. This model played a crucial role in analyzing traffic patterns and identifying anomalies indicative of DDoS attacks.

Figure 5 compares the detection accuracy with and without data compression across eight distinct datasets. The vertical axis represents accuracy as a percentage. The blue bar indicates accuracy without data compression, while the red bar signifies accuracy with data compression. Notably, data compression enables high detection accuracy for all datasets ranging from 86.1% to 95.6%. The decrease in accuracy due to data compression is relatively small, ranging from 1.08% to 3.22%.

Our solution not only effectively detects DDoS attacks but also enables rapid mitigation through the programmable capabilities of P4. Upon detecting an attack, the network switches can dynamically update their forwarding rules to drop or redirect malicious traffic, ensuring the continuity of legitimate traffic flow. Our findings demonstrate that the P4-based solution, enhanced with data compression, significantly maintains network performance while effectively detecting and mitigating DDoS attacks.

2) False Data Injection Attack: Our second case study focuses on the detection of false data injection attacks (FDIA). FDIA is a critical security threat in which an attacker manipulates the sensor measurements or control commands, potentially leading to harmful consequences, such as incorrect state estimations, meter readings, or control signals, leading to failure to report system anomalies, incorrect system control decisions, false alarms, and even equipment damage or power outages [22].

We simulated a Supervisory Control and Data Acquisition (SCADA) System network comprising five DNP3 outstations and one DNP3 master station. The network traffic included both normal operational data and maliciously crafted data packets representing FDIA attempts. Our P4-based security solution, integrated with data compression, utilized a decision tree model to pinpoint suspicious data patterns and deviations from expected behavior.

Figure 6 shows the comparative performance of our detection module implemented in two different platforms: the baseline system, implemented using Python on a PC, and the P4 Security Solution implemented in the p4 switch, over

TABLE III: Decision Tree Model Features

Network Feature	Network Feature	Power System Feature	Power System Feature
srcPort	avgFlowBytes	VoltageMean	VoltageDiff
dstProt	counterPacket	VoltageMin	VoltageMagnitude
freqIIN	freqFuncCode	VoltageMax	VoltageChangeRate

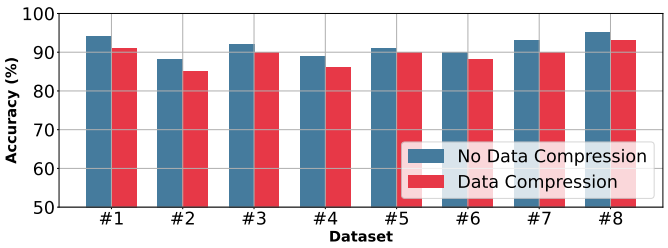

Fig. 5: Comparison of DDoS detection accuracy with and without data compression.

a timeline from 0 to 4032. The anomaly detection scores, which are the aggregation of the predictions of multiple decision trees and their weights, are plotted for both systems with the baseline depicted in blue and the P4 solution in red. The Baseline method exhibits greater consistency in its anomaly detection scores, maintaining a relatively stable and low variability throughout the observed time period. Conversely, the P4 Security Solution shows more significant variability, which could indicate a more sensitive or complex method that perhaps tries to adapt to changing data patterns over time. Despite this variability, both methods register a significant increase in anomaly detection scores at time index 3232, marking a clear response to an event, the launch of a False Data Injection Attack (FDIA), which is generated based on the theorem from [22]. This sharp increase demonstrates the capability of both methods to recognize and react to a significant cyber-security threat within the system.

Figure 7 compares key machine learning performance metrics for models trained with and without data compression. Notably, while data compression resulted in a consistent degradation in model performance, it is essential to highlight that the difference is small. Accuracy, for instance, experienced the smallest relative drop, declining from approximately 0.890 to 0.865, a decrease of about 2.81%. Meanwhile, precision, recall, and F1-score exhibited slightly larger performance gaps between compressed and uncompressed conditions. Specifically, precision decreased by approximately 3.19%, recall by approximately 3.50%, and the F1-score by about 3.01%.

The results indicate that although data compression does affect model performance, the extent of this impact is

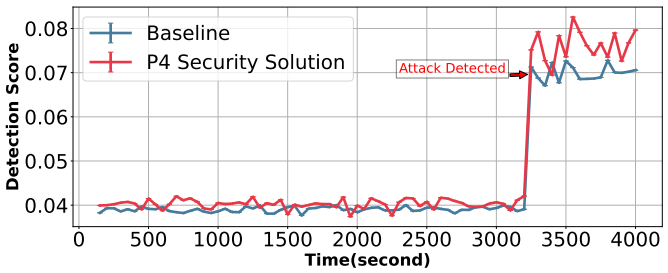

Fig. 6: Comparative analysis of anomaly detection scores over time.

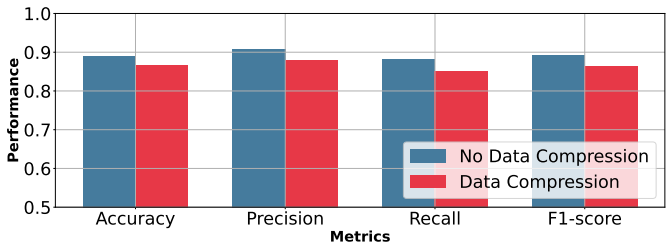

Fig. 7: Comparative performance of the FDIA detection module with and without data compression.

minimal, with all metrics remaining above 0.851 even under compression. This implies that data compression could be a viable approach for substantially reducing storage and computational costs, with only a slight impact on key performance measures. Moving forward, we plan further to explore the tradeoffs between compression ratio and model performance to achieve an optimal balance.

Limitations and Discussion. Enhancing performance for specific attacks may require redesigning and redeploying customized structures, increasing system complexity and development time. Additionally, the range of detectable attacks is limited by the localized view based on traffic inspected by P4 switches. This limitation could be mitigated by integrating a centralized controller or enabling inter-switch communication through customizable and stateful protocols, providing a more comprehensive network view and improving detection accuracy.

V. Conclusion and Future Works

We utilize line-rate programmable data planes for high-performance intrusion detection in smart grid operations. Our approach includes a novel data structure optimized for time series data and integrates data compression techniques. This enables efficient in-network security solutions, correlating insights from both network and application layers, which are essential for detecting DDoS and false data injection attacks.

In our future work, we aim to expand the scope of our research by integrating P4 switches with communication networks and power system simulators to evaluate their performance and usability across diverse cyber-physical scenarios. Additionally, we plan to enhance our anomaly detection capabilities by incorporating advanced machine learning algorithms, such as Long Short-Term Memory (LSTM) networks and Vector AutoRegressive (VAR) models. Furthermore, we seek to broaden our detection capabilities to encompass a broader range of attack vectors by integrating semi-supervised learning and data augmentation strategies. These initiatives will bolster the adaptability, resilience, and efficacy of our framework against evolving threats in smart grid operations.

Acknowledgment. This work was supported by the National Science Foundation (NSF) under Grant CNS-2247721, CNS-2247722, CNS-2144513, and EEC-2113903. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of NSF.

References

[1] Y. Zhang, L. Wang, W. Sun, R. C. Green II, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Transactions on Smart Grid*, 2011.
[2] G. Gilchrist, "Secure authentication for DNP3," in *Proc. of 2008 IEEE Power and Energy Society General Meeting*, 2008.
[3] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera, "DNPSec: Distributed network protocol version 3 (DNP3) security framework," in *Advances in Computer, Information, and Systems Sciences, and Engineering*, 2006.
[4] A. Goyal, H. Daumé, and G. Cormode, "Sketch algorithms for estimating point queries in NLP," in *Proc. of EMNLP-CoNLL '12*, 2012.
[5] S. Geravand and M. Ahmadi, "Bloom filter applications in network security: A state-of-the-art survey," *Computer Networks*, 2013.
[6] M. Zhang, G. Li, S. Wang, C. Liu, , A. Chen, H. Hu, G. Gu, Q. Li, M. Xu, and J. Wu, "Poseidon: Mitigating volumetric ddos attacks with programmable switches," in *Proc. of NDSS 2020*, 2020.
[7] H. Zhou and G. Gu, "Cerberus: Enabling efficient and effective in-network monitoring on programmable switches," in *Proc. of the 2024 IEEE Symposium on Security and Privacy (SP)*, 2023.
[8] Z. Liu, H. Namkung, G. Nikolaidis, J. Lee, C. Kim, X. Jin, V. Braverman, M. Yu, and V. Sekar, "Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric DDoS attacks with programmable switches," in *Proc. of USENIX Security 21*, 2021.
[9] J. Xing, Q. Kang, and A. Chen, "Netwarden: Mitigating network covert channels while preserving performance," in *Proc. of USENIX Security 20*, 2020.
[10] D. Barradas, N. Santos, L. Rodrigues, S. Signorelloy, F. M. V. Ramos, and A. Madeira, "Flowlens: Enabling efficient flow classification for ML-based network security applications," in *Proc. of NDSS 2021*, 2021.
[11] H. Zhou, S. Hong, Y. Liu, X. Luo, W. Li, and G. Gu, "Mew: Enabling large-scale and dynamic link-flooding defenses on programmable switches," in *Proc. of 2023 IEEE Symposium on Security and Privacy (SP)*, 2023.
[12] R. Vestergaard, Q. Zhang, and D. E. Lucani, "Generalized deduplication: Bounds, convergence, and asymptotic properties," in *Proc. of 2019 IEEE GLOBECOM*, 2019.
[13] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, 2014.
[14] H. Lin, C. Chen, J. Wang, J. Qi, D. Jin, Z. T. Kalbarczyk, and R. K. Iyer, "Self-healing attack-resilient PMU network for power system operation," *IEEE Transactions on Smart Grid*, 2018.
[15] Y. Qu, G. Chen, X. Liu, J. Yan, B. Chen, and D. Jin, "Cyber-resilience enhancement of PMU networks using software-defined networking," in *Proc. of 2020 IEEE SmartGridComm*, 2020.
[16] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, and P. Sarigiannidis, "A unified deep learning anomaly detection and classification approach for smart grid environments," *IEEE Transactions on Network and Service Management*, 2021.
[17] C. Hannon, D. Deka, D. Jin, M. Vuffray, and A. Y. Lokhov, "Real-time anomaly detection and classification in streaming PMU data," in *Proc. of 2021 IEEE Madrid PowerTech*, 2021.
[18] G. Zhou, Z. Liu, C. Fu, Q. Li, and K. Xu, "An efficient design of intelligent network data plane," in *Proc. of USENIX Security 23*, 2023.
[19] D. Barradas, N. Santos, L. Rodrigues, S. Signorello, F. M. Ramos, and A. Madeira, "Flowlens: Enabling efficient flow classification for ML-based network security applications." in *Proc. of NDSS 21*, 2021.
[20] M. Fehér, N. Yazdani, M. T. Hansen, F. E. Vester, and D. E. Lucani, "Smart meter data compression using generalized deduplication," in *Proc. of GLOBECOM 2020*, 2020.
[21] W. Wang, M. Moshref, Y. Li, G. Kumar, T. S. E. Ng, N. Cardwell, and N. Dukkipati, "Poseidon: Efficient, robust, and practical datacenter CC via deployable INT," in *Proc. of NSDI 23*, 2023.
[22] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, 2011.