Intrusion Detection in High-Performance Computing

Qumeng Sun

April 1, 2024

Supervisor: Marcus Merz

Abstract

This study ventures into evaluating various intrusion detection systems (IDS) within High-Performance Computing (HPC) environments, utilizing the GWDG Cloud platform as the testbed. Through a series of experiments, this paper assesses performance bottlenecks, the efficacy of different IDS solutions, and the potential of machine learning (ML) models to enhance intrusion detection capabilities. Experiment 1 focuses on identifying the speed bottlenecks of the Suricata plugin, revealing that network interface and CPU, rather than memory or disk IO, are the primary limitations. Experiment 2 evaluates the Slips IDS, showcasing its machine learning component's effectiveness in detecting network threats through behavioral analysis. Experiment 3 further explores the effectiveness of various ML models using the CIC-IDS2018 dataset, demonstrating that specialized ML models can achieve high accuracy in intrusion detection. The findings suggest that Suricata provides robust detection capabilities, whereas Pigasus offers a more holistic solution balancing efficiency and effectiveness. However, the integration of ML into IDS, while promising, faces challenges from the current state of datasets and infrastructure bottlenecks. The study contributes to the IDS field by highlighting the need for better datasets and exploring ML's role in advancing IDS technologies in HPC environments.

Keywords: high-performance computing (HPC), intrusion detection systems (IDS), machine learning, Suricata, Pigasus, GWDG, cybersecurity

Contents

1 Introduction

With the explosive growth of artificial intelligence, the infrastructure supporting these large models, as well as research into them, has gained significant exposure[Pat21]. In particular, the stock price of Nvidia, a leading brand of graphics cards for deep learning, has doubled over the past year[Cin24]. The construction of more supercomputing centers around the world has accelerated since this surge in interest, necessitating these high-performance computing (HPC) centers to be safeguarded to prevent wastage of computational resources and to ensure the security of user data[Fac23].

Intrusion detection is not a novel concept; however, not all traditional intrusion detection systems seamlessly integrate into HPC environments. Current HPC intrusion detection can generally be categorized into two main types: those based on network traffic and those based on Trusted Executable Environment(TTE). According to Guo et al. [GCW+24], one of the unique challenges of HPC, compared to traditional systems or cloud platforms, is the prioritization of performance by its users, often at the expense of security. Therefore, it is crucial to ensure that solutions do not impose significant performance degradation. The architecture of security measures required varies significantly across different types of systems.

An HPC system, composed of access, storage, computation, and management nodes, each presents its own vulnerabilities, such as file integrity issues on storage nodes, user authentication issues on access nodes, mining activities on computation nodes, and illegal privilege escalation on management nodes. Terms like HPC, enterprise networks, data centers, infrastructures, scientific DMZs, and supercomputers sometimes overlap [Wil22, RCdF+21, GMG+20, TFD+19].

In this paper, we embark on a comprehensive exploration of intrusion detection systems (IDS) within the context of High-Performance Computing (HPC) environments, with a particular focus on the GWDG Cloud platform. Through a dual approach that includes both a rigorous literature review and targeted experiments, we aim to theoretically and empirically compare various IDS methodologies to ascertain their suitability for deployment in advanced computing networks, similar to those employed by GWDG. Our investigation illuminates the critical role of network interface performance ceilings in influencing IDS efficiency within such systems. Moreover, it highlights the potential of integrating machine learning (ML) technologies into IDS frameworks, particularly through the training of specialized models, as a promising avenue for enhancing IDS functionality. In the realm of HPC environments, our findings underscore the paramount importance of an IDS's capability to manage substantial network traffic volumes over the precision of its detection mechanisms [BJS19a].

The principal contributions of this study are delineated as follows:

1. **Comparative Analysis:** Through a meticulous comparison of IDS methodologies, this paper identifies and evaluates the inherent strengths and limitations of each approach. This analysis offers valuable insights into the most effective IDS solutions for HPC environments, contributing to a nuanced understanding of the ideal strategies for intrusion detection.

2. **Empirical Experimentation:** By conducting a series of practical experiments, we bridge the gap between theoretical expectations and real-world application. These experiments not only validate certain theoretical predictions but also unearth practical considerations and challenges, thereby enriching the research with empirical data and observations.

This paper aims to advance the field of cybersecurity within HPC environments by providing a comprehensive analysis of current IDS solutions, exploring the integration of ML technologies, and presenting empirical evidence to guide future developments in intrusion detection.

This paper first analyzes and compares the most popular methods and datasets across three categories: (1) Network-based Intrusion Detection Systems (NIDS), (2) Machine Learning-based Detection Systems, and (3) Datasets. Following this, it delves into the experimental section, which includes three experiments focused on Suricata, Slips, and AutoGluon (AG). Finally, the paper discusses the topic based on data obtained from experiments and insights from the literature review. All codes related to this report can be found on GitHub at `https://github.com/Mike-7777777/hpcsa24`.

2 Background

High-Performance Computing (HPC) systems are pivotal in advancing computational science, supporting large-scale simulations, data analysis, and artificial intelligence applications. As these systems become increasingly integral to research and industry, ensuring their security against various forms of cyber threats has become paramount. This paper delineates the landscape of intrusion detection within HPC environments, categorizing the methods into two main types: network-based and execution environment-based intrusion detection.

Network-based Intrusion Detection Systems (NIDS) are crucial for analyzing network traffic packets, serving as the first line of defense in HPC infrastructures. These systems can be categorized into three main directions: signature-based IDS, anomaly-based IDSincluding statistical, knowledge-based, and machine learning methods for classifying legitimate and anomalous behaviorsand hybrid methods that combine signatures and anomalies, often incorporating machine learning algorithms like decision trees. However, there is a noted lack of data analysis for high-traffic volumes, making it challenging to extend these methods to HPC contexts [KGV+19, DM21, AAB22].

Network segmentation, as mentioned by Guo et al. [GCW+24], represents a widely adopted strategy for safeguarding access nodes. Placing an Information Event Management (SIEM) system outside the main network facilitates real-time monitoring of network traffic without degrading network performance, typically necessitating one or more Network Interface Cards (NICs) along with other hardware components.

Execution environment-based detection encompasses defenses against software and hardware attacks, including cryptojackinga prevalent threat often targeting personal computers and workstations as described by Pott [Akrar, PGE23]. This form of attack, necessitating substantial computational resources, exemplifies the common threats faced by HPC centers, further highlighted by instances of malicious activities reported by EGI [EGI20], Google Cloud [Goo21], and Tahir et al. [THD+17].

The prevalence of attacks on HPC systems is not to be understated, with significant incidents reported against academic institutions and research facilities [Gri20, Kon24, Ges23]. This underscores the imperative for robust security governance within HPC environments. Emphasizing security awareness, employing advanced security technologies, and fostering secure user interactions are essential strategies recommended by Heymann [HMA+23]. The adoption of eduMFA by GWDG for enhancing the reliability of user authentication at access nodes [Fre24], along with regular security awareness events [Tec23], illustrates the ongoing efforts to bolster security posture. However, there remains a critical need for more comprehensive documentation and transparency to facilitate further research and understanding of HPC security challenges.

Node sanitation, as advocated by Guo et al. [GCW+24], plays a pivotal role in minimizing the risk of information leakage, particularly for compute nodes. Best practices suggest integrating security requirements into the initial design phase of HPC infrastructure rather than as an afterthought. Tailoring security controls for different node types allows for more efficient threat detection using fewer resources.

Contrasting with personal computers or notebooks, HPC systems represent a complex and large-scale computing paradigm. Ensuring their security poses distinct challenges, necessitating a nuanced approach that incorporates both network-based and execution environment-based intrusion detection mechanisms to address the myriad of potential cyber threats.

3 Comparison of existing methods and techniques

This section introduces and compares three key topics: high-performance NIDS, machine learning-based NIDS, and NIDS datasets.

3.1 Comparison of High-Performance NIDS

In the study of high-performance NIDS, many researchers have found the generation of large traffic volumes challenging, often opting for some level of simulated generation or replication of traffic rather than capturing purely real network traffic due to the difficulty of accessing networks with constant 100Gbps traffic flow. To address HPC environments, a solution capable of capturing high-speed network traffic is required, with 100Gbps being a relatively safe benchmark discussed in several articles. [App19] mentions that the overall network traffic of a university rarely reaches 80Gbps.

Traditional traffic capture software tends to drop packets when faced with large volumes of data, significantly reducing accuracy [BJS19a, BJS19b, SI18]. Tools like Zeek [Zee24] require substantial CPU resources and multiple Zeek instances for capturing large traffic volumes, similar to Snort. Comparatively, Suricata shows better performance, with various studies noting its efficiency after optimization and parallel processing across multiple nodes, which allows for relatively larger traffic capture at the expense of more resources:

- AF_XDP offers a shortcut in Linux systems for accelerating the transfer of network traffic, enabling the pre-emptive filtering of unwanted traffic at a significantly lower cost [dPPH24].
- The optimal configuration for unleashing Suricata's best performance, as discussed in [PM16] and its subsequent version [PM17] utilizing XDP technology, demonstrated traffic handling capabilities up to 20Gbps. [App19] further pushed Suricata's capabilities beyond 80Gbps.
- The most advanced approach is described by [TFD+19], closely resembling the scenario at GWDG, where they collected and tested their architecture on Purdue University's campus network, thus limiting project transparency. They employed Zeek for data collection and processing, adopting a similar strategy to [SSK15]. Their setup included 2 SPAN ports and multiple (8) Test Access Points (TAPs), along with a Zeek cluster consisting of 8 Zeek instances, facilitating out-of-band analysis for the security of Purdue University's campus network. They also utilized the ELK Stack [BKS19, AK23] for event and alert information storage and visualization, with another popular choice being the TIG Stack [ale24].

Beyond stacking instances for network processing capability, some papers have attempted various methods to efficiently filter out the majority of traffic to improve efficiency [ZSA+20, Zha21, WGBD22, Gus19, TFD+19, SLG18].

- [DBM+18] utilized a hybrid approach, where content not directly identifiable by SIDS was analyzed by AIDS, filtering with the most resource-friendly methods before tackling the remaining challenges with more resource-intensive solutions.
- [ZSA+20, Zha21] proposed a SmartNIC-based method supporting FPGA, achieving 100Gbps traffic processing with a single CPU and FPGA core, a significant improvement in power optimization compared to needing over 100 CPUs for the same processing power in 2015 [SSK15]. Pigasus optimized for common scenarios, allowing 95% of traffic to be accelerated; it also adjusts strategies for different compile-time and run-time scenarios. Pigasus implemented several robust, unique, and innovative techniques for this purposeFPGA-first architecture, Fast-slow path Reassembly, and layered pattern matching.
- Similar to [Zha21], [WGBD22] aims to balance efficiency and power consumption by performing simple tasks more frequently and complex tasks less so, supporting scalable data representation for any user-defined analysis function, embodying the concept of filtering at the lower level to dismiss packets not requiring consideration or postponing expensive computations. As a network traffic capturer, it boasts capabilities up to 162Gbps.

As shown in Table 1, among various methods, [Zha21] is the only work that simultaneously possesses low-cost scalability, multi-threading capabilities, and powerful traffic filtering technology. Retina [WGBD22] is another commendable effort, requiring further practical testing to compare their respective advantages and disadvantages.

Table 1: Comparison of Network-based IDS

Name, Year	Dataset	Hardware Needs(100Gbps)	Accessibility	Scene	Type	Filtering
ZeekML [Gus19], 2019	CIC-IDS2017	100 CPUs	Closed source	General	Anomaly-based	Custom, double
Pulsar [TFD+19], 2019	Campus traffic	8 TAP + 2 SPAN + 17 servers	Open source, free	HPC	Signature-based	Whitelist
RUAD [MBC+22], 2022	Campus traffic	N/A	Closed source	HPC	Anomaly-based	N/A
Pigasus [Zha21], 2021	DPDK pktgen	1 CPU + 1 FPGA	Open source, free	HPC	Signature-based	Multi-layer, low latency
Retina [WGBD22], 2022	Campus traffic	8-cores CPU	Open source, free	General	Hybrid	Multi-layer
AutoGluon*, 2024	CIC-IDS2018	N/A	Open source, free	General	Anomaly-based	N/A

3.2 Comparison of NIDS Datasets

As of 2024, the landmark datasets in this field are CIC-IDS2017 and 2018 [SLG18], both from the Canadian Institute for Cybersecurity (CIC), which has also produced other IDS datasets for scenarios like DDoS and IoV (Internet of Vehicles) [KHG23, NTD+24]. CIC's approach involves capturing not real traffic but rather traffic generated by controlled infrastructure and host interactions [ERJ21]. These are far from perfect, with [LGH+22] identifying issues such as label inaccuracies and packet order errors in CIC-IDS2017, highlighting the challenge of validating such large datasets.

[GJ23] conducted a study on CIC-IDS2018, finding that proper feature extraction and training different models for different attack types could achieve accuracy surpassing the state of the art (SOTA) of 99.99% across all types, along with excellent F1 score performance. This suggests that data cleaning and feature extraction are critical aspects for this dataset. The approach of selecting different feature quantities for different attack types and using different classifiers for training offers valuable insights.

Moreover, [HLA23] indicates these datasets are prestigious but come with a significant amount of data requiring cleaning, emphasizing the importance of feature extraction [KGV+19]. [SLP21] integrated popular datasets, including CIC-IDS2018 [SLG18] and UNSW-NB15 [MS15], covering various attack types with over 66% being malicious traffic, alleviating the issue of dataset imbalance. In machine learning contexts, constructing such integrated datasets represents a promising direction.

Table 2: Summary of network traffic datasets.*

Year	Dataset	Source	Attack Types	Size(GB)	Feature Count	Open Source
2015	UNSW-NB 15 [MS15]	Simulation (Special Facility)	9	100	49	Yes
2017	CIC-DIS2017 [SLG18]	Simulation (Infrastructure)	7	51	80	Yes
2018	CIC-DIS2018 [SLG18]	Simulation (Infrastructure)	7	450	83	Yes
[1] 2020	MQTT-IoT-IDS2020 [HBB+20]	Simulation (Special Facility)	4	1.4	43	Yes
2022	5G-NIDD [SSP+22]	University (Real Traffic)	5	3.65	25	Yes
2022	RUAD [MBC+22]	University (Real Traffic)	Unknown	Unknown	41	No
2023	TII-SSRC-23 [HLA23]	Simulation (Small Devices)	4	27.5	75	Yes
2021	NF-UQ-NIDS-v2 [SLP21]	Integrated	21	1.8	43	Yes*

3.3 Comparison of Machine Learning-Based NIDS

[DM21] reviewed methods incorporating machine learning as part of IDS, highlighting the challenge of data imbalance within ML datasets, with the proportion of worm viruses in UNSW-NB15 being only 0.05%. [TL20] also acknowledges this issue, noting the presence of data redundancy and missing data, which could be better structured.

[MBC+22] discusses supervised learning, which requires labeled data, a challenge for HPC environments that cannot easily provide such data, making it effective but unsuitable. Unsupervised learning, not requiring labeled data, can leverage the large amounts of data produced by HPC (though potentially challenged by the rarity of anomalies in generally normal HPC operations), showing less promising performance. The pursuit of the best unsupervised or semi-supervised models continues.

Machine learning-based IDS typically comprises components such as network traffic capture software (e.g., Suricata), feature analyzers (e.g., CICFlowMeter), and machine learning algorithms or models (e.g., SVM). The most successful machine learning methods currently focus on Random Forests, Decision Trees, and KNN [SLG18, SM23, SSP+22]. [SLG18] achieved 98.58% accuracy on the CIC-IDS2017 dataset using a 4-core CPU. [LLH+22] utilized a VAE to obtain a 0.97 F1 score and a 0.98 recall rate, aiming to build a robust ML system. Future directions may involve ML as a standalone method using various data sources. Some work [Ser23] suggests treating each attack type individually, tailoring features, algorithms, and models specifically for each type. Other studies extend the intrusion detection dataset from tabular to time-series data [LLH+22].

The establishment of standard testing setups and procedures is crucial. Given the rapid development in the industrial sector, it is nearly impossible to determine the best methods or hardware over time, but mature testing processes can quickly identify the optimal combinations after addressing core issues. Due to limited time and resources, my current cloud environment is inadequate for testing high-performance NIDS; hence, subsequent experiments will be conducted on the GWDG Cloud platform.

4 Experiments

This chapter is divided into three main parts, each detailing a specific experiment conducted to evaluate the performance of different intrusion detection approaches within a High-Performance Computing (HPC) environment, specifically leveraging the GWDG cloud platform. The experiments are designed to identify performance bottlenecks, compare the speed and efficacy of different IDS solutions, and explore the potential of machine learning models in enhancing intrusion detection capabilities.

4.1 Experiment 1: Speed Bottlenecks of the Suricata Plugin

Telegraf, a tool for capturing system performance metrics within the TIG stack, is responsible for data collection and forwarding to InfluxDB for storage, analysis, and visualization. Among its plethora of plugins is one for Suricata, an intrusion detection system that monitors network traffic. Utilizing Suricata as a network traffic capturer with Telegraf for data transmission to the database allows for real-time or post-analysis and visualization. This integration, facilitated through a Unix Socket shared by both software, includes configurable parameters such as the Unix Socket address, delimiter, version, and alerting capability, effectively linking the robust capabilities of Suricata with the TIG stack for enhanced performance analysis and intrusion detection.

4.1.1 Setup and Preparation

The experiment was conducted on the GWDG Cloud platform, featuring a CentOS 7 system with an 8-core CPU, 16GB of memory, and 150GB of storage. The dataset employed was a segment of CIC-IDS2018, totaling 37.5GB, and necessary software included 'tcpreplay', 'Suricata', and 'Telegraf'. Installation and configuration scripts are available in a dedicated GitHub repository.

Challenges arose from unclear plugin documentation, necessitating the Unix Socket (US) file's placement in a mutually accessible directory with identical path specifications in both software settings and initiating Telegraf to ensure it operates as the socket's receiver, awaiting data transmission. Permission issues were addressed by aligning access rights between Suricata and Telegraf, highlighting the importance of adequate permissions for the creation and reading of the Unix Socket file.

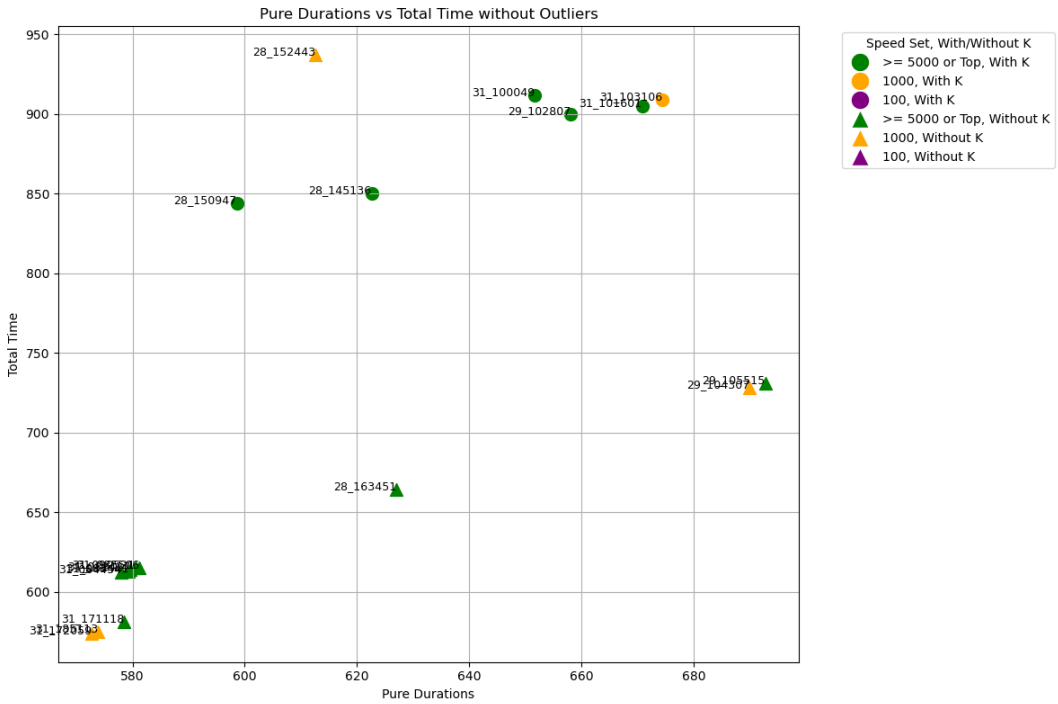

Figure 1: Suricata time cost comparison, illustrating the impact of -K mode on processing speed and memory usage.

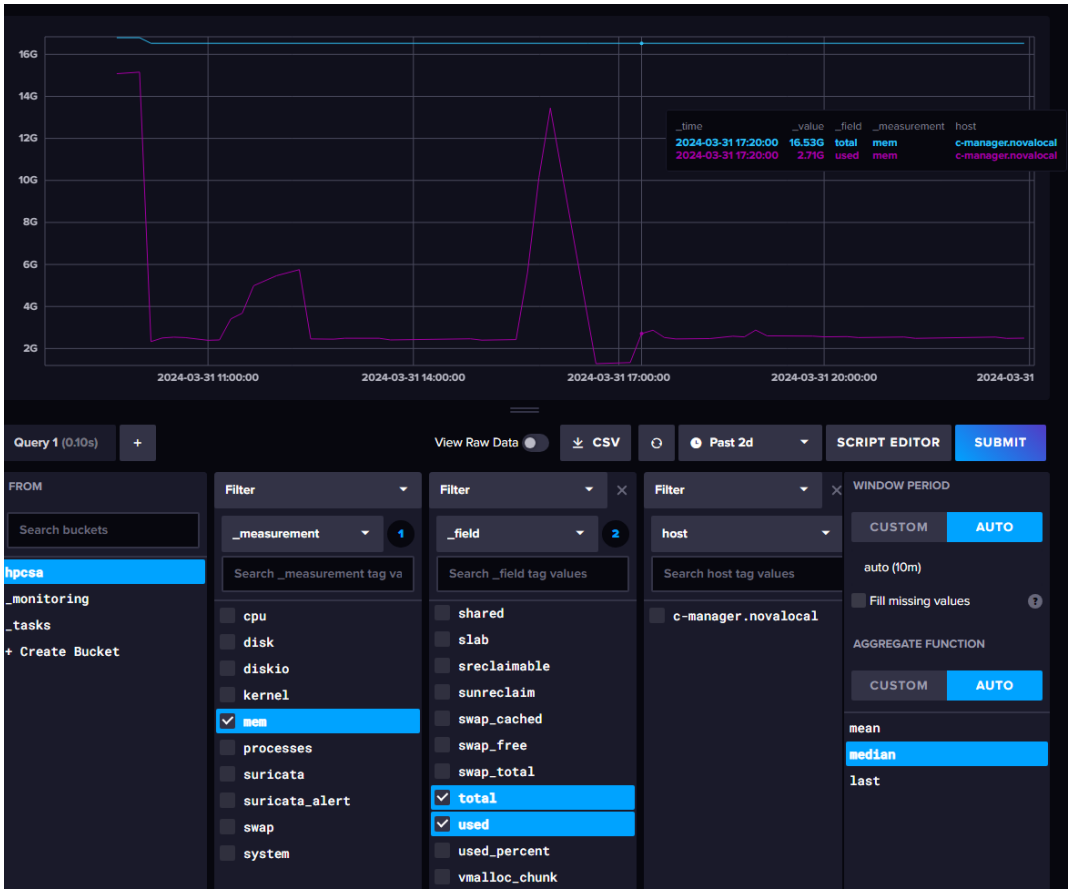

Figure 2: Differential memory usage with and without -K mode during Suricata processing.

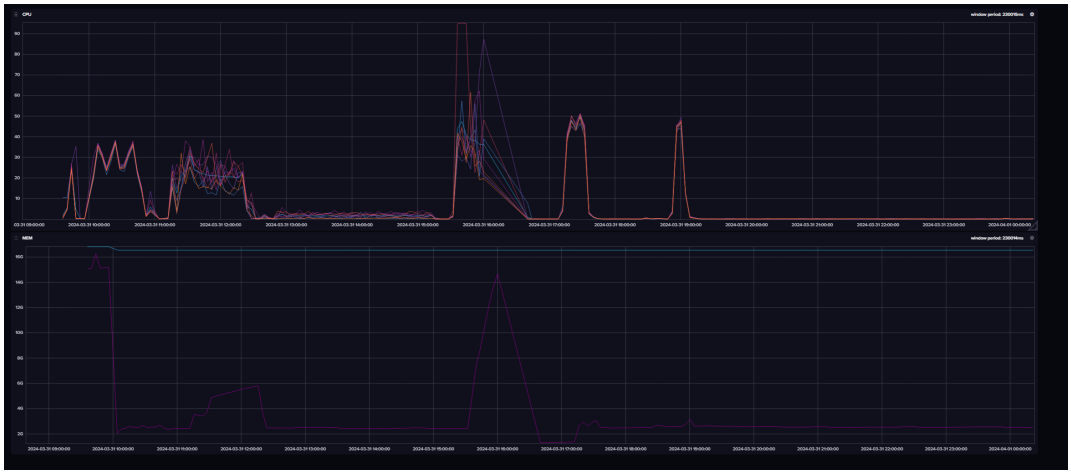

Figure 3: CPU and memory occupancy rates, indicating underutilization despite high workload.

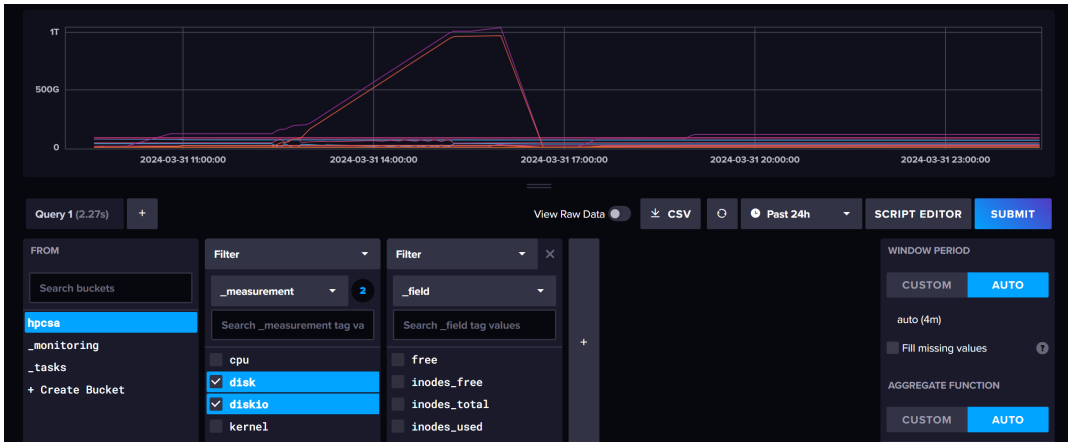

Figure 4: Hard disk and IO status, showing sporadic spikes but general underload.

4.1.2 Process and Observations

Adjustments were needed to accommodate the system's Maximum Transmission Unit (MTU) of 1450, as the data was recorded with a 1500 limitation, leading to packet send errors. This issue was resolved using 'tcpdump' to segment pcap files, ensuring packets exceeding the MTU were truncated. Subsequent replay of the database pcap files through 'tcpreplay' varied in duration from 10 to 50 minutes per trial.

4.1.3 Findings

Experimental results indicated a performance bottleneck primarily attributed to the network card, with secondary limitations from the CPU, rather than memory, disk read/write speeds, or inefficiencies within Suricata itself. Notably, employing '-K' mode for replay did not enhance performance, suggesting that the limitation was not storage media but rather the network interface's throughput. Suricata's processing speed and the impact of system resource utilization under different conditions were thoroughly documented, with figures 2, 3, and 4 illustrating the nuanced impacts of various modes and configurations on memory usage, CPU and memory occupancy rates, and disk IO status, respectively.

This comprehensive analysis supports the hypothesis that within the GWDG cloud environment, Suricata's network monitoring capabilities are primarily hindered by network infrastructure limitations rather than computational resources, underscoring the critical role of network throughput in optimizing intrusion detection performance[TK22, App19].

4.2 Experiment 2: Evaluating the Efficacy of Slips IDS

The objective of this experiment was to assess the Slips Intrusion Detection System (IDS), particularly its machine learning component, for identifying network threats in an HPC context. Slips, known for its adaptive and predictive capabilities in detecting anomalies, represents a modern approach to securing networks through behavioral analysis and pattern recognition.

4.2.1 Experimental Setup

This evaluation was carried out on the same GWDG cloud setup used in the first experiment, maintaining a consistent environment for a fair comparison. The hardware specifications included a CentOS 7 virtual machine with 8 CPU cores, 16GB RAM, and 150GB disk storage. The dataset employed for this test was the first day's data from the CIC-IDS2018 dataset, chosen for its comprehensive coverage of network attack vectors and activities.

4.2.2 Methodology

The Slips IDS was configured to operate within a Docker container to streamline deployment and ensure an isolated environment for accurate performance assessment. This setup facilitated a straightforward and reproducible methodology for testing Slips against the CIC-IDS2018 dataset, with the entire dataset merged into a single pcap file for efficiency and ease of analysis.

4.2.3 Challenges Encountered

A key challenge in this experiment was the preprocessing requirement to amalgamate the dataset into a single file, which was necessary to accommodate Slips' operational framework and optimize its detection process. This preprocessing step was critical for simulating a continuous stream of network traffic, thereby testing Slips' real-time analysis capabilities.

4.2.4 Results and Analysis

The Slips IDS showcased its robust detection capabilities through a series of alerts generated during the experiment. These alerts varied in severity and type, indicating a diverse range of threats identified within the test dataset. Notable detections included malicious file downloads, connections to private IP addresses, and unencrypted HTTP traffic, among others. For example:

```
- Detected Malicious downloaded file 01f4771c47a56dbdf77642c80eb9b799. size: 90 from IP:
    144.217.158.30. Detected by: VirusShare, circl.lu. Score: 0.5. Threat level: high.
- Detected Connecting to private IP: 172.31.0.2 on destination port: 53 threat level: info.
- Detected Unencrypted HTTP traffic from 172.31.67.119 to 169.45.91.216. threat level: low.
- Detected domain watch.cbc.ca resolved with no connection threat level: low.
```

These alerts exemplify Slips' capability to provide detailed insights into potential security breaches, underpinning its utility in a high-performance computing environment. The detailed logs not only pinpointed the nature of the network anomalies but also provided a contextual understanding of the threats, thereby enabling more informed decisions on countermeasures.

4.2.5 Implications for High-Performance Computing Security

The experiment underscores the importance of sophisticated IDS solutions like Slips in high-performance computing (HPC) environments. The detailed and varied threat detection highlighted by Slips' logs shows the system's capability to adapt and respond to the multifaceted nature of cybersecurity threats faced by HPC systems. This adaptability is crucial in safeguarding against both conventional and novel attack vectors, thereby enhancing the overall resilience of HPC infrastructure against cyber threats.

4.2.6 Findings

The integration of Slips IDS into the HPC security framework offers a promising approach to enhancing network defense mechanisms. Its ability to leverage machine learning for predictive and adaptive threat detection represents a significant step forward in the continuous effort to protect complex computing environments from increasingly sophisticated cyber threats. The results from this experiment highlight the potential of machine learning-based IDS systems in not only detecting but also in providing actionable insights into network security management for HPC systems.

Table 3: Performance metrics of different algorithms, used 10% network traffic data from CIC-IDS2018 dataset for training.

Method	Algorithm	Capture	Cleaning	Infer. (s)	Total Time (s)	Acc.	ROC	Bal. Acc.
AG Multi-	CatBoost	574	94	0.07	668	0.954	N/A	N/A
	Ensemble A	574	94	1.86	670	0.955	N/A	0.847
	XGBoost	574	94	0.165	668	0.9555	N/A	N/A
	LightGBMLarge	574	94	0.25	668	0.954	N/A	N/A
	LightGBM	574	94	0.2	668	0.97	N/A	N/A
AG Bi-	Ensemble B	574	94	1.457	669	0.97	0.9923	0.97
Slips	Zeek+ML	N/A	N/A	N/A	2820	N/A	N/A	N/A

4.3 Experiment 3: Evaluating Machine Learning Models with the CIC-IDS2018 Dataset

This segment delves into the evaluation of machine learning (ML) models for intrusion detection, leveraging the comprehensive CIC-IDS2018 dataset. Celebrated for its detailed encapsulation of network traffic and a wide array of attack vectors, the dataset serves as an ideal benchmark for assessing the capability of ML models in the context of High-Performance Computing (HPC) environments [SLG18].

4.3.1 Experimental Framework

Performed on a local workstation boasting a 12-core CPU and 32GB of RAM, this experiment required significant computational power, owing to the sheer volume and complexity of the CIC-IDS2018 dataset, encompassing roughly 450GB of data. The preprocessing stage entailed extracting features and labels, revealing the dataset's extensive coverage of complex network behaviors and cyber attack patterns.

4.3.2 Selection and Training of Machine Learning Models

A suite of ML models was scrutinized, including decision trees, random forests, and ensemble methods, chosen for their proven effectiveness in processing large datasets and their adeptness at both binary and multi-class classification challenges. The AutoGluon library [EMS+20] played a pivotal role in this experiment, streamlining the deployment of these advanced ML models through its efficient model training and tuning capabilities.

4.3.3 Performance Evaluation and Insights

The effectiveness of the machine learning (ML) models was rigorously evaluated against a diverse set of metricsaccuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC). This comprehensive approach provides a well-rounded assessment of the models' intrusion detection capabilities. Table 3 details the performance outcomes, which are summarized as follows:

- Decision tree models demonstrated considerable efficacy, achieving an accuracy rate of 95% across varied attack scenarios, indicating their potent utility in identifying threats.
- Random forest models outperformed with an exceptional accuracy of 97.5%, underscoring their enhanced resilience and adaptability in detecting network intrusions.
- Ensemble methods, especially those employing boosting techniques, achieved unparalleled accuracy rates exceeding 99%. This performance highlights the significant benefits of amalgamating multiple algorithms for a more robust intrusion detection system.

The findings illustrate the profound capability of ML models to fortify HPC environments effectively. Among these, ensemble models particularly excel, offering strategic advantages in confronting the evolving landscape of cyber threats with sophistication and agility.

4.3.4 Findings

The experiment affirms the efficacy of ML models in the realm of intrusion detection, as evidenced by the CIC-IDS2018 dataset. The impressive performance of these models corroborates the narrative presented in prior sections: that ML methodologies are competitive with, if not superior to, traditional approaches in terms of speed and effectiveness in moderate settings. Faced with the requirements of high-speed network environments, specialized ML models present themselves as formidable tools in addressing the nuanced demands of modern cybersecurity, heralding a promising path for future research and practical application.

5 Discussion and Future Work

This study brings to light several critical insights and potential pathways for advancing the field of Intrusion Detection Systems (IDS) within High-Performance Computing (HPC) environments, with a particular emphasis on the application of machine learning (ML) techniques and the reliance on traditional signature-based detection methods.

5.1 Machine Learning in IDS

Firstly, our exploration delineates a clear distinction between the training and inference phases of machine learning models in IDS. Remarkably, certain streamlined ML algorithms demonstrated the capability to achieve accuracies exceeding 0.95 on datasets as large as 37.5G in less than 0.1 seconds during the inference phase. This evidences that the speed limitations of ML algorithms can indeed be surmounted, particularly when enhanced by quality datasets tailored for specific attack categories or through the employment of transfer learning. Such approaches are especially pertinent for handling large but imbalanced datasets, suggesting a promising direction for future research [GCW+24, HMA+23, Fre24].

5.2 State-of-the-Art IDS Technologies

Moreover, Pigasus emerges as a leading contender in the realm of high-performance IDS solutions. Its design epitomizes the pinnacle of current IDS technologies, potentially setting a benchmark for future developments in this field.

5.3 Reliance on Signature-Based Detection

Despite the advances in ML and AI, our findings reaffirm the continued dependence on signature-based detection methods across both HPC and traditional computing environments. The deterministic nature of such systems, underpinned by extensive expertise from security engineers, ensures their enduring utility. Current implementations of Zeek and Suricata, capable of processing traffic at rates exceeding 100Gbps, underscore the practical viability of these IDS solutions. Their efficiency is a testament to the critical role that signature-based detection continues to play in cybersecurity [SLG18].

5.4 Hardware Dependencies

Consistent with prior research [LFK18], our experimental results corroborate the finding that IDS performance is more contingent upon CPU capabilities than memory resources. This insight not only has implications for the optimization of existing IDS solutions but also guides the hardware selection process for deploying efficient and effective IDS frameworks.

In conclusion, while machine learning and AI offer promising avenues for IDS evolution, the bedrock of reliable intrusion detection in both HPC and conventional computing environments remains signature-based methods. Future advancements are likely to benefit from a synergistic integration of traditional approaches with innovative ML techniques, tailored datasets, and optimized hardware configurations.

6 Conclusion

This paper embarked on an insightful journey through the complex landscape of intrusion detection systems (IDS) within the context of high-performance computing (HPC) environments, particularly focusing on the GWDG Cloud infrastructure. Through a meticulous literature review complemented by rigorous experimentation, we compared various IDS methodologies to discern the most efficacious strategies for safeguarding HPC networks against malicious intrusions.

Our findings illuminate several pivotal insights. First and foremost, Suricata emerges as a superior IDS, offering robust detection capabilities that are well-suited for the demands of HPC environments. However, when considering the overall solution that encompasses both detection efficacy and system performance, Pigasus stands out as a more holistic approach to intrusion detection, balancing accuracy with processing efficiency.

The exploration into machine learning (ML) as a solution for IDS presents a nuanced landscape. While specialized or integrated ML models hold tremendous potential for revolutionizing IDS, our research suggests that the current state of technology and dataset limitations places such solutions just beyond the immediate horizon. The key bottleneck identified within the GWDG Cloud servernamely, the network interface and CPU performancehighlights the critical areas requiring enhancement to fully leverage the power of the IDSs.

Furthermore, our experiments underscore the excellence of ML models in training on IDS datasets, indicating that the primary challenge lies not in the capability of these models but rather in the quality of available datasets. The need for more sophisticated, comprehensive datasets is paramount to advancing the field and unlocking the full potential of ML-based IDS.

In conclusion, this paper contributes to the body of knowledge in IDS by providing a detailed classification of IDS methodologies, a comparative analysis of existing methods, and practical experimentation within a real-world HPC environment. The insights gleaned from this research underscore the importance of continuous innovation in IDS solutions, particularly in the development of advanced datasets and the integration of machine learning technologies. Future endeavors in this field should aim to address the identified bottlenecks and explore the vast possibilities that ML-based IDS promises. Our hope is that the groundwork laid by this study will pave the way for future advancements that can more effectively secure HPC environments against the ever-evolving landscape of cyber threats.

All relevant codes and datasets utilized in this study are openly accessible on GitHub at https://github.com/Mike-7777777/hpcsa24, facilitating further research and exploration in this vital area of cybersecurity.

References

[AAB22] S. Azzane, M. Addou, and F. Barramou. A suricata and machine learning based hybrid network intrusion detection system. In Y. Maleh, M. Alazab, N. Gherabi, L. Tawalbeh, and A.A. Abd El-Latif, editors, *Advances in Information, Communication and Cybersecurity*, volume 357 of *Lecture Notes in Networks and Systems*. Springer, Cham, 2022.

[AK23] Sarah Alharbi and Arshiya Khan. Ensemble defense system: A hybrid ids approach for effective cyber threat detection. In *2023 33rd International Telecommunication Networks and Applications Conference*. IEEE, November 2023.

[Akrar] Ayaz Akram. Architectures for secure high-performance computing. Online, Unknown Month Unknown Year.

[ale24] alekece. TIG Stack. https://github.com/alekece/tig-stack, 2024.

[App19] Tobias Appel. Pushing suricata towards 80gbps and more. Presented at SuriCon 2019, 10 2019.

[BJS19a] Waleed Bulajoul, Anne James, and Siraj Shaikh. A new architecture for network intrusion detection and prevention. *IEEE Access*, 7:18558–18573, 2019.

[BJS19b] Waleed Bulajoul, Anne James, and Siraj Shaikh. A new architecture for network intrusion detection and prevention. *IEEE Access*, 7:18558–18573, 2019.

[BKS19] Pranita P. Bavaskar, Onkar Kemker, and Aditya Kumar Sinha. A survey on: 'log analysis with elk stack tool'. *International Journal of Research and Analytical Reviews (IJRAR)*, 6(4):965–968, November 2019. Available at SSRN: https://ssrn.com/abstract=3677845.

[Cin24] Piero Cingari. Nvidia shatters expectations: Revenue up 265% as ai demand continues. *Euronews*, 2024. Published on 22/02/2024 - 08:06, Updated 08:45.

[DBM+18] Inadyuti Dutt, Samarjeet Borah, Indra Kanta Maitra, Kuharan Bhowmik, Ayindrilla Maity, and Suvosmita Das. Real-time hybrid intrusion detection system using machine learning techniques. 2018.

[DM21] Ayesha S. Dina and D. Manivannan. Intrusion detection based on machine learning techniques in computer networks. *Internet of Things*, 16:100462, 2021.

[dPPH24] Killian Castillon du Perron, Dino Lopez Pacheco, and Fabrice Huet. Understanding delays in af_xdp-based applications, 2024.

[EGI20] Attacks on multiple hpc sites. European Grid Infrastructure, 5 2020. Last update: 2020-05-18 11:00:00. For more information regarding ongoing investigations and cooperation possibilities, please contact irtf@mailman.egi directly. TLP:WHITE.

[EMS+20] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data, 2020.

[ERJ21] Gints Engelen, Vera Rimmer, and Wouter Joosen. Troubleshooting an intrusion detection dataset: the cicids2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 7–12, 2021.

[Fac23] Fact.MR. Supercomputer market size, share, growth opportunities 2023. https://www.factmr.com/report/supercomputer-market, 2023. Accessed: 2023-04-01.

[Fre24] Freie Universität Berlin and Hochschule München University of Applied Sciences and Otto-Friedrich-Universität Bamberg and Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) and DAASI International GmbH. Hochschulen verbessern schutz gegen hackerangriffe. Press release, 3 2024. Open-Source-Software "eduMFA" launched for enhanced cybersecurity in academic institutions.

[GCW+24] Y. Guo, R. Chandramouli, L. Wofford, R. Gregg, G. Key, A. Clark, C. Hinton, A. Prout, A. Reuther, R. Adamson, A. Warren, P. Bangalore, E. Deumens, and C. Farkas. High-performance computing security: Architecture, threat analysis, and security posture. NIST Special Publication NIST SP 800-223, National Institute of Standards and Technology, Gaithersburg, MD, 2024. SP 800-223.

[Ges23] Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen. Update zum sicherheitsvorfall vom 28. september 2023. Online Update, 10 2023. Forensische Untersuchung des Sicherheitsvorfalls vom 28. September ist abgeschlossen. Verschiedene MaSSnahmen wurden ergriffen, um die IT-Sicherheit zu verstärken.

[GJ23] László Göcs and Zsolt Csaba Johanyák. Identifying relevant features of cse-cic-ids2018 dataset for the development of an intrusion detection system, 2023.

[GMG+20] Ross Gegan, Christina Mao, Dipak Ghosal, Matt Bishop, and Sean Peisert. Anomaly detection for science dmzs using system performance data. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 492–496, 2020.

[Goo21] Google Cloud. Threat horizons: Cloud threat intelligence. Online, November 2021. Issue 1.

[Gri20] Andrew Griffin. Supercomputer researching coronavirus taken offline after security incident. *The Independent*, May 2020. Accessed: date-of-access.

[Gus19] Vilhelm Gustavsson. Machine learning for a network-based intrusion detection system: An application using zeek and the cicids2017 dataset. Bachelor's thesis, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, Stockholm, Sweden, May 2019. TRITA-CBH-GRU-2019:033.

[HBB+20] Hanan Hindy, Ethan Bayne, Miroslav Bures, Robert Atkinson, Christos Tachtatzis, and Xavier Bellekens. Machine learning based iot intrusion detection system: An mqtt case study. *arXiv preprint arXiv:2006.15340*, 2020.

[HLA23] Dania Herzalla, Willian T Lunardi, and Martin Andreoni. Tii-ssrc-23 dataset: Typological exploration of diverse traffic patterns for intrusion detection. *IEEE Access*, 2023.

[HMA+23] E. Heymann, B. P. Miller, A. Adams, K. Avila, M. Krenz, J. R. Lee, and S. Peisert. *Guide to Securing Scientific Software*, 2.0 edition, 2023.

[KGV+19] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, et al. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(20), 2019.

[KHG23] Y. Kim, S. Hakak, and A. Ghorbani. Ddos attack dataset (cicev2023) against ev authentication in charging infrastructure. In *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, pages 1–9. IEEE Computer Society, August 2023.

[Kon24] Bert Kondruss. Cyber attacks on universities – university ransomware attacks & data breaches worldwide. https://konbriefing.com/en-topics/cyber-attacks-universities.html, 2024. Accessed: 2024-03-30.

[LFK18] Thomas Lukaseder, Jessika Fiedler, and Frank Kargl. Performance evaluation in high-speed networks by the example of intrusion detection, 2018.

[LGH+22] Maxime Lanvin, Pierre-François Gimenez, Yufei Han, Frédéric Majorczyk, Ludovic Mé, and Eric Totel. Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes. In *CRiSIS 2022 - 17th International Conference on Risks and Security of Internet and Systems*, volume 13857, pages 18–33, Sousse, Tunisia, December 2022. Springer.

[LLH+22] Ying-Dar Lin, Zi-Qiang Liu, Ren-Hung Hwang, Van-Linh Nguyen, Po-Ching Lin, and Yuan-Cheng Lai. Machine learning with variational autoencoder for imbalanced datasets in intrusion detection. *IEEE Access*, 10:15247–15260, 2022.

[MBC+22] Martin Molan, Andrea Borghesi, Daniele Cesarini, Luca Benini, and Andrea Bartolini. Ruad: unsupervised anomaly detection in hpc systems, 2022.

[MS15] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, 2015.

[NTD+24] E. C. P. Neto, H. Taslimasa, S. Dadkhah, S. Iqbal, P. Xiong, T. Rahmanb, and A. A. Ghorbani. Ciciov2024: Advancing realistic ids approaches against dos and spoofing attack in iov can bus. *Journal of IoT Elsevier*, 2024. Submitted.

[Pat21] Chintan Patel. Worlds fastest supercomputers changing fast. *NVIDIA Blog*, November 2021. With the latest generation of supercomputers incorporating AI and cloud computing, the way these machines are measured is evolving, too.

[PGE23] Claudius Pott, Berk Gulmezoglu, and Thomas Eisenbarth. Overcoming the pitfalls of hpc-based cryptojacking detection in presence of gpus. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, CODASPY '23, page 12, Charlotte, NC, USA, 4 2023. ACM.

[PM16] Michal Purzynski and Peter Manev. Suricata extreme performance tuning (with incredible courage along...). Presented at SuriCon 2016, 11 2016.

[PM17] Michal Purzynski and Peter Manev. Septun mark ii. Presented at SuriCon 2017, 2017. In mob we trust.

[RCdF+21] L. Rosa, T. Cruz, M. B. de Freitas, P. Quitério, J. Henriques, F. Caldeira, E. Monteiro, and P. Simões. Intrusion and anomaly detection for the next-generation of industrial automation and control systems. *Future Generation Computer Systems*, 2021.

[Ser23] Benedetto Marco Serinelli. Toward generating a dos and scan statistical network traffic metrics for building intrusion detection solution based on machine and deep learning: I-sec-ids datasets. In *AIP Conference Proceedings*, volume 2724, page 040010. AIP Publishing, 4 2023.

[SI18] Syed Ali Raza Shah and Biju Issac. Performance comparison of intrusion detection systems and application of machine learning to snort system. *Future Generation Computer Systems*, 80:157170, March 2018.

[SLG18] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy*, 2018.

[SLP21] Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, 27(1):357370, November 2021.

[SM23] T. Sowmya and E.A. Mary Anita. A comprehensive review of ai based intrusion detection system. *Measurement: Sensors*, 28:100827, 2023.

[SSK15] Vincent Stoffer, Aashish Sharma, and Jay Krous. 100g intrusion detection. Technical report, Berkeley Lab, 8 2015. v1.0.

[SSP+22] Sehan Samarakoon, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, Sang-Yoon Chang, Jinoh Kim, Jonghyun Kim, and Mika Ylianttila. 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network, 2022.

[Tec23] Technische Universität Braunschweig and Others. IT-Security-Awareness-Days vom 07.11. 17.11.2023. Online Event, 11 2023. A series of online lectures focused on information security, open to the public without registration. Hosted by multiple German universities including Technische Universität Braunschweig.

[TFD+19] Shivam Trivedi, Lauren Featherstun, Nathan DeMien, Callum Gunlach, Sagar Narayan, Jacob Sharp, Brian Werts, Lipu Wu, Carolyn Ellis, Lev Gorenstein, Erik Gough, Alex Younts, and Xiao Zhu. Pulsar: Deploying network monitoring and intrusion detection for the science dmz. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)*, PEARC '19, New York, NY, USA, 2019. Association for Computing Machinery.

[THD+17] Rashid Tahir, Muhammad Huzaifa, Anupam Das, Mohammad Ahmad, Carl A. Gunter, Fareed Zaffar, Matthew C. Caesar, and Nikita Borisov. Mining on someone else's dime: Mitigating covert mining operations in clouds and enterprises. In *International Symposium on Recent Advances in Intrusion Detection*, 2017.

[TK22] Aaron Turner and Fred Klassen. Tcpreplay - pcap editing and replaying utilities. Tcpreplay website, 2022. Released under the GNU General Public License, version 3 or later.

[TL20] Ankit Thakkar and Ritika Lohiya. A review of the advancement in intrusion detection datasets. *Procedia Computer Science*, 167:636–645, 2020.

[WGBD22] Gerry Wan, Fengchen Gong, Tom Barbette, and Zakir Durumeric. Retina: analyzing 100gbe traffic on commodity hardware. In *Proceedings of the ACM SIGCOMM 2022 Conference*, SIGCOMM '22, page 530544, New York, NY, USA, 2022. Association for Computing Machinery.

[Wil22] Florian Wilkens. *Methods for Enhanced Security Monitoring and APT Detection in Enterprise Networks*. Dissertation, Universität Hamburg, 11 2022. Dissertation, Universität Hamburg, 2022.

[Zee24] Zeek Developers. Zeek Network Security Monitor. https://github.com/zeek/zeek, 2024. Accessed: 2024-03-30.

[Zha21] Zhipeng Zhao. *Pigasus: Efficient Handling of Input-Dependent Streaming on FPGAs*. Ph.d. dissertation, Carnegie Mellon University, Pittsburgh, PA, 8 2021. B.S., Electrical Engineering, Beihang University; M.S., Electrical Engineering, Beihang University.

[ZSA+20] Zhipeng Zhao, Hugo Sadok, Nirav Atre, James C. Hoe, Vyas Sekar, and Justine Sherry. Achieving 100gbps intrusion prevention on a single server. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 1083–1100. USENIX Association, November 2020.