Cybersecurity Concerns of Artificial Intelligence Applications on High-Performance Computing Systems

Rishabh Saxena, Aadesh Baskar, Sameer Haroon,
Sameed Hayat, Oleksandr Shcherbakov, Kerem Kayabay, Dennis Hoppe
High-Performance Computing Center Stuttgart (HLRS)
University of Stuttgart
Stuttgart, Germany
e-mail: {firstname.lastname}@hlrs.de

***Abstract*—The High-Performance Computing (HPC) landscape is undergoing profound changes with developments in fast-growing domains such as Artificial Intelligence (AI), cloud, edge computing, and quantum computing. The growth of AI has particularly impacted the relatively isolated HPC realm, bringing in new user communities like start-ups that don't want to fall behind and are increasingly dependent on foundational models trained by a handful of companies. However, the rapidly growing AI technology landscape introduces security vulnerabilities to the HPC world, which hesitates to install and maintain potentially unstable software. This paper is a first step towards enabling secure AI workloads on HPC systems by investigating AI security vulnerabilities using the AI Lifecycle. We then organize the challenges for HPC centres through the lens of the Technology-Organization-Environment (TOE) framework. Lastly, we discuss the differences between AI security concerns and mitigation strategies on HPC and other systems, and outline future work towards secure AI workloads on HPC systems.**

Keywords-High-Performance Computing (HPC), Artificial Intelligence (AI), AI Security Vulnerabilities, TOE Framework

I. Introduction

Supercomputers are the fastest computers of their time, and have long been geared towards solving complex, time-intensive problems. As Strohmaier *et al.* [1] notes, the traditional focus on floating-point intensive technical applications is no longer sufficient to survive in the market. The HPC landscape is undergoing profound changes with the emergence of Machine Learning (ML) and Deep Learning (DL), cloud and edge computing, and quantum computing. This paper looks at the growth of AI and the need for HPC to embrace these technologies and attract new user communities while ensuring a high level of security. This is crucial to remain an attractive computing platform for Small and Medium-Sized Enterprise (SMEs), start-ups, and industry.

Why is the growth in AI relevant for supercomputing? There are actually two sides to the coin: First, AI needs the processing power of HPC, which is, after tackling technical barriers, a straightforward task. Second, HPC should leverage AI to improve classical simulations and system operation. This task is quite challenging because it predominantly requires expertise in both, AI and HPC.

Updating most of today's HPC systems to support AI workflows is a challenge, as it opens up the relatively isolated HPC realm, bringing it out of its secure bubble to a higher, and still relatively unknown, level of security risks. Moreover, many HPC system administrators focus on traditional HPC application areas like engineering and chemistry, which makes it difficult for them to fully understand the specific needs of emerging user communities, such as those in AI. This is especially true for widely used AI frameworks (e.g., TensorFlow and PyTorch) that are part of the rapidly evolving ecosystem of AI software and libraries, and are in stark contrast to the limited legacy software that administrators maintain on traditional HPC systems, over which they have much greater control and experience. Therefore, there is some resistance in installing and maintaining software from the AI realm that is potentially unstable or may have security vulnerabilities, as well as allowing such software to train and execute potentially malicious or exploitable AI models.

Nevertheless, ways must be found to enable AI workloads on HPC systems. If not, there is a growing risk that the academic world, along with start-ups and SMEs, will continue to fall behind and become increasingly dependent on the foundational models, or their powered-down versions, provided by bigger companies [2]. It is not possible for SMEs to build up and train their own counterparts to foundational models, without access to federal or academic supercomputing resources. To this end, HPC experts and AI experts must jointly develop solutions that allow using pure AI applications and workflows on HPC and thus enabling seamless, hybrid HPC/AI workflows. The technical obstacles include, for example, making the entire AI software stack available for HPC systems (e.g., via containerisation), evaluating AI-specific usage patterns, and cybersecurity aspects. This paper focuses on the cybersecurity concerns for running large-scale AI applications on HPC systems.

In order to acquaint the reader with the foundations, the paper first leads into a quick review of each of the main concepts, namely, HPC, AI on HPC, and cybersecurity on HPC. Then a thorough investigation is carried out on the technical areas of concern that threaten or undermine the usage of ML workflows on HPC, followed by potential challenges at an organisational level for HPC centres, through the lens of the TOE framework [3]. Finally, we review the potential problems and solutions presented in the paper, and discuss how our findings relate to research in the state of the art, and what future work could lead on from this paper.

II. Literature Review

The literature review explores the role of HPC systems, their integration with AI, and cybersecurity considerations, while identifying related work and existing knowledge gaps.

A. HPC

1) The Importance of HPC for Diverse User Communities: HPC is utilized across industry [4] and academia, with use cases as diverse as simulating fluid flows in the turbulence around aeroplanes to the fluid flow of blood through the human heart [5], from carrying out genome sequencing in biology to molecular simulations in nuclear fusion [6], and from predictions ranging from weather forecasts, financial markets, and the spread of diseases and pandemics [7] [8]. Most of these diverse use cases can be classified into two basic classes of problems, tracking and simulating the interactions of a large system of individual particles, and solving forms of partial differential equations. Often, solving these problems results in solving a Linear Algebraic system [9].

2) An overview of an HPC system: A supercomputer, or HPC cluster, derives its processing power from aggregating and coordinating huge number of individual computational systems. It not only orchestrates the parallel execution of users' programs, called codes, over these systems, but also handles many users and their codes simultaneously [9]. These individual systems, or nodes, of an HPC system vary according to their tasks. Login nodes provide initial access to the system, as well as basic storage and standardised interfaces, such as to the scheduling system (sometimes running on dedicated scheduler or head nodes) [10]. The Scheduler allocates users the access to compute (or worker) nodes. These are resource heavy nodes equipped to do the heavy lifting of executing application codes, and themselves come in different flavours, such as pure Central Processing Unit (CPU) nodes, mixed CPU and Graphics Processing Unit (GPU) nodes, pure General-Purpose Graphics Processing Unit (GPGPU) nodes, and data transfer nodes [11]. All the nodes making up an HPC cluster use a choice of high-performance interconnect to distribute data and instructions amongst themselves, such as InfiniBand or Gigabit Ethernet [12].

The user must design their program keeping in mind the parallel architecture of a supercomputer, from the level of multiple cores in a single processor, to multiple processors in a single node, and finally scaling up to the nodes required or available on the HPC system [11]. The design of the program must also understand and make use of the memory architecture of the system, with hybrid models of shared memory and distributed memory paradigms available on most HPC systems. Ultimately, it is the design of the application code, including the exploitation of parallel frameworks such as Message Passing Interface (MPI) and Open Multi-Processing (OpenMP), that determines how efficiently it can harness the power of the HPC system [13].

B. AI on HPC

The rise in the level of AI model complexity and size to exponential levels poses an unprecedented computational requirement, thus the adoption of HPC resources [14] [15]. Modern AI models, especially Large Language Models (LLMs), contain hundreds of billions of parameters that far exceed the memory capacity of a single GPU [15] [16]. This growth, coupled with expansion in training datasets, creates a number of technical challenges that only HPC is well-poised to tackle.

The driving technical requirements of AI's need for HPC are many. First, the model size already far exceeds single GPU memory capability and requires distributed computing approaches [16] [17]. Second, very large data sets already used for training exceed single machine memory and storage capability [14] [17]. Third, this computational intensity of training resulted in prohibitively long training times on conventional hardware [16] [17].

Moreover, research on AI does require long hyperparameter tuning and thus many training runs with different settings are needed. HPC clusters, therefore, provide the best environment for these parallel experiments due to their high functionality in job scheduling and resource management systems [16] [17].

The high computational demands of AI are challenging existing computing platforms. AI workloads are already driving the architecture of new HPC hardware, particularly in the construction of higher-end, more powerful, and more efficient GPUs and dedicated AI accelerators [16] [17]. Software frameworks evolve to better cope with distributed AI training and inference on HPC clusters, with innovation in techniques such as model parallelism and pipeline parallelism [16] [17]. This convergence pushes the frontiers of both AI and HPC to handle the ever-increasing scale and complexity of AI models and datasets [18], also raising significant concerns, such as cybersecurity.

C. Cybersecurity on HPC

Similar to the general purpose Information Technology (IT) systems, HPC systems face a variety of threats that can affect their confidentiality, integrity, and availability. Some examples are stealing of compute cycles, unauthorized access, Denial of Service (DoS), data breaches and leakage, misuse of compute power, and alteration of code. When comparing HPC with general purpose IT systems, there are differences in their functions, software and hardware stack, the user community and the workflow.

Peisert [19] considers these differences and presents the challenges and opportunities in implementing cybersecurity for HPC. A single ingress and egress point between the cluster and the external world makes it easy to monitor and restrict the traffic. Not all nodes in the cluster are directly accessible by the users. The users connect to the login nodes to submit the jobs and data transfer nodes to pull the data from external sources. The login nodes and data transfer nodes are placed behind firewalls or protected by Access Control Lists in the routers or switches. They might be accessible only through secure protocols like SSH for login and GridFTP for data transfer [20].

The compute nodes can only be used by submitting a job to the resource manager and are not directly accessible by

TABLE I: AI VULNERABILITIES IN SECONDARY STUDIES

Study	Identified Vulnerabilities
Huq *et al.* [26]	Training data poisoning, trojaning, LeftOverLocals
Familoni [27]	Adversarial attacks, data breaches, deepfakes, distributed DoS, phishing, cyber conflicts, evolving malware, data poisoning
Roshanaei *et al.* [28]	Adversarial attacks, data poisoning, model stealing, model inversion, infrastructure attacks
Blowers and Williams [29]	Steganographic attack, evolving malware, deep fakes
Kaloudi and Li [30]	Evasive malware, evolving malware, voice synthesis, social bots, adversarial training
Muñoz-González and Lupu [31]	Data poisoning, exploratory attacks, evasive attacks, availability violation, data stealing
Hu *et al.* [32]	Data breach, data biases/fake data, sensor spoofing attack, image scaling attack, data poisoning attack, adversarial attacks, availability attack, data stealing, model stealing, AI framework backdoors

the user. The resource manager that allocates the nodes and schedules the jobs can use its own authentication mechanism. A simple example of such an authentication mechanism is Munge used by Slurm to encode the user credentials of a calling process and decode them in a remote node [21]. Multi-tenancy in HPC enables jobs from multiple users to run at the same time in the cluster. Even if a given node remains exclusive to a particular user's job, all the nodes in the cluster will be connected to the same network. Prout *et al.* [22] offers a solution for this problem by implementing network policies based on user and group IDs of the application processes. Since the HPC providers can support users from various institutions and SMEs in the same cluster, the need for proper configuration of file-system access control is crucial. Discretionary Access Control is configured by the owners of the file to restrict permissions to their file and Mandatory Access Control is configured by the system administrators [23].

Since the primary goal of HPC systems is to offer very high compute power, the overheads from security tools are not acceptable. This presents a challenge in directly using security tools available from the general-purpose web/software development ecosystem. However, the world of HPC is witnessing serious change due to requirements emerging from diverse user communities. One such trend is the increased adoption of containers that provide reproducibility, flexibility, and portability in shipping applications. The usage of containers can provide extra attack surface and can be risky in multi-tenant HPC clusters [24]. Keller Tesser and Borin [25] stress on the importance of unprivileged user containers to reduce the risks associated with using containers in multi-tenant systems. The following sections review such vulnerabilities, focusing on the AI domain's requirements.

D. Related Work and Gaps

Before discussing the cybersecurity concerns of large-scale AI applications on HPC systems, we should discuss the secondary studies with similar objectives (Table I). Huq *et al.* [26] survey the cloud-based GPU threats and their impact on AI, HPC, and Cloud Computing. The report explores potential attacks against AI using GPUs. Familoni [27] reviews the cybersecurity concerns in AI systems. After presenting the vulnerabilities, the paper points out the challenges in securing AI systems, including human factors and the lack of explainability and transparency in AI systems. Roshanaei *et al.* [28] identify the defensive mechanisms and frameworks after specifying the potential threats to AI systems. Following an introduction to potential vulnerabilities, Blowers and Williams [29] emphasize the design considerations for secure AI/ML architectures. Kaloudi and Li [30] focus on the intentional use of AI for harmful purposes, classifying the attack stages and objectives in a cyber threat framework with defensive approaches. Muñoz-González and Lupu [31] introduce a threat model that organizes the ML vulnerabilities by attacker's capability, goal, and knowledge. Hu *et al.* [32] map the attacks on AI systems to the AI lifecycle comprising data collection, data preprocessing, training, inference, and integration phases.

As AI systems become larger, driven by competition among a handful of large companies, it is critical that start-ups and SMEs also have access to the computational power needed to train and deploy foundational models [33]. Furthermore, researchers also need the computational capability to evaluate these large models [34]. Therefore, we need secure HPC infrastructures to train, evaluate, and deploy large-scale AI systems. To the best of our knowledge, the literature lacks studies that organize large-scale AI vulnerabilities into an ML lifecycle framework from the HPC perspective, and map the challenges HPC centres face in solving AI system vulnerabilities. The next section organizes large-scale AI system vulnerabilities on HPC and classifies the challenges for HPC centres.

III. AI Cybersecurity for HPC Systems

This section examines the cybersecurity risks in the ML lifecycle on HPC systems and the challenges of addressing AI vulnerabilities using the TOE framework.

A. Potential Risks in the ML Lifecycle on HPC

Since the ML lifecycle involves multiple steps and use-cases, multiple points of attack can be exploited by potential bad actors. This subsection briefly details a non-exhaustive list of security risks associated with ML pipelines, with an emphasis on how HPC is particularly exposed to such risks.

- **Problem Definition**: The first vulnerability that must be addressed, even before looking into technical security risks, is that of the usage of HPC resources for malicious use-cases or ill-posed applications. Blauth et al. [35] mention various categories of malicious uses of AI, such as social engineering models, misinformation and fake news, hacking, and warfare-related AI. These risks are significant because detecting the development of these models requires manual oversight. This necessitates a more stringent review of projects and code on HPC systems at computing centres, along with periodic checks to ensure only relevant tasks are performed.
- **Data Exploration**: Development of ML systems usually begins with an Exploratory Data Analysis (EDA) phase [36],

where the users and developers understand the composition of the data and problem that needs to be solved. Since HPC is a component of the pipeline, and not the only available infrastructure during the development lifecycle, most ML development takes place in a heterogeneous computing environment [37]. In such scenarios, a mixture of traditional HPC, cloud, and edge computing is used to distribute different phases of the lifecycle. Because EDA is an iterative and experimental phase, it often requires developers to connect their local systems to HPC or cloud systems outside the normal job-based scheduling environment. With the steady increase in model size and training requirements [38], HPC environments have become more relevant for EDA.

Security risks during EDA on cloud systems has also become equally relevant [39], and as such, also extends to the HPC environment. Since EDA on HPC requires opening ports to the outside world, this presents a unique challenge where security vulnerabilities throughout the chain of connections may affect the source HPC system. Where most cloud providers deploy their EDA environments through containerization, these methods become difficult to implement in a batch-scheduling system. The most famous containerization engine, Docker, requires root-access, which presents a security risk when provided within a shared, HPC environment. On the other hand, development of rootless containers, such as Singularity/Apptainer [40], are not well integrated with other systems. We further discuss the security issues with container runtimes in HPC environments in the next section, under technological challenges. As such, EDA on HPC systems is usually more time-consuming task, in order to maintain security.

- **Data Ingestion**: Similar to providing an environment for EDA, HPC infrastructure must also allow for transportation and ingestion of vast amounts of data, especially for AI/ML development. This risk is mitigated in the cloud using encryption at rest, transmission and source, along with lifecycle features. Since most ML development in the cloud uses object storage [41], this differs from the traditional HPC approach. Connecting these systems is challenging because higher bandwidth data transmission requires multiple steps between the source and destination, increasing vulnerabilities [42], [43], including man-in-the-middle attacks [44].
- **Data Engineering**: Even when the data can be securely moved around different storage resources throughout the pipeline, further risks exists that can be exploited by bad actors. Once the data is at rest, engineering and utilizing this data for further processing becomes even more important. Kumar et al. [45] show that there are various security risks involved with data pipelines, specifically in cloud systems, such as risks involving confidentiality (access to the data), integrity (tampering with the data), availability (DoS), risks involving authentication and access-control (since most cloud data pipelines are built with a singular authentication mechanism), and other minor risks such as data location, multi-tenancy and backup of data. These risks also extend to HPC storage systems, where the storage system must also deal with these security risks. Adversarial attacks via malicious actors, such as poisoning attacks [46] can cause loss of data integrity, both for cloud and HPC systems.
- **Model Training**: Another attack vector is the training and code execution of models. ML pipelines either train a model from scratch using multiple libraries, such as TensorFlow, PyTorch, and Scikit-learn. Although these libraries have active development teams to patch discovered vulnerabilities, they still possess a variety of security risks. [47]. When these libraries are used to train a model, there may be open back-doors that allow bad actors to execute malicious code. Apart from pre-training models, pre-trained models hosted on various repositories may also contain malicious code embedded into the model file itself, such as backdoor code, weight poisoning attacks, and falsified model description [48]. Although root privileges are generally unavailable on HPC systems, any cloud-HPC ML pipeline may have privileged steps that allow such spillover.
- **Model Evaluation**: Another major step in the ML pipeline is the evaluation of pre-trained models. In this step, the developers usually look at evaluating the model against a test or live dataset, and predict the performance of such models. Major security issues posed during this step are evasion attacks and model inversion attacks [46], where the bad actor might poison the dataset for evaluation, in order to falsify the final output, or simply switch the model output entirely. These attacks can cause falsified information to be used when using these models in the real world. This is a particularly difficult problem within the HPC environment since HPC resources are expensive, and falsified evaluation results from ML training may cause excessive usage of resources.
- **Model Deployment**: Toward the end of an ML pipeline, before monitoring and maintenance, is usually the deployment of the model in a production environment. Here, the usual security risks of any cloud environment [45] become automatically relevant. Apart from these, model ML specific attacks that are relevant at these steps are surveyed by Chen et al. [49], where they mention attacks such as distributed DoS attacks on deployed models, model inversion and extraction attacks (where the output of the model is used to replicate the model by prompting it with different datasets), membership inversion attacks [46] (where the attacker can generate the underlying dataset of the model, along with other parameters, by repeatedly querying the model), as well as injecting malicious code during batch inferencing of ML models. There may be threats present if any attacker gains access to a GPU session, even after the GPU session has ended, by extracting the information execution on GPUs [26]. In case these models are being hosted within the HPC environment, this may lead to a loss of confidential information and other secrets. Lastly, as models get larger and more complex, it becomes harder and more computationally expensive to thoroughly evaluate them before deploying to production. Large-scale models bring many opportunities, but additional care is necessary

for critical domains such as law [2].

- **Monitoring and Maintenance**: The production ML models are usually ephemeral because the model performance degrades over time [50]. Continuous Integration (CI) and Continuous Delivery (CD) have increasingly become prominent methodologies to automate software development in the industrial landscape, as well in the HPC/AI domain. This is because CI/CD plays an important role in automated monitoring to track the model performance in real-time, automated retraining, and rollbacks. Since CI/CD require multiple components, such as a central version control repository like Git, and execution platforms such as runners, there are multiple vectors of attack available for bad actors. To encapsulate this pipeline, most CI/CD tools use Docker containers, which come with their own vulnerabilities [51] such as insecure configurations, privilege escalation, and changing of container permissions through exploits. Apart from this, other vulnerabilities, such as running malicious code within the CI/CD pipeline is another risk factor, which is compounded when HPC is involved as a component in the pipeline. If the initial code being tested and built is compromised, the privilege provided to the runners might spill over the infection to the HPC system, thereby creating a security issue for the entire-cluster.

In the next subsection, we will look at why these risks are difficult to solve, even when they may be known.

B. Challenges in Addressing AI Vulnerabilities on HPC

To address AI vulnerabilities on HPC, we use the TOE framework [3], which explains how three contextual factors—technology, organization, and environment—affect an organization's adoption and implementation of innovations.

1) Technological challenges: While AI applications bring immense potential to HPC environments, integrating these innovations introduces several technological challenges.

- **Increasing Spectrum of Hardware Components**: Modern HPC systems are incorporating a growing variety of hardware components to enhance computational power, energy efficiency, and specialized processing capabilities. These components can range from traditional CPUs and GPUs to more specialized hardware like Tensor Processing Units (TPUs) and even quantum processors. The inclusion of such diverse and sometimes exotic hardware increases the complexity of managing security across the entire HPC environment. Each type of hardware component in an HPC system may have unique security requirements. For instance, GPUs and TPUs optimized for parallel processing might have different memory management vulnerabilities compared to CPUs. Exotic and cutting-edge hardware components in HPC systems may have unique firmware and micro-architectural vulnerabilities that are less well understood or documented. Attackers can exploit these low-level vulnerabilities through techniques like side-channel attacks (exploit information gained from the physical implementation of a computer system rather than vulnerabilities in the code itself [52]), row hammer attacks [53] (hardware vulnerability in DRAM memory), or Spectre [54], and Meltdown-type [55] exploits (exploit speculative execution - a performance optimization in modern CPUs - to access unauthorized memory). The challenge is to ensure robust and properly managed security configurations for each hardware type. This includes avoiding conflicts or vulnerabilities and consistently identifying, patching, and protecting against vulnerabilities on various devices, often requiring specialized knowledge.
- **Performance-Security Trade-offs**: HPC applications are designed to maximize performance, as the scalability of these systems means that any performance loss also scales significantly. Consequently, HPC users value security measures only when they come with a tolerable performance penalty [24]. To achieve optimal performance, HPC systems often operate as shared environments where multiple tenants can access shared resources, such as access nodes and certain network layers. This is in contrast to cloud systems, which are predominantly virtualized. In cloud environments, each tenant or user has isolated virtualized compute and network resources, reducing the risk of cross-tenant interference or data leakage.
- **Evolution of AI, Big Data, and HPC Software Ecosystems**: AI, Big Data, and HPC have evolved within distinct software ecosystems, each optimized for different goals and environments. AI software ecosystems are built around cloud-native, containerized environments with frameworks like TensorFlow, PyTorch, and Keras. Big Data ecosystems, such as Apache Hadoop and Spark, are designed for distributed storage and processing of vast datasets. Meanwhile, supercomputer ecosystems focus on HPC with specialized libraries and frameworks like MPI and OpenMP optimized for parallel processing. The divergence in software ecosystems creates significant challenges when integrating AI and Big Data workflows with HPC environments. The AI and Big Data frameworks often lack the native compatibility with HPC-specific software and libraries. Managing dependencies and ensuring version compatibility across these ecosystems is a non-trivial task. AI and Big Data frameworks evolve rapidly with frequent updates and new releases, whereas HPC software stacks may rely on more stable, tested versions. Ensuring compatibility between different versions, libraries, and tools without exposing the system to vulnerabilities or performance issues is a considerable challenge.
- **Cloud-Native ML Frameworks and HPC Security Compatibility**: The distributed ML libraries and frameworks, such as TensorFlow, PyTorch, Horovod or Ray, have been developed primarily with cloud infrastructure assumptions in mind. However, these frameworks rely on the inherent isolation provided by cloud virtualization for security and require users to manage infrastructure-level security settings [56]. In an HPC environment, where such virtualized isolation is often absent, deploying these frameworks securely becomes challenging. The lack of compatibility with HPC security requirements means these frameworks may inadvertently

expose vulnerabilities when deployed in non-virtualized environments. This creates a challenge of either adapting these frameworks or fundamentally redesigning the HPC environment to support them securely.

- **Maturity of Distributed ML Libraries and Frameworks:** As mentioned above, distributed ML libraries and frameworks are still in relatively early stages of their development lifecycle, tending to prioritize performance and scalability over security, and leading to a lack of robust built-in security features. For example, they may not have mature mechanisms for handling control or secure communication, which are critical in multi-tenant HPC environments. This creates vulnerabilities that could be exploited in environments where sensitive data is processed. They may also focus heavily on performance optimization and may employ shortcuts or assumptions that do not hold in more secure or controlled environments like HPC. For instance, assuming trusted environments and thus lacking robust isolation between processes, increases the risk of side-channel attacks or data leakage.
- **Security Issues with Container Runtimes in HPC Environments**: Most container runtimes (software responsible for running containers, managing container images, and providing necessary tools and libraries to support containerized applications), such as Docker, traditionally require root (administrator) privileges to manage containers, which poses a significant security risk in HPC environments. Running containers with root privileges can lead to a potential exploitation where malicious users can gain unauthorized access to the underlying host system. This is particularly concerning in multi-tenant HPC setups, where ensuring isolation and security between different users and their workloads is crucial. HPC-oriented container runtimes like Apptainer (formerly Singularity) and Podman are designed to address some of these security concerns by allowing containers to run in a "rootless" mode, which avoids requiring root privileges. However, these runtimes rely heavily on user namespaces (a Linux kernel feature that allows a process and its children to have a different view of the system's user and group IDs; this enables root privileges within the namespace without granting those privileges on the host system) to provide isolated environments. Recent history has shown that user namespaces have been subject to vulnerabilities, such as CVE-2022-0492, CVE-2022-0185, CVE-2021-22555 where a flaw in the user namespace handling could lead to privilege escalation. Such vulnerabilities undermine the security guarantees provided by rootless containerization in HPC environments. Alternatively, udocker [57] is a unique container runtime that operates entirely in user space, meaning it does not require root or system-level privileges to execute. This design significantly reduces the risk of privilege escalation attacks, a common concern with other containerization tools that rely on elevated privileges. Since udocker runs without needing system privileges, it is well-suited for environments where users do not have administrative rights, such as shared HPC systems. While udocker provides enhanced security by running entirely in user space, this approach can lead to performance penalties. The runtime achieves container-like isolation by emulating container features through techniques such as tracing or intercepting system calls (both are normally used to monitor, control or debug the behaviour of processes). These techniques, while effective at maintaining isolation without elevated privileges, can introduce significant overhead, especially for I/O-intensive HPC applications.

2) Organizational challenges: Beyond the technological complexities, securing AI applications in HPC environments also involves overcoming significant organizational challenges.

- **Managing Multiple Systems for Diverse User Groups**: HPC centres often cater to a wide range of users with varying computational needs, such as researchers, data scientists, and engineers from different domains. As a result, a single centre may deploy multiple types of systems, including traditional HPC clusters, AI-specific accelerators, Big Data analytics platforms, and GPU-based systems for deep learning. This diversity in system types creates significant challenges in terms of system management and security. For instance, AI and Big Data platforms may require more frequent updates and may have different access control mechanisms compared to traditional HPC clusters. Coordinating these security needs across different systems while maintaining a consistent security posture becomes a challenge.
- **Continuous Infrastructure Upgrades to Maintain Cutting-Edge Capabilities**: As HPC centres continuously update their infrastructures with newer hardware, they may inadvertently introduce new security vulnerabilities. Each new piece of hardware, whether it's a next-generation CPU, GPU, or a specialized accelerator, comes with its own set of firmware, drivers, and software dependencies. These components could have undiscovered or recently discovered vulnerabilities that can be exploited by malicious actors, especially if proper security assessments and patches are not promptly applied. The diversity of hardware in an upgraded HPC environment inherently increases the attack surface. New components and systems require additional configurations, libraries, and tools, which may not always be fully vetted for security. An attacker could exploit inconsistencies or gaps in security configurations, especially in environments where legacy systems are mixed with newer hardware. Frequent hardware upgrades also expose HPC centres to supply chain risks. As they procure new components from different vendors, there is a risk of introducing compromised hardware or firmware that could be exploited.
- **Elevated Risk of Insider Threats in HPC Systems**: HPC systems often handle highly valuable computational resources and sensitive data, such as proprietary research, government data, or confidential business analytics. This makes them prime targets for insider threats, where individuals with legitimate access may misuse their privileges for unauthorized purposes, either for personal gain, sabotage, or espionage. The high-stakes environment of HPC makes

the impact of insider attacks particularly severe, potentially resulting in substantial financial loss, reputational damage, or compromised research integrity. Due to the collaborative nature of HPC environments, where researchers, scientists, and external collaborators often require access to various systems and data, managing user privileges becomes highly complex. Many HPC centres provide access to shared resources, which can be exploited by insiders if not managed properly. The lack of fine-grained access controls or monitoring capabilities can allow malicious insiders to access sensitive information or disrupt system operations unnoticed. HPC centre staff, such as researchers and system administrators, usually have high technical expertise. This technical proficiency means that insiders who wish to conduct malicious activities might be able to bypass standard security controls, manipulate logs, or exploit unpatched vulnerabilities without easily being detected. The insider's deep understanding of the system architecture and potential weak points makes it more challenging for security teams to detect and prevent insider threats. While implementing strict security policies can help mitigate insider threats, doing so in HPC environments is challenging due to the need for flexible and rapid access to resources by different user groups. Security measures that are perceived as too restrictive can hinder research productivity and lead to user resistance or attempts to circumvent controls, inadvertently creating security loopholes.

- **Lack of Security Awareness Among HPC Users**: Users in HPC environments, such as researchers, data scientists, and engineers, are typically focused on maximizing ease of use and achieving research or computational results as quickly as possible. Security is often seen as an impediment to their workflows rather than a necessity. This mindset can lead to risky behaviours, such as sharing passwords, using weak or repetitive credentials, ignoring security updates, or circumventing security protocols that they perceive as hindrances. Given their focus on productivity and achieving results, users may resist the implementation of strict security controls, such as multifactor authentication, strict access controls, or frequent password changes. Such controls are often viewed as burdensome and time-consuming, leading users to find workarounds or ignore policies altogether. This resistance can undermine organizational efforts to maintain a secure HPC environment. Many users assume that the responsibility for security lies solely with HPC administrators and IT security teams. This overreliance creates a gap in the overall security posture of the organization, as users may fail to recognize that their actions - such as downloading unverified software, neglecting to patch their applications, or mishandling sensitive data - can directly impact the security of the entire HPC system.

3) Environmental challenges: While organizational challenges focus on user behaviour and policy management, the environmental context addresses broader issues stemming from the shared and increasingly diverse nature of HPC systems and their network security practices.

- **Delegated Security Risks in HPC**: Traditionally, HPC service providers have delegated the responsibility for secure network usage to end users, assuming that users will manage their own network security measures. This model relies heavily on users being knowledgeable and proactive about securing their connections, data transfers, and communications. However, this assumption does not always hold true, especially given the wide range of technical expertise among users in academic and research settings. The primary users of today's HPC systems are academic researchers, scientists, and students who often focus on their research objectives rather than on implementing robust security practices. Many of these users assume that the underlying HPC system and its network are inherently secure, leading to a lack of precaution when developing software or transferring sensitive data.
- **Increased Application Diversity from HPC and AI Convergence**: The convergence of HPC and AI significantly expands the variety of applications running on HPC systems. Traditional HPC workloads, such as large-scale simulations and complex scientific calculations, are now being combined with AI-driven applications like deep learning, natural language processing, and data analytics. This convergence results in a more diverse set of software, libraries, and frameworks that need to be managed within the same HPC environment. The introduction of AI workloads brings new security challenges, as many AI frameworks and libraries were originally developed with cloud environments in mind and may lack the rigorous security controls required in HPC settings. The increased diversity in applications can lead to conflicting dependencies, security vulnerabilities, and unintentional exposure of sensitive data. Managing the security of these diverse applications is particularly challenging when they rely on different security models and practices.
- **Growing Target for Sensitive Data and Malicious Applications**: HPC systems are increasingly used to process and analyse sensitive data, such as genomic information, climate modelling data, defence simulations, and proprietary research. As a result, these systems have become attractive targets for cyberattackers [59] who seek to steal, manipulate, or exfiltrate valuable information. The aggregation of sensitive data in HPC environments heightens the risk of breaches, particularly if adequate security measures are not in place to protect data in storage, transit, and processing. Many HPC systems operate in a shared environment where users from different institutions, research centres, and even commercial entities collaborate. This openness, while fostering innovation and scientific progress, also increases the risk of insider threats and unauthorized access to sensitive data. Attackers may exploit this shared nature to infiltrate systems, elevate privileges, and access data that they are not authorized to see.

IV. Discussion and Conclusions

The last section compares security concerns across different computing paradigms, proposes strategies to mitigate potential

TABLE II: AI THREAT MITIGATION STRATEGIES FOR CLOUD-BASED GPU SYSTEMS AND HPC SYSTEMS WITH GPU ACCELERATION

Mitigation Strategy	Cloud-Based GPU Systems based on [26]	HPC Systems with GPU Acceleration
Advanced Virtualization Security	Use Hypervisors and VMs to virtualize GPU and system usage.	Virtualized platforms with low performance overhead and virtual networks on HPC [58]
Robust Kernel Isolation	Using vGPUs to mitigate manipulation attacks through APIs.	Same as general systems, as well as prevent privilege escalation by regularly updating underlying images.
Enhanced Memory Management	Prevent memory snooping and leakage through randomization, encryption and clearing.	Same as general system, as well as anomaly detection on memory usage.
Driver and Firmware Security	Rigorous system for patching and driver management to stay on top of vulnerabilities.	Using safe underlying frameworks for GPU execution, and active scanning for CUDA/ROCm vulnerabilities.
Secure Code Execution Frameworks	Verifying code before executing and keeping up-to-date with underlying frameworks.	Same as general systems.
GPU Usage Monitoring and Anomaly Detection	Deep monitoring of GPU resources to counteract cryptojacking, distributed DoS or overconsumption of resources using AI/ML techniques.	Including monitoring details within the pipeline to correlate jobs with resource usage to detect anomalies.
Application-Level Security Measures	Validating input data before running AI/ML workloads to mitigate model/data poisoning and evasion.	Same as general systems.
Hardware Security Modules for Sensitive Operations	HSMs offer higher security than GPUs, and should be used for critical tasks.	Induction and inclusion of HSM partitions in HPC clusters.
Access Control Policies	Role-Based Access Control for GPUs to reduce security leaks due to unauthorized access.	Integration of HPC access policies, and peripheral system access policies for more fine-grained resource-level access control.
Education and Awareness	Provide training for GPU-based security issues.	Same as general systems, with an added emphasis on the HPC architecture.

attacks on HPC, and addresses limitations and areas for future research.

A. AI Security Concerns Across Computing Paradigms

While cloud, edge, and HPC environments each have unique challenges for securing AI applications, HPC systems face specific security risks due to their focus on performance and scale. AI workloads in HPC need massive computational power, often distributed across thousands of nodes. Ensuring security at such a scale, especially when running distributed ML algorithms, is a significant challenge. At scale, attack vectors like data poisoning, adversarial inputs, and model inversion become more feasible, particularly if the underlying HPC infrastructure lacks robust, AI-specific security measures. The literature does not, to our knowledge, sufficiently organize large-scale AI vulnerabilities within an ML lifecycle framework from an HPC perspective, nor does it address the specific challenges HPC centers face in mitigating these vulnerabilities. Our research fills this gap by exploring how AI security concerns manifest in HPC environments.

Edge computing environments, where AI inference is performed closer to data sources, also face unique challenges such as physical tampering, localized DoS attacks, and limited computational resources for robust security protocols. HPC systems, which typically handle large datasets in centralized facilities, must ensure the integrity and confidentiality of data across multiple storage and processing layers, with particular attention to data in transit and at rest. Physical security in HPC involves safeguarding large-scale data centres, whereas edge environments require securing numerous distributed devices, each with a potentially greater risk of compromise.

Cloud environments rely heavily on virtualization and multi-tenancy to maximize resource utilization, which provides strong isolation mechanisms. However, HPC systems often prioritize performance and thus avoid extensive virtualization, opting instead for shared access to physical hardware. This lack of virtualization increases the risk of side-channel attacks and resource contention vulnerabilities. So, addressing these risks requires incorporating best practices from cloud security, but adapting them to the specific needs of HPC. Accordingly, the next section compares the AI threat mitigation strategies for cloud and HPC systems.

B. Strategies for Solving Potential Attacks on HPC

We look at some threat mitigation strategies in Table II, based on the work done by Huq *et al.* [26], with a particular focus for AI on GPU-accelerated HPC partitions. Nevertheless, HPC environments are increasingly integrating GPUs, TPUs, and other accelerators to enhance AI processing capabilities. The integration of these diverse resources requires a more nuanced security strategy that addresses the specific risks associated with each type of hardware.

Therefore, usage of purpose-built tools to monitor and infer incursions should be used to create dedicated pipelines for cybersecurity. For example, NVIDIA Morpheus [60] uses pre-trained ML models within a pipeline framework to collect cybersecurity information, and detect anomalous behaviours across a data centre. In addition, Burstein [61] presents the Data Processing Unit (DPU) architecture for accelerating infrastructure processes, and taking them off the main CPU of the processing nodes. Vilalta *et al.* [62] show the combination of DPU and Morpheus to isolate the cybersecurity mechanisms from the host machines, allowing for smarter analysis of traffic on clusters. These modifications should bring the overall security of AI applications on HPC higher. In the final section, we discuss the limitations of the work done, as well as what future steps can be taken to improve above this analysis.

C. Limitations and Future Work

As with any study, our research has limitations. First, we do not employ a systematic literature review approach, although we use established frameworks to map the studies. Second, while we discuss the mitigation strategies for AI security risks on HPC, we do not cover reproducing the threats or validating

the mitigations. Third, we do not provide an agenda for HPC centres by ranking the vulnerabilities according to criticality or offering a secure AI technology adoption roadmap. Instead, this study takes the first step toward secure AI applications on HPC systems by introducing the threat landscape and mapping the challenges. Future studies should address these limitations to lower the computation barrier for start-ups, SMEs, and researchers by enabling a secure HPC-integrated computing environment for AI applications.

REFERENCES

[1] E. Strohmaier, J. J. Dongarra, H. W. Meuer, and H. D. Simon, "The marketplace of high-performance computing," *Parallel Computing*, vol. 25, no. 13, pp. 1517–1544, Dec. 1999.

[2] R. Bommasani *et al.*, *On the Opportunities and Risks of Foundation Models*, arXiv:2108.07258 [cs], Jul. 2022. DOI: 10.48550/arXiv.2108.07258.

[3] J. Baker, "The technology–organization–environment framework," *Information Systems Theory: Explaining and Predicting Our Digital Society, Vol. 1*, pp. 231–245, 2012.

[4] S. K. P. A. and R. S. Rao Kunte, "ABCD Analysis of Industries Using High-Performance Computing," *International Journal of Case Studies in Business, IT, and Education*, pp. 448–465, Jun. 2023. DOI: 10.47992/IJCSBE.2581.6942.0282.

[5] J. Li, S. Wang, S. Rudinac, and A. Osseyran, "High-performance computing in healthcare: An automatic literature analysis perspective," *Journal of Big Data*, vol. 11, no. 1, p. 61, May 2024, ISSN: 2196-1115. DOI: 10.1186/s40537-024-00929-2.

[6] A. Cavelan, R. M. Cabezón, M. Grabarczyk, and F. M. Ciorba, "A Smoothed Particle Hydrodynamics Mini-App for Exascale," en, in *Proceedings of the Platform for Advanced Scientific Computing Conference*, arXiv:2005.02656 [cs], Jun. 2020, pp. 1–11. DOI: 10.1145/3394277.3401855.

[7] K. Lee and S. Lee, "Knowledge Structure of the Application of High-Performance Computing: A Co-Word Analysis," en, *Sustainability*, vol. 13, no. 20, p. 11 249, Oct. 2021, ISSN: 2071-1050. DOI: 10.3390/su132011249.

[8] A. Rousset, B. Herrmann, C. Lang, and L. Philippe, "A survey on parallel and distributed multi-agent systems for high performance computing simulations," *Computer Science Review*, vol. 22, pp. 27–46, Nov. 2016, ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2016.08.001.

[9] T. Sterling, M. Brodowicz, and M. Anderson, *High Performance Computing: Modern Systems and Practices*, en. Morgan Kaufmann, Dec. 2017, Google-Books-ID: qOHIBAAAQBAJ, ISBN: 978-0-12-420215-3.

[10] E. Suarez *et al.*, "Modular Supercomputing Architecture: A success story of European *R&D*," JSC, Tech. Rep. 9, 2022, p. 24. DOI: 10.5281/zenodo.6508394.

[11] P. García-Risueño and P. Ibáñez, "A review of High Performance Computing foundations for scientists," *International Journal of Modern Physics C*, vol. 23, Jul. 2012. DOI: 10.1142/S0129183112300011.

[12] A. Akram, *Architectures for secure high performance computing*, en, UC Davis.

[13] H. Jin *et al.*, "High performance computing using MPI and OpenMP on multi-core parallel systems," *Parallel Computing*, Emerging Programming Paradigms for Large-Scale Scientific Computing, vol. 37, no. 9, pp. 562–575, Sep. 2011, ISSN: 0167-8191. DOI: 10.1016/j.parco.2011.02.002.

[14] J. Kaplan *et al.*, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[15] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[16] D. Narayanan *et al.*, "Efficient large-scale language model training on gpu clusters using megatron-lm," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–15.

[17] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "Zero: Memory optimizations toward training trillion parameter models," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2020, pp. 1–16.

[18] J. Sevilla *et al.*, "Compute trends across three eras of machine learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2022, pp. 1–8.

[19] S. Peisert, "Security in high-performance computing environments," *Commun. ACM*, vol. 60, no. 9, pp. 72–80, Aug. 2017, ISSN: 0001-0782. DOI: 10.1145/3096742.

[20] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The science dmz: A network design pattern for data-intensive science," in *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2013, pp. 1–10. DOI: 10.1145/2503210.2503245.

[21] M. A. Jette and T. Wickberg, "Architecture of the slurm workload manager," in *Job Scheduling Strategies for Parallel Processing*, D. Klusáček, J. Corbalán, and G. P. Rodrigo, Eds., Cham: Springer Nature Switzerland, 2023, pp. 3–23, ISBN: 978-3-031-43943-8.

[22] A. Prout *et al.*, "Enhancing hpc security with a user-based firewall," in *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, 2016, pp. 1–4. DOI: 10.1109/HPEC.2016.7761641.

[23] T. Hou, T. Wang, D. Shen, Z. Lu, and Y. Liu, "Autonomous security mechanisms for high-performance computing systems: Review and analysis," in *Adaptive Autonomous Secure Cyber Systems*, S. Jajodia *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 109–129, ISBN: 978-3-030-33432-1. DOI: 10.1007/978-3-030-33432-1_6.

[24] Y. Guo *et al.*, *High-performance computing security:* Feb. 2024. DOI: 10.6028/nist.sp.800-223.

[25] R. Keller Tesser and E. Borin, "Containers in hpc: A survey," *The Journal of Supercomputing*, vol. 79, no. 5, pp. 5759–5827, Mar. 2023, ISSN: 1573-0484. DOI: 10.1007/s11227-022-04848-y.

[26] N. Huq, P. Lin, R. Reyes, and C. Perine, "A survey of cloud-based gpu threats and their impact on ai, hpc, and cloud computing," Trend Research, Tech. Rep., 2024.

[27] B. T. Familoni, "Cybersecurity challenges in the age of ai: Theoretical approaches and practical solutions," *Computer Science & IT Research Journal*, vol. 5, no. 3, pp. 703–724, 2024.

[28] M. Roshanaei, M. R. Khan, and N. N. Sylvester, "Navigating ai cybersecurity: Evolving landscape and challenges," *Journal of Intelligent Learning Systems and Applications*, vol. 16, no. 3, pp. 155–174, 2024.

[29] M. Blowers and J. Williams, "Artificial intelligence presents new challenges in cybersecurity," in *Disruptive Technologies in Information Sciences IV*, SPIE, vol. 11419, 2020, pp. 75–81.

[30] N. Kaloudi and J. Li, "The ai-based cyber threat landscape: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–34, 2020.

[31] L. Muñoz-González and E. C. Lupu, "The security of machine learning systems," *AI in Cybersecurity*, pp. 47–79, 2019.

[32] Y. Hu *et al.*, "Artificial intelligence security: Threats and countermeasures," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.

[33] K. Strier, J. Clark, and S. Khareghani, *Measuring compute capacity: A critical step to capturing AI's full economic potential*, OECD AI Policy Observatory, Accessed: 2023-08-30, Feb. 2022.

[34] N. Ahmed, M. Wahed, and N. C. Thompson, "The growing influence of industry in ai research," *Science*, vol. 379, no. 6635, pp. 884–886, 2023.

[35] T. F. Blauth, O. J. Gstrein, and A. Zwitter, "Artificial intelligence crime: An overview of malicious use and abuse of ai," *IEEE Access*, vol. 10, pp. 77 110–77 122, 2022. DOI: 10.1109/ACCESS.2022.3191790.

[36] M. Komorowski, D. C. Marshall, J. D. Salciccioli, and Y. Crutain, "Exploratory data analysis," in *Secondary Analysis of Electronic Health Records*. Cham: Springer International Publishing, 2016, pp. 185–203, ISBN: 978-3-319-43742-2. DOI: 10.1007/978-3-319-43742-2_15.

[37] D. Milojicic, P. Faraboschi, N. Dube, and D. Roweth, "Future of hpc: Diversifying heterogeneity," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 276–281. DOI: 10.23919/DATE51398.2021.9474063.

[38] Epoch AI, *Parameter, compute and data trends in machine learning*, Accessed: 2024-08-26, 2022.

[39] M. Ye *et al.*, *Enabling performant and secure eda as a service in public clouds using confidential containers*, 2024. arXiv: 2407.06040 [cs.CR].

[40] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PloS one*, vol. 12, no. 5, e0177459, 2017.

[41] M. Factor, K. Meth, D. Naor, O. Rodeh, and J. Satran, "Object storage: The future building block for storage systems," in *2005 IEEE International Symposium on Mass Storage Systems and Technology*, IEEE, 2005, pp. 119–123.

[42] N. C. Rajasekar and C. O. Imafidon, "Exploitation of vulnerabilities in cloud-storage," *GSTF Journal on Computing (JoC)*, vol. 1, no. 2, 2014.

[43] M. Blanc, K. Guerin, J.-F. Lalande, and V. Le Port, "Mandatory access control implantation against potential nfs vulnerabilities," in *2009 International Symposium on Collaborative Technologies and Systems*, IEEE, 2009, pp. 195–200.

[44] B. Bhushan, G. Sahoo, and A. K. Rai, "Man-in-the-middle attack in wireless and computer networking—a review," in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*, IEEE, 2017, pp. 1–6.

[45] P. R. Kumar, P. H. Raj, and P. Jelciana, "Exploring data security issues and solutions in cloud computing," *Procedia Computer Science*, vol. 125, pp. 691–697, 2018, The 6th International Conference on Smart Computing and Communications, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.12.089.

[46] A. Paracha, J. Arshad, M. B. Farah, and K. Ismail, "Machine learning security and privacy: A review of threats and countermeasures," *EURASIP Journal on Information Security*, vol. 2024, no. 1, pp. 1–23, 2024.

[47] N. S. Harzevili, J. Shin, J. Wang, and S. Wang, *Characterizing and understanding software security vulnerabilities in machine learning libraries*, 2022. arXiv: 2203.06502 [cs.SE].

[48] W. Jiang *et al.*, "An empirical study of artifacts and security risks in the pre-trained model supply chain," in *Proceedings of the 2022 ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses*, 2022, pp. 105–114.

[49] H. Chen and M. A. Babar, "Security for machine learning-based software systems: A survey of threats, practices, and challenges," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, 2024.

[50] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*, Springer, 2004, pp. 286–295.

[51] A. Martin, S. Raponi, T. Combe, and R. Di Pietro, "Docker ecosystem – vulnerability analysis," *Computer Communications*, vol. 122, pp. 30–43, 2018, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2018.03.011.

[52] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16*, Springer, 1996, pp. 104–113.

[53] M. Seaborn and T. Dullien, "Exploiting the dram rowhammer bug to gain kernel privileges," *Black Hat*, vol. 15, no. 71, p. 2, 2015.

[54] P. Kocher *et al.*, "Spectre attacks: Exploiting speculative execution," in *40th IEEE Symposium on Security and Privacy (S&P'19)*, 2019.

[55] M. Lipp *et al.*, "Meltdown: Reading kernel memory from user space," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

[56] Martín Abadi *et al.*, *Using tensorflow securely / tensorflow documentation*.

[57] J. Gomes *et al.*, "Enabling rootless linux containers in multi-user environments: The udocker tool," *Computer Physics Communications*, vol. 232, pp. 84–97, 2018, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2018.05.021.

[58] M. Scheerman *et al.*, *Secure platform for processing sensitive data on shared hpc systems*, 2021. arXiv: 2103.14679 [cs.CR].

[59] E. C. Security and I. R. Team, *Attacks on multiple hpc sites*, Accessed:2024-08-29.

[60] NVIDIA Corporation, *Nvidia morpheus*, Accessed: 2024-08-30, 2024.

[61] I. Burstein, "Nvidia data center processing unit (dpu) architecture," in *2021 IEEE Hot Chips 33 Symposium (HCS)*, 2021, pp. 1–20. DOI: 10.1109/HCS52781.2021.9567066.

[62] R. Vilalta *et al.*, "Providing anomalous behaviour profiling by extending smartnic transceiver support in packet-optical networks," in *2024 Optical Fiber Communications Conference and Exhibition (OFC)*, 2024, pp. 1–3.