A Survey of AI Vulnerability Detection in Distributed Systems and Cloud Environments

www.surveyx.cn

Abstract

This survey paper explores the transformative role of Artificial Intelligence (AI) in modern computing, particularly focusing on vulnerability detection and distributed systems within cloud environments. The integration of AI technologies, such as Large Language Models (LLMs), has significantly advanced cybersecurity measures, enabling more effective threat identification and mitigation. AI-driven solutions have demonstrated substantial improvements in performance and scalability, bridging the gap between AI capabilities and user needs. In the domain of fault tolerance and network security, robust mechanisms are highlighted as critical for maintaining system reliability, especially in high-performance computing (HPC) and large-scale data processing applications. The survey identifies a need for more robust frameworks to enhance model interpretability and address ethical concerns, emphasizing the importance of interdisciplinary collaboration in edge AI research. It underscores the necessity of developing adaptive frameworks that balance safety, privacy, and performance, while addressing potential misuse of LLMs. Future research should focus on holistic vulnerability handling processes, adaptive models for evolving threats, and integration with existing security frameworks. The survey advocates for fostering open and responsible AI evaluation practices to advance the field and ensure the security of digital infrastructures.

1 Introduction

1.1 Significance of AI in Modern Computing

Artificial Intelligence (AI) has emerged as a transformative force in modern computing, significantly influencing various scientific and engineering fields [1]. Its integration across diverse domains has driven advancements in automation, efficiency, and decision-making processes. The rapid growth of Internet of Things (IoT) devices highlights the demand for efficient sensor architectures, a need that AI technologies adeptly fulfill [2]. AI's impact is particularly evident in automating data processing and enhancing experimental workflows, particularly in the integration of scientific instruments with computational resources [3].

Moreover, AI's convergence with Operating Systems (OS) enhances performance and security, optimizing system operations and mitigating vulnerabilities [4]. The flexibility of cloud infrastructure, which allows rapid resource provisioning for on-demand computational tasks, is further augmented by AI's capability to manage scalability challenges in complex operations [5].

Despite the advancements, a gap persists between the capabilities of foundational AI models and their practical applications, often requiring traditional software development methods [6]. Bridging this gap is crucial for realizing AI's full potential in delivering tailored solutions across various sectors. As AI continues to evolve, its role in reshaping computing paradigms and pushing technological boundaries remains undeniable.

Generated by SurveyX.cn.

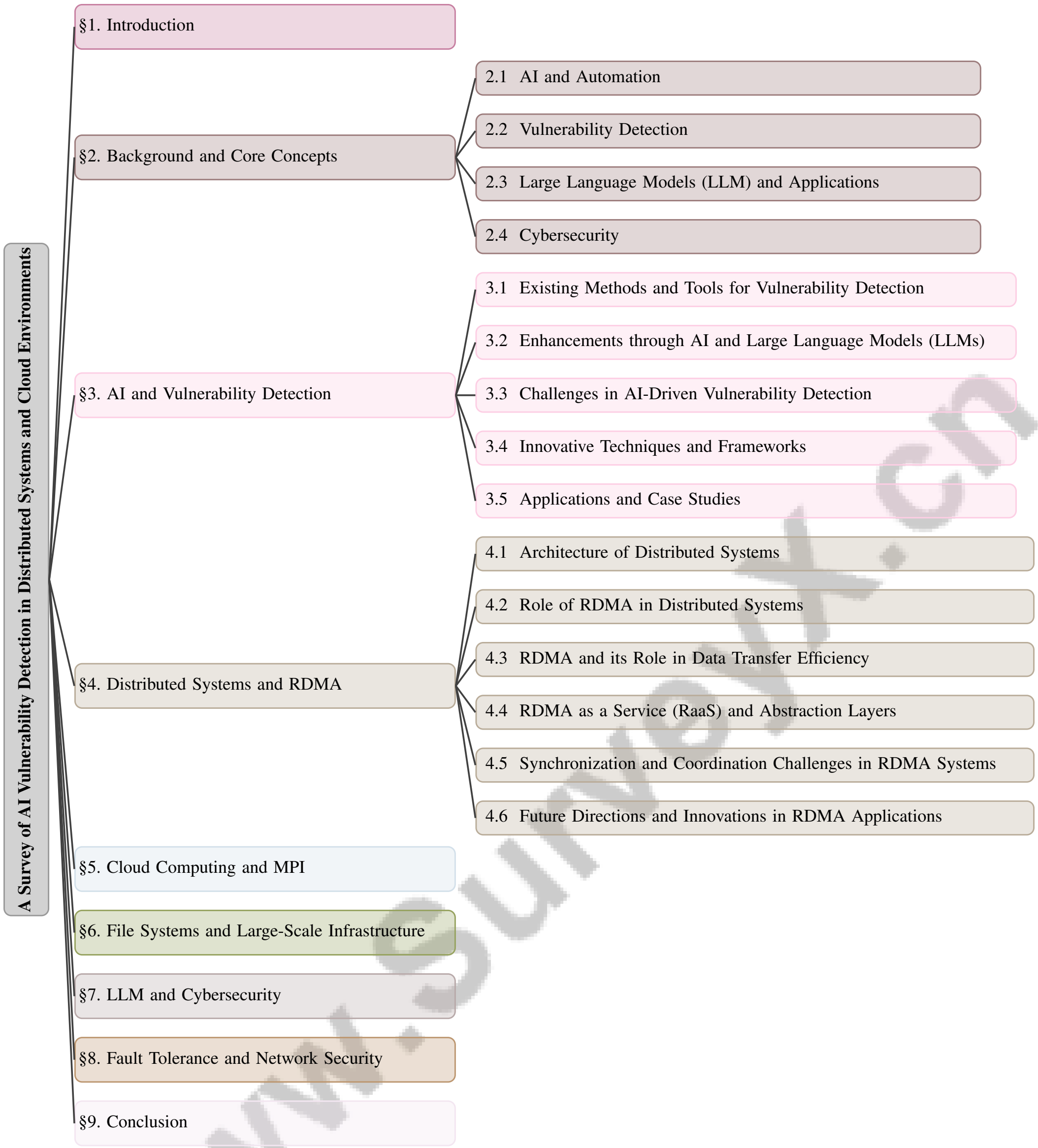

A Survey of AI Vulnerability Detection in Distributed Systems and Cloud Environments
§1. Introduction
§2. Background and Core Concepts
2.1 AI and Automation
2.2 Vulnerability Detection
2.3 Large Language Models (LLM) and Applications
2.4 Cybersecurity
§3. AI and Vulnerability Detection
3.1 Existing Methods and Tools for Vulnerability Detection
3.2 Enhancements through AI and Large Language Models (LLMs)
3.3 Challenges in AI-Driven Vulnerability Detection
3.4 Innovative Techniques and Frameworks
3.5 Applications and Case Studies
§4. Distributed Systems and RDMA
4.1 Architecture of Distributed Systems
4.2 Role of RDMA in Distributed Systems
4.3 RDMA and its Role in Data Transfer Efficiency
4.4 RDMA as a Service (RaaS) and Abstraction Layers
4.5 Synchronization and Coordination Challenges in RDMA Systems
4.6 Future Directions and Innovations in RDMA Applications
§5. Cloud Computing and MPI
§6. File Systems and Large-Scale Infrastructure
§7. LLM and Cybersecurity
§8. Fault Tolerance and Network Security
§9. Conclusion

Figure 1: chapter structure

1.2 Interconnectedness of AI and Cybersecurity

The intricate relationship between AI and cybersecurity is essential for safeguarding digital infrastructures. Notably, the education sector witnessed a staggering 576% increase in phishing attacks from 2021 to 2022, emphasizing the need for robust cybersecurity measures [7]. The complexity and diversity of AI introduce significant challenges regarding the reuse of AI artifacts, further necessitating the establishment of strong cybersecurity frameworks to protect these technologies [1].

The potential misuse of Large Language Models (LLMs) as vectors for malware delivery illustrates the dual-edged nature of AI technologies, serving both as tools for innovation and as threats [8]. This duality calls for a comprehensive approach to integrating AI into cybersecurity practices, ensuring that advances in AI do not inadvertently compromise system security [9].

Additionally, the effectiveness of current flaw reporting practices for LLMs poses challenges, particularly in fostering a culture of responsible disclosure for AI systems [10]. Addressing these issues requires an interdisciplinary approach, underscoring the need for a structured framework to assess trust in LLM-based automation agents [11]. This integrated perspective is crucial for developing resilient cybersecurity strategies that leverage AI's potential while mitigating associated risks.

AI-generated, for reference only.

1.3 Structure of the Survey

This survey is systematically organized to provide a comprehensive exploration of AI-driven vulnerability detection within distributed systems and cloud environments. It begins with an **Introduction**, which establishes the significance of AI in modern computing, its interconnectedness with cybersecurity, and the overall structure of the paper. The subsequent **Background and Core Concepts** section delves into foundational technologies and concepts, including AI's role in automation, techniques for vulnerability detection, and the importance of distributed systems.

The third section, **AI and Vulnerability Detection**, examines how AI enhances vulnerability detection processes, discussing existing methods, challenges, and innovative frameworks. Following this, the **Distributed Systems and RDMA** section addresses the architecture of distributed systems and the role of Remote Direct Memory Access (RDMA) in improving data transfer efficiency.

The fifth section, **Cloud Computing and MPI**, focuses on resource management and the application of Message Passing Interface (MPI) for parallel computing in cloud environments. The sixth section, **File Systems and Large-Scale Infrastructure**, analyzes the significance of file systems and the challenges associated with managing extensive datasets and complex computations.

The survey further investigates **LLM and Cybersecurity**, highlighting the application of large language models in detecting and mitigating security threats. The eighth section, **Fault Tolerance and Network Security**, discusses strategies for maintaining system reliability and ensuring data integrity. Finally, the **Conclusion** summarizes key findings and insights, reflecting on the current state and future directions of AI, vulnerability detection, and distributed systems in cloud environments.The following sections are organized as shown in Figure 1.

2 Background and Core Concepts

2.1 AI and Automation

AI significantly enhances automation and efficiency across computing environments, notably in high-performance computing (HPC) and distributed systems. Its integration into operating systems is vital for optimizing performance and security amidst growing system complexities [4]. ColonyOS exemplifies AI's role in improving computational task execution via distributed executors across heterogeneous platforms [12]. In machine learning, TensorFlow facilitates algorithm execution on diverse devices, crucial for large-scale tasks [13], while Scavenger cloud service optimizes training time and cost in distributed learning environments [14]. AI chain engineering streamlines operations by assembling workflows without traditional programming [6].

AI's impact extends to distributed machine learning, where parallelization and model coherence are crucial due to increasing data demands [15]. The MIT SuperCloud dataset highlights AI's role in addressing resource allocation and energy consumption challenges in HPC [16]. Additionally, AI's application in national security underlines the need for transparency and trust in framework development [17].

AI is transforming automation in industrial IoT, edge computing, and hybrid cloud systems, driven by AI and machine learning advancements, communication technologies like 5G and 6G, and OS integration for real-time applications such as immersive video conferencing and autonomous vehicles. AI techniques enhance innovative applications and infrastructure sustainability, paving the way for adaptive computing systems addressing societal challenges [4, 18, 19, 20, 21].

2.2 Vulnerability Detection

Detecting software system vulnerabilities is crucial for digital infrastructure security. Automated methodologies are necessary to identify software vulnerabilities (SVs) that pose risks like data breaches and system failures [22]. Traditional methods, such as honey files, are less effective against sophisticated threats like ransomware, prompting the evolution of detection techniques [23]. Large Language Models (LLMs) have emerged as effective tools for vulnerability detection, enabling precise assessments and descriptions [24]. They facilitate automated penetration testing, essential for comprehensive assessments [25]. The high-dimensional nature of network traffic data, especially

AI-generated, for reference only.

regarding DoS attacks, presents challenges for machine learning algorithms, necessitating advanced approaches to maintain accuracy [26].

The complexity of modern software environments, where vulnerabilities propagate through interconnected components, complicates detection [27]. Initiatives like the MIT Supercloud Dataset foster innovative AI/ML approaches for analyzing datacenter operations, enhancing resource utilization and security [16]. Automated benchmarks for identifying software source code vulnerabilities are critical for advancing detection methodologies [28].

Federated learning systems' vulnerability, where gradient updates can lead to data reconstruction attacks, underscores the need for robust strategies to protect data privacy [29]. Integrating AI into OS functionalities, such as memory management and intrusion detection, is vital for optimizing security measures and managing software complexities [4].

2.3 Large Language Models (LLM) and Applications

Large Language Models (LLMs) have become integral across domains, advancing fields like telecommunications and cybersecurity. They optimize network performance, especially with emerging 6G technologies, enhancing data processing and decision-making [30]. In cybersecurity, LLMs are used for intrusion detection, malware analysis, and phishing detection, providing robust threat mitigation mechanisms. Frameworks like Automated Progressive Red Teaming (APRT) automate vulnerability identification, enhancing security posture [31]. LLM architectures, categorized into encoder-only, encoder-decoder, and decoder-only models, require tailored approaches for distinct challenges in vulnerability detection and repair [32].

Evaluating LLMs involves comprehensive benchmarks, including datasets with 5,000 code samples from synthetic and real-world projects [33]. The Big-Vul dataset, derived from real-world projects documented in the Common Vulnerabilities and Exposures (CVE) database, offers insights into software vulnerabilities, facilitating secure system development [24].

Training LLMs in heterogeneous network environments presents challenges requiring innovative solutions to improve efficiency and scalability. Research on distributed training clusters addresses these challenges and proposes enhancements for LLM scalability [34]. FedRDMA demonstrates LLMs' potential to optimize federated learning infrastructures by improving data transfer in cross-silo scenarios [35].

LLMs play a crucial role in protecting sensitive information through 'data defenses', empowering data owners to prevent LLMs from inferring sensitive information [36]. Transparency in LLM operations is critical, focusing on model reporting, evaluation results, explanations, and uncertainty communication [20]. These efforts are vital for building trust in AI-based automation agents, addressing trust categories, challenges, and considerations for LLM frameworks [11].

The survey of compound AI systems explores archetypes and approaches to LLM-based end-to-end optimization, providing a structured understanding of effective LLM integration into broader AI systems [37]. Through diverse applications, LLMs drive innovation and provide critical solutions across multiple domains, contributing to advancements in cybersecurity and beyond. Understanding their capabilities and limitations is essential for leveraging their full potential while mitigating associated risks [38].

2.4 Cybersecurity, Fault Tolerance, and Network Security

In distributed and cloud environments, cybersecurity, fault tolerance, and network security are crucial for data integrity and system reliability. The complexity and scale of these environments necessitate robust protective measures against potential vulnerabilities. The rise of data breaches and insider threats in cloud systems highlights the need for comprehensive cybersecurity strategies, especially in sensitive fields like biomedical research, where existing methods often fall short [39].

The evolution of distributed systems, particularly in HPC, introduces new fault tolerance challenges. Ensuring robustness is critical for IoT devices prone to instability [40]. Recovery mechanisms for distributed iterative solvers are essential for computational reliability in exascale systems [41]. The shift towards high-bandwidth networks, driven by AI workloads in data centers and HPC clusters, exacerbates network latency issues, affecting communication-intensive applications [42].

Network security is integral to distributed environments, where inefficient data movement can create vulnerabilities. Limitations of existing database architectures in utilizing memory and compute resources under high-performance demands necessitate innovative approaches to enhance security and performance [43]. Effective communication and coordination among autonomous entities are crucial for mitigating security vulnerabilities in distributed systems [44].

Security policies in distributed systems must incorporate robust mechanisms to control information flow between processes handling sensitive data [45]. The challenges of implementing Federated Computing (FC) systems, enabling collaborative processing while ensuring data privacy and compliance with regulatory frameworks, emphasize secure data handling practices [46].

As distributed and cloud environments evolve, developing sophisticated strategies in cybersecurity, fault tolerance, and network security becomes increasingly critical due to growing complexity, heterogeneity, and security threats. Advanced techniques, such as AI for vulnerability detection and repair, alongside efficient algorithms for secure computations in distributed systems, are essential for mitigating risks and enhancing resilience. These approaches focus on real-time threat detection and preemptive measures, ensuring vulnerabilities are identified and addressed before deployment, safeguarding the integrity and privacy of distributed networks against accidental and malicious faults [47, 27, 44]. Leveraging innovative technologies and comprehensive frameworks, these systems can achieve resilience against evolving cyber threats, ensuring reliable operation of critical infrastructures.

3 AI and Vulnerability Detection

Category	Feature	Method
Existing Methods and Tools for Vulnerability Detection	Collaborative Strategies	DSMP[44]
Enhancements through AI and Large Language Models (LLMs)	Security Integration	ACE[6], ND[48], DD[36]
Innovative Techniques and Frameworks	Performance Enhancement Resilience and Fault Tolerance Security and Compliance	TSoR[49] CDC[40] DMSPE[45]
Applications and Case Studies	Cybersecurity Techniques System Enhancement	N/A[23] rDLB[50], N/A[14], CVM[51], CDPA[29]

Table 1: This table summarizes the various methodologies and tools employed in vulnerability detection, highlighting existing strategies, advancements through AI and Large Language Models (LLMs), and innovative techniques. It categorizes these approaches into four main areas: existing methods, AI and LLM enhancements, innovative frameworks, and applications, providing a comprehensive overview of the current landscape and future directions in cybersecurity.

Exploring the intersection of Artificial Intelligence (AI) and vulnerability detection necessitates examining current methodologies and tools that form this domain's foundation. Understanding these approaches is vital for appreciating the advancements AI introduces. As illustrated in **??**, the hierarchical structure of AI and vulnerability detection highlights existing methodologies, AI and LLM enhancements, challenges, innovative techniques, and applications. This figure categorizes the field into five main areas: existing methods and tools, AI enhancements, challenges, innovative frameworks, and applications, each with detailed subcategories. Table 1 presents a comprehensive summary of methods and tools used in vulnerability detection, detailing the evolution from traditional techniques to advanced AI-driven solutions. Additionally, Table 2 provides a detailed comparison of various methods employed in vulnerability detection, illustrating their optimization strategies, target environments, and unique features. The following subsection discusses established methods and tools, emphasizing their roles, effectiveness, and operational contexts, thereby paving the way for a detailed discussion on the enhancements brought about by AI and Large Language Models (LLMs).

3.1 Existing Methods and Tools for Vulnerability Detection

The field of vulnerability detection is characterized by diverse methodologies and tools tailored to address specific challenges in identifying security threats in software and systems. A comprehensive framework categorizes existing research into algorithm development, application domains, and evaluation metrics, facilitating a structured understanding of the field [44]. This classification enables systematic exploration of vulnerability detection techniques and their effectiveness across various contexts. Figure 2 illustrates this hierarchical classification, providing a visual representation of how these methods and tools are organized.

In distributed systems, methodologies such as Distributed Self Management Protocols (DSMP) enhance decision-making for intrusion detection and resource allocation by enabling entities to share local security information [44]. Collaborative approaches are essential for identifying vulnerabilities in complex environments where data and resources are spread across multiple nodes.

Integrating network architectures like NetDAM, which combines memory access with Ethernet communication, exemplifies efforts to improve data processing efficiency, thus supporting effective vulnerability detection processes [48]. Efficient data handling is critical in large-scale systems where timely identification of security threats is paramount.

Specialized file systems such as SentryFS generate and strategically place honey files throughout the file system. By continuously updating their content and metadata, SentryFS enhances the attractiveness of these files to ransomware, serving as decoys to detect and mitigate malicious activities [23]. This underscores the importance of deception techniques in modern vulnerability detection strategies.

Benchmarking efforts analyzing six open-source models trained for vulnerability detection against general-purpose LLMs like CodeBERT and GPT-4 provide insights into model effectiveness and areas for improvement [52]. These benchmarks are crucial for evaluating the performance of various tools and methodologies in real-world scenarios.

The application of multiscale models in distributed computing environments, as demonstrated in experiments involving canal system models and hydrology applications, highlights the potential of diverse programming languages and frameworks to tackle complex vulnerability detection challenges [53]. This emphasizes the need for flexible solutions capable of operating across multiple scales and domains.

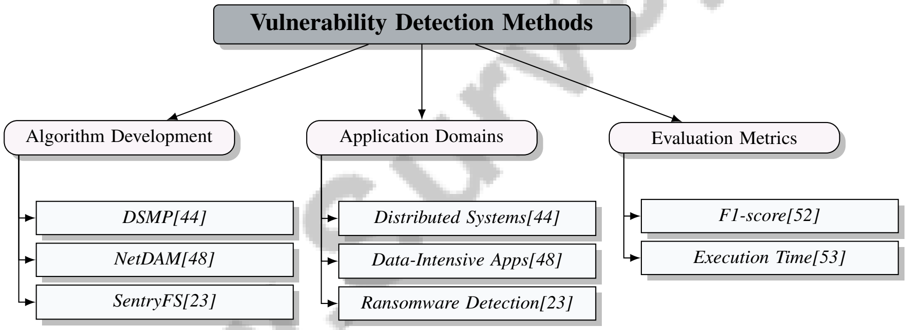


Figure 2: This figure illustrates the hierarchical classification of existing methods and tools for vulnerability detection, categorizing them into algorithm development, application domains, and evaluation metrics.

3.2 Enhancements through AI and Large Language Models (LLMs)

Integrating AI and LLMs into vulnerability detection processes has significantly improved the accuracy, efficiency, and adaptability of security threat identification. LLMs, such as ChatGPT, have shown significant advancements in automated vulnerability detection and repair, often outperforming traditional methods [21]. These models utilize advanced data analysis and fault diagnosis to optimize network performance, especially in complex environments like 6G networks [48].

The Scavenger innovation combines conventional parallel scaling with stochastic gradient descent noise insights to estimate time and cost across different configurations, enhancing vulnerability detection processes [14]. Additionally, integrating AI into operating system frameworks has improved resource efficiency and system agility, further supporting LLM-based solutions [6].

In distributed systems, scalable architectures that integrate multiple technologies and machine learning techniques facilitate real-time anomaly detection, improving the accuracy and speed of vulnerability detection [23]. The CoRD networking architecture, which allows efficient OS-level control over RDMA communication, enhances data processing capabilities, thereby strengthening the overall security framework [48].

LLMs are categorized into architectures such as encoder-only, encoder-decoder, and decoder-only models, each affecting their cybersecurity applications [37]. Despite their potential, LLMs generally underperform compared to traditional transformer-based models in some aspects of vulnerability detection, particularly in vulnerability description [52]. Comprehensive benchmarks have been established to evaluate LLMs' capabilities across various software security tasks, providing insights for future improvements [38].

Secure prompting mechanisms have been explored for enhancing vulnerability detection in LLM-integrated systems, particularly against prompt injection attacks [29]. Proposed data defenses leverage adversarial prompt injections to obscure sensitive information, preventing LLMs from making accurate inferences [36]. These methods enhance LLM robustness in cybersecurity applications, protecting sensitive data.

The integration of AI and LLMs into vulnerability detection processes significantly enhances security threat identification and mitigation. This is achieved through capabilities such as analyzing source code across multiple programming languages, generating interactive honeypots for attacker engagement, and employing machine learning techniques to detect vulnerabilities before software deployment, ultimately improving cybersecurity effectiveness [54, 55, 27, 56, 57]. By optimizing training processes and adopting comprehensive evaluation frameworks, these technologies continue to bolster the resilience of software systems against evolving cyber threats.

3.3 Challenges in AI-Driven Vulnerability Detection

Implementing AI in vulnerability detection presents multifaceted challenges spanning technical, operational, and infrastructural domains. A primary technical challenge is the inadequacy of existing dynamic loop self-scheduling methods, which lack fault tolerance and rely on reactive strategies, limiting their effectiveness in handling failures [50]. Researchers also face difficulties supervising the LLM optimizer to produce accurate parameter adjustments, complicated by interdependent parameters within these systems [37].

Operational challenges are exacerbated by significant issues in existing benchmarks, such as mislabeling and inadequate dataset quality, adversely affecting model training and performance [52]. These deficiencies hinder the development of robust AI models capable of accurately detecting vulnerabilities. Additionally, the overhead introduced by multiscale frameworks in distributed environments often goes inadequately measured by current benchmarks, leading to inefficiencies in performance assessment [53].

Infrastructurally, the absence of hardware-level cache coherence in distributed systems complicates buffer management and concurrency control, critical for maintaining system integrity and performance [43]. Furthermore, the unreliable nature of Internet of Things (IoT) devices and wireless networks poses challenges, as these systems are prone to high recovery overheads and the loss of time-sensitive information during failures [40].

Addressing these challenges is essential for advancing AI-driven vulnerability detection and ensuring robust cybersecurity measures in increasingly complex digital environments. By leveraging innovative AI solutions and comprehensive frameworks, the cybersecurity field can adapt to the rapidly evolving landscape of cyber threats. These advancements, particularly through LLMs and machine learning techniques, enable automated vulnerability detection, malware analysis, and proactive defense strategies. As cyber attacks become more sophisticated, developing diverse and representative datasets, along with enhancing interpretability in AI models, is crucial for improving threat detection and response capabilities. This multifaceted approach not only addresses current vulnerabilities but also lays the groundwork for future research and robust security measures [17, 27, 58, 59, 56].

3.4 Innovative Techniques and Frameworks

The evolution of AI-driven vulnerability detection has been significantly propelled by innovative techniques and frameworks designed to enhance security threat identification and mitigation. One advancement is Coded Distributed Computing (CDC), which ensures continuity of deep neural network (DNN) computations amidst failures, maintaining operational integrity in distributed environments [40]. This method is vital for distributed systems where fault tolerance is essential.

Prompt engineering techniques in benchmark evaluations have enhanced the efficacy of LLMs by structuring prompts for better contextualization. This approach improves models' ability to detect and mitigate adversarial threats and enhances the security of systems integrating LLMs. Research indicates that tailored prompts can lead to more accurate identification of software vulnerabilities and improve performance in various cybersecurity tasks such as malware analysis and network intrusion detection [60, 6, 32, 59, 28]. These advancements are crucial for protecting AI models from malicious exploitation and ensuring reliable operation in cybersecurity applications.

In network technologies, the TSoR technique allows applications to utilize standard POSIX sockets while benefiting from Remote Direct Memory Access (RDMA) performance enhancements without requiring code modifications [49]. This innovation facilitates the seamless integration of RDMA's high-performance capabilities into existing systems, enhancing data transfer efficiency and supporting effective vulnerability detection.

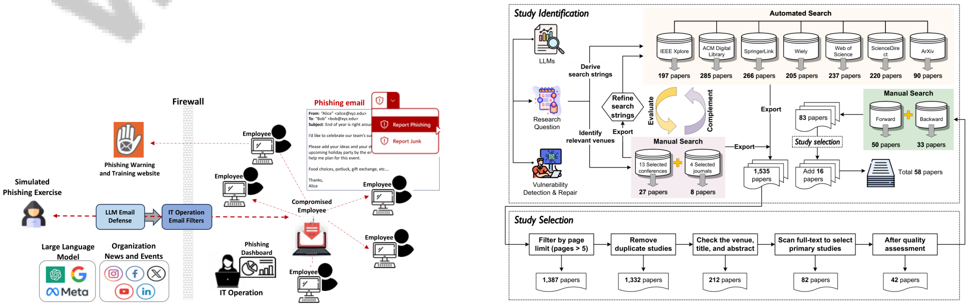
The Distributed Machine Security Policy Enforcement (DMSPE) framework models distributed systems with security policies incorporating filter functions to control information flow and ensure compliance through local verification [45]. This framework is crucial for maintaining data integrity and compliance in complex distributed environments where rigorous enforcement of security policies is necessary.

Furthermore, the architecture for distributed shared-memory databases that separates compute and memory nodes allows for independent scaling and enhanced performance through RDMA networking [43]. This architecture provides a foundation for developing robust systems capable of efficiently managing distributed data and computations.

SentryFS exemplifies the effectiveness of dynamically adaptive honey files that align with evolving ransomware strategies, improving early detection chances [23]. Additionally, the rDLB approach effectively utilizes idle processing times for task rescheduling, minimizing failure impacts and ensuring timely task completion [50].

The survey of compound AI systems introduces a framework that draws analogies from program analysis to understand how LLM optimizers can be prompted to optimize compound AI systems [37]. This framework is critical for advancing the optimization of complex AI-driven systems.

Innovative techniques and frameworks in recent literature signify substantial progress in AI-driven vulnerability detection within cybersecurity. These advancements leverage machine learning and natural language processing to identify security vulnerabilities before software deployment, analyze compliance with security standards, and enhance vulnerability repair processes. LLMs are being utilized for various cybersecurity tasks, including malware analysis and network intrusion detection, although challenges such as limited training datasets and the need for more interpretable models persist. Collectively, these developments offer robust, efficient, and scalable solutions tailored to the dynamic and increasingly sophisticated landscape of cybersecurity threats [56, 59, 27]. By leveraging cutting-edge technologies and methodologies, the field continues to enhance the security posture of digital infrastructures against an ever-changing threat landscape.



From: "Alice" <alice@xyz.edu>
To: "Bob" <bob@xyz.edu>
Subject: End of year is right arou

I'd like to celebrate our team's su

Please add your ideas and your st
upcoming holiday party by the er
help me plan for this event.

Food choices, potluck, gift exchange, etc....

Thanks,
Alice

(a) Phishing Email Scam Prevention and Detection System[7]

(b) Automated and Manual Search Process for Identifying and Selecting Studies on Vulnerability Detection and Repair[32]

Figure 3: Examples of Innovative Techniques and Frameworks

AI-generated, for reference only.

As shown in Figure 3, the integration of AI for vulnerability detection has ushered in innovative techniques and frameworks, exemplified by two distinct systems. The first, a Phishing Email Scam Prevention and Detection System, employs a combination of firewalls, simulated phishing exercises, LLMs, and IT operations to meticulously analyze and filter potentially harmful emails. By leveraging LLM Email Defense and IT Operation Email Filters, it effectively identifies and neutralizes phishing attempts, even when a compromised employee inadvertently receives a malicious email. The second example illustrates an Automated and Manual Search Process designed to identify and select studies on vulnerability detection and repair. This process is structured into three main phases: Study Identification, Study Selection, and Study Selection, beginning with automated searches across renowned databases such as IEEE Xplore and ACM Digital Library. Together, these examples highlight the cutting-edge methodologies and strategic frameworks that AI brings to the forefront of vulnerability detection and cybersecurity [7, 32].

3.5 Applications and Case Studies

AI's application in vulnerability detection spans various domains, showcasing its potential to enhance cybersecurity measures and operational efficiency. A notable case study involves AI-driven solutions in smart microgrid systems, where distributed systems consist of multiple prosumers interacting with a centralized controller. This setup highlights AI's role in optimizing resource allocation and intrusion detection rates, thereby improving overall performance [51].

In cloud computing environments, adopting distributed shared-memory architectures supported by Remote Direct Memory Access (RDMA) has led to significant performance and scalability improvements in database systems. This is particularly beneficial in scenarios demanding high data throughput and low latency, as evidenced by notable improvements in simulation performance achieved through unified cloud-enabled architectures [43]. These advancements underscore AI's role in facilitating efficient data management and processing in complex cloud environments.

The Scavenger cloud service exemplifies AI's capability to optimize training time and cost in distributed machine learning tasks, achieving up to a 2× reduction in training time and over 50

In federated AI-enabled infrastructures, the CDPA framework effectively mitigates data reconstruction attacks while preserving model utility and reducing communication costs. This approach is particularly valuable in critical infrastructures where data privacy and energy efficiency are paramount [29]. The integration of AI in these systems ensures robust data protection and operational sustainability.

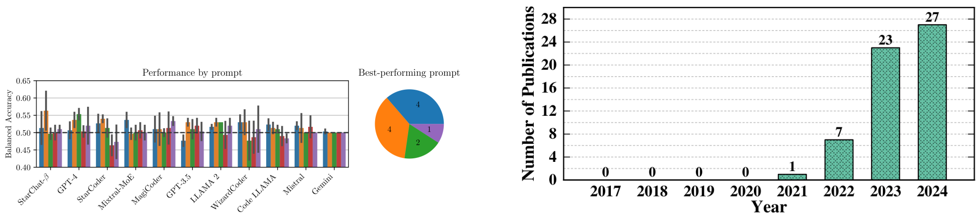
The evaluation of AI applications in distributed systems, as demonstrated by the MIT SuperCloud dataset, reveals varying performance across jobs with different GPU allocations, highlighting the necessity for optimized scheduling and resource management to maximize AI-driven process efficiency [16]. These insights are crucial for advancing AI's role in managing large-scale computational tasks.

Moreover, AI in distributed machine learning has led to significant advancements in systems and techniques, despite ongoing challenges in performance and fault tolerance [15]. This survey emphasizes the need for continuous innovation to address the complexities of distributed environments and enhance the robustness of AI-driven solutions.

The SentryFS system significantly enhances the effectiveness of honey files against modern ransomware by adapting to their evolving strategies, demonstrating AI's role in deception-based vulnerability detection strategies [23]. Additionally, the rDLB approach has shown significant improvements in robustness and performance, allowing applications to tolerate up to P-1 processor failures and achieving execution times up to seven times faster under latency perturbations compared to methods without rDLB [50].

These case studies and applications illustrate AI's transformative impact on vulnerability detection, providing innovative solutions that enhance the security and resilience of digital infrastructures. By leveraging AI's capabilities, organizations can better protect against evolving cyber threats and ensure system integrity. Future research should focus on refining machine learning models to reduce false positives, exploring additional data sources for enhanced context, and developing sophisticated algorithms for anomaly detection [61].

As shown in Figure 4, the application of AI in vulnerability detection has garnered significant attention, as illustrated by recent studies and case analyses. The first figure, "Performance by prompt," highlights the comparative performance of various AI models, such as StarChat-, GPT-4,



(a) Performance by prompt[62]

(b) Number of Publications Over Time[32]

Figure 4: Examples of Applications and Case Studies

StarCoder, Mixtral-MoE, and MagiCoder, across different prompts. This bar chart uses color coding to differentiate the models, with the y-axis indicating their balanced accuracy, thus offering insights into the efficacy of these models in handling diverse vulnerability detection tasks. Meanwhile, the second figure, "Number of Publications Over Time," tracks the burgeoning academic interest in this domain, showcasing a marked increase in publications from 2023 onwards. This trend underscores the growing recognition and exploration of AI's potential in enhancing vulnerability detection capabilities, as researchers and practitioners seek to leverage AI technologies to preemptively identify and mitigate security threats. Together, these figures encapsulate the dynamic landscape of AI applications in vulnerability detection, emphasizing both technological advancements and the expanding body of research dedicated to this critical area [62, 32].

Feature	Distributed Self Management Protocols (DSMP)	NetDAM	SentryFS
Optimization Approach	Collaborative Decision-making	Data Processing Efficiency	Deception Techniques
Target Environment	Distributed Systems	Network Architectures	File Systems
Unique Feature	Local Security Sharing	Memory Access With Ethernet	Honey Files

Table 2: A comparative analysis of three distinct methods for vulnerability detection, highlighting their optimization approaches, target environments, and unique features. The table contrasts Distributed Self Management Protocols (DSMP), NetDAM, and SentryFS, elucidating their respective strategies such as collaborative decision-making, data processing efficiency, and deception techniques.

4 Distributed Systems and RDMA

In distributed systems, the architecture is fundamental to optimizing performance and ensuring reliability, influencing resource management and communication protocols like Remote Direct Memory Access (RDMA). RDMA enhances data movement efficiency across distributed systems, a topic explored further in the following subsection, which discusses the architecture of these systems and RDMA's contributions.

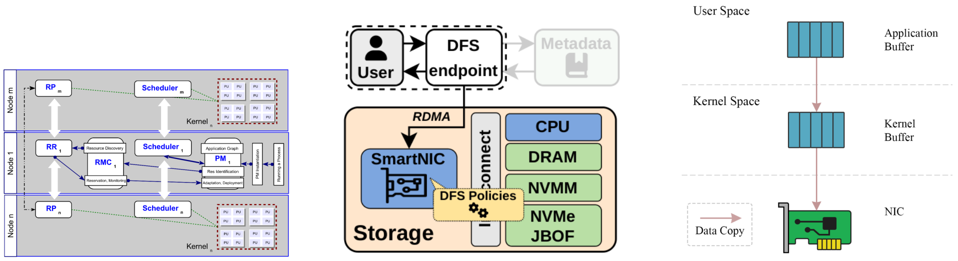
4.1 Architecture of Distributed Systems

Distributed systems architecture optimizes task execution across interconnected nodes, offering scalability, fault tolerance, and resource management benefits. A key element is communication protocols such as RDMA, which reduces latency by enabling direct memory access without OS intervention, crucial for elastic applications [63]. Fault-tolerant mechanisms like PartRePer-MPI enhance resilience by partially replicating MPI processes, ensuring rapid recovery from failures while maintaining low overheads [64, 65]. Robust self-scheduling methods such as rDLB optimize resource allocation, even amidst processor failures [50].

Efficient task distribution is supported by dual-component systems like DIR net, which enhances reliability through real-time fault detection and recovery [66]. Distributed systems also facilitate large dataset computation across nodes, leveraging methods like distributed Principal Subspace Analysis (PSA) for scalable data processing [67]. These architectures employ decentralized resource management strategies, enhancing performance and scalability [68].

As depicted in Figure 5, the architecture of distributed systems involves hierarchical resource management, SmartNIC integration with DFS policies, and NIC data transfer processes. These

AI-generated, for reference only.



(a) Resource Management System Architecture[69]

(b) SmartNIC with DFS Policies[70]

(c) Data Transfer Process in NIC[68]

Figure 5: Examples of Architecture of Distributed Systems

elements underscore the importance of efficient resource management and data transfer for optimal system performance [69, 70, 68].

4.2 Role of RDMA in Distributed Systems

RDMA enhances data transfer efficiency in distributed systems by enabling high-throughput, low-latency communication through direct memory access, crucial for HPC and large-scale data environments [63]. Innovations like Efficient RDMA-based Synchronous Mirroring (E-SM) exemplify RDMA's potential in enhancing data replication reliability [63]. However, security vulnerabilities in RDMA and NVMe-oF protocols pose significant threats, necessitating robust security measures [71].

RDMA continues to advance distributed computing by facilitating efficient data transfer and addressing performance and synchronization challenges. Innovations like the Remote Fetching Paradigm (RFP) enhance IOPS, while RDMA as a Service (RaaS) simplifies deployment [72, 73, 74].

4.3 RDMA and its Role in Data Transfer Efficiency

RDMA significantly improves data transfer efficiency by reducing overhead through direct memory access, enabling high-speed communication in distributed systems [75]. Innovations like continuity hashing and Talos enhance RDMA's role by ensuring consistent and efficient data operations [76, 77]. Systems like Storm and Nova-LSM demonstrate RDMA's ability to support complex operations with minimal latency [78, 79].

RDMA's adaptability is seen in systems like Modularis, which allows minimal code alterations for platform changes [80]. Innovations like rFaaS and IRN further enhance RDMA's efficiency by reducing communication overhead and improving packet loss handling [81, 82]. Pilot-Data and Erda optimize data transfer by separating logical data units and ensuring atomicity in RDMA writes [83, 84].

Challenges in RDMA include synchronization issues, addressed by advancements like KRCORE and Collie, which optimize connection establishment and performance anomaly detection [85, 86]. RDMA's technical aspects, such as reduced software overhead and advanced flow control, significantly enhance data transfer efficiency, supporting critical applications like machine learning and remote storage [87, 73].

4.4 RDMA as a Service (RaaS) and Abstraction Layers

RaaS simplifies RDMA integration by abstracting complex low-level operations, enhancing CPU and memory utilization, and providing a user-friendly interface for developers [49]. Abstraction layers, involving hardware and software innovations, ensure high performance and scalability [73]. TSoR exemplifies seamless RDMA integration, enhancing distributed applications' performance without significant infrastructure changes [49].

Innovations like Remote Direct Cache Access (RDCA) and new mutual exclusion primitives further optimize RDMA performance by enhancing bandwidth utilization and ensuring fairness [86, 88].

Justitia highlights the role of software-based abstraction layers in optimizing RDMA performance and resource management [89].

RaaS and its abstraction layers streamline RDMA integration, enhancing distributed systems' efficiency and scalability across various applications [90, 72].

4.5 Synchronization and Coordination Challenges in RDMA Systems

Synchronization and coordination in RDMA systems are challenged by managing high-speed data transfers while ensuring coherence and performance. Congestion control dynamics pose significant challenges, with simplified models often leading to inefficiencies [91]. Innovations like ALock reduce congestion but face challenges with high contention [92].

As illustrated in Figure 6, the key challenges and methods in synchronization and coordination within RDMA systems are highlighted, emphasizing critical areas such as congestion control, synchronization methods, and infrastructure challenges. Advanced congestion management techniques like Dart's one-RTT convergence are crucial for maintaining throughput, although infrastructure variability affects efficiency [93]. Reconfigurable atomic transaction commit methods reduce resource requirements and improve fault tolerance but must be managed carefully to remain effective [94].

Limitations in Distributed Network Processors (DNP) impact bandwidth and latency, necessitating optimization for improved RDMA system performance [95].

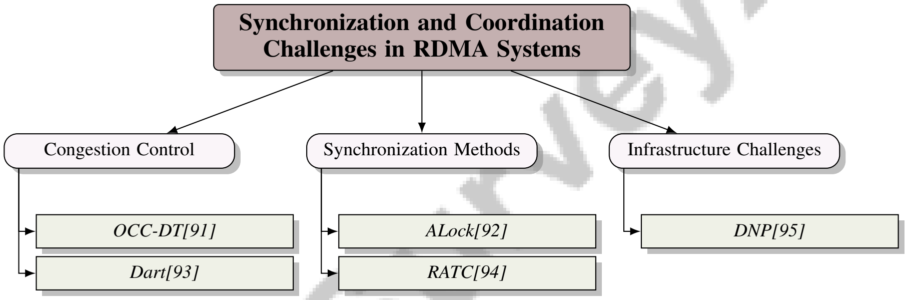

Synchronization and Coordination Challenges in RDMA Systems
Congestion Control
Synchronization Methods
Infrastructure Challenges
OCC-DT[91]
Dart[93]
ALock[92]
RATC[94]
DNP[95]

Figure 6: This figure illustrates the key challenges and methods in synchronization and coordination within RDMA systems, highlighting congestion control, synchronization methods, and infrastructure challenges.

4.6 Future Directions and Innovations in RDMA Applications

Future RDMA applications will focus on enhancing performance, scalability, and integration across diverse environments. Innovations in RNICs and dynamic performance isolation, like in Justitia, promise better integration with programmable NICs [96, 89]. Robust synchronization algorithms and KRCORE optimization in virtualized environments will advance RDMA's applicability [88, 85].

Fault tolerance innovations, like dynamic replication strategies in PartRePer-MPI, will enhance RDMA systems' resilience [64]. Expanding search spaces for anomaly detection will improve RDMA implementations' robustness [86].

RDMA's future is characterized by optimizing performance, enhancing integration, and expanding applicability. Innovations like the Remote Fetching Paradigm (RFP) and NP-RDMA will ensure RDMA's continued role in high-performance computing and data-intensive tasks [87, 74].

5 Cloud Computing and MPI

As cloud computing advances, enhancing its efficacy, especially in high-performance computing (HPC), is crucial. A central focus is developing scalable resource management strategies to optimize resource utilization and ensure efficient operations across various applications. This section explores scalable resource management in cloud computing, emphasizing its importance and innovative solutions to address resource allocation and management challenges in dynamic cloud environments.

AI-generated, for reference only.

5.1 Scalable Resource Management in Cloud Computing

Optimizing computational efficiency through scalable resource management is vital for seamless HPC operations. Techniques like Nova-LSM's dynamic range management improve scalability and performance by adapting data distribution and storage [79]. This is essential for managing the complexities of large-scale data processing.

The effectiveness of scalable resource management is demonstrated in multi-GPU clusters, which enhance the execution of computationally intensive tasks through parallel processing, thus handling extensive datasets and complex computations efficiently [97]. These strategies are further supported by comprehensive evaluations of fault tolerance techniques and resource management strategies across diverse HPC applications [98].

Tools like Sapper highlight the significance of human-AI collaboration in optimizing resource allocation and management in cloud computing [6]. These tools facilitate effective resource scaling, ensuring cloud infrastructures meet the growing demands of AI-driven applications.

Scalable resource management encompasses innovative data management techniques, advanced computational resources, and collaborative development tools. These strategies enhance cloud environments' ability to manage modern computational tasks, such as machine learning training and big data analytics, by optimizing resource allocation and ensuring secure data handling. This results in efficient service delivery, reduced training times and costs, and addresses critical data security concerns in sensitive applications like biomedical research [14, 39].

5.2 MPI in Parallel Computing

The Message Passing Interface (MPI) is integral to enabling parallel computing in cloud-based systems, facilitating efficient communication and workload distribution across multiple nodes. Its architecture enhances data exchange and task coordination, crucial for executing complex computational tasks in parallel environments. MPI addresses challenges in large-scale systems, such as energy efficiency and resiliency, while optimizing communication locality to improve application performance and reduce energy consumption. By employing advanced process placement strategies that minimize communication costs and account for potential node failures, MPI streamlines resource management and enhances the performance of parallel applications, particularly in HPC systems [99, 69, 100, 75, 65].

Distributed TensorFlow MPI (DTMPI) exemplifies MPI's effectiveness in cloud-based parallel computing by distributing workloads across multiple devices, enhancing machine learning tasks' scalability and efficiency [101]. This approach leverages MPI's communication protocols to synchronize processes and manage data flow efficiently.

The TOpology and Fault Aware (TOFA) process placement method further optimizes MPI's performance by minimizing communication costs and accounting for potential node failures, improving resilience and efficiency in MPI-based applications [65]. This optimization is crucial for maintaining high performance in cloud environments, where resource availability and network conditions vary.

MPI's role is also evident in algorithms for distributed principal subspace analysis, facilitating communication among distributed nodes [67]. Additionally, integrating MPI within containerized environments, such as those utilizing HPE Cray MPI, allows users to create and run MPI-based applications with optimal performance [102].

MPI's robust communication framework and ability to manage distributed workloads make it indispensable for parallel computing in cloud-based systems. By enhancing resource utilization and process coordination through advanced mapping techniques, MPI effectively meets the increasing demands of HPC applications in modern cloud environments. This optimization reduces communication costs and completion times—by up to 31

5.3 Enhancing Performance with RDMA and MPI

Integrating Remote Direct Memory Access (RDMA) and MPI enhances performance in cloud environments by optimizing data transfer and communication processes critical for HPC applications. RDMA's high bandwidth and low latency capabilities, combined with MPI's comprehensive framework for managing distributed workloads, create a powerful synergy that improves overall system

AI-generated, for reference only.

performance. This combination enables efficient resource utilization and enhances input/output operations per second (IOPS), as demonstrated by advancements like the Remote Fetching Paradigm (RFP), which leverages RDMA's superior in-bound read capabilities to optimize data retrieval processes, yielding significant performance gains in applications like in-memory key-value stores [89, 73, 87, 74, 103].

RDMA minimizes CPU overhead and accelerates data transfer speeds, as exemplified by Erda, which reduces Non-Volatile Memory (NVM) writes by approximately 50

MPI enhances parallel computing by efficiently handling distributed tasks, as seen in implementations like CryptMPI, which optimizes encrypted MPI communication using AES-GCM through multi-threading and pipelining. This ensures data integrity and privacy while maintaining high performance, crucial for secure cloud computing applications. MPI's ability to optimize communication patterns and resource allocation significantly reduces communication costs and improves energy efficiency, particularly in large-scale systems. Recent advancements, such as topology-aware process placement, have demonstrated that strategic mapping can decrease job completion times by up to 31

Platforms like rFaaS utilize RDMA to accelerate Function-as-a-Service (FaaS) applications, optimizing resource allocation and invocation paths for latency-sensitive tasks. CoRD introduces a novel high-performance networking approach that eliminates the need for traditional kernel bypass methods, allowing efficient OS-level management of the RDMA dataplane, enhancing high-performance cloud applications' overall efficiency [4, 104, 34, 105, 68].

Experiments on advanced infrastructures like the LUMI supercomputer highlight the effectiveness of MPI implementations in cloud environments. Studies emphasize the critical role of optimized communication protocols in enhancing performance, particularly through deploying containerized MPI applications. By leveraging tailored container solutions and advanced process placement strategies, researchers have achieved significant improvements in job completion times and reduced communication costs, underscoring the importance of efficient resource allocation and data management in HPC [99, 100, 106, 102, 65]. The integration of RDMA and MPI not only enhances performance but also addresses modern traffic patterns and performance requirements, as highlighted by surveys comparing various RDMA implementations with emerging alternatives.

The combination of RDMA and MPI provides a powerful framework for enhancing performance in cloud environments, enabling efficient data transfer and communication processes that support the growing demands of HPC applications. By integrating the strengths of cloud computing and advanced optimization techniques, cloud systems can enhance scalability, efficiency, and reliability, particularly for resource-intensive tasks like machine learning model training. This integration allows for optimal cluster configurations, significantly reducing training times and costs while addressing potential security risks and improving data migration across different cloud environments. Furthermore, adopting innovative frameworks leveraging emerging technologies, such as quantum computing and AI, further enhances cloud systems' performance and adaptability in meeting complex workloads' demands [107, 14, 19, 39, 108].

5.4 Security and Fault Tolerance in Cloud-MPI Environments

Ensuring security and fault tolerance in cloud environments utilizing MPI is paramount for maintaining the integrity and reliability of HPC applications. Integrating secure communication protocols within MPI frameworks protects sensitive data from unauthorized access and ensures data confidentiality. CryptMPI exemplifies this by implementing fast encrypted MPI communication using AES-GCM, leveraging multi-threading and pipelining to maintain high performance while securing data exchanges [109].

Secure data migration between cloud providers is critical for maintaining data integrity and confidentiality. A systematic approach to data migration protects sensitive information during transfer, reducing data breach risks and ensuring compliance with data protection regulations [107]. This method is vital for organizations managing large data volumes in cloud infrastructures.

Fault tolerance is equally important in cloud-MPI environments, where system reliability must be maintained despite potential failures. The MATCH benchmark suite provides a comprehensive framework for evaluating different MPI fault tolerance techniques, guiding the development of efficient designs tailored to specific scenarios [98]. This benchmarking is essential for identifying effective strategies to enhance system resilience.

The Horizon set management solution, utilizing the Session model, reduces deadlock risks in MPI applications without significant performance impacts. This approach offers better scalability compared to traditional methods, ensuring operational efficiency under challenging conditions [110]. Additionally, the TOFA method improves MPI job completion times and reduces job abort rates, demonstrating effectiveness in optimizing performance and resilience in HPC environments [65].

Flexibility in cloud architectures for resource provisioning based on data sensitivity is crucial for addressing security and fault tolerance measures. By utilizing encryption and other protective mechanisms, cloud environments can dynamically adjust resource allocation to meet specific security requirements [39]. This adaptability is vital for effectively managing varying workloads while safeguarding sensitive data.

Collectively, integrating secure communication protocols, comprehensive data migration strategies, and advanced fault tolerance techniques enhances the reliability and security of HPC services in cloud-MPI environments. Secure data migration mechanisms safeguard sensitive information during transfers, while libraries like CryptMPI ensure encrypted communication without significantly compromising performance. Implementing Byzantine error detection expands fault tolerance, enabling systems to manage a wider range of errors beyond traditional fail-stop scenarios. This multifaceted approach ensures high levels of service availability, security, and operational efficiency in cloud-MPI environments [107, 109, 111]. By leveraging advanced technologies and comprehensive evaluation frameworks, these systems effectively address challenges associated with modern cloud-based infrastructures.

5.5 Integration of Advanced Technologies

Integrating advanced technologies within cloud computing environments is pivotal for enhancing performance, scalability, and reliability across diverse applications. Utilizing RDMA (Remote Direct Memory Access) improves data transfer efficiency and reduces latency in cloud infrastructures. RDMA enables direct memory access between nodes without CPU intervention, accelerating data movement and optimizing cloud-based operations [49]. This technology is particularly beneficial in scenarios requiring high-throughput and low-latency communication, such as HPC and large-scale data processing.

Adopting containerization technologies, like Kubernetes, facilitates the orchestration and management of cloud resources, enhancing the scalability and flexibility of cloud services. Kubernetes' networking model, integrated with RDMA, provides transparency and performance improvements for containerized applications, allowing them to leverage advanced networking capabilities without significant modifications to existing infrastructure [49].

Incorporating advanced machine learning frameworks, such as TensorFlow and PyTorch, within cloud environments exemplifies the integration of cutting-edge technologies. These frameworks enable executing complex machine learning models at scale, leveraging cloud resources to accelerate training and inference processes. Distributed computing techniques, supported by technologies like MPI, enhance task parallelization, improving the efficiency and speed of machine learning operations in cloud settings [101].

The implementation of advanced security protocols, such as CryptMPI, ensures the protection of sensitive data within cloud-MPI environments. By employing encryption techniques like AES-GCM, CryptMPI maintains data confidentiality and integrity while minimizing performance overhead, enabling secure and efficient communication in cloud-based applications [109].

Integrating advanced technologies in cloud computing environments is crucial for enhancing performance and meeting the increasing demands of modern applications, particularly in sectors like big data analytics and artificial intelligence, where cost-effective resource provisioning and secure data handling are essential. Innovative architectures ensuring data security during transit, in use, and at rest are vital for sensitive fields like biomedical research. Moreover, optimization techniques for selecting cloud configurations can significantly reduce training time and costs for machine learning models, highlighting the importance of balancing parallel and statistical efficiency in achieving optimal performance [14, 39]. By leveraging innovations in networking, containerization, machine learning, and security, cloud systems can achieve greater efficiency, scalability, and reliability, ensuring they remain at the forefront of technological advancement.

6 File Systems and Large-Scale Infrastructure

6.1 Importance of File Systems in Large-Scale Infrastructures

File systems are integral to the operational efficiency of large-scale infrastructures, facilitating data storage, retrieval, and management in high-performance computing (HPC) environments. They are designed to handle vast data volumes, ensuring accessibility for complex computations. However, central filesystems often encounter inefficiencies and high loads in large-scale parallel jobs, leading to performance degradation [112]. This underscores the need for resilient file system architectures capable of managing data flow and minimizing bottlenecks.

File systems must meet the high throughput and low latency demands of modern applications, particularly as the need for efficient data access grows. Innovations like remote direct memory access (RDMA) and offloading storage policies to programmable SmartNICs have significantly reduced latency and CPU utilization. Distributed file systems (DFS) for edge devices further illustrate the complexity of developing robust architectures that adapt to high mobility and network partitioning, ensuring data accessibility across diverse computing environments. Optimizing file system performance is crucial to preventing I/O bottlenecks and enhancing overall system efficiency [69, 113, 100, 70, 112].

File systems also support application scalability. As computational demands escalate, file systems must evolve to maintain consistent performance amid fluctuations in data volume or user concurrency. Technologies like RDMA and SmartNICs enhance data transfer efficiency and reduce latency, while distributed file systems designed for edge environments tackle challenges related to mobility and network partitioning. Effective resource management in many-core systems is critical for optimizing resource allocation and enhancing system performance [69, 113, 76, 70, 112]. This scalability is vital for maintaining operational efficiency in dynamic, resource-intensive settings.

6.2 Challenges in Managing Extensive Datasets

Managing extensive datasets in large-scale infrastructures presents significant challenges due to the inherent complexity and scale of the data. A primary challenge is the accurate identification and analysis of anomalies, crucial for effective data-driven root cause analysis. These methods rely on modeling complex interactions and transitions between states for timely anomaly detection and mitigation [114]. Efficient handling of vast data volumes is essential to prevent disruptions.

Interference and performance degradation in memory access patterns also pose challenges. Analyzing interference in RDMA and local memory systems reveals significant impacts on memory access latencies and throughputs [115]. Such interference can lead to bottlenecks and reduced performance, particularly in high-throughput, low-latency environments.

Robust mechanisms for managing the vast data generated by large-scale computing operations are essential. The MIT SuperCloud dataset, with over one million job statistics and time series data from CPU and GPU usage, exemplifies the scale of data management required [16]. Advanced data management strategies and infrastructure are necessary to support HPC applications.

Addressing these challenges involves implementing advanced anomaly detection techniques, optimizing memory access for efficient data retrieval, and establishing robust data handling capabilities. These challenges highlight the need for sophisticated architectures, such as those employing Flume agents for log data storage and machine learning models for anomaly prediction, alongside security measures to guard against emerging threats [38, 59, 39, 7, 116]. Addressing these challenges is crucial for ensuring reliability and performance in HPC environments.

6.3 Solutions for Complex Computations

Innovative solutions are essential for addressing complex computational challenges in large-scale systems, enhancing both performance and scalability. A file-based communication architecture leveraging local filesystems instead of a central filesystem improves performance in large-scale parallel jobs by reducing central resource load and mitigating bottlenecks [112].

Research aims to expand the set of sub-operators to support additional analytics types, optimizing performance across various execution environments [80]. By developing versatile and efficient analytics tools, large-scale systems can better accommodate diverse computational tasks, enhancing their capacity to process extensive datasets and execute complex algorithms effectively.

These strategies enhance computational capabilities through advanced resource management, AI model deployment, and secure cloud architectures. Frameworks like HPCFair facilitate efficient reuse of AI models and datasets, while innovative resource management schemes such as ElCore optimize resource allocation in many-core systems, and secure cloud architectures safeguard sensitive data [39, 1, 117, 69].

6.4 Advancements in File System Technologies

Recent advancements in file system technologies have enhanced data management capabilities within large-scale computing environments. These advancements focus on improving computation and communication efficiencies, vital for the scalability and performance of HPC systems [118]. Local filesystems for message transfers minimize reliance on central filesystems, alleviating bottlenecks and boosting performance. Secure copy protocols (scp) ensure efficient and secure data transfers across distributed nodes [112].

These advancements are crucial for managing the escalating complexity and scale of modern computing tasks. Integrating advanced file system architectures with optimized data transfer mechanisms enables higher throughput and lower latency, supporting complex computational workloads. This strategy lays a robust foundation for future advancements in data management and processing within HPC environments. Innovative resource management techniques, such as hybrid adaptive resource discovery in many-core systems, enhance resource utilization and enable efficient orchestration of complex computational tasks. Addressing challenges related to diminishing returns in large-scale distributed training by optimizing hardware configurations and parallelization strategies paves the way for substantial improvements in system performance and adaptability [1, 118, 69].

6.5 Performance Metrics and Benchmarks

Evaluating file system performance in large-scale infrastructures requires comprehensive metrics and benchmarks that capture the complexities of data management and processing. Throughput measures the volume of data processed within a specified timeframe, crucial for executing large-scale parallel jobs efficiently. Byun et al.'s approach illustrates a significant reduction in central filesystem load, enhancing throughput and scalability by distributing data management tasks across local filesystems [112].

Latency reflects the time required for data to travel from source to destination within the file system. Low latency is essential for maintaining responsiveness in HPC environments, particularly in scenarios demanding rapid data access and processing. Secure copy protocols (scp) for inter-node communications minimize latency through efficient data transfers across distributed nodes [112].

Scalability is also crucial; file systems must handle increasing data volumes and user demands without compromising performance. Byun et al.'s approach, which reduces reliance on central filesystems, exemplifies how scalability can be achieved through innovative data management strategies [112].

Benchmarks provide standardized evaluations of file system performance, simulating real-world workloads to assess capabilities under varying conditions. Utilizing performance benchmarks allows organizations to identify performance bottlenecks and fine-tune file system configurations, essential for adapting to evolving computing demands. Advanced techniques can significantly reduce training times and costs, highlighting the impact of precise configuration adjustments. Comprehensive analysis tools reveal application interactions with file systems, enabling proactive issue resolution and enhancing overall system performance [14, 119, 69].

Comprehensive performance metrics and benchmarks are vital for evaluating and enhancing file system performance in large-scale infrastructures. Prioritizing throughput, latency, scalability, and standardized benchmarks enables organizations to improve file systems to manage intricate HPC tasks.

This strategy facilitates efficient resource sharing and utilization in future many-core systems, ensuring robust performance across diverse applications, as evidenced by analyses of parallel I/O patterns and novel file-based communication architectures. Implementing these strategies can significantly improve overall system performance, particularly in large-scale computing environments where resource contention and filesystem overload are prevalent [112, 119, 69].

7 LLM and Cybersecurity

The intersection of Large Language Models (LLMs) and cybersecurity is rapidly advancing, offering innovative solutions to complex cyber threats. As cyber threats grow more sophisticated, LLMs enhance threat detection and mitigation strategies, particularly in phishing and malware defense.

7.1 Phishing and Malware Detection

LLMs significantly enhance cybersecurity by identifying and thwarting phishing and malware threats. The sophistication of LLM-generated phishing attacks necessitates advanced detection techniques and comprehensive training programs to strengthen defenses [7]. Utilizing natural language processing, LLMs analyze communication to detect malicious patterns and anomalies. In malware detection, generative AI techniques enable LLMs to analyze extensive datasets for malware signatures and behaviors, facilitating proactive threat mitigation [9]. This capability is essential for maintaining digital infrastructure integrity, allowing rapid identification of and response to emerging threats. Effective utilization of LLMs in cybersecurity depends on structured frameworks for responsible vulnerability disclosure, fostering trust and collaboration between developers and security professionals [10].

7.2 Network Management and Intrusion Detection

LLMs enhance network management and intrusion detection by processing extensive network data to detect patterns and anomalies indicative of threats [54, 38, 59]. Automating routine tasks like traffic monitoring and anomaly detection improves operational efficiency and reduces human error. In intrusion detection, LLMs analyze network logs and communication patterns to identify subtle signs of malicious activity, enhancing automated detection of vulnerabilities and real-time responses [54, 27]. Furthermore, LLMs optimize network configurations by recommending adjustments based on historical data and current conditions, advancing security through strategies like intent-based networking [120, 121].

7.3 Integration of LLMs in Cybersecurity

Integrating LLMs into cybersecurity enhances threat detection and mitigation. LLMs excel in processing large data volumes, improving vulnerability identification and threat detection [122]. Their ability to generate realistic responses is utilized in honeypot systems to understand attacker behavior and craft defense strategies [28]. Benchmarks are vital for evaluating LLM safety and reliability, promoting transparency and reproducibility in network management [28]. Insights from these benchmarks drive future research on LLM-augmented vulnerability detection systems, enhancing accuracy and efficiency [122].

7.4 Ethical and Security Challenges

The integration of LLMs into cybersecurity presents ethical and security challenges. The potential misuse of LLMs as proxies for malware attacks poses significant risks, necessitating stringent oversight and robust security measures [8]. Excessive trust in AI systems can lead to harmful behaviors if not managed properly, highlighting the need for transparency and accountability [11]. High computational costs and limitations in contextual understanding further complicate LLM application in cybersecurity, emphasizing the need for ongoing research and vigilance [21].

7.5 Future Directions and Research Opportunities

The future of LLMs in cybersecurity involves developing proactive defense strategies and refining alignment strategies to minimize misuse risks [56]. Integrating safety measures during model training enhances LLM resilience against adversarial attacks [56]. Developing robust datasets improves LLM accuracy in detecting cyber threats, and enhancing interpretability fosters trust among security professionals [59]. Integrating LLMs with other security technologies offers holistic cybersecurity solutions, addressing the multifaceted nature of modern challenges and ensuring LLMs evolve as a cornerstone of digital security [59].

8 Fault Tolerance and Network Security

8.1 Importance of Fault Tolerance in System Reliability

Fault tolerance is vital for maintaining reliability in high-performance computing (HPC) and distributed systems. The complexity of modern computing systems demands robust fault tolerance mechanisms to prevent system failures and ensure uninterrupted operations. Ineffective fault detection, isolation, and recovery can severely impact performance and lead to system breakdowns [66]. Innovations such as the NetDAM architecture, which minimizes communication overhead and enhances data processing efficiency, demonstrate effective solutions for high-bandwidth applications [48].

Robust strategies are essential for managing load imbalances in scientific applications that are computationally intensive, often due to irregular parallel tasks and HPC systems' susceptibility to faults [50]. Techniques like the rDLB approach enable efficient task rescheduling and resource optimization, ensuring consistent performance amid failures.

In distributed networks, localizing security compliance checks to individual processes enhances scalability and manageability, simplifying security verification while maintaining operational integrity against threats [67]. Continuous cybersecurity updates, including honey file strategies, are crucial for countering evolving threats and bolstering system resilience [23].

Advanced technologies like RedN facilitate complex offloads on commodity RDMA NICs, enhancing performance and fault resilience without the need for expensive SmartNICs. As systems grow, challenges such as energy efficiency and resilience to component failures become prominent. Effective fault tolerance strategies, including advanced error detection and optimized process placement, are crucial for minimizing disruptions and ensuring seamless operations across interconnected nodes [123, 65, 111].

In dynamic and resource-intensive environments, fault tolerance techniques like user-level failure mitigation (ULFM) and in-memory data replication enhance system resilience, enabling applications to recover gracefully from failures without losing progress. These strategies bolster application robustness, optimize resource utilization, and reduce recovery times, ensuring efficient performance in the face of failures [124, 123, 64, 125, 65]. Leveraging advanced technologies and evaluation frameworks, systems can enhance performance, minimize downtime, and improve resilience against failures and cyber threats.

8.2 Strategies for Achieving Fault Tolerance

Effective strategies for achieving fault tolerance are crucial for maintaining reliability and performance in distributed systems, especially with dynamic task scheduling and potential process failures. Enhancing the Message Passing Interface (MPI) with fault tolerance capabilities specific to Deep Learning algorithms ensures uninterrupted operations despite faults [126].

Implementing robust dynamic load balancing (rDLB) methods proactively manages task scheduling, optimizing resource utilization and maintaining system stability during execution [50]. This adaptability allows distributed systems to dynamically respond to varying workloads and resource conditions, enhancing fault tolerance.

The DIR net architecture exemplifies the integration of built-in recovery actions and user-defined strategies for various fault scenarios, ensuring effective responses to unexpected disruptions while maintaining operational continuity [66].

AI-generated, for reference only.

In data protection, SentryFS employs an AI scoring agent to assess the risk of write operations, creating virtual clones to prevent data loss and enhance resilience against corruption [23]. Future research should focus on optimizing workload distribution and enhancing fault tolerance mechanisms, alongside developing systems that autonomously adapt to fluctuating resource conditions [15]. Addressing vulnerabilities in technologies like the NVMe-over-Fabrics (NVMe-oF) protocol is also critical for safe deployment in production environments [71].

Integrating advanced fault tolerance strategies, such as MPI enhancements, robust dynamic load balancing, and proactive data protection measures, is vital for ensuring resilience and reliability in distributed systems. Utilizing advanced AIOps techniques, particularly those enhanced by developments in large language models (LLMs), enables effective failure management and sustains operational efficiency amid increasing complexity in computing environments. These methodologies ensure high availability and reliability of large-scale distributed software systems while addressing challenges like cross-platform generality and task flexibility. Additionally, incorporating artificial intelligence in vulnerability detection and prevention enhances robustness by identifying and mitigating potential security risks pre-deployment, supporting a comprehensive approach to failure management and operational resilience [127, 27].

8.3 Network Security Practices

Network security practices are critical for maintaining the integrity and reliability of distributed systems, particularly in environments vulnerable to process failures and malicious attacks. Implementing fault tolerance mechanisms that adapt recovery strategies to specific application needs is a key aspect of network security [124], ensuring operational continuity during disruptions.

The hybrid fault tolerance method significantly enhances performance and fault tolerance in distributed optimization tasks, even amid node failures [128]. This approach is vital for systems requiring uninterrupted operations despite challenges.

The Legio framework provides an effective network security solution by intercepting MPI calls, allowing applications to continue execution by excluding failed processes from computation [129]. This capability ensures sustained performance and reliability in the face of multiple process failures. The Horizon set approach further enhances system resilience through its modular and scalable fault management solutions for MPI applications [110].

However, existing fault-tolerant MPI libraries encounter challenges in leveraging efficient native MPI communications, which can hinder performance [64]. Addressing these issues is crucial for improving the scalability and efficiency of fault-tolerant practices in larger clusters.

The DIR net architecture enhances network security practices by providing dual-component efficiency in fault detection and recovery, allowing for user-defined recovery strategies tailored to specific application requirements [66]. Despite these advancements, limitations remain in current network security practices, particularly regarding scalability in larger clusters and the necessity for enhanced security measures in RDMA protocols. Experiments have shown that existing NVMe-oF security mechanisms are inadequate against identified vulnerabilities, underscoring the need for improved security measures in RDMA protocols [71].

9 Conclusion

The survey elucidates the profound influence of Artificial Intelligence (AI) on contemporary computing landscapes, with a particular focus on vulnerability detection and distributed systems in cloud environments. AI advancements, especially in Large Language Models (LLMs), have revolutionized cybersecurity by enhancing threat detection and response capabilities. The integration of AI solutions has notably improved system performance and scalability, as evidenced by innovations like Prompt Sapper, which aligns AI functionalities with user requirements to deliver tailored services.

Addressing fault tolerance and network security, the survey highlights the critical need for robust frameworks that ensure system reliability. Cutting-edge strategies in distributed systems bolster resilience and adaptability, essential for high-performance computing and large-scale data processing. Despite progress in machine learning, particularly within Federated Learning, challenges remain in evaluating Federated Computing systems in practical settings.

AI-generated, for reference only.

Significant advancements notwithstanding, there is an imperative for frameworks that enhance model interpretability and address ethical considerations. The survey underscores the value of interdisciplinary collaboration in edge AI research and the formulation of adaptive frameworks that balance safety, privacy, and performance. Additionally, the potential misuse of LLMs underscores the necessity for ongoing research to address security vulnerabilities and ensure safe deployment.

Future research should focus on comprehensive vulnerability management processes that integrate continual learning and efficient data collection. Exploring adaptive models to counter evolving malware threats and integrating LLMs with existing security frameworks are crucial for advancing cybersecurity. Furthermore, optimizing workflow execution through enhanced system integration with model repositories and persistent service tasks presents new research avenues.

The survey also emphasizes fostering a culture of open and responsible AI assessment, with an emphasis on standardized documentation and improved flaw reporting tools. As AI and distributed systems continue to evolve, these research directions will be instrumental in advancing the field, ensuring the resilience and security of digital infrastructures.

References

[1] Towards seamless management of ai models in high-performance computing.

[2] Bhanuka Manesha Samarasekara Vitharana Gamage and Vishnu Monn Baskaran. Simulation and analysis of distributed wireless sensor network using message passing interface, 2020.

[3] Rafael Vescovi, Ryan Chard, Nickolaus Saint, Ben Blaiszik, Jim Pruyne, Tekin Bicer, Alex Lavens, Zhengchun Liu, Michael E. Papka, Suresh Narayanan, Nicholas Schwarz, Kyle Chard, and Ian Foster. Linking scientific instruments and hpc: Patterns, technologies, experiences, 2022.

[4] Yifan Zhang, Xinkui Zhao, Ziying Li, Jianwei Yin, Lufei Zhang, and Zuoning Chen. Integrating artificial intelligence into operating systems: A comprehensive survey on techniques, applications, and future directions, 2024.

[5] José L. Risco-Martín, Kevin Henares, Saurabh Mittal, Luis F. Almendras, and Katzalin Olcoz. A unified cloud-enabled discrete event parallel and distributed simulation architecture, 2023.

[6] Zhenchang Xing, Qing Huang, Yu Cheng, Liming Zhu, Qinghua Lu, and Xiwei Xu. Prompt sapper: Llm-empowered software engineering infrastructure for ai-native services, 2023.

[7] Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. Large language model lateral spear phishing: A comparative study in large-scale organizational settings, 2024.

[8] Mika Beckerich, Laura Plein, and Sergio Coronado. Ratgpt: Turning online llms into proxies for malware attacks, 2023.

[9] Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, Norbert Tihanyi, Tamas Bisztray, and Merouane Debbah. Generative ai in cybersecurity: A comprehensive review of llm applications and vulnerabilities, 2025.

[10] Sean McGregor, Allyson Ettinger, Nick Judd, Paul Albee, Liwei Jiang, Kavel Rao, Will Smith, Shayne Longpre, Avijit Ghosh, Christopher Fiorelli, Michelle Hoang, Sven Cattell, and Nouha Dziri. To err is ai : A case study informing llm flaw reporting practices, 2024.

[11] Sivan Schwartz, Avi Yaeli, and Segev Shlomov. Enhancing trust in llm-based ai automation agents: New considerations and future challenges, 2023.

[12] Johan Kristiansson. Colonyos – a meta-operating system for distributed computing across heterogeneous platform, 2024.

[13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

[14] Sahil Tyagi and Prateek Sharma. Scavenger: A cloud service for optimizing cost and performance of ml training, 2023.

[15] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S. Rellermeyer. A survey on distributed machine learning, 2019.

[16] Siddharth Samsi, Matthew L Weiss, David Bestor, Baolin Li, Michael Jones, Albert Reuther, Daniel Edelman, William Arcand, Chansup Byun, John Holodnack, Matthew Hubbell, Jeremy Kepner, Anna Klein, Joseph McDonald, Adam Michaleas, Peter Michaleas, Lauren Milechin, Julia Mullen, Charles Yee, Benjamin Price, Andrew Prout, Antonio Rosa, Allan Vanterpool, Lindsey McEvoy, Anson Cheng, Devesh Tiwari, and Vijay Gadepally. The mit supercloud dataset, 2021.

[17] Erik Blasch, James Sung, Tao Nguyen, Chandra P. Daniel, and Alisa P. Mason. Artificial intelligence strategies for national security and safety standards, 2019.

[18] Elisa Bertino and Sujata Banerjee. Artificial intelligence at the edge, 2020.

[19] Deming Chen, Alaa Youssef, Ruchi Pendse, André Schleife, Bryan K Clark, Hendrik Hamann, Jingrui He, Teodoro Laino, Lav Varshney, Yuxiong Wang, et al. Transforming the hybrid cloud for emerging ai workloads. *arXiv preprint arXiv:2411.13239*, 2024.

[20] Q. Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap, 2023.

[21] Frank Joublin, Antonello Ceravola, Joerg Deigmoeller, Michael Gienger, Mathias Franzius, and Julian Eggert. A glimpse in chatgpt capabilities and its impact for ai research, 2023.

[22] Van Nguyen, Surya Nepal, Tingmin Wu, Xingliang Yuan, and Carsten Rudolph. Safe: Advancing large language models in leveraging semantic and syntactic relationships for software vulnerability detection, 2024.

[23] Abdul Rahim Saleh, Gihad Al-Nemera, Saif Al-Otaibi, Rashid Tahir, and Mohammed Alkhatib. Making honey files sweeter: Sentryfs – a service-oriented smart ransomware solution, 2021.

[24] Xin Yin, Chao Ni, and Shaohua Wang. Multitask-based evaluation of open-source llm on software vulnerability, 2024.

[25] Mete Keltek, Rong Hu, Mohammadreza Fani Sani, and Ziyue Li. Boosting cybersecurity vulnerability scanning based on llm-supported static application security testing, 2024.

[26] Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, Gérôme Bovet, and Gregorio Martínez Pérez. Transfer learning in pre-trained large language models for malware detection based on system calls, 2024.

[27] Steve Kommrusch. Artificial intelligence techniques for security vulnerability prevention, 2019.

[28] Kumar Shashwat, Francis Hahn, Xinming Ou, Dmitry Goldgof, Lawrence Hall, Jay Ligatti, S. Raj Rajgopalan, and Armin Ziaie Tabari. A preliminary study on using large language models in software pentesting, 2024.

[29] Zehang Deng, Ruoxi Sun, Minhui Xue, Sheng Wen, Seyit Camtepe, Surya Nepal, and Yang Xiang. Leakage-resilient and carbon-neutral aggregation featuring the federated ai-enabled critical infrastructure, 2024.

[30] Sifan Long, Fengxiao Tang, Yangfan Li, Tiao Tan, Zhengjie Jin, Ming Zhao, and Nei Kato. 6g comprehensive intelligence: network operations and optimization based on large language models, 2025.

[31] Bojian Jiang, Yi Jing, Tianhao Shen, Tong Wu, Qing Yang, and Deyi Xiong. Automated progressive red teaming, 2024.

[32] Xin Zhou, Sicong Cao, Xiaobing Sun, and David Lo. Large language model for vulnerability detection and repair: Literature review and the road ahead, 2024.

[33] Avishree Khare, Saikat Dutta, Ziyang Li, Alaia Solko-Breslin, Rajeev Alur, and Mayur Naik. Understanding the effectiveness of large language models in detecting security vulnerabilities, 2024.

[34] Fei Yang, Shuang Peng, Ning Sun, Fangyu Wang, Yuanyuan Wang, Fu Wu, Jiezhong Qiu, and Aimin Pan. Holmes: Towards distributed training across clusters with heterogeneous nic environment, 2024.

[35] Zeling Zhang, Dongqi Cai, Yiran Zhang, Mengwei Xu, Shangguang Wang, and Ao Zhou. Fedrdma: Communication-efficient cross-silo federated llm via chunked rdma transmission, 2024.

[36] William Agnew, Harry H. Jiang, Cella Sum, Maarten Sap, and Sauvik Das. Data defenses against large language models, 2024.

[37] Matthieu Lin, Jenny Sheng, Andrew Zhao, Shenzhi Wang, Yang Yue, Yiran Wu, Huan Liu, Jun Liu, Gao Huang, and Yong-Jin Liu. Llm-based optimization of compound ai systems: A survey, 2024.

[38] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly, 2024.

[39] Naweiluo Zhou, Florent Dufour, Vinzent Bode, Peter Zinterhof, Nicolay J Hammer, and Dieter Kranzlmüller. Towards confidential computing: A secure cloud architecture for big data analytics and ai, 2023.

[40] Ramyad Hadidi, Jiashen Cao, and Hyesoon Kim. Creating robust deep neural networks with coded distributed computing for iot systems, 2021.

[41] Christopher J. Vogl, Zachary Atkins, Alyson Fox, Agnieszka Miedlar, and Colin Ponce. Modifying the asynchronous jacobi method for data corruption resilience, 2024.

[42] Siyuan Shen, Langwen Huang, Marcin Chrapek, Timo Schneider, Jai Dayal, Manisha Gajbe, Robert Wisniewski, and Torsten Hoefler. Llamp: Assessing network latency tolerance of hpc applications with linear programming, 2024.

[43] Ruihong Wang, Jianguo Wang, Stratos Idreos, M. Tamer Özsu, and Walid G. Aref. The case for distributed shared-memory databases with rdma-enabled memory disaggregation, 2022.

[44] Michael Hilker. Distributed self management for distributed security systems, 2008.

[45] Jean Quilbeuf, Georgeta Igna, Denis Bytschkow, and Harald Ruess. Security policies for distributed systems, 2013.

[46] René Schwermer, Ruben Mayer, and Hans-Arno Jacobsen. Federated computing – survey on building blocks, extensions and systems, 2024.

[47] Donald Rozinak Beaver. Security, fault tolerance, and communication complexity in distributed systems, 2021.

[48] Kevin Fang and David Peng. Netdam: Network direct attached memory with programmable in-memory computing isa, 2021.

[49] Yulin Sun, Qingming Qu, Chenxingyu Zhao, Arvind Krishnamurthy, Hong Chang, and Ying Xiong. Tsor: Tcp socket over rdma container network for cloud native computing, 2023.

[50] Ali Mohammed, Aurelien Cavelan, and Florina M. Ciorba. rdlb: A novel approach for robust dynamic load balancing of scientific applications with parallel independent tasks, 2019.

[51] Ingo Müller, Renato Marroquín, Dimitrios Koutsoukos, Mike Wawrzoniak, Sabir Akhadov, and Gustavo Alonso. The collection virtual machine: An abstraction for multi-frontend multi-backend data analysis, 2020.

[52] Yuejun Guo, Constantinos Patsakis, Qiang Hu, Qiang Tang, and Fran Casino. Outside the comfort zone: Analysing llm capabilities in software vulnerability detection, 2024.

[53] Joris Borgdorff, Mariusz Mamonski, Bartosz Bosak, Krzysztof Kurowski, Mohamed Ben Belgacem, Bastien Chopard, Derek Groen, Peter V. Coveney, and Alfons G. Hoekstra. Distributed multiscale computing with muscle 2, the multiscale coupling library and environment, 2013.

[54] Hakan T. Otal and M. Abdullah Canbaz. Llm honeypot: Leveraging large language models as advanced interactive honeypot systems, 2024.

[55] Andrew A Mahyari. Harnessing the power of llms in source code vulnerability detection, 2024.

[56] Jing Cui, Yishi Xu, Zhewei Huang, Shuchang Zhou, Jianbin Jiao, and Junge Zhang. Recent advances in attack and defense approaches of large language models, 2024.

[57] Jamal Al-Karaki, Muhammad Al-Zafar Khan, and Marwan Omar. Exploring llms for malware detection: Review, framework design, and countermeasure approaches, 2024.

[58] Thomas Reinhold, Philipp Kuehn, Daniel Günther, Thomas Schneider, and Christian Reuter. Extrust: Reducing exploit stockpiles with a privacy-preserving depletion system for inter-state relationships, 2023.

[59] Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. Large language models for cyber security: A systematic literature review, 2024.

[60] Wenxiao Zhang, Xiangrui Kong, Conan Dewitt, Thomas Braunl, and Jin B. Hong. A study on prompt injection attack against llm-integrated mobile robotic systems, 2024.

[61] Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, Yunjun Gao, Yingcai Wu, and Shouling Ji. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents, 2024.

[62] Benjamin Steenhoek, Md Mahbubur Rahman, Monoshi Kumar Roy, Mirza Sanjida Alam, Hengbo Tong, Swarna Das, Earl T. Barr, and Wei Le. To err is machine: Vulnerability detection challenges llm reasoning, 2025.

[63] Arash Tavakkol, Aasheesh Kolli, Stanko Novakovic, Kaveh Razavi, Juan Gomez-Luna, Hasan Hassan, Claude Barthels, Yaohua Wang, Mohammad Sadrosadati, Saugata Ghose, Ankit Singla, Pratap Subrahmanyam, and Onur Mutlu. Enabling efficient rdma-based synchronous mirroring of persistent memory transactions, 2018.

[64] Sarthak Joshi and Sathish Vadhiyar. Partreper-mpi: Combining fault tolerance and performance for mpi applications, 2023.

[65] Ioannis Vardas, Manolis Ploumidis, and Manolis Marazakis. Improving the performance and resilience of mpi parallel jobs with topology and fault-aware process placement, 2021.

[66] Vincenzo De Florio. The dir net: A distributed system for detection, isolation, and recovery, 2015.

[67] Arpita Gang, Bingqing Xiang, and Waheed U. Bajwa. Distributed principal subspace analysis for partitioned big data: Algorithms, analysis, and implementation, 2021.

[68] Ioannis Argyroulis. Recent advancements in distributed system communications, 2021.

[69] Javad Zarrin, Rui L. Aguiar, and Joao Paulo Barraca. Decentralized resource discovery and management for future manycore systems, 2017.

[70] Salvatore Di Girolamo, Daniele De Sensi, Konstantin Taranov, Milos Malesevic, Maciej Besta, Timo Schneider, Severin Kistler, and Torsten Hoefler. Building blocks for network-accelerated distributed file systems, 2022.

[71] Konstantin Taranov, Benjamin Rothenberger, Daniele De Sensi, Adrian Perrig, and Torsten Hoefler. Nevermore: Exploiting rdma mistakes in nvme-of storage applications, 2022.

[72] Zhi Wang, Xiaoliang Wang, Zhuzhong Qian, Baoliu Ye, and Sanglu Lu. Rdmavisor: Toward deploying scalable and simple rdma as a service in datacenters, 2018.

[73] Xinhao Kong. *Towards Large-Scale RDMA Networks without Performance Anomalies*. PhD thesis, Duke University, 2024.

[74] Maomeng Su, Mingxing Zhang, Kang Chen, Yongwei Wu, and Guoliang Li. Rfp: A remote fetching paradigm for rdma-accelerated systems, 2015.

AI-generated, for reference only.

[75] Jiuxing Liu, Weihang Jiang, Pete Wyckoff, Dhabaleswar K. Panda, David Ashton, Darius Buntinas, William Gropp, and Brian Toonen. Design and implementation of mpich2 over infiniband with rdma support, 2003.

[76] Xinxin Liu, Yu Hua, and Rong Bai. Consistent rdma-friendly hashing on remote persistent memory, 2021.

[77] Michalis Vardoulakis, Giorgos Saloustros, Pilar González-Férez, and Angelos Bilas. Using rdma for efficient index replication in lsm key-value stores, 2021.

[78] Stanko Novakovic, Yizhou Shan, Aasheesh Kolli, Michael Cui, Yiying Zhang, Haggai Eran, Liran Liss, Michael Wei, Dan Tsafrir, and Marcos Aguilera. Storm: a fast transactional dataplane for remote data structures, 2019.

[79] Haoyu Huang and Shahram Ghandeharizadeh. Nova-lsm: A distributed, component-based lsm-tree key-value store, 2021.

[80] Dimitrios Koutsoukos, Ingo Müller, Renato Marroquín, Ana Klimovic, and Gustavo Alonso. Modularis: Modular relational analytics over heterogeneous distributed platforms, 2021.

[81] Marcin Copik, Konstantin Taranov, Alexandru Calotoiu, and Torsten Hoefler. rfaas: Enabling high performance serverless with rdma and leases, 2023.

[82] Radhika Mittal, Alexander Shpiner, Aurojit Panda, Eitan Zahavi, Arvind Krishnamurthy, Sylvia Ratnasamy, and Scott Shenker. Revisiting network support for rdma, 2018.

[83] Andre Luckow, Mark Santcroos, Ashley Zebrowski, and Shantenu Jha. Pilot-data: An abstraction for distributed data, 2013.

[84] Xinxin Liu, Yu Hua, Xuan Li, and Qifan Liu. Write-optimized and consistent rdma-based nvm systems, 2019.

[85] Krcore: a microsecond-scale rdma control plane for elastic computing.

[86] Xinhao Kong, Yibo Zhu, Huaping Zhou, Zhuo Jiang, Jianxi Ye, Chuanxiong Guo, and Danyang Zhuo. Collie: Finding performance anomalies in rdma subsystems, 2023.

[87] Huijun Shen, Guo Chen, Bojie Li, Xingtong Lin, Xingyu Zhang, Xizheng Wang, Amit Geron, Shamir Rabinovitch, Haifeng Lin, Han Ruan, Lijun Li, Jingbin Zhou, and Kun Tan. Np-rdma: Using commodity rdma without pinning memory, 2023.

[88] Jacob Nelson-Slivon, Lewis Tseng, and Roberto Palmieri. Technical report: Asymmetric mutual exclusion for rdma, 2022.

[89] Yiwen Zhang, Yue Tan, Brent Stephens, and Mosharaf Chowdhury. Rdma performance isolation with justitia, 2019.

[90] Benjamin Brock, Yuxin Chen, Jiakun Yan, John D. Owens, Aydın Buluç, and Katherine Yelick. Rdma vs. rpc for implementing distributed data structures, 2019.

[91] Tarannum Khan, Saeed Rashidi, Srinivas Sridharan, Pallavi Shurpali, Aditya Akella, and Tushar Krishna. Impact of roce congestion control policies on distributed training of dnns, 2022.

[92] Amanda Baran, Jacob Nelson-Slivon, Lewis Tseng, and Roberto Palmieri. Alock: Asymmetric lock primitive for rdma systems, 2024.

[93] Jiachen Xue, Muhammad Usama Chaudhry, Balajee Vamanan, T. N. Vijaykumar, and Mithuna Thottethodi. Dart: Divide and specialize for fast response to congestion in rdma-based datacenter networks, 2019.

[94] Manuel Bravo and Alexey Gotsman. Reconfigurable atomic transaction commit (extended version), 2019.

AI-generated, for reference only.

[95] Andrea Biagioni, Francesca Lo Cicero, Alessandro Lonardo, Pier Stanislao Paolucci, Mersia Perra, Davide Rossetti, Carlo Sidore, Francesco Simula, Laura Tosoratto, and Piero Vicini. The distributed network processor: a novel off-chip and on-chip interconnection network architecture, 2012.

[96] Waleed Reda, Marco Canini, Dejan Kostić, and Simon Peter. Rdma is turing complete, we just did not know it yet!, 2021.

[97] Guandong Lu, Runzhe Chen, Yakai Wang, Yangjie Zhou, Rui Zhang, Zheng Hu, Yanming Miao, Zhifang Cai, Li Li, Jingwen Leng, and Minyi Guo. Distsim: A performance model of large-scale hybrid distributed dnn training, 2023.

[98] Luanzheng Guo, Giorgis Georgakoudis, Konstantinos Parasyris, Ignacio Laguna, and Dong Li. Match: An mpi fault tolerance benchmark suite, 2021.

[99] Adam Lev-Libfeld, Alex Margolin, and Amnon Barak. Open-mpi over mosix: paralleled computing in a clustered world, 2019.

[100] Noah Lewis, Jean Luca Bez, and Suren Byna. I/o in machine learning applications on hpc systems: A 360-degree survey. *arXiv preprint arXiv:2404.10386*, 2024.

[101] Abhinav Vishnu, Charles Siegel, and Jeffrey Daily. Distributed tensorflow with mpi, 2017.

[102] Alfio Lazzaro. Enabling message passing interface containers on the lumi supercomputer, 2024.

[103] Konstantin Taranov, Fabian Fischer, and Torsten Hoefler. Efficient rdma communication protocols, 2022.

[104] Maksym Planeta, Jan Bierbaum, Michael Roitzsch, and Hermann Härtig. Cord: Converged rdma dataplane for high-performance clouds, 2023.

[105] Yufeng Xin, Ilya Baldin, Jeff Chase, and Kemafor Ogan. Leveraging semantic web technologies for managing resources in a multi-domain infrastructure-as-a-service environment, 2014.

[106] Adrian Bazaga and Michal Pitonak. Performance evaluation of an algorithm-based asynchronous checkpoint-restart fault tolerant application using mixed mpi/gpi-2, 2018.

[107] Ismail Hababeh. Data migration among different clouds, 2015.

[108] Bryan Ford. Icebergs in the clouds: the other risks of cloud computing, 2012.

[109] Abu Naser, Cong Wu, Mehran Sadeghi Lahijani, Mohsen Gavahi, Viet Tung Hoang, Zhi Wang, and Xin Yuan. Cryptmpi: A fast encrypted mpi library, 2020.

[110] Roberto Rocco, Gianluca Palermo, and Daniele Gregori. Fault awareness in the mpi 4.0 session model, 2023.

[111] Dmitry Mogilevsky and Sean Keller. Safempi - extending mpi for byzantine error detection on parallel clusters, 2005.

[112] Chansup Byun, Jeremy Kepner, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Peter Michaleas, Julie Mullen, Andrew Prout, Antonio Rosa, Siddharth Samsi, Charles Yee, and Albert Reuther. Large scale parallelization using file-based communications, 2019.

[113] R. Copstein and F. Dotti. Distributed file system for an edge-based environment, 2020.

[114] Chao Liu, Kin Gwn Lore, and Soumik Sarkar. Data-driven root-cause analysis for distributed system anomalies, 2018.

[115] Kazuichi Oe. Analysis of interference between rdma and local access on hybrid memory system, 2020.

[116] Swapneel Mehta, Prasanth Kothuri, and Daniel Lanza Garcia. A big data architecture for log data storage and analysis, 2018.

[117] David Brayford, Sofia Vallecorsa, Atanas Atanasov, Fabio Baruffa, and Walter Riviera. Deploying ai frameworks on secure hpc systems with containers, 2019.

[118] Jared Fernandez, Luca Wehrstedt, Leonid Shamis, Mostafa Elhoushi, Kalyan Saladi, Yonatan Bisk, Emma Strubell, and Jacob Kahn. Hardware scaling trends and diminishing returns in large-scale distributed training, 2024.

[119] Andrew Turner, Dominic Sloan-Murphy, Karthee Sivalingam, Harvey Richardson, and Julian Kunkel. Analysis of parallel i/o use on the uk national supercomputing service, archer using cray lassi and epcc safe, 2019.

[120] Shin-Yeh Tsai and Yiying Zhang. A double-edged sword: Security threats and opportunities in one-sided network communication, 2019.

[121] Dimitrios Michael Manias, Ali Chouman, and Abdallah Shami. Semantic routing for enhanced performance of llm-assisted intent-based 5g core network management and orchestration, 2024.

[122] Fangzhou Wu, Qingzhao Zhang, Ati Priya Bajaj, Tiffany Bao, Ning Zhang, Ruoyu "Fish" Wang, and Chaowei Xiao. Exploring the limits of chatgpt in software security applications, 2023.

[123] Michael Treaster. A survey of fault-tolerance and fault-recovery techniques in parallel systems, 2005.

[124] Rizwan A. Ashraf, Saurabh Hukerikar, and Christian Engelmann. Shrink or substitute: Handling process failures in hpc systems using in-situ recovery, 2018.

[125] Lukas Hübner, Demian Hespe, Peter Sanders, and Alexandros Stamatakis. Restore: In-memory replicated storage for rapid recovery in fault-tolerant algorithms, 2023.

[126] Vinay Amatya, Abhinav Vishnu, Charles Siegel, and Jeff Daily. What does fault tolerant deep learning need from mpi?, 2017.

[127] Lingzhe Zhang, Tong Jia, Mengxi Jia, Yifan Wu, Aiwei Liu, Yong Yang, Zhonghai Wu, Xuming Hu, Philip S. Yu, and Ying Li. A survey of aiops for failure management in the era of large language models, 2024.

[128] Aaditya Prakash. Measures of fault tolerance in distributed simulated annealing, 2012.

[129] Roberto Rocco, Davide Gadioli, and Gianluca Palermo. Legio: Fault resiliency for embarrassingly parallel mpi applications, 2021.

AI-generated, for reference only.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.

AI-generated, for reference only.