Middle East Journal of Applied Science & Technology (MEJAST)
Volume 8, Issue 4, Pages 118-128, October-December 2025

Intelligent Automation Framework for Multi-Vendor Network Infrastructure Using AI and Open-Source Tools

Mirza Golam Rasul[1], Mst. Sahela Rahman[2] & Md. Rabiul Islam[3]*

[1,2]*Department of Computer Science & Engineering, Pundra University of Science & Technology, Gokul, Bogura-5800, Bangladesh.* [3]*Department of Computer Science & Engineering, International Islami University of Science and Technology Bangladesh, Dhaka-1349, Bangladesh.*
*Corresponding Author (Md. Rabiul Islam) Email: mdrabiulislam521@gmail.com**

DOI: https://doi.org/10.46431/mejast.2025.8410

Article Received: 15 October 2025 *Article Accepted: 26 December 2025* *Article Published: 29 December 2025*

ABSTRACT

The growing variety of hardware, protocols, and vendor-specific configurations poses significant challenges to maintaining network infrastructure in multi-vendor environments. Conventional manual methods of network management are no longer adequate because of their restricted scalability, slow deployment times, and high mistake rates. The intelligent automation methodology presented in this research aims to standardize and accelerate network administration in diverse vendor ecosystems. Critical operations like VLAN provisioning, firmware upgrades, configuration backup, real-time monitoring, and self-healing are automated by the framework through the integration of open-source tools like Ansible, Netmiko, Nornir, Prometheus, and Grafana, as well as AI-driven fault detection. The system was evaluated using a realistic multi-vendor prototype that included devices from Juniper, Cisco, and MikroTik. According to experimental data, our proposed network minimizes human error through intelligent automation, greatly increases recovery rates (94–95%) from network outages, and cuts execution time by more than 90%. Scalability is supported by the system's tiered architecture, and fault tolerance is improved by including predictive maintenance. Additionally, by moving away from manual configuration and towards NetDevOps methods, which promote automation, scripting and proactive monitoring, the framework transforms the role of network experts.

Keywords: Network Automation; Multi-Vendor Infrastructure; Ansible; Self-Healing Network; NetDevOps; AI Monitoring; Intelligent Network Infrastructure; Prometheus Monitoring; Scalable Network; Open-Source Tools; Automatic Fault Detection.

1. Introduction

Heterogeneous devices from several vendors, including Cisco, Juniper, and MikroTik, are becoming more and more common in modern workplace networks. This multi-vendor approach adds a great deal of operational complexity, even while it provides flexibility and prevents vendor lock-in. Network engineers must maintain in-depth knowledge of each vendor's unique operating systems, command-line interfaces, and configuration standards [1]. Traditional manual network administration covering tasks like VLAN creation, firmware updates, and monitoring that is time-consuming, error-prone, and unscalable. As businesses expand, these constraints become crucial because they raise the possibility of configuration errors, performance deterioration, and delayed incident response [2]. The demand for reliable, scalable, and automated network management is further increased by the growth of cloud services, IoT devices, and dynamic workloads. Recent advancements in real-time monitoring frameworks (like Prometheus, Grafana) and open-source network automation (like Ansible, Netmiko) have enabled proactive management and AI-driven fault detection across distributed systems. But many of these systems lack the unified control required for completely autonomous, cross-platform network administration, and are either fragmented or vendor-specific [3].

The continued dominance of manual interventions results in ineffective operations, security flaws, and higher operating expenses. A uniform, intelligent automation system that can function flawlessly across devices from different vendors, allow for real-time monitoring and predictive maintenance, and reduce manual overhead that is desperately needed. This study introduces a flexible network which is an intelligent, vendor-agnostic automation framework that transforms traditional network administration. The framework, which was constructed with

popular open-source tools like Ansible, Netmiko, Nornir, Prometheus, and Grafana, automates common activities like dynamic IP allocation, firmware updates, VLAN formation, and backup and restoration procedures. In order to react proactively to network failures, it also incorporates AI-based fault detection and self-healing technologies. So, we set out our work with objectives:

a) To develop an automated framework that reduces manual effort and human error in network administration.

b) To enable open-source tools to provide vendor-agnostic automation for fault recovery, monitoring, and provisioning.

c) To develop and assess a self-healing system driven by AI that anticipates and constantly fixes network outages.

d) To illustrate how an intelligent automation model changes the job of network engineers and increases efficiency.

This is how the rest of the paper is structured. Section 2 summarizes the state-of-the-art solutions and current gaps in the relevant work in network automation and multi-vendor management. The suggested network framework's layered architecture and design principles are presented in Section 3, along with information on its vendor-agnostic automation methodology. The experimental setup, including the multi-vendor testbed, chosen tools, and implementation workflow, is explained in Section 4. A comprehensive synopsis of the main conclusions and contributions is provided in Section 5, which brings the article to a conclusion.

2. Related Works

This section covers recent research on multi-vendor interoperability solutions, network automation frameworks, and the use of AI in network issue detection. Numerous studies highlight the drawbacks of conventional network administration and suggest intelligent automation as a solution. Coito et al. [4] presented a modular and adaptive framework for intelligent automation, focusing on real-time data exchange and the use of intelligent agents for dynamic system control. Although the methodology is conceptually sound and cross-domain relevant, its applicability to multi-vendor infrastructures is limited due to the absence of empirical validation and detailed implementation in complex, real-world environments. A thorough review of network automation tools and methods is given by Muhammad and Munir [5], who place a special emphasis on open-source platforms such as Ansible and Python-based libraries. However, its utility in complex, heterogeneous environments is limited, as it lacks in-depth analysis of automation in multi-vendor or enterprise-scale settings. Kakade [6] studies intelligent automation techniques with the goal of improving IT operations' performance and agility, with a focus on AI-driven decision-making and workflow simplification. The benefits of automation across industries are viewed from a strategic level in this research. However, its usefulness for system designers is constrained by the lack of technical detail, practical evaluation, and relevance to network infrastructure environments.

Sohail [7] emphasizes an interdisciplinary approach that blends networking, artificial intelligence, and system engineering ideas in his early discussion of network management automation. Nevertheless, its earlier times, lack of empirical evidence, and inability to integrate modern tools limit its applicability to the quickly changing, multi-vendor, AI-driven network systems of today. A thorough technical analysis of network automation in Internet of Things settings highlighting significant obstacles as security threats, device heterogeneity, and

scalability. The paper's generalization is limited by its primary focus on IoT-specific limitations and lack of wider applicability to enterprise-grade multi-vendor network infrastructures. Mazin et al. [8] explain how to effectively manage and set up third-party network devices across vendors using Python-based automation with tools like Netmiko. The focus is on device-level engagement, with little attention paid to scalability, centralized orchestration, or AI integration, all of which are critical for overseeing massive multi-vendor systems. The implementation of multi-vendor 5G Open RAN systems is experimentally studied by Mehran et al. [9], who point out the operational challenges and possible advantages of attaining vendor interoperability. It provides scant advice for enterprise network automation frameworks that are more general than telecom-specific use cases. Industry-specific information regarding operational issues is provided in the paper [10] about the organizational challenges associated with managing software ecosystems that involve several providers. It does not, however, offer specific technical or automation solutions and adopts a business-oriented approach, which limits its direct applicability to network infrastructure automation.

However, the majority of research either concentrates on domain-specific settings like IoT and 5G, single-vendor environments, or theoretical approaches; there remains a lack of a unified, scalable solution for real-world, multi-vendor network infrastructures. Furthermore, although automation and AI integration are widely recognized to be important, few studies successfully integrate these components into a cohesive, self-healing, end-to-end framework. This disparity highlights the need for a comprehensive system such as our proposed framework, which delivers centralized management, intelligent fault recovery, and cross-vendor automation tailored for dynamic, enterprise-scale networks.

3. Research Methodology

Systematic literature review and experimental implementation are the two main research methods used in this study on the automation of network configuration and management from diverse vendor ecosystems.

3.1. Systematic Literature Review

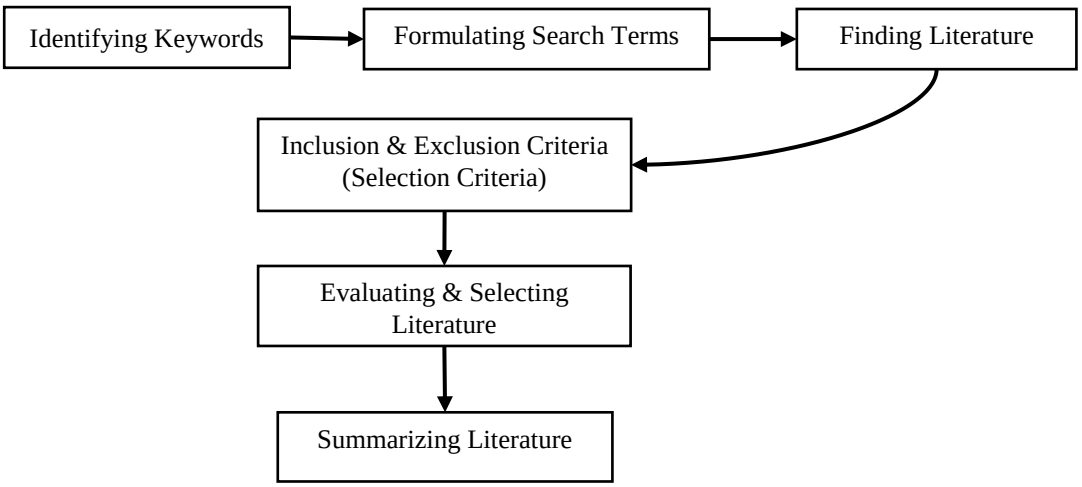

Figure 1. Systematic literature review process

A systematic literature review (SLR) was conducted to identify challenges, limitations, and research gaps in multi-vendor network automation and intelligent network management, with particular emphasis on the drawbacks

of manual network infrastructures. The review process is depicted in the block diagram in Figure 1, which is followed by keyword identification, search phrase formulation, literature discovery, and criterion selection for particular literature, literature evaluation, and summarization of the chosen literature. Relevant research encompassing publications from 2010 to 2025 were gathered from several public online databases, illustrated in Table 1. Network Automation, Multi-Vendor Networks, AI-based Network Management, Self-Healing Networks, Ansible Automation, and NetDevOps were all included in a keyword-based search method.

Table 1. Systematic Literature Review Protocol

Component	**Description**
Databases	IEEE Xplore; Scopus; ScienceDirect (IFAC-PapersOnLine); SpringerLink; Google Scholar
Time Period	Mainly 2010–2025, but exception was some 90s paper for base knowledge
Search Strategy	Keyword based
Inclusion Criteria	Peer-reviewed studies; automation-focused; experimental or architectural
Exclusion Criteria	Blogs; white papers; vendor materials; irrelevant or outdated studies
Review Outcome	Identification of limitations in manual multi-vendor network management

A selected collection of articles about network automation, multi-vendor interoperability, and AI-driven monitoring was qualitatively examined after duplicates were eliminated and predetermined inclusion and exclusion criteria were applied. The suggested framework's architectural design, tool choice, and AI integration approach were all directly influenced by the review's findings.

3.2. Experimental Setup

The implementation of network infrastructure, system design, and tool selection make up the experimental setup.

Table 2. Required tools and technologies

Hardware Requirements	**Device**	**Vendor**	**Purpose**
	MikroTik Routers	MikroTik	Dynamic routing, Firewall automation
	Cisco Switches & Routers	Cisco	Layer 2/3 network connectivity
	Juniper Devices	Juniper	Policy-based security and automation
	Cloud-based Servers	Multi-vendor	Hosting automation and AI-driven monitoring
Software & Automated Framework	**Tools/ Software platforms**	**Purpose**	
	Ansible	Automated network provisioning and updates	
	Netmiko	Python-based CLI automation	
	Nornir	Large-scale network automation	
	EVE-NG	Network simulation and testing	
	Virtual box	Virtualized network infrastructure testing	
	Grafana & Prometheus	Real-time network monitoring and analytics	

Table 2 lists the hardware, software, and automated frameworks needed for an intelligent automated network infrastructure for a multi-vendor environment. The software tools and platforms were used to create seamless automation, and the previously mentioned physical network devices were used for testing and validation. Figure 2

illustrates all of the implementation processes that comprised our suggested infrastructure. In the first step, VLANs were dynamically created and configured across all network devices using Python and Ansible scripts. The second phase involved setting up configuration backups and implementing automated firmware (router) updates using Ansible playbooks.

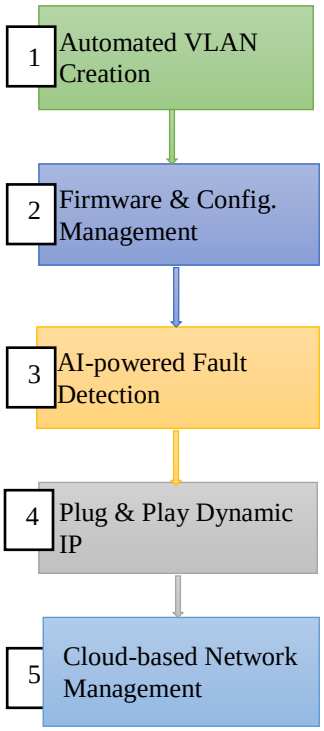

Figure 2. Implementation Steps of the Proposed Infrastructure

An AI-based detection system that dynamically modified network configurations in response to failure occurrences was then put into place to examine the network logs. The integration of new devices is then done using an IP allocation method based on the dynamic host control protocol. Ultimately, a system for automatic synchronization and network management was created on the cloud.

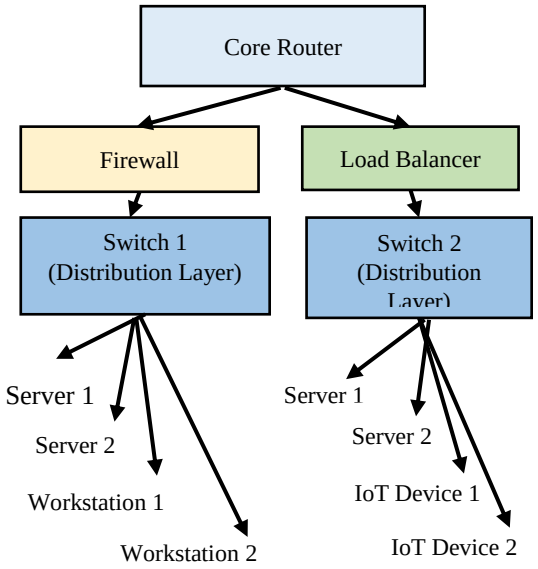

Figure 3. A prototype of an extensible automated network infrastructure

3.2.1. Multi-Vendor Testbed Description

The experimental testbed, which consists of MikroTik (2 routers, 4 switches), Cisco (1 router, 4 switches), and Juniper (1 router, 4 switches), was created to mimic a genuine multi-vendor environment (Table 3).

Table 3. Testbed Configuration

Component	**Details**
Devices per Vendor	• MikroTik: 2 routers, 4 switches • Cisco: 1 router, 4 switches • Juniper: 1 router, 4 switches
Network Size & Topology	• Three-tier architecture (core–distribution–access) • 4 subnets • 16 endpoint devices (servers, workstations, IoT nodes) • Cloud-hosted monitoring & automation servers
Failure Injection	• Router shutdowns (hardware failure) • VLAN misconfigurations (human error) • Switch port disabling (link failure) • Repeated under varying load conditions
Evaluation Metrics	• Recovery time (MTTD, MTTR) • Success rate of automated corrective actions • Effectiveness of orchestration and self-healing

Four subnets, sixteen endpoint devices (servers, workstations, and Internet of Things nodes), and cloud-hosted monitoring/automation servers comprised the three-tier core–distribution–access architecture. Controlled failures, such as switch port disabling, VLAN misconfigurations, and router shutdowns, were routinely injected under different loads to verify resilience. In order to ensure reproducibility and quantitative validation of fault tolerance and self-healing, recovery time (MTTD, MTTR), automated success rate, and orchestration efficacy were examined.

3.3. AI-Based Fault Detection and Predictive Maintenance

For proactive monitoring and automated self-healing in multi-vendor network settings, the suggested framework incorporates an AI-based fault detection and predictive maintenance module. Using lightweight machine learning models: Logistic Regression and Isolation Forest trained in Python on both historical and current telemetry data, a hybrid supervised and semi-supervised anomaly detection strategy was constructed, as described in Algorithm 1. Key characteristics of the training data included CPU/memory utilization, interface status, packet drop rate, log event frequency, and temperature threshold breaches. The training data included both normal operation and injected fault scenarios.

Algorithm 1. AI-Based Fault Detection and Automated Recovery

Input: Live network telemetry T
Output: Automated fault recovery actions

1: Initialize trained ML models (Logistic Regression, Isolation Forest)
2: Collect telemetry metrics from network devices
3: Extract feature vector F from T
4: Compute anomaly score A using ML models
5: if A > predefined threshold θ then
6: Identify fault type and affected components
7: Trigger corresponding Ansible playbook
8: Execute recovery action (reroute, restore, restart, isolate)
9: end if
10: Log event and recovery outcome

Telemetry feeds are continuously processed to calculate anomaly scores during runtime. The system automatically initiates Ansible playbooks for corrective measures, such as traffic rerouting, configuration restoration, interface restart, or isolation of problematic segments, when the score surpasses a certain level. Standard performance criteria, including fault detection accuracy, recovery success rate, Mean Time to Detect (MTTD), and Mean Time to Recover (MTTR), were used to assess the efficacy of an AI-assisted method. These metrics are compiled in Table 4.

Table 4. Performance Evaluation Metrics for AI-Based Fault Detection

Metric	Description
Fault Detection Accuracy (%)	• Ratio of correctly detected faults to total faults. • Measures detection reliability
Recovery Success Rate (%)	• Percentage of faults successfully resolved automatically • Evaluates self-healing effectiveness
Mean Time to Detect (MTTD)	• Average time to identify a fault • Assesses responsiveness
Mean Time to Recover (MTTR)	• Average time to restore normal operation • Measures recovery efficiency

Together, Algorithm 1 and Table 4 validate the predictive maintenance capability and its effect on lowering manual intervention and recovery time by showcasing the explicit learning approach, training data, feature set, and evaluation criteria.

4. Result and Discussion

We deployed an intelligent automation framework that is scalable and capable of managing heterogeneous, multi-vendor network settings through a thorough experimental setup that included a variety of tools and

technologies. According to Figure 3, the system architecture is divided into four modular layers: the distribution layer, the access layer, the cloud & monitoring layer, and the core network layer. Because of its tiered architecture, the system may grow horizontally while preserving compatibility among devices made by Cisco, Juniper, and MikroTik, among other vendors.

4.1. Performance Evaluation

We compared our automated workflows with conventional manual methods in order to evaluate operational efficiency. Table 5 presents the findings. The findings demonstrate that automation minimizes manual error rates and drastically cuts down on execution time for all crucial processes. Using our system, VLAN provisioning which previously took more than an hour to accomplish manually was finished in only five minutes with reliable, error-free results.

Table 5. Efficiency Comparison

Task	Manual Execution Time	Automated Execution Time	Efficiency Gain
VLAN Creation (100 VLANs)	1 hour	5 minutes	92% faster
Router Firmware Update (10 devices)	2 hours	10 minutes	91% faster
Backup & Restore	30 minutes	2 minutes	93% faster

4.2. AI-Based Self-Healing and Recovery

To assess the AI-based self-healing capabilities of the system, we replicated typical network failures. Table 6 demonstrates that the system's recovery success rates were high for all sorts of failure.

Table 6. Recovery Rate Evaluation

Type of Failure	Manual Recovery Time	Automated Recovery Time	Success Rate
Router Failure	30 minutes	2 minutes	94%
VLAN Misconfiguration	20 minutes	1 minute	95%
Switch Port Failure	25 minutes	3 minutes	92%

By successfully lowering mean time to recovery (MTTR) from tens of minutes to a few minutes, the automated recovery system enhanced service continuity and network availability. By guaranteeing consistent configurations and enabling proactive backups, the cloud-based synchronization engine significantly expedited recovery.

4.3. Key Comparison with Existing Work

We performed a comparison with important current research initiatives in order to verify the originality and significance of our suggested intelligent automation framework. Although previous studies have examined many facets of network automation, the majority of these studies are still theoretical, vendor-specific, or have limited practical relevance. Fung [1] and Sohail [7], two early foundational efforts, provided basic models but lacked scalability and implementation. Despite proposing modular and AI-enhanced solutions, studies like Coito et al. [4]

and Kakade [6] were unable to demonstrate vendor compatibility or real-world deployment. Mazin et al. [3] and Muhammad & Munir [5] focused on open-source tools such as Ansible or Netmiko, although they were limited to simple scripting or single-device automation without performance measurement or self-healing. Wider application was limited by Mehran et al.'s [9] concentration on 5G Open RAN scenarios. On the other hand, our suggested architecture integrates AI-driven self-healing with technologies like Ansible, Netmiko, Nornir, Prometheus, and Grafana to provide a consistent, scalable, and vendor-agnostic automation solution. It achieves recovery success rates above 94% and reduces execution time by over 90%, outperforming manual settings.

This solution is a strong, future-ready addition to intelligent network automation since, as far as we know, it is the first end-to-end, experimentally validated solution that automates VLAN provisioning, firmware updates, configuration backup, monitoring, and predictive maintenance across Cisco, Juniper, and MikroTik devices.

5. Conclusion

Our framework efficiently automates crucial network tasks like VLAN provisioning, firmware updates, backup and restoration, real-time monitoring, and self-healing operations by combining popular open-source tools like Ansible, Netmiko, Nornir, Prometheus, and Grafana with AI-based fault detection mechanisms. The effectiveness and dependability of the system were confirmed by the experimental findings. In contrast to conventional manual techniques, the suggested solution showed excellent recovery success rates of up to 95% for a variety of network problems and decreased execution time by over 90% for critical operations. These enhancements show how the framework may reduce human error, improve fault tolerance, and guarantee service continuity in real-time business settings. AI-driven analytics allow for proactive incident response and predictive maintenance, while the framework's layered, modular architecture guarantees scalability and cross-vendor compatibility. Furthermore, the technology converts network engineers from manual operators to automation architects and proactive maintainers, signaling a dramatic move towards NetDevOps techniques.

6. Future Recommendations

Future improvements shall be concentrated on the suggestions below:

1. Voice-activated system configuration ensuring through Natural Language Processing (NLP).
2. Blockchain-based security audits to verify the transparency, integrity, and trust.
3. Use of Reinforcement Learning for intelligent and dynamic network maintenance.
4. Can introduction of regenerative self-healing capabilities to improve system resilience.
5. The framework supports scalability and adapts to expand the enterprise requirements.

Declarations

Source of Funding

The authors declare that no specific grant from any funding agency in the public, commercial, or not-for-profit sector was received for this study.

Competing Interests Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Consent for publication

The authors declare that they consented to the publication of this study.

Availability of data and materials

The datasets generated and analysed during the current study are available from the corresponding author upon reasonable request.

Institutional Review Board Statement

Not applicable for this study.

Informed Consent

Not applicable for this study.

Acknowledgements

The authors appreciate the welcoming atmosphere at work. Authors also want to thank all of their co-workers and supporters.

References

[1] Fung, H.S. (1989). Defining intelligent network services in a multivendor distributed environment. In 1989 IEEE Global Telecommunications Conference and Exhibition: Communications Technology for the 1990s and Beyond, Pages 132–136, IEEE.

[2] Choi, B. (2021). Introduction to python network automation. Introduction to Python Network Automation: The First Journey, Pages 1–21, Apress.

[3] Mazin, A.A., Abidin, H.Z., Mazalan, L., & Mazin, A.M. (2023). Network automation using python programming to interact with multiple third-party network devices. In 2023 10th International Conference on Information Technology, Computer, and Electrical Engineering, Pages 59–64, IEEE.

[4] Coito, T., Viegas, J.L., Martins, M.S., Cunha, M.M., Figueiredo, J., Vieira, S.M., & Sousa, J.M. (2019). A novel framework for intelligent automation. IFAC-PapersOnLine, 52: 1825–1830.

[5] Muhammad, T., & Munir, M. (2023). Network automation. European Journal of Technology, 7: 23–42.

[6] Kakade, A. (2023). Optimizing performance and agility through intelligent automation strategies. International Numerical Journal of Machine Learning and Robots, 7: 1–10.

[7] Sohail, S. (2010). Automation of network management with multidisciplinary concepts. International Journal of Computer Technology and Applications, 11.

[8] Mazin, A.A., Abidin, H.Z., Mazalan, L., & Mazin, A.M. (2023). Network automation using python programming to interact with multiple third-party network devices. In 2023 10th International Conference on Information Technology, Computer, and Electrical Engineering, Pages 59–64, IEEE.

[9] Mehran, F., Turyagyenda, C., & Kaleshi, D. (2024). Experimental evaluation of multi-vendor 5G open RANs: Promises, challenges, and lessons learned. IEEE Access.

[10] Rani, A., Mishra, D., & Omerovic, A. (2022). Multi-vendor software ecosystem: Challenges from company perspective. World Conference on Information Systems and Technologies, Pages 382–393, Springer International Publishing.