

Article

Network Orchestration Framework Design Using AI-Driven Automation and Cybersecurity

Tasneem Annahti ¹, Albandari Alsumayt ^{1,*} and Majid Alshammari ²

¹ Saudi Aramco Cybersecurity Chair, Networks and Communications Department, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia; 2250500216@iau.edu.sa

² Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia; m.alshammari@tu.edu.sa

* Correspondence: afaalsumayt@iau.edu.sa

Abstract

This paper addresses human error in network orchestration systems and the high cost and resource requirements of integrating artificial intelligence (AI) for network orchestration. It proposes a framework for implementing an AI decision-maker and automation. Data are fed into the AI decision-maker to trigger designated automation robots' tasks or notify IT specialists to gradually implement automated robots, ensuring efficient resource use, reducing costs, and enhancing productivity. We evaluated the proposed method in a simulation with genuinely uncertain outcomes, across 20 independent runs: the AI decision-maker reached 78.3% accuracy against an estimated 79.1% achievable ceiling, and the proposed framework reduced operational cost by $61.4 \pm 0.7\%$ relative to fully manual operation—the best of six operating policies in the training environment—while an explicit sensitivity guard, rather than the learned model, accounts for the absence of security incidents; under distribution shift, the framework retains $43.2 \pm 0.8\%$ savings, second only to a hand-tuned rule-based router that requires environment-specific threshold calibration. However, the proposed method requires an IT specialist to implement it properly, and the AI model's accuracy depends on the amount of input data. In the end, we recommend that future work conduct a study focused on AI decision-makers, test the proposed method on real-world companies, and implement AI decision-makers across various departments to cover a broader range of the company's systems.

Keywords: network orchestration; gradual integration; task automation; AI decision-maker; cybersecurity; risk assessment



Academic Editors: Francesco Buccafurri and Sara Lazzaro

Received: 8 June 2026

Revised: 19 July 2026

Accepted: 21 July 2026

Published: 27 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Network orchestration is the process by which the network controller performs tasks within the network, such as setting up devices. It's basically directing the machines into performing tasks. We could compare it to an orchestra, which consists of instrumentalists who play together and is led by the maestro, sometimes referred to as a conductor, who leads the rest of the group for the symphony. The maestro is the network controller that orchestrates network tasks, and, to keep it secure, the symphony is the security. It is the network controller's goal to achieve during configurations. Network orchestration is crucial in the IT industry, as it delivers reliable, fast service, applies configurations across multiple network domains, and helps companies adapt to new technologies more quickly. The orchestration makes it easier to control, monitor, and configure from a single point. It

reduces time and costs associated with manual network configuration. Moreover, network orchestration is important for organizational performance, ensuring direct, smooth communication across the network regardless of the size of the entities involved, and enabling them to allocate work, align their objectives and strategies, and achieve a common goal [1].

Orchestration tasks require heavy, time-intensive tasks to be done in a short time. When humans handle these tasks fully, errors can occur, such as misconfigured settings that could disrupt network operations. So it is important to integrate various technologies to assist with human tasks and minimize human involvement, requiring it only for the important parts. Today, automation is expanding across industries to address issues such as human error, increased costs, and time [2,3]. Using automation in the network security industry has proven its success over time. In particular, in network orchestration, automation has been implemented to address issues such as inconsistent policy configurations, errors caused by manual configuration, and the complexity of integrating various security functions [4]. Most errors have a human factor in common; reducing the human factor in network orchestration would help reduce errors and ensure network security [5,6]. Organizations have been combining artificial intelligence (AI) with automation to increase accuracy and to have a more intelligent decision-maker who makes decisions based on an analysis of historical data, so using AI decision-making would improve performance [7].

This paper has contributed by proposing a framework to overcome network orchestration challenges, especially challenges related to human-factor errors. The paper proposes a framework protocol that uses intelligent automation and artificial intelligence (AI) decision-making. It outputs a decision based on the data provided to the orchestration network, triggering the automation process and allowing the robot to handle the orchestration task. If the AI's output does not include this orchestration task in the automation task list, it triggers the IT employee. The employee can either perform this task manually or create an automated task for it. This is a crucial step in saving time and cost. The following points summarize the paper contribution:

- The paper proposes a framework to overcome network orchestration challenges, particularly those related to human-factor errors. It uses intelligent automation and artificial intelligence (AI) decision-making.
- The paper proposes a framework that uses gradual implementation to reduce cost and time.
- The paper proposes a framework that uses a human-in-the-loop approach.

The paper is organized as follows. In Section 2, we discuss the previous related works of literature on implementing AI and automation in network orchestration; in Section 3, we present an in-depth discussion of the proposed methodology and the rationale behind each design choice; in Section 4, we describe the experiment and simulation conducted to validate the proposed framework and report its results in terms of decision-maker accuracy against an estimated achievable ceiling, cost comparisons against four alternative operating policies including a rule-based router, robustness under distribution shift, and a sensitivity analysis of the cost model; and finally, in Section 5, we discuss the limitations of the proposed method, outline directions for future work, and summarize the findings of the paper.

2. Related Works

In this section, we will review related work by other researchers and the methodologies they used to deploy automation in network orchestration. Automation has been implemented across various industries to address different issues, but the common goal they share is to reduce costs, time, and human error. When tasks rapidly increase in both volume and scope, they tend to consume time, cost, and resources; therefore, companies

seek solutions to address these issues and save on budget, and automation is the answer to their needs. Along with human errors, the cost of human errors sometimes reflects badly on the company and might have major consequences; therefore, companies tend to deploy automation to decrease the human factor's intervention to avoid costly mistakes and make use of the employee's time and energy in some other tasks that cannot be automated [8].

The researchers Sajid et al. (2021) [9] have deployed automation to address cyber deception and security orchestration to defend against rising cyber threats and vulnerabilities, as attack environments evolve rapidly. It has been observed that the traditional security model has failed to fulfill its role in detecting and mitigating advanced, sophisticated threats. To address this matter, their research has proposed a system for cyber deception orchestration and automation that generates and customizes a deception playbook based on the attacker's behavior, ensuring the deception structure aligns with the identified threat. Then the orchestration automation begins creating a deceptive environment that tricks the malware into adapting to and engaging with it. However, this methodology requires an expert to set up and mitigate the system. It also suffers from a generalizability limitation, as SODA focuses solely on malware deception and cannot be guaranteed to succeed in other security challenges.

Also, Belgaum et al. (2021) [10] have implemented various techniques with artificial intelligence to enhance performance and reduce errors and human involvement, such as implementing machine learning (ML) to predict security threats that could cause the system to fail, and the AI-based management mitigates the detected or predicted threats and generates new policies to assure continuity of learning. The advantages of implementing the method include reduced human intervention, improved system response and performance, and scalability for managing large-scale network environments.

In the 5G network environment, Manda (2022) and Moubayed et al. (2022) [11,12] have worked on implementing AI-driven techniques to reduce the human environments by letting the AI make all the decisions, and that has saved a lot of time and energy along with enhancing the performance, but the method implemented suffers from maintaining data privacy, and like the previously mentioned research, it is also vulnerable to security threats and demand and expensive resources.

Moreover, Brighenti et al. (2022) [13] have implemented automated firewall configuration to mitigate human error, which can make the network vulnerable to unauthorized access. The automation has ensured a consistent, secure configuration, adaptability without human intervention, and the generation of firewall rules derived from the network's requirements. By implementing the proposed method, they reduced the risk of inconsistent configuration and the time required to configure firewalls, making updates more efficient and ensuring that security policies were followed. Lastly, this method is scalable to large networks of large companies where manual configuration is consuming. Although this method might require an expert to set up the automation system, in large networks it can sometimes affect speed and performance due to constant firewall configuration adjustments. They tested this method on a virtual network, and it has proven successful. However, it might not work as well in a traditional network, and its success relies on the accuracy of the models used; the models must be complete and flawless to avoid any security loopholes.

Issues have been addressed in the cloud environment, where they identified challenges in balancing task scheduling and performance robustness. To address this issue, Zhong et al. (2022) [14] have implemented machine learning-based orchestration to enhance performance by classifying ML techniques into supervised and unsupervised categories and adapting to new orchestration tasks. The ML, based on the given data, optimizes scheduling and improves resource allocation, which helps reduce human intervention and allows automated decision-making to take over. Implementing this solution has come at

the cost of increased demand for specialized resources, and the integration complexity requires expertise; lastly, the performance of machine learning techniques depends on the quality of the data used for model training.

Bringhenti et al. (2023) [15] have also addressed the problem of manual configuration, which leads to human errors and time-consuming processes. As companies have grown larger, the network size has increased, making manual configuration more complex and more vulnerable to security threats. They proposed automation to overcome these challenges. Their research has reviewed existing methodologies for enhancing the efficiency and performance of network configurations and security and compared the limitations of each. The methodologies their research reviewed to ensure a robust protection structure included optimization algorithms, AI-driven automation, formal verification, and configuration management automation. The optimization algorithm in network automation configuration has reduced setup time and costs by eliminating the need for expertise and automatically analyzing configurations to allocate optimal settings, thereby reducing human error. It maintains a balance between security requirements and resource constraints; it ensures only the necessary resources are implemented, and it is adaptable to factors such as network threats and changing conditions. However, this method fails to scale effectively, as large networks increase computational costs, limiting its applicability.

The orchestration of traditional cloud-based systems is hampered by the need to manage edge devices due to bandwidth constraints. The nature of dynamic edge environments poses scalability challenges. Therefore, the paper of Anuraj (2023) [16] introduced a solution to address these issues and enhance the edge computing swarms. They integrated agent-based system, where edge devices are presented as automation agents, depends on local decision-making to overcome the delay issue; decentralizing the orchestration process allowed the system to continue from where it left off in case of an error causing the agent to fail, where each automation agent is working separately to exchange data, utilize resources, and that have increased the performance and scalability, as the system can add as many edge devices as wanted, and that does not affect or degrade the performance. However, implementing this solution is challenging, especially in maintaining consistency across edge agents; the decentralization introduces new security threats and vulnerabilities that must be monitored and addressed.

Furthermore, Bringhenti et al. (2024) [17] have implemented AI-driven automation to address critical human errors in manual configuration, especially in large networks. They used automation, combined with formal verification and optimization, to define the network layout and the allocation scheme, as well as the communication protection system (CPS) protection rules. They presented the solution using the Maximum Satisfiability Modulo Theories (MaxSMT) problem, which finds the most suitable solution. Their proposed solution has improved the network's efficiency, performance, scalability, and security.

Recently, artificial intelligence (AI) has been implemented across many fields to address a range of issues. Kaul (2024) [18] has explored the use of AI and machine learning (ML) to address network orchestration issues; in their research, they have used Kubernetes management, which automates tasks across clusters. To address the limitations of Kubernetes, such as its constant need for human intervention and its demands for. They implemented ML to predict events by analyzing resource usage and using regression analysis, and deep learning to detect suspicious behaviors in the system. These improvements have enhanced the system's performance and efficiency, and the AI not only detects suspicious behaviors but also self-heals, making it more flexible and adaptable in stressful environments. Although implementing AI requires significant resources, it can be difficult to interpret, and AI is also vulnerable to many attacks; therefore, it requires constant training, which is expensive to maintain.

In another study, Nadaf et al. (2025) [19] used AI-driven automation to address the complex infrastructure of 5G and beyond, where manual resource configuration and management of dynamic resource demand across a large number of devices became difficult. They implemented machine learning (ML) to analyze and predict network behavior, leveraging historical data and real-time insights. This has enhanced decision-making to support proactive resource management and automatically optimizes performance by configuring resources. This research has demonstrated its success in a large company, where machine learning requires substantial resources to achieve high accuracy; however, deploying the same methodology in a smaller company might not yield the same results.

Shamsuzzoha and Pelkonen (2025) [2] have proposed using robotic process automation (RPA) to address the issue of employees having workloads of repetitive, exhaustive tasks, which, when overloaded, lead humans to make errors, take longer, be less creative, or take on more tasks. Having the RPA model take over would be an adequate solution for both employees and managers. The RPA would handle tasks more quickly than humans, with higher quality and fewer errors. RPA models would be a good assist to employees in getting their tasks done; instead of having to do the entire task, they can just take a quick overview of the robot's work. This model would ensure the work is done efficiently.

Another solution for reducing human error is to integrate artificial intelligence (AI). Kothandapani (2025) [20] has integrated an AI-driven model and a large language model (LLM) in financial tasks to reduce errors and increase speed and efficiency. In finance, they mostly handle large volumes of data and tasks that must be completed as quickly as possible while ensuring accuracy. However, this rush causes employees to make mistakes that must be addressed later, which can be costly and waste time and resources. Therefore, having an AI-driven model and an LLM to assist with heavy tasks would help reduce errors, increase completion speed, and eliminate resource waste. However, integrating AI-driven models has introduced a bias limitation; their training data has led to algorithmic bias. Having a biased model is crucial; some AI models can drift toward bias during training, and this can be addressed by making adjustments or reconfigurations, or by adjusting the dataset. It all depends on the dataset used for training. The bias issue should be addressed to improve the AI model's reliability.

With technology that makes human lives easier, people tend to over-rely on it and disregard issues such as bias. Romeo and Conti (2026) [21] have focused their study on the importance of managing automation bias to avoid critical errors and poor performance. They proposed integrating cognitive forcing functions (CFFs) to prompt users to validate and review the generated output. This ensures human involvement in tasks such as reviewing output and decision-making. This increases integrity and reduces over-reliance on the automation model.

Tayal (2026) [22] has reviewed the intelligent frameworks in industry 4.0, and proposed an automation model along with the AI decision making, to change the industry experience for the employees, where now the proposed framework can analyze and predict strategies before starting. This method works as a hybrid model, it is still requires human involvement to oversee the output and results, this method succeeded in reducing the work overload on employees and improved the transaction efficiency in completion, however, the limitation of this method is that in order to be integrated and configured they requires a technical expert employee to set it up.

Automation has been used in various fields for a range of goals and objectives. Varnisha et al. (2026) [23] have deployed it in the IoT field. For smart homes to enhance accessibility and user experience. They proposed AI-driven, voice-controlled automation that receives commands via speech. However, it does not perform well in the presence of background noise. Automation roles differ across fields, but they all share the same goal:

reducing errors and increasing efficiency and speed. Also, Sidhu and Sharma (2026) [24] have focused on enhancing the environments of 5G and IoT, increasing speed and reducing manual configuration. An AI-driven automation model can be integrated to handle repetitive tasks; however, the issue of vulnerability to cybersecurity attacks remains.

Badhel et al. (2026) [25] proposed a solution for smart homes: an AI decision-maker and automation to save energy. Their proposed model works by having the AI decide when to turn the device on or off and which device to turn on or off, and then sending the request to the automation model to handle the device. This method shows potential to reduce costs and conserve energy. However, it requires a strong internet connection to communicate with smart devices.

Small and medium-sized enterprises (SMEs) handle a lot of manual file handling, including Excel and PDF files. Therefore, Lihansisco and Wiputra (2026) [26] have proposed an RPA and AI model to automate report generation, in which the RPA robot prepares the files, and the AI handles text proofreading and content. This method would reduce the time and energy employees spend completing tasks, giving them more free time and energy to focus on tasks that require human judgment. However, the limitation is that the model does not eliminate errors; it only reduces them, so it would still require human review of the outputs.

Automation has been integrated to get the tasks done efficiently and reduce human error and human involvement to save employees' time and energy on the critical tasks; automation has also been integrated to help focus on employees' well-being by monitoring their biosignals, and ensuring that they are in good health physically and mentally. Valtonen et al. (2026). [27] deployed automation, AI, and machine learning (ML) to measure employees' eye movement and heart rate posture to monitor their health physically and mentally and to assist them in case they are in a stressful environment or if they are feeling stressed or overwhelmed. To get the job done, the ML model detects their biosignals and checks if they are not normal; then it sends a request to the automation to start where it assesses the employees, sending them structures, loading for them information, holding off their tasks or meetings, and assessing them into distressing to calm them to help them get the tasks done in a good state of mind. This helps us understand that the automation is not trying to replace employees or just free up employees' time to start another task, but also to help them maintain a healthy lifestyle. However, this method's limitation is that the employee's private data is exposed to the ML model; the security level must be high to maintain privacy.

Mgbemele (2026) [28] has evaluated the use of automation and ML against the MITRE ATT&CK framework. They proposed an AI-driven detection solution that keeps pace with global security threats and maintains the current system's alert status for potential attacks. The method has succeeded and keeps pace with new worldwide attacks, alerting the system when they occur, thereby increasing security. However, it has a limitation in that the alerts that the AI provides do not give away a lot of information, so the human employee who reviews and traces back the alert has to look thoroughly to find out what the alert was for, as it does not give away a lot of information about it.

Nguyen et al. (2026) [29] have focused their study on the security, orchestration automation, and response (SOAR) technologies. They wanted an efficient solution that has a fast response and still uses a human-in-the-loop framework where human involvement is still required in the transaction, but it is with minimal steps. Therefore, they propose a method of automated decision-making deep reinforcement learning DRL, this framework generates an analysis, report, and strategies and responses for the system security. They also have an explainable AI chat4XAI where the employees and users can interact with the

AI as a chatbot, and get more details about the output generated, however, the limitation of this matter is that it requires expertise.

To eliminate manual order handling, Cyril (2025) [30] integrated RPA, freeing employees to focus on other tasks and increasing task completion speed with minimal errors. However, this method has the limitation that it can only handle the prespecified rules; if any new issue arises, the robot cannot handle it.

Pokorny (2026) [31] has reviewed AI models that can enhance the performance of AI-driven automation. They conducted analysis experiments on the following models: Random Forest, Multi-Layer Perceptron (MLP), and XGBoost. They proposed a 4-tier automation framework to handle orchestration responses. The framework is based on the NIST guidelines, and the automation model performs tasks such as isolating the network system; however, it lacks testing against adversarial attacks.

Kulkarni et al. (2026) [32] have analyzed the IoT environment, and AI-driven models are commonly used; one such model is the Random Forest Classifier. It is suitable for network intrusion detection, as is the XGBoost model. The RFC also performs better than linear models at handling network traffic issues, as evidenced by its higher accuracy and lower false-positive rates. It also reduces resource waste by optimizing costs and operations. However, they are still vulnerable to Adversary attacks.

Moreover, Nagaraja et al. (2026) [33] have focused on the financial system, and having a well-established AI model is crucial to the success of the framework proposed in this paper. We are using network orchestration automation that follows pre-programmed rules; having AI is adequate for new issues that are not pre-programmed, enabling detection and alerting, and has helped reduce operational disruption. Its response policies have limitations, which constrain its ability to defend against threats.

Li et al. (2026) [34] have pointed out that having the framework send out many alerts to the IT specialists, and to have the alert reveal information is a crucial step in enhancing the performance and not wasting time, therefore, using a random forest classifier would send the alerts with standardized logged data that is useful to handle the alerts, however, the method still requires human involvement in reviewing the data alerts and handle them.

Deploying only a rule-based system to handle network orchestration responses might cause systems to crash when a new, unconfigured rule is added, resulting in a large synthetic dataset. To handle it, Bhavana et al. (2026) [35] propose MATLAB 26.1 simulation to handle the large dataset generated, and they tested it among ML models such as the random forest classifier, and they have succeeded in enhancing the performance efficiency in the network, where the random forest classifier model provides a high rate of model accuracy; however, it has a limitation of having a high cost.

Cloud-edge systems often omit vital components, such as API gateways. As a solution, Peng et al. (2026) [36] proposed a hybrid, AI-based orchestration with an Action Masked Deep Q-Network (AMDQN). This method allows the system, instead of ignoring vitals, to prompt for routing directions. This method has enhanced network performance and reduced waste of time and resources.

Vashisht et al. (2026). [37] proposed a framework that is an agentic intelligence-assisted ML fusion framework. It is based on a random forest classifier, where it works by analyzing system behaviors and predicting the next behaviors; the framework also includes a layer that overviews the decision and validates it with the AI agent, where it acts based on the standardized safety rules that were configured by the IT employee. The framework increased the operations speed over the network, however, it has a limitation of adaption to multi-agent systems.

The 5G network environment can sometimes cause network slowdowns due to poor planning, as it relies on slice-level resource reallocation. Therefore, Iqbal and

Nakao (2026) [38] proposed the 5ERASI framework, which automates slice-level operations and orchestrates resource allocation. They also use random forests and principal component analysis (PCA) to optimize data and increase operational speed, as well as undersampling techniques (ENN). These should all enhance the environment orchestration; however, they have limitations or require specialized resources for testing.

Cloud-based networks could be vulnerable to data breaches and exploitation. Ahmed and Yuan (2026) [39] have proposed integrating a zero-trust AI-driven model with multiple ML models, linear regression (LR) as the baseline, a feed-forward neural network (NN), Random Forests (RF) as estimators, XGBoost, and AES-128 encryption. This method improves the security and performance of models; however, it incurs local storage overhead.

To address challenges in Industrial Internet of Things (IIoT) environments and Industry 5.0, Govindarajan et al. (2026) [40] propose the Ageis-5 framework, which uses multi-classifier models, random forest, SVM, and XGBoost, and a dynamic weighting strategy system that regulates between classifier models based on real-time performance. This reduces false alarms and enhances accuracy; however, it has limitations when testing on hardware devices.

The machine learning integration might pose challenges, such as clustering. To mitigate these challenges, Boufarh et al. (2026) [41] proposed implementing the Mixed Effects Random Forest (MERF) combined with the Artificial Bee Colony (ABC) algorithm, which aims to enhance data operations and output quality and increase accuracy; however, it has the limitation of being computationally complex.

Table 1 represents a comprehensive overview of all the related works mentioned in this section, considering the main focus of the paper, their proposed solution, and the advantages and limitations of the proposed solution.

Table 1. Related works summary.

Ref	Year	Main Focus	Proposed Solution	Advantages	Limitations
[8]	2021	Reducing cost, time and human error	Automation	Reduce human factor intervention	High implementation costs, and various security vulnerabilities
[9]	2021	Cyber deceptions and security orchestration	Automation	Tricking malware into adapting to deceptive environments	Require experts, and generalization
[10]	2021	Enhance the performance and reduce errors and human involvement	Various techniques with AI	Reduced human intervention, increased system performance and scalability	Depends on the data quality
[11]	2022	5G network environment	AI-driven techniques	Saved time, energy and enhance performance	Vulnerable to security threats and demand expensive resources
[12]	2022	5G network environment	AI-driven techniques	Saved time, energy and enhance performance	Vulnerable to security threats and demand expensive resources
[13]	2022	Human errors in misconfiguration	Automation	Reduce the risk of inconsistent configuration and the time of configuration	Might not work as well in traditional networks, and it depends on the accuracy of the models used

Table 1. Cont.

Ref	Year	Main Focus	Proposed Solution	Advantages	Limitations
[14]	2022	Tasks and performance robustness in cloud environment	ML-based orchestration	Enhance performance, adaptability, and reduce human decision-making	Demand special resources, requires expertise, and ML performance depends on the quality of data
[15]	2023	Manual configurations	AI-driven automation	Robust protection structure, reduces time and cost, and does not require expertise	Fails to scale effectively in large networks, limited applicability
[16]	2023	Orchestration of traditional cloud-based systems	Agent-based system	Increased performance and scalability	The decentralization's security threats and vulnerabilities
[17]	2024	Manual configurations	AI-driven automation	Improve network efficiency, performance, scalability, and security	Unable to manage large networks
[18]	2024	Network orchestration issues	AI, ML, and Kubernetes management	Improve performance and efficiency of the system	Requires expensive resources, interpretation difficulties, and AI is vulnerable to various attacks
[19]	2025	Complex infrastructure of 5G and beyond	AI-driven automation	Enhance decision-making and optimize performance	Deploying the same methodology on a smaller company might not guarantee the same output
[2]	2025	Reduce human involvement	RPA model	Reduce errors and time	Requires employee's expertise
[20]	2025	Reduce errors in finance field	AI-Driven model and LLM	Reduce cost and increase speed	AI model is vulnerable to data bias
[21]	2026	Addressing automation bias	Cognitive forcing functions (CFFs)	Reduce bias	Lack qualitative analysis
[22]	2026	Intelligent frameworks in industry 4.0	Automation	Improve efficiency and reduce overload	Configuration requires expertise
[23]	2026	Smart homes gadgets	AI-Driven and voice controlled automation	Enhance user experience and accessibility	Vulnerable to background speech
[24]	2026	5G and IoT environment	AI-Driven automation	Reduce manual configuration	Vulnerable to cybersecurity attacks
[25]	2026	Preserve energy in smart homes	AI and automation	Mitigate preserving energy	Requires good internet connection
[26]	2026	Manual handling of report file formats, such as Excel and PDF	RPA and AI model	Reduce errors	Does not eliminate errors but only reduces them
[27]	2026	Healthy workplace environment	ML model and automation	Mitigate the employees' healthy lifestyle	Privacy matter
[28]	2026	MITRE ATT&CK	AI drive of attack detection	Increase security	AI alerts do not provide a lot of details
[29]	2026	SOAR technologies	Automated decision making and deep reinforcement learning (DRL)	Fast response	Requires expertise

Table 1. Cont.

Ref	Year	Main Focus	Proposed Solution	Advantages	Limitations
[30]	2026	Reduce manual involvement in Order handling	RPA model	Increase speed and reduce errors	Only handles pre-programmed rules
[31]	2026	Review AI models	A 4-tier automation framework	Improve efficiency	Vulnerable to adversarial attacks
[32]	2026	IoT environments	AI-Driven models	Enhance accuracy and reduce false positive	Adversary attacks vulnerabilities
[33]	2026	Financial system environment	network orchestration automation	Decrease operation disruption	response policies lacking strong defense against cyber attacks
[34]	2026	AI models false positive alerts	Random Forest classifier	Robustness	Requires human data monitoring
[35]	2026	Rule based system	MATLAB simulation	Enhance performance	High cost
[36]	2026	Cloud edge based networks	Hybrid orchestration AI based	Enhance operations speed and reduces waste of resources	Lack of dynamic testing
[37]	2026	6G networks environments	Agentic intelligence assisted ML fusion framework	Increase speed of operations	Limited adaption to multi agent systems
[38]	2026	5G network environments	5ERASI which is an automation based framework	Increase speed of operations	Requires special resources for testing
[39]	2026	Cloud-based networks vulnerabilities	zero trust, AI-Driven model, multi ML model and AES-128 encryption	Increase security	Local storage overhead
[40]	2026	Industrial Internet of Things environments (IIoT) and Industry 5.0	Ageis-5	Reduces false alerts and enhances accuracy	Lack hardware testing
[41]	2026	Machine learning integration challenges	Mixed Effects Random Forest (MERF)	Increase accuracy	Complex computations

Therefore, in this paper, we propose a network orchestration deployment. Based on the previous research discussed in this section, automation is a good solution to deploy to mitigate human error interventions, detect malware, ensure consistent orchestration handling, increase speed, and enhance performance; however, to handle the resources without increasing the cost, we have an AI decision maker that would assist the human employee, and human involvement would still be essential for the important decision-making tasks.

3. Proposed Methodology and Materials

In this section, we propose a method to address human error in network orchestration, the high cost and resource requirements of integrating AI, and the new security threats and vulnerabilities that have emerged with its integration. In this paper, we propose to only implement AI for decision-making, then the decision made will go to one of the previously developed automated tasks, and in case the description does not find a match for the available automated tasks, then it goes to the IT specialist for a review, analysis, or development if needed, based on the decision received; where the AI-decision has been made based on an analysis of historical data. This way, we will overcome the challenges of high costs and limited resources, ensuring that only the required resources

are deployed. Previous research has fully implemented AI agents, eliminating human intervention, opening the door to new security threats and vulnerabilities that require simultaneous attention. In this paper, we propose gradually deploying automation, part by part, to ensure that only the necessary resources are used and that each new threat and vulnerability is addressed one at a time. This approach allows us to focus on each issue and prevent it from being lost in a large scope.

Over time, researchers have shown that gradually integrating robotic automation is more effective than fully integrating it, thereby avoiding waste of resources. Full integration would also take longer in production, since it would first require thorough analysis to ensure resources are used correctly and without waste, followed by development and testing, which might disrupt production or, in other words, delay production efficiency. By contrast, gradual implementation allows production to improve steadily and gives development a comprehensive overview of the challenges to be addressed, as we can see how the test performs in the live environment. Previous researchers have explored the idea of partially relying on automation, arguing that automation should be part of our jobs, not all of them; it should help us, not take over. Integrating automation guarantees consistent execution of a specific task in decision-making, and it should be deployed wisely to ensure its long-term effectiveness. Businesses would mostly care about reducing cost when integrating automation, but in fact, it does more than reduce cost; it can enhance production and quality in a short time [42–44].

The AI plays a crucial role in decision-making to save a lot of time on making employees conduct meetings and analyses to come up with an analysis report to make a decision; this way, they can focus their energy on the other stuff the robot cannot do, and for decision-making, they can leave it to the robot. Integrating AI decision-makers has proven effective for businesses across various fields, including healthcare. Researchers have deployed it to analyze medical records, producing diagnoses from the data to reduce errors and improve accuracy. And the doctors still provide their final opinions; they do not fully depend on AI decision-makers; they use them more as assistants that help, ease, and speed up their day-to-day tasks [45]. Furthermore, businesses have been integrating AI decision-makers to analyze data and produce insightful reports, which can be deployed across departments, such as the marketing department, to analyze customer behavior and support business operations [46]. The AI provides analysis reports and decisions based on historical data to help the employees with their tasks. When the employee receives the AI response, they know how to use the report based on their expertise. The report provides facts, and the employee's job is to know how to make use of the generated report [47–49].

We proposed a similar approach: integrating AI decision-makers and automation into the job to make tasks easier and more accurate, without letting them take over completely. The method we propose aims to overcome issues arising from human error in network orchestration. Based on the analysis of related previous work, researchers have shown that implementing AI agents incurs high costs and requires specialized resources and expertise in integration.

The techniques to be deployed in the methodology include integrating artificial intelligence (AI) decision-makers into the orchestration process. The AI model is trained on historical data such as network logs, security alerts, intrusion attempts, and device configurations; at runtime, it analyzes each incoming orchestration task and produces a decision on how to handle it. Then the AI model produces a report and a decision on how to handle the task. However, an IT specialist needs to use the data generated by the AI model effectively. Furthermore, we integrate automation robots to perform repeated sequential tasks. This automation helps save employees' time and ensure error-free tasks and continuous quality. The AI model triggers the automation tasks.

The model type that would be suitable for the approach we are proposing for the AI decision-maker, which is the supervised learning model using the Random Forest Classifier, is the most suitable for the decision-making-based agent because it allows input data such as categorical data, it provides a high accuracy, it can identify the important features, and it's flexible to environments such as orchestrator tasks, and it's a more fitted choice than other models that have slow processing in interpretations such as Gradient Boosted Trees or Neural Networks that are a suitable fit for more complex environments. For training the model, the data used include network logs, device configurations, and alerts. The model requires a large amount of training data to achieve a high accuracy rate; this choice is validated empirically against alternative classifiers, including gradient boosting and a neural network, in Section 4.2.

This paper proposes a solution for gradually integrating automation, moving step by step to develop and integrate automation tasks based on an analysis of the AI decision and an IT specialist's review. The decision determines whether the task is repetitive and should be automated, or whether it is less expensive to integrate it manually to avoid wasting resources. Therefore, we do not fully eliminate human involvement; rather, it is reduced over time until there is no need for human intervention. As shown in previous research, gradually implementing automation has been effective [50–52].

Figure 1 shows a graphic diagram of the proposed method. Figure 2 shows a more detailed flowchart of the proposed method. Moreover, based on the figures, the real-time data are the orchestration system jobs, such as updating devices, configuring new devices, and configuring IP addresses. The jobs are queued and processed one by one through the AI decision-maker. If a job is found among the automation tasks, it runs the automation task. If it is not found or if the behavior is suspicious, the AI decision-maker sends the decision to the IT specialist for review and manual intervention. If the AI decision indicates that this is a repetitive task that should be automated, then the IT specialist develops an automation task. Otherwise, if it is a special case that occurs only once, there is no need to develop an automation task and waste resources; therefore, prompt the employee to intervene and execute the task manually. Algorithm 1 details the proposed framework. The following steps detail how the framework works:

1. Real-time data on the orchestration system starts the jobs, such as updating devices, configuring new devices, configuring IP addresses, etc.
2. The jobs are queued to the AI decision-maker
3. Based on the output of the AI decision-maker, one of the following options is taken:
 - (a) If it is found among the automation tasks, then proceed to run the automation task
 - (b) If it is not found or if it is a suspicious behavior, then the AI decision-maker sends the decision to the IT specialist for review and to prompt manual intervention
 - (c) If the AI decision indicates that this is a repetitive task that should be automated, then the IT specialist develops an automation task; otherwise
 - (d) If it was a special case occurring as a one-time thing, then no need to develop an automation task and waste resources
4. Repeat for the next real-time data on the orchestration system

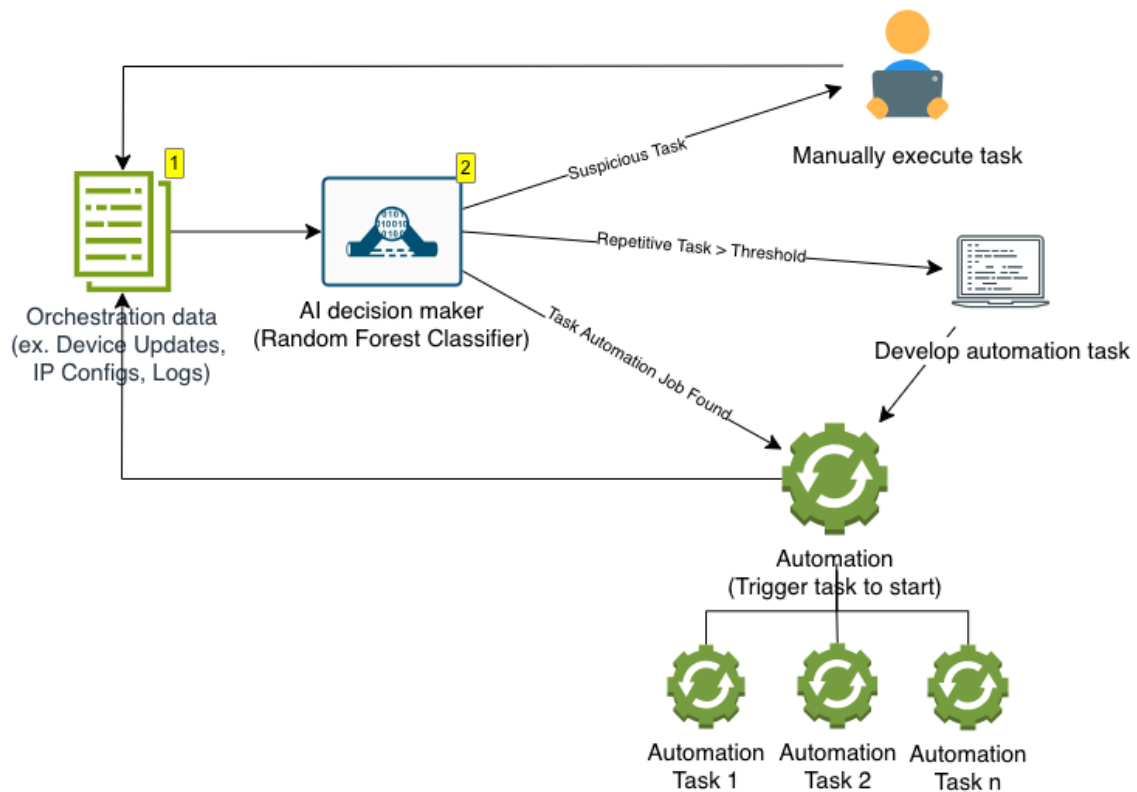


Figure 1. Diagram of the proposed method.

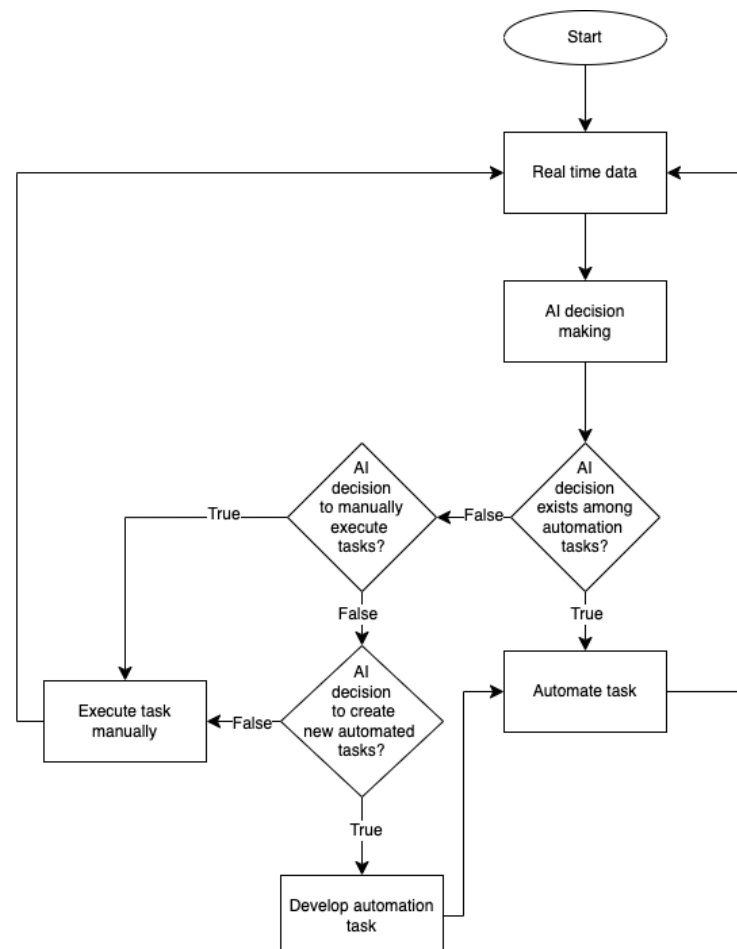


Figure 2. Flowchart of the proposed method.

Algorithm 1: Intelligent Network Orchestration Algorithm

```

Data: Network Orchestration Data
Data_Unprocessed  $\leftarrow$  Network_Orchestration_Data;
foreach Data to Data_Unprocessed do
    if Data.sensitive then
        | Alert_IT(Manual_Handle); // protective guard - precedes the classifier
    else
        Output_Decision  $\leftarrow$  AI_decision_Maker(Data);
        switch Output_Decision do
            case "Automation_Task_Found" do
                | Start_Automation_Task();
            end
            case "Automation_Task_NOTFound" do
                | Alert_IT(Manual_Handle);
            end
            case "Repetitive_Task" do
                | Pending[Task]  $\leftarrow$  Pending[Task] + 1;
                if Pending[Task]  $\geq$  Threshold then
                    | Alert_IT(Review_and_Develop_Automation_Task);
                else
                    | Alert_IT(Manual_Handle);
                end
            end
            case "Otherwise" do
                | Alert_IT();
            end
        end
    end
end

```

4. Experiment and Simulation

To validate the proposed framework, we implemented the AI decision-maker and the gradual-automation logic in Python 3.11 (scikit-learn 1.4, NumPy 1.26, SciPy 1.11) and simulated 30 operational days of an orchestration system under six alternative operating policies, reporting all headline results as means with 95% confidence intervals over 20 independent runs. The experiment is designed to answer four questions: (i) how accurately can the decision-maker predict the operationally optimal action when that action is genuinely uncertain; (ii) how does the proposed framework compare against meaningful alternatives, including a rule-based router and non-incremental automation strategies, once error and infrastructure costs are accounted for; (iii) how robust are the conclusions to the cost-model parameters and to the framework's own hyperparameters; and (iv) how does the framework behave when the operating environment shifts away from the one it was trained on? The full source code, all figures, and the raw result logs are openly available on Zenodo (see the Data Availability Statement).

4.1. Experimental Setup and Data Generation

Because real orchestration logs from enterprises are typically confidential, the experiment uses a synthetic task generator. In the revised design, each task carries two kinds of variables. Observable features are what a real orchestrator records in its logs and are the only inputs given to the classifier: task type, device type, priority, historical frequency, configuration complexity, whether a matching automation already exists, the number of similar past jobs, and a sensitivity flag. Latent outcome variables are never shown to the

classifier: whether the task will actually recur in the future, whether an automation of it would actually work reliably, and whether automating it would carry a residual operational risk. The latent variables are drawn stochastically—they are correlated with, but not determined by, the observable features (for example, the probability of true recurrence increases smoothly with the observed historical frequency). The ground-truth label of each task is the action a perfectly informed operator would take given the latent outcome: use an existing automation, develop a new automation, or execute manually. Because the latent variables are stochastic given the observables, the label is not a deterministic function of the classifier’s inputs, and no classifier can reach 100% accuracy. We estimate the achievable ceiling by Monte-Carlo re-sampling of the latent variables: the most likely optimal action given the observables agrees with the realized optimal action in approximately 79.1% of cases. Reported accuracies must therefore be read against this ceiling rather than against 100%. We distinguish explicitly between training data—a historical set of 5000 labeled tasks representing the organization’s past logs—and runtime input data—the stream of new tasks arriving during the 30 simulated days, for which the framework must make decisions online. To assess generalization, we additionally generate a distribution-shifted test population representing a “younger”, more security-oriented network: a different task-type mix, higher configuration complexity, and lower historical frequencies. The classifier is trained only on the in-distribution historical data and is never retrained. Table 2 lists the cost-model parameters, their default values, and the rationale for each. The defaults are working assumptions rather than measured constants; for this reason every influential parameter is varied over a wide range in the sensitivity analysis of Section 4.5, and the qualitative conclusions are evaluated across that range rather than at the point values. Unlike the previous version of this experiment, the cost model now includes error costs on both sides: manual tasks incur a 3% human-error probability with 30 min of rework; automated tasks can fail and require fixing; automations developed for tasks that were actually infeasible remain fragile and fail in 30% of their executions; auto-executing a high-risk task can trigger an incident requiring 180 min of remediation; and the AI-based policies additionally pay a daily infrastructure cost and an amortized model-maintenance effort. The fully manual baseline is therefore no longer error-free.

Table 2. Cost-model parameters, default values, and rationale.

Parameter	Default	Rationale
Manual task effort	12 min	Mid-range of common orchestration changes (6–20 min varied in sensitivity analysis)
Automated task effort	0.5 min	Robot execution plus logging overhead
Automation development	60 min	One playbook/script per task signature (30–180 min varied)
IT-specialist rate	1.20 USD/min	≈72 USD/h fully-loaded IT labour rate
Human error prob./rework	3%/30 min	Manual misconfiguration and its correction (0–10% varied)
Robot failure prob./fix	0.5%/15 min	Failure of a well-built automation
Fragile automation failure/fix	30%/20 min	Failure rate when the automated task was in fact infeasible
Incident prob./remediation	10%/180 min	Auto-execution of a high-risk (sensitive) task going wrong
AI infrastructure	12 USD/day	Model hosting and inference compute
AI maintenance	≈0.5 h/week	Model monitoring and retraining effort, amortised daily
Specialist review	10 min	Review of one development recommendation
Specialist accuracy	$p = 0.85$	Probability the review judgment is correct (0.5–1.0 swept in Section 4.5)

Two mechanisms of the proposed framework are made explicit in this revision. First, a *protective guard rule*: any task whose observable sensitivity flag is set is routed to the IT specialist, and this rule takes precedence over the classifier’s output and over the automation library. The guard is applied identically in the rule-based, full-AI, and proposed policies, and its consequences are attributed to the rule itself rather than to the learned model. Second, an *imperfect, costed specialist review*: when a task signature has accumulated the required number of development recommendations, the specialist reviews the accumulated evidence; the review consumes 10 min of specialist time, and the specialist’s judgment of whether the candidate is genuinely recurring and feasible is correct with probability p (default $p = 0.85$), independent of the unobservable ground truth otherwise. Declined signatures may be re-reviewed after further recommendations accumulate and are excluded from automation after two declines. The specialist therefore no longer has privileged access to the latent variables, and the review is no longer free. Finally, all results in Sections 4.3 and 4.4 are reported as means with 95% confidence intervals over 20 independent simulation runs with different random seeds, with all policies evaluated on identical task streams within each run (common random numbers).

4.2. Accuracy of the AI Decision-Maker

Table 3 and Figure 3 compare five classifiers trained on the observable features: logistic regression, a single decision tree, a Random Forest, gradient boosting, and a multi-layer-perceptron (MLP) neural network. The Random Forest attains the best in-distribution test accuracy (78.3%, macro-F1 0.775), marginally ahead of gradient boosting (78.0%) and logistic regression (77.5%), and clearly ahead of the MLP (69.9%), which is data-hungry relative to the 4000 training examples available. Crucially, the Random Forest’s accuracy sits within one percentage point of the estimated 79.1% achievable ceiling—the model has extracted nearly all the signal that the observable features carry about the latent outcome. On the distribution-shifted test set, the Random Forest retains 81.3% accuracy without retraining. These results justify the Random Forest choice made in Section 3 on empirical rather than only qualitative grounds.

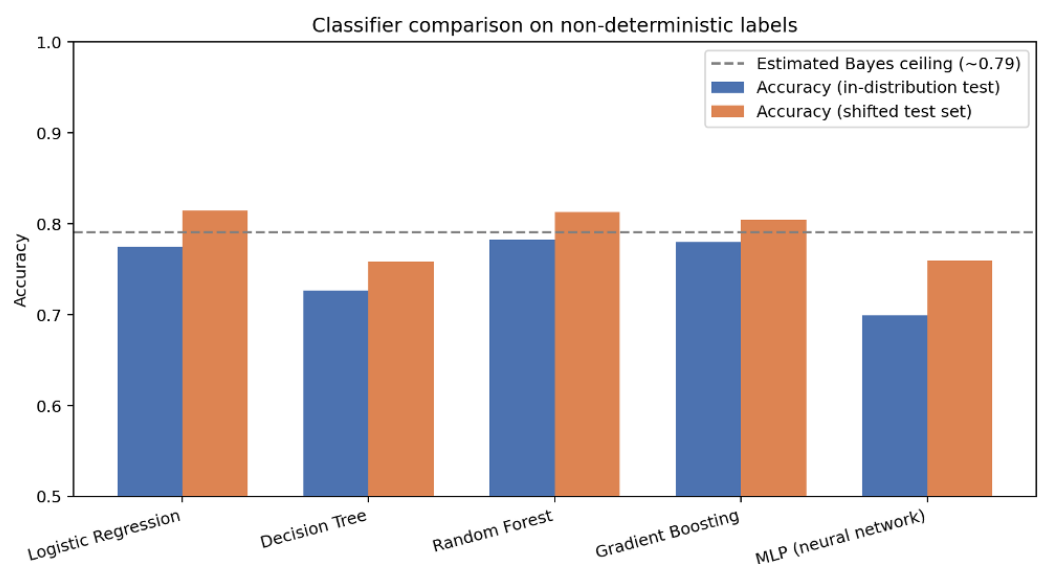


Figure 3. Classifier accuracy on the in-distribution and shifted test sets, against the estimated achievable ceiling.

Figure 4 shows the Random Forest confusion matrix. Errors concentrate where the underlying uncertainty lives: tasks whose recurrence or feasibility is genuinely borderline are confused between the automation classes and manual execution, while the two au-

tomation classes are almost never confused with each other. Figure 5 reports validation curves over the two Random Forest hyperparameters. Accuracy is flat between 25 and 400 trees and peaks on a broad plateau for maximum depths of 8–12, degrading for deeper, overfitting-prone settings; the values used in the paper (200 trees, depth 12) lie on this plateau, and the conclusions are insensitive to the exact choice.

Table 3. Classifier comparison on the non-deterministic labels (best values in bold).

Model	Acc. (Test)	F1 (Test)	Acc. (Shifted)	F1 (Shifted)	Fit Time (s)
Logistic Regression	0.775	0.766	0.815	0.686	0.03
Decision Tree	0.727	0.720	0.758	0.645	0.01
Random Forest	0.783	0.775	0.813	0.699	0.53
Gradient Boosting	0.780	0.773	0.805	0.695	1.06
MLP (neural network)	0.699	0.684	0.760	0.621	5.27

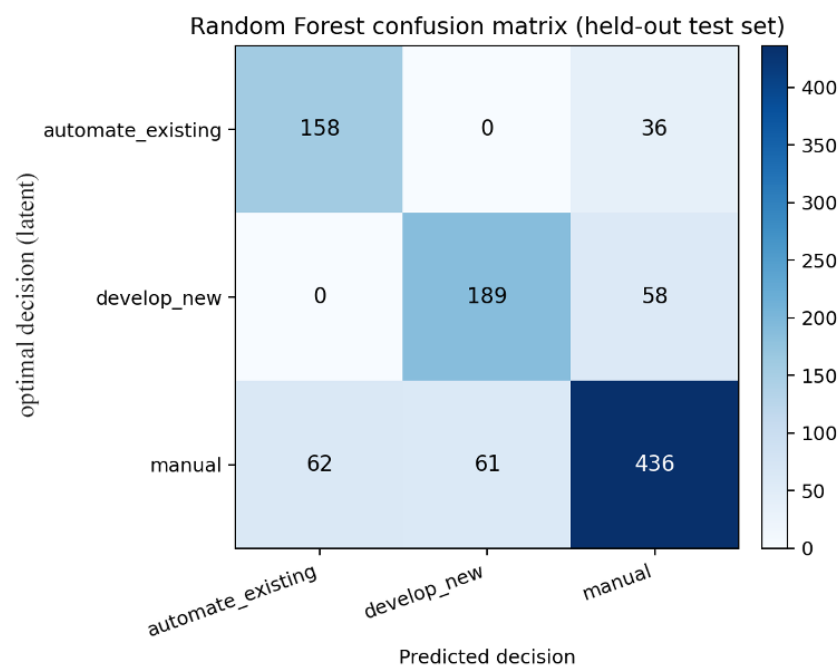


Figure 4. Confusion matrix of the Random Forest decision-maker on the held-out test set.

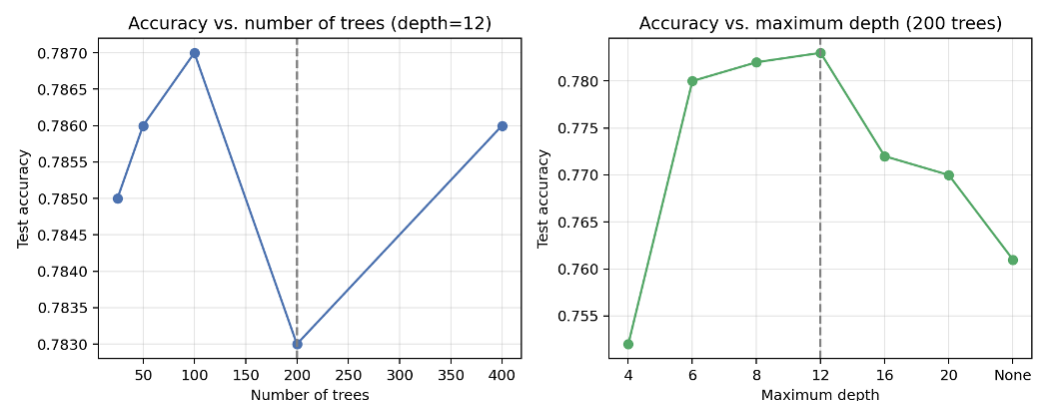


Figure 5. Validation curves for the Random Forest hyperparameters; dashed lines mark the values used.

4.3. Policy Comparison Under Error and Infrastructure Costs

The runtime simulation compares six operating policies over 30 days with 80 incoming tasks per day (2400 tasks): (a) fully manual execution; (b) a rule-based router that applies

fixed thresholds to the observable features and needs no machine learning; (c) automate everything, which develops an automation for every new task signature at first sight and executes all tasks automatically, including sensitive ones; (d) automate everything + guard, identical to (c) except that the protective sensitivity guard of Section 4.1 is applied; (e) full AI, in which the classifier routes every task and automations are developed immediately on first recommendation, with no human review; and (f) the proposed framework, in which the classifier routes tasks, an automation is developed only after the same signature has been recommended five times and the imperfect, costed specialist review of Section 4.1 approves it, and tasks routed to the human stay with the human. Table 4 and Figure 6 report means $\pm$ 95% confidence intervals over 20 runs. Every policy that carries the guard records zero incidents; the unguarded automate-everything policy records 49.0 ± 3.6 incidents, and adding the guard alone recovers its savings from 40.0% to 51.1%. Incident prevention is therefore a property of the guard rule on an observable feature, not of the learned model. The proposed framework attains the best in-distribution savings ($61.4 \pm 0.7\%$), ahead of the rule-based router ($59.8 \pm 0.7\%$) by a margin that exceeds the confidence intervals, and it is the only policy with near-zero wasted development (0.5 ± 0.3). Section 4.4 examines how this ranking changes when the environment shifts.

Table 4. Policy comparison over 30 days, in-distribution environment (2400 tasks); mean $\pm$ 95% CI over 20 runs ($p = 0.85$).

Policy	Hours	Cost (USD)	Incidents	Wasted dev.	Savings vs. Manual
Fully manual	515.4 $\pm$ 1.7	37,109 $\pm$ 125	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	—
Rule-based router	206.9 $\pm$ 3.9	14,900 $\pm$ 281	0.0 $\pm$ 0.0	5.5 $\pm$ 1.4	59.8 $\pm$ 0.7%
Automate everything (no guard)	309.1 $\pm$ 13.1	22,252 $\pm$ 942	49.0 $\pm$ 3.6	35.5 $\pm$ 2.0	40.0 $\pm$ 2.5%
Automate everything + guard	252.1 $\pm$ 5.3	18,151 $\pm$ 381	0.0 $\pm$ 0.0	27.8 $\pm$ 1.6	51.1 $\pm$ 1.0%
Full AI (no review)	213.4 $\pm$ 4.3	15,882 $\pm$ 312	0.0 $\pm$ 0.0	5.2 $\pm$ 1.3	57.2 $\pm$ 0.8%
Proposed (gradual) ¹	191.8 $\pm$ 3.9	14,325 $\pm$ 278	0.0 $\pm$ 0.0	0.5 $\pm$ 0.3	61.4 $\pm$ 0.7%

¹ Bold indicates the best-performing policy in this environment.

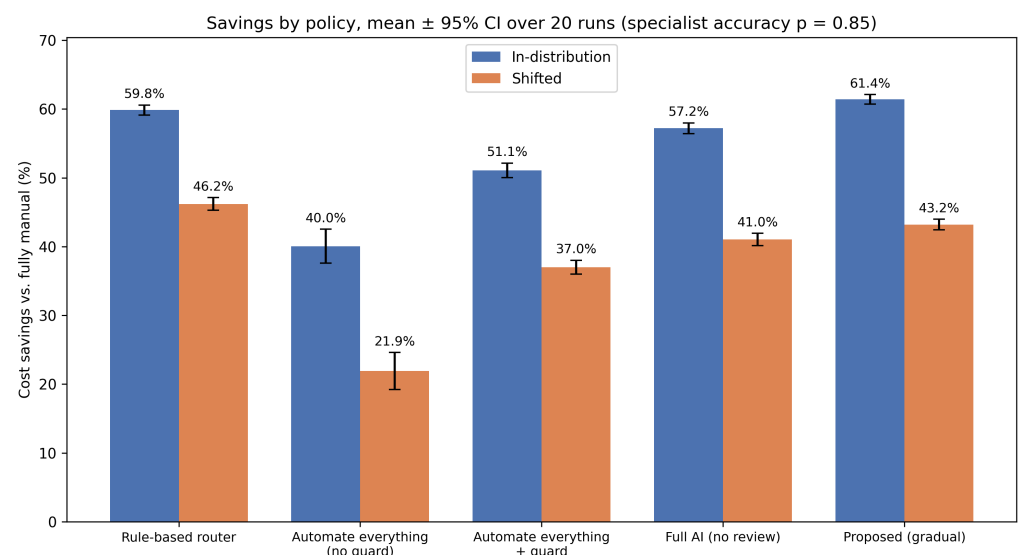


Figure 6. Savings by policy in the in-distribution and shifted environments; bars show means $\pm$ 95% CI over 20 runs ($p = 0.85$).

4.4. Robustness to Distribution Shift

To separate the contribution of the learned decision-maker from that of the incremental strategy, all six policies are re-run in the shifted environment of Section 4.1, without re-

training the classifier and without re-tuning the rule thresholds. Table 5 and Figure 7 show the outcome. Under distribution shift, the ranking changes: the rule-based router leads ($46.2 \pm 0.9\%$), followed by the proposed framework ($43.2 \pm 0.8\%$); the difference exceeds the confidence intervals. The single-run advantage of the proposed framework reported in the previous revision does not survive multi-seed evaluation, and we correct that claim here. Unguarded automate-everything collapses to 21.9% with 75.3 ± 3.9 incidents, while its guarded counterpart retains 37.0%. The proposed framework's remaining advantages under shift are structural rather than cost-based: it requires no environment-specific threshold tuning (the router's thresholds are, by construction, calibrated to the training environment and would require manual re-tuning in practice), and it retains near-zero wasted development (0.1 ± 0.2 versus 7.2 ± 1.0).

Table 5. Policy comparison over 30 days, distribution-shifted environment; mean $\pm$ 95% CI over 20 runs ($p = 0.85$).

Policy	Hours	Cost (USD)	Incidents	Wasted dev.	Savings vs. Manual
Fully manual	515.4 ± 1.7	$37,109 \pm 125$	0.0 ± 0.0	0.0 ± 0.0	—
Rule-based router	277.3 ± 5.0	$19,968 \pm 363$	0.0 ± 0.0	7.2 ± 1.0	$46.2 \pm 0.9\%$
Automate everything (no guard)	402.6 ± 14.5	$28,989 \pm 1044$	75.3 ± 3.9	53.9 ± 2.6	$21.9 \pm 2.7\%$
Automate everything + guard	324.7 ± 5.5	$23,379 \pm 399$	0.0 ± 0.0	44.0 ± 1.9	$37.0 \pm 1.0\%$
Full AI (no review)	296.9 ± 5.0	$21,889 \pm 361$	0.0 ± 0.0	7.2 ± 1.1	$41.0 \pm 0.9\%$
Proposed (gradual)	285.6 ± 3.7	$21,076 \pm 269$	0.0 ± 0.0	0.1 ± 0.2	$43.2 \pm 0.8\%$

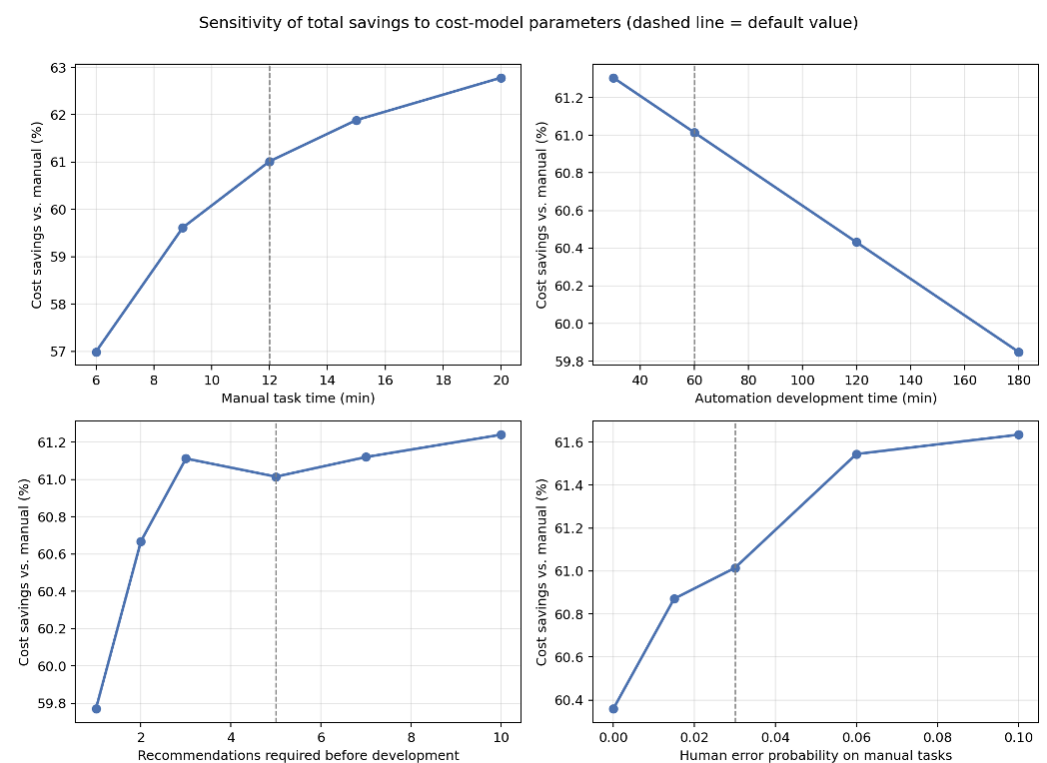


Figure 7. Sensitivity of the proposed framework's savings to the cost-model parameters and to the development threshold; dashed lines mark the default values.

4.5. Sensitivity and Robustness Analysis

Figure 7 varies the four most influential parameters one at a time and re-computes the proposed framework's savings against the manual baseline. Savings range from 57.0% (when a manual task takes only 6 min) to 62.8% (when it takes 20 min); they decline only mildly, from 61.3% to 59.8%, as the development cost of one automation is increased six-fold

from 30 to 180 min; and they are almost flat in the human-error probability, because both the proposed framework and the baseline contain manual work. The development threshold—the number of repeated recommendations required before an automation is built—changes savings by less than 1.5 percentage points between 1 and 10; the value of 5 used in the paper is therefore not a tuned constant but a mid-range choice, and small deviations from it are immaterial. In no examined configuration does the proposed framework lose its advantage over the manual baseline, and its qualitative ranking among the policies is likewise preserved.

Figure 8 varies the specialist’s assessment accuracy p from 0.5 (uninformative) to 1.0 (oracle). The effect on total savings is below 0.3 percentage points in both environments, and wasted development rises only from ≈ 0.5 to ≈ 1 as p falls to 0.5. The reason is structural: under the task semantics of Section 4.1, most of the automation library is populated by registering pre-existing automations, and only four to five development reviews occur per 30-day run, so the reviewer’s reliability has little room to influence the totals. We state this explicitly rather than attributing the framework’s performance to the reviewer: in environments where most automations must be developed from scratch, the reviewer’s reliability and cost would weigh far more heavily.

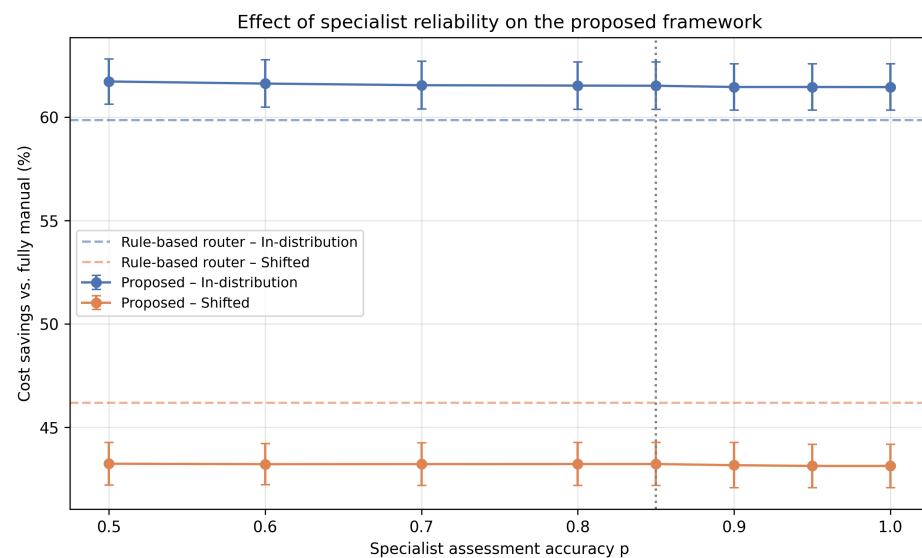


Figure 8. Effect of specialist assessment accuracy p on the proposed framework’s savings; dashed lines mark the rule-based router’s mean savings.

4.6. Dynamics of the Gradual Rollout

Figure 9 shows the proposed framework’s daily task mix and Figure 10 the growth of its automation library in a representative run. The library is empty at deployment (day 0). During the first operational day, the framework already routes recurring tasks with existing automations to the robots, so the library ends day 1 with 18 signatures while 55 of the 80 tasks are still executed manually; the values reported for each day are end-of-day states. Automated execution then grows as recommendations accumulate past the review threshold, stabilizing after roughly two weeks at around 83 signatures, while sensitive and one-off tasks continue to be routed to the IT specialist throughout—by design, manual work never disappears entirely.

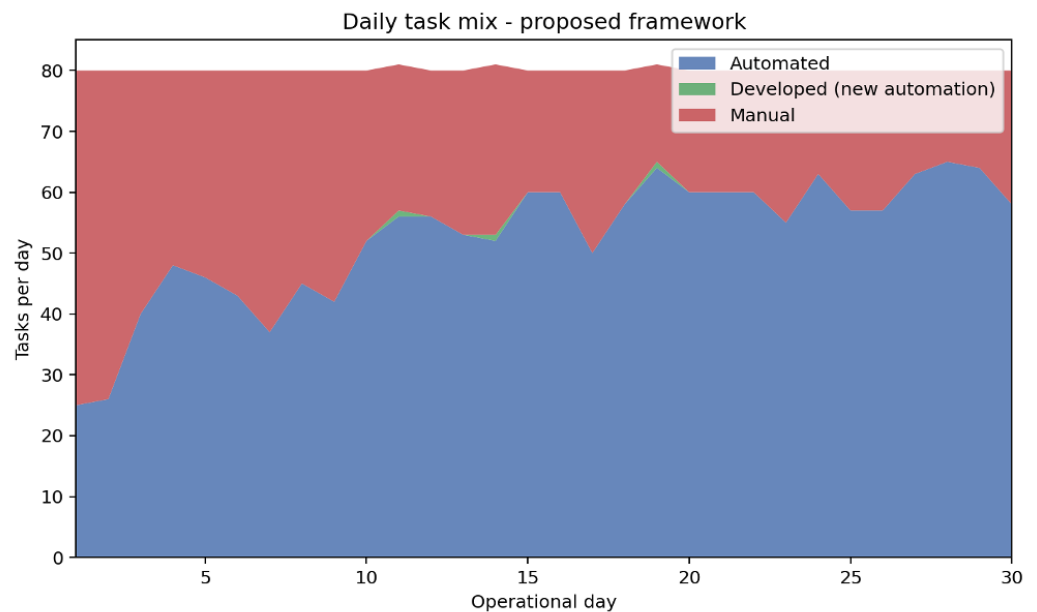


Figure 9. Daily task mix under the proposed framework.

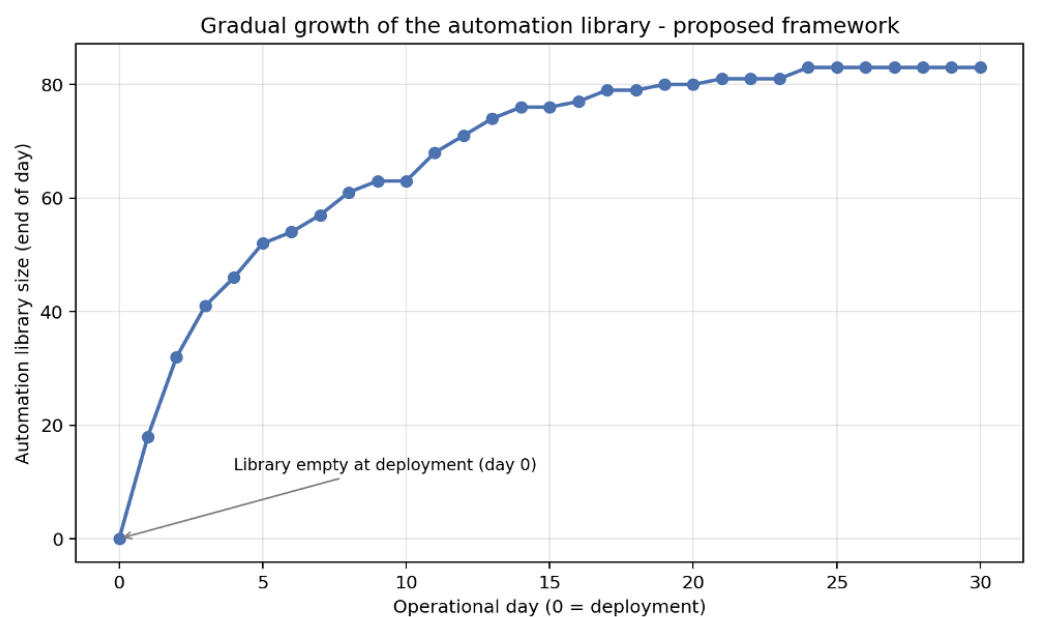


Figure 10. Automation library growth from an empty state at deployment; values are end-of-day.

4.7. Discussion and Threats to Validity

Three limitations qualify these results. First, the data remain synthetic: although the latent-variable design removes the circularity between labeling and features, the generator's structure was still specified by the authors, and a real deployment would train the decision-maker on the organization's own historical logs, whose feature–outcome relationships may be weaker or stronger than simulated. The distribution-shift experiment partially mitigates, but does not eliminate, this threat. Second, the cost model, while now including human errors, automation failures, incidents, wasted development, and AI infrastructure and maintenance, still simplifies reality—for example, effort is uniform within each category rather than scaling with task complexity; the sensitivity analysis bounds, but does not remove, the resulting uncertainty. Third, in a production environment the framework would ingest runtime tasks through the orchestrator's event bus or northbound APIs and would require periodic retraining as the network evolves; the maintenance cost included

here approximates, but cannot fully anticipate, that operational burden. Within these limits, the experiment supports the paper's central claims: the decision-maker extracts nearly all available predictive signal; the explicit sensitivity guard—not the learned model—eliminates incidents; the gradual, reviewed strategy keeps wasted development near zero and achieves the best in-distribution savings; and under distribution shift it remains competitive with, though no longer ahead of, a router hand-tuned to the environment.

5. Conclusions and Future Work

In this paper, we proposed a network orchestration framework that combines an AI decision-maker with gradual task automation to address the high cost, heavy resource demands, and human-error risks of manual orchestration. The framework analyses real-time orchestration data using a Random Forest Classifier, which classifies each incoming task into one of three actions: execute the task using an existing automation, request that an IT specialist develop a new automation, or escalate the task for manual execution. By developing automations only after the AI has repeatedly recommended the same task signature, the framework expands its automation library step by step rather than in a single costly rollout, preserving human judgment for sensitive or one-off jobs.

To validate the framework, we ran a 30-day simulation in which ground-truth decisions depend on latent outcomes the classifier never observes, evaluated over 20 independent runs with 95% confidence intervals. The Random Forest decision-maker reached 78.3% accuracy against a 79.1% achievable ceiling and outperformed four alternative classifiers. Among six operating policies—including a rule-based router, unguarded and guarded immediate automation, and full AI without review—the proposed framework achieved the highest in-distribution savings ($61.4 \pm 0.7\%$) with near-zero wasted development, and $43.2 \pm 0.8\%$ under distribution shift, where only a router hand-tuned to the environment exceeded it. The absence of incidents in all guarded policies is attributable to an explicit sensitivity rule rather than to the learned model, and the results are stable across wide ranges of every cost parameter and of the specialist's reliability.

The proposed method has limitations that should be acknowledged. First, the framework still requires an IT specialist to review the AI recommendations and develop new automations, which means it reduces rather than eliminates the need for human expertise. Second, the accuracy of the AI decision-maker depends on the volume and quality of the historical data used to train it. Third, introducing AI and automation components into the orchestration pipeline creates new attack surfaces that adversaries could target, including the AI model and the automation robots themselves; these surfaces must be monitored and protected by the IT team alongside existing network defenses.

For future work, we recommend three directions. First, a dedicated study on the AI decision-maker is needed to improve its accuracy and robustness, as the framework partly depends on the quality of its recommendations. This study should explore deeper architectures on larger, real-world datasets and analyze the model's behavior under adversarial conditions. Second, the proposed method should be tested in real-world enterprises that operate large orchestration workloads, so that the AI can be trained on authentic historical logs and operational issues that do not appear in synthetic simulations can be identified and addressed. Third, multiple specialized AI decision-makers could be deployed across different departments of the organization, allowing each agent to handle its own portion of the network while communicating with the others; this would extend the framework's coverage to a broader range of the company's systems.

Author Contributions: Conceptualization, T.A.; methodology, T.A.; software, M.A.; validation, M.A.; formal analysis, M.A.; investigation, M.A. and A.A.; resources, M.A. and A.A.; data curation, M.A.; writing—original draft preparation, T.A.; writing—review and editing, M.A. and A.A.; visualization, M.A.; supervision, A.A.; project administration, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: We would like to thank the Saudi ARAMCO Cybersecurity Chair for funding this project.

Data Availability Statement: The simulation source code, result logs, and figures supporting the findings of this study are openly available on Zenodo at <https://doi.org/10.5281/zenodo.21434165> accessed on 20 July 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Roehrich, J.K.; Kalra, J.; Squire, B.; Davies, A. Network orchestration in a large inter-organizational project. *J. Oper. Manag.* **2023**, *69*, 1078–1099. [[CrossRef](#)]
2. Shamsuzzoha, A.; Pelkonen, S. A robotic process automation model for order-handling optimization in supply chain management. *Supply Chain Anal.* **2025**, *9*, 100102. [[CrossRef](#)]
3. Bhardwaj, V.; Kumar, M. Transforming higher education with robotic process automation: Enhancing efficiency, innovation, and student-centered learning. *Discov. Sustain.* **2025**, *6*, 356. [[CrossRef](#)]
4. Firdaouss, L.; Ayoub, B.; Manal, B.; Ikrame, Y. Automated VPN configuration using DevOps. *Procedia Comput. Sci.* **2022**, *198*, 632–637. [[CrossRef](#)]
5. Alozie, C.E.; Ajayi, O.O.; Akerele, J.I.; Kamau, E.; Myllynen, T. The role of automation in site reliability engineering: Enhancing efficiency and reducing downtime in cloud operations. *J. Cloud Oper. Autom. Forthcom.* **2025**, *11*, 160–181. [[CrossRef](#)]
6. Ajiga, D.; Okeleke, P.A.; Folorunsho, S.O.; Ezeigweneme, C. The role of software automation in improving industrial operations and efficiency. *Int. J. Eng. Res. Updat.* **2024**, *7*, 22–35. [[CrossRef](#)]
7. Mrida, M.S.H.; Rahman, M.A.; Alam, M.S. AI-Driven Data Analytics and Automation: A Systematic Literature Review of Industry Applications. *Strateg. Data Manag. Innov.* **2025**, *2*, 21–40. [[CrossRef](#)]
8. Ribeiro, J.; Lima, R.; Eckhardt, T.; Paiva, S. Robotic process automation and artificial intelligence in industry 4.0—A literature review. *Procedia Comput. Sci.* **2021**, *181*, 51–58. [[CrossRef](#)]
9. Sajid, M.S.I.; Wei, J.; Abdeen, B.; Al-Shaer, E.; Islam, M.M.; Diong, W.; Khan, L. Soda: A system for cyber deception orchestration and automation. In Proceedings of the 37th Annual Computer Security Applications Conference, Virtual, 6–10 December 2021; pp. 675–689.
10. Belgaum, M.R.; Alansari, Z.; Musa, S.; Alam, M.M.; Mazliham, M. Impact of artificial intelligence-enabled software-defined networks in infrastructure and operations: Trends and challenges. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 66–73. [[CrossRef](#)]
11. Manda, J.K. AI-driven Network Orchestration in 5G Networks: Leveraging AI and Machine Learning for Dynamic Network Orchestration and Optimization in 5G Environments. *Educ. Res. (IJMCER)* **2022**, *4*, 356–365.
12. Moubayed, A.; Shami, A.; Al-Dulaimi, A. On end-to-end intelligent automation of 6G networks. *Future Internet* **2022**, *14*, 165. [[CrossRef](#)]
13. Bringhenti, D.; Marchetto, G.; Sisto, R.; Valenza, F.; Yusupov, J. Automated firewall configuration in virtual networks. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1559–1576. [[CrossRef](#)]
14. Zhong, Z.; Xu, M.; Rodriguez, M.A.; Xu, C.; Buyya, R. Machine learning-based orchestration of containers: A taxonomy and future directions. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 217. [[CrossRef](#)]
15. Bringhenti, D.; Marchetto, G.; Sisto, R.; Valenza, F. Automation for network security configuration: State of the art and research trends. *ACM Comput. Surv.* **2023**, *56*, 57. [[CrossRef](#)]
16. Anuraj, B. Agent-based orchestration on a swarm of edge devices. In Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems, Neuchâtel, Switzerland, 27–30 June 2023; pp. 199–202.
17. Bringhenti, D.; Sisto, R.; Valenza, F. Automating VPN configuration in computer networks. *IEEE Trans. Dependable Secur. Comput.* **2024**, *22*, 561–578. [[CrossRef](#)]
18. Kaul, D. AI-Driven Self-Healing Container Orchestration Framework for Energy-Efficient Kubernetes Clusters. *Emerg. Sci. Res.* **2024**, *2*, 1–13. [[CrossRef](#)]
19. Nadaf, S.R.; Yarlagadda, S.B.; Rath, H.K.; Mishra, G. A Cognitive Resource Aware Orchestration of Network Services for 5G and beyond. In Proceedings of the 26th International Conference on Distributed Computing and Networking, Hyderabad, India, 4–7 January 2025; pp. 1–10.

20. Kothandapani, H.P. Ai-driven regulatory compliance: Transforming financial oversight through large language models and automation. *Emerg. Sci. Res.* **2025**, *12*, 12–24.
21. Romeo, G.; Conti, D. Exploring automation bias in human–AI collaboration: A review and implications for explainable AI. *AI Soc.* **2026**, *41*, 259–278.
22. Tayal, C. Assessing the Scope of Optimising Operational Roles Through Intelligent Automation Frameworks. In Proceedings of the 2026 IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 5–7 January 2026; IEEE: Piscataway, NJ, USA, 2026; pp. 0063–0068.
23. Varnisha, S.; Srivarshini, S. AI-Driven Voice-Controlled Virtual Assistant Framework for Smart Home Automation. In Proceedings of the 2026 International Conference on Visual Analytics and Data Visualization (ICVADV), Tamil Nadu, India, 24–26 February 2026; IEEE: Piscataway, NJ, USA, 2026; pp. 1344–1351.
24. Sidhu, S.K.S.; Sharma, A. The Evolution of Network Management: From Manual Configurations to Intelligent Automation. *Int. J. Netw. Manag.* **2026**, *36*, e70031.
25. Badhel, I.; Ingle, H.; Ingle, S.; Ingle, T.; Ingole, K.; Gupta, I.; Desai, S. Automation of Power Management for Electrical Appliances Image Processing. In Proceedings of the 2026 International Conference on Emerging Trends and Innovations in ICT (ICEI), Pune, India, 9–11 January 2026; IEEE: Piscataway, NJ, USA, 2026; pp. 1–5.
26. Lihansisco, J.; Wiputra, R. Intelligent Automation System for Multi-Format Document Processing. In Proceedings of the 2026 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 11–13 March 2026; IEEE: Piscataway, NJ, USA, 2026; pp. 1–6.
27. Valtonen, A.; Kimpimäki, J.P.; Savela, N. Employee wellbeing: A computational review on the consequences of workplace automation. *Technovation* **2026**, *152*, 103424. [\[CrossRef\]](#)
28. Mgbemele, A.F. Advancing Cyber Threat Detection through SIEM-Based Automation and MITRE ATT&CK Aligned Analytics: A Systematic Review. *Asian J. Res. Comput. Sci.* **2026**, *19*, 233–254. [\[CrossRef\]](#)
29. Nguyen, P.; Rauniyar, A.; Bartel, J.; Laufer, J.; Dalamagkas, C.; Pohl, K. AI-driven digital twin-based security orchestration, automation and response for critical infrastructures. *Autom. Softw. Eng.* **2026**, *33*, 61. [\[CrossRef\]](#)
30. Cyril, R. ADAPTIVE ZERO-TRUST ORCHESTRATION FOR HYBRID MULTI-CLOUD ENVIRONMENTS USING AI-DRIVEN BEHAVIORAL ANALYTICS. *Res. J. Educ. Sci. Technol.* **2025**, 2581–7795.
31. Pokorný, L. Optimizing AI-Driven Anomaly Detection and Automated Response Orchestration for Multi-Sector Cyber Defense: A Comparative Framework for Military, Critical Infrastructure, Corporate, and Financial Systems. Zenodo, 2026. Available online: <https://zenodo.org/records/19425158> (accessed on 20 July 2026).
32. Kulkarni, I.; Rathee, G.; Kerrache, C.A.; Jaglan, N. Adaptive ML-blockchain framework for intrusion detection and trust management in IoT networks. *Sustain. Cities Soc. Adv.* **2026**, *2*, 100055. [\[CrossRef\]](#)
33. Nagaraja, M.B.; Palavali, D.R.; Jain, A.; Hosamani, R.Y. Self-Healing Security Orchestration for Financial Fraud Systems: Autonomous Incident Response and Infrastructure Stabilization. In Proceedings of the SoutheastCon 2026, Huntsville, AL, USA, 12–15 March 2026; IEEE: Piscataway, NJ, USA, 2026; pp. 1–7.
34. Li, B.; Guo, M.; Jia, Y.; Zeng, T.; Liu, X. Design of Cloud Platform Alert Monitoring and Automatic Analysis System Based on Random Forest Algorithm. *Array* **2026**, *29*, 100694. [\[CrossRef\]](#)
35. Bhavana, D.; Chaudhari, S.; Atreya, S.C. Ensemble Learning for Resource Allocation in 5G Networks using MATLAB Simulation Environment. In Proceedings of the 2026 Third International Conference on Networking and Communications (ICNWC), Chennai, India, 7–8 April 2026; IEEE: Piscataway, NJ, USA, 2026; pp. 1–7.
36. Peng, K.; Liu, X.; Han, D.; Hu, Y.; Hu, M.; Cai, C.; Xiong, Z. Hybrid Orchestration of AI Services and Microservices in Cloud-Edge Collaboration. *IEEE Trans. Mob. Comput.* **2026**, *25*, 9382–9400. [\[CrossRef\]](#)
37. Vashisht, S.; Rani, S.; Feng, H. Multimodal and Agentic Intelligence–Driven ML Fusion for Sustainable 6G Network Slicing. *IEEE Open J. Commun. Soc.* **2026**, *7*, 1643–1653. [\[CrossRef\]](#)
38. Iqbal, M.; Nakao, A. 5ERASI: 5G Service-Enabled Resource Allocation and Sub-Slice Isolation for Open RAN. *IEEE Access* **2026**, *Early Access*.
39. Ahmed, M.; Yuan, J.S. AI-Driven Hybrid Architecture for Secure, Reconstruction-Resistant Multi-Cloud Storage. *Future Internet* **2026**, *18*, 70. [\[CrossRef\]](#)
40. Govindarajan, V.; Ahmed, F.; Faheem, Z.B.; Bilal, M.; Ayadi, M.; Ali, J. Aegis-5: A Hybrid Ensemble Framework for Intrusion Detection in Industry 5.0 Driven Smart Manufacturing Environment. *ACM Trans. Auton. Adapt. Syst.* **2026**. [\[CrossRef\]](#)
41. Boufarh, R.; Boursas, F.; Bakri, M.; Djabri, A. Factor of safety prediction for high road embankments using mixed effects random forest and bee colony optimization. *Sci. Rep.* **2026**, *16*, 6003. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Van Der Aalst, W.M. Hybrid Intelligence: To automate or not to automate, that is the question. *Int. J. Inf. Syst. Proj. Manag.* **2021**, *9*, 5–20. [\[CrossRef\]](#)
43. Lacity, M.C.; Willcocks, L.P. Becoming strategic with intelligent automation. *MIS Q. Exec.* **2021**, *20*, 7. [\[CrossRef\]](#)

44. Lee, S.C.; Nadri, C.; Sanghavi, H.; Jeon, M. Eliciting user needs and design requirements for user experience in fully automated vehicles. *Int. J. Hum.-Interact.* **2022**, *38*, 227–239.
45. Lipińska, J. *THE USE AND APPLICATION OF ARTIFICIAL INTELLIGENCE (AI) IN DECISION-MAKING PROCESSES*; Scientific Papers of Silesian University of Technology; Organization & Management/Zeszyty Naukowe Politechniki Slaskiej, Seria Organizacji i Zarzadzanie; Silesian University of Technology: Gliwice, Poland, 2024.
46. Badmus, O.; Rajput, S.A.; Arogundade, J.B.; Williams, M. AI-driven business analytics and decision making. *World J. Adv. Res. Rev.* **2024**, *24*, 616–633. [[CrossRef](#)]
47. Lai, V.; Chen, C.; Liao, Q.V.; Smith-Renner, A.; Tan, C. Towards a science of human-ai decision making: A survey of empirical studies. *arXiv* **2021**, arXiv:2112.11471.
48. Lai, V.; Chen, C.; Smith-Renner, A.; Liao, Q.V.; Tan, C. Towards a science of human-AI decision making: An overview of design space in empirical human-subject studies. In Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, Chicago, IL, USA, 12–15 June 2023; pp. 1369–1385.
49. Reverberi, C.; Rigon, T.; Solari, A.; Hassan, C.; Cherubini, P.; Cherubini, A. Experimental evidence of effective human–AI collaboration in medical decision-making. *Sci. Rep.* **2022**, *12*, 14952. [[CrossRef](#)] [[PubMed](#)]
50. Mohanasundaram, P. WAREHOUSE AUTOMATION: TRANSFORMING SUPPLY CHAINS FOR THE DIGITAL ERA. *Int. J. Inf. Technol. Manag. Inf. Syst.* **2025**, *16*, 1077–1093. [[CrossRef](#)]
51. Duggal, A.S.; Malik, P.K.; Gehlot, A.; Singh, R.; Gaba, G.S.; Masud, M.; Al-Amri, J.F. A sequential roadmap to Industry 6.0: Exploring future manufacturing trends. *IET Commun.* **2022**, *16*, 521–531.
52. Chen, Z.; Xie, M.; Zu, Q.; Abdulfattokhov, S. Electrical Automation Intelligent Control System Based on Internet of Things Technology. *Electrica* **2023**, *23*, 329–337. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.