Trusted Microservice Orchestration for Secure Edge Computing in Industrial Cyber-Physical Systems

Redowan Mahmud, Jiong Jin, Jonathan Kua, Mahbuba Afrin, Sajib Mistry, and Aneesh Krishna

Abstract

The rapid penetration of Industrial Internet of Things (IIoT) and computing technologies has significantly transformed the landscape of Industrial Cyber-Physical Systems (ICPS). Recent disruptive paradigms, such as Cloud-Fog Automation, have advocated for the open deployment of ICPS at the cloud/fog layers. These paradigms aim to enhance system modularity and efficiency while leveraging the computing capabilities of cloud/fog technologies. The vast volume and diversity of data generated by ICPS can easily overwhelm traditional cloud solutions due to bandwidth and latency challenges. Edge-based analytics offers a promising solution by processing data closer to the source, yet deploying advanced analytics at the edge is challenging due to the computational, memory, and energy limitations of edge devices. Recently, microservices have emerged as an ideal candidate for providing lightweight and modular analytic services in ICPS. However, microservices introduce severe security vulnerabilities and operational complexities, particularly in heterogeneous and dynamic industrial settings. Existing microservice orchestration platforms are designed for stable cloud environments, hence they are unable to meet the unique security and adaptability requirements of industrial edge computing. To this end, we propose a novel vulnerability-aware microservice orchestration framework for secure edge computing in ICPS. We first conceptualize a trust model that detects behavioral anomalies in microservices, then perform extensive simulations using a state-of-the-art platform to validate the effectiveness of our trust model in enhancing cyber-physical security and system resilience in decentralized industrial edge computing environments.

Introduction

The widespread adoption of Industrial Internet of Things (IIoT) technologies and computing paradigms have accelerated the architectural advancements of Industrial Cyber-Physical Systems (ICPS). Recently, a new digitalized industrial automation architecture, known as "Cloud-Fog Automation" has been proposed [1]. This new network-centric architecture disrupts the traditional International Society of Automation (ISA)-95 model by proposing the open infrastructural deployment of industrial automation systems at the cloud/fog layers. Cloud-Fog Automation aim to ultimately synergize the convergence of communications, computing, and control towards next-generation ICPS, which will significantly transform the ICPS landscape across various domains such as smart manufacturing, smart transport, smart agriculture, and so forth.

In a typical cloud/fog-based industrial automation system, it is common to have a wide array of interconnected sensors, actuators, and smart machines, which are deployed to enhance operational efficiency, reduce costs, and improve overall system productivity. For instance, IIoT devices in a modern manufacturing plant monitor equipment health in real-time, predict maintenance requirements, and adjust production schedules dynamically to minimize downtime. The data generated by ICPS is characterized by its high volume, velocity, and variety, which necessitate real-time processing to derive actionable insights. Immediate data analysis becomes even more crucial for anomaly detection, predictive maintenance, and adaptive control systems. However, it is inadequate to rely solely on cloud-based computing resources for real-time processing due to latency challenges and bandwidth constraints. Transmitting vast amounts of data to centralized servers for processing further adds unacceptable delays in time-sensitive and mission-critical applications for industrial operations [2].

The advent of edge computing has popularized edge-based data analytics, which is an attractive solution to address the limitations inherent to cloud computing by bringing computation closer to data sources. Industrial applications and services achieve lower latency by processing data at the network's edge (on devices or local servers) with reduced bandwidth usage and enhanced data flow controls. Edge computing enables real-time analytics and decision-making, which significantly elevate the efficiency of ICPS. However, implementing advanced data analytics at the edge also introduces significant challenges, primarily due to resource and energy constraints in edge devices [3]. These devices often possess limited computational power, memory, and energy

Digital Object Identifier: 10.1109/MNET.2025.3541032
Date of Current Version: 15 July 2025
Date of Publication: 11 February 2025

Redowan Mahmud, Mahbuba Afrin, Sajib Mistry, and Aneesh Krishna are with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, WA 6102, Australia; Jiong Jin (corresponding author) is with the School of Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia; Jonathan Kua is with the School of Information Technology, Deakin University, Geelong, VIC 3220, Australia.

resources, making them difficult to host complex analytics that are traditionally hosted on more robust and powerful cloud servers.

Microservices offer a promising approach to overcome the limitations of edge computing by decomposing applications into smaller, independent services that can be readily deployed, scaled, and managed individually [4]. This modular approach aligns well with the resource-constrained nature of edge devices, allowing for more efficient utilization of available resources and facilitating continuous deployment and scalability of services. However, adopting microservices at the industrial edge introduces new security vulnerabilities and complexities in ICPS. The heterogeneity of edge devices (from simple sensors to complex machineries) requires microservices to operate across diverse hardware and software environments, thus complicating deployment and operations [5]. Dynamic network topologies (e.g., equipment that frequently connects and disconnects to/from the network) further add to the challenge by causing fluctuations in service availability. The distributed and ever-changing environment expands the attack surface, making microservices susceptible to cyber-physical security threats such as unauthorized access or malware infiltration. Therefore, the effective coordination and orchestration of microservices is paramount in ICPS.

Motivation

Security breaches in ICPS can lead to catastrophic consequences, such as endangering personnel, causing significant equipment damage, damaging organizational reputation and impacting revenues. The interconnected nature of modern industrial automation systems means that a single vulnerability in a microservice or an edge node can compromise the entire network or lead to a damaging cascading effect across the entire system. A notable example is the catastrophic 2021 Colonial Pipeline ransomware attack incident, where cyber-criminals accessed critical control systems and shut down the largest fuel pipeline in the United States. This led to widespread fuel shortages and significant economic disruption, illustrating how a single compromise can dramatically impact interconnected industrial systems on a large scale [6].

However, traditional security measures often fall short of microservice orchestration for edge computing in ICPS as they are tailored for centralized, monolithic architectures with clear boundaries, and they rely on perimeter defenses such as firewalls [1]. Additionally, edge computing facilitates data processing in a remote/less secure environment, rendering centralized security approaches ineffective, and exposing edge nodes to physical tampering and resource constraints which limits the implementation of robust security measures. The shift to microservices also exponentially increases the attack surface, as each independent service communicating over networks becomes a potential entry point for attackers [7]. The evolving threat landscape, as characterized by advanced attack techniques and frequent changes to microservices and edge nodes, further undermines traditional security measures.

The data generated by ICPS is characterized by its high volume, velocity, and variety, which necessitate real-time processing to derive actionable insights.

Therefore, there is a pressing need to investigate and implement a vulnerability-aware trusted microservice orchestration framework in ICPS infrastructure. This approach must integrate security directly into the orchestration process, proactively addressing vulnerabilities and establishing trust among the various interconnected components. Through continuous monitoring, this approach should be able to detect and address potential threats before they are being exploited. The orchestration framework needs to ensure that only verified and authorized entities have access to the network. This includes employing robust authentication and authorization mechanisms, secure communication protocols, encryption, and adopting zero-trust security models. These measures collectively fortify the security framework, enhance system resilience and reliability, and defend against the evolving landscape of security threats within the smart industrial ecosystem.

Existing Works and Limitations

Microservice orchestration platforms such as Kubernetes, Docker Swarm, and Apache Mesos have gained popularity due to their automation capabilities in application deployment, scaling, and management. Although these platforms performed excellently in stable, resource-abundant cloud computing environments, their effectiveness is challenged in dynamic and resource-scarce environments, such as edge computing in ICPS. For example, Kubernetes stands out for its robust automation, yet it struggles in edge computing due to its reliance on consistent network conditions, often lacking in volatile settings [8]. Despite security extensions, such as network policies and service meshes, Kubernetes still fails to manage trust among dispersed microservices, which ensures ICPS reliability. While Docker Swarm is user-friendly due to its straightforward clustering and scheduling capabilities, it falls short in edge environments that demand high security. It lacks the necessary features to safeguard against threats in decentralized networks. Although Apache Mesos is adept at managing resources in large-scale data centres, it similarly fails to offer the quick adaptability required for the unpredictable nature of industrial edge computing [9]. Additionally, these frameworks' built-in current orchestration policies do not support a zero-trust security model, which is essential for verifying all access requests in highly dynamic and vulnerable networks. They also lack integrated encryption and privacy protection mechanisms, both during data transit and at rest, exposing sensitive information to potential interception [8].

The integration of contemporary security measures in edge computing significantly impacts the flexibility of microservices architectures. For example, the use of sidecar proxies within secure service meshes introduces additional computational and memory overhead, which leads to higher latency, and constraints their ability to rapidly adapt in dynamic environments [10]. Similarly, the embedding of identity verification and

compliance checks into CI/CD pipelines of microservice orchestration reduces the modularity and independent deployment capabilities of microservices, which subsequently results in slower deployment cycles and increased management complexities in heterogeneous edge environments [5]. Hardware-based security modules on edge nodes require customization to facilitate microservice orchestration, which incur additional overhead in their deployment, portability, and replication. Additionally, there is a notable deficiency in real-time security monitoring and anomaly detection in hardware-assisted security protocols for microservices. The aforementioned limitations impede timely anomaly detection in edge computing-enabled ICPS, leaving them vulnerable to attacks that exploit undetected weaknesses in microservices [7].

As edge environments are susceptible to rapid security threats, they require continuous oversight and monitoring while orchestrating microservices on unreliable computing nodes. Considering the limitations of existing frameworks and techniques in securing multi-tenant edge environments, where infrastructure sharing among multiple stakeholders demands stringent isolation, this necessity becomes even more critical. Our proposed solution addresses such a pivotal issue by making innovative contributions as follows.

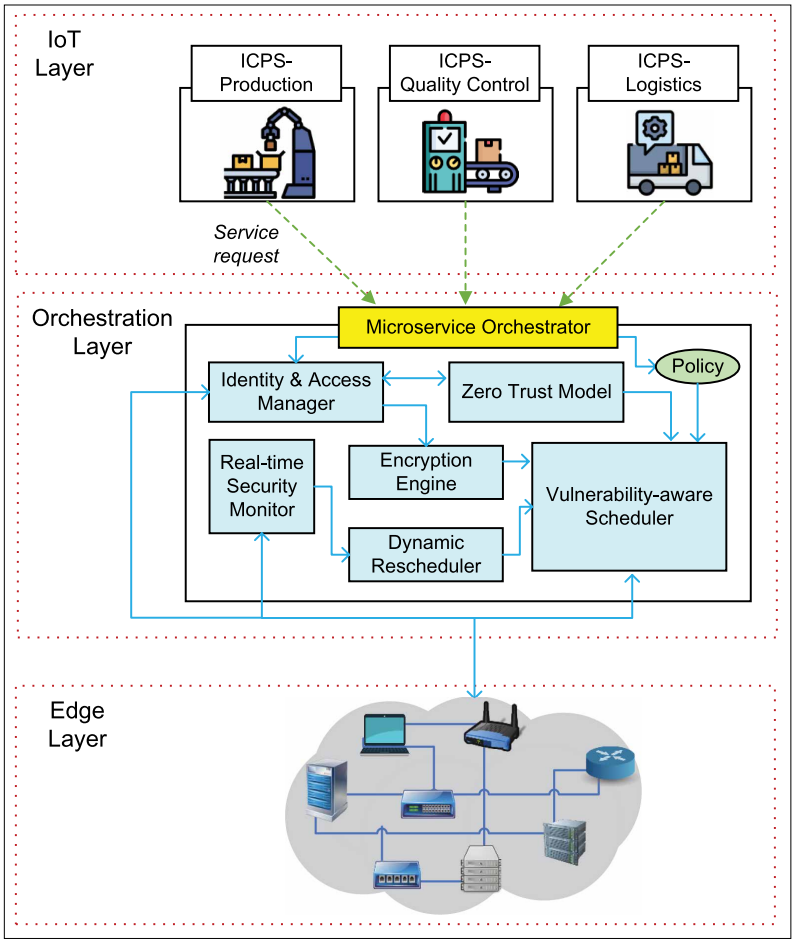

FIGURE 1. System overview of our proposed trusted microservice orchestration framework, illustrating the seamless interactions across the IoT, orchestration, and edge layers. The *IoT layer* initiates service requests from distinct ICPS operations, including Production, Quality Control, and Logistics. The *Orchestration layer* is embodied on a Microservice Orchestrator. It manages service requests using Identity and Access Management, Real-time Security Monitoring, and Encryption strategies. These modules are underpinned by a Zero-trust security model. The *Edge layer* supports the network infrastructure for data flow and processing throughout the system.

Our Contributions

In this paper, we propose a novel architectural framework for vulnerability-aware and trusted microservice orchestration for industrial edge computing. Our framework is equipped with defined functionalities tailored to address the unique challenges of ICPS. It is designed to enhance the secure deployment and management of microservices on resource-constrained edge nodes, ensuring robust performance while mitigating security risks inherent to decentralized networks.

At the core of our framework is a trust model, which we have designed to quantify behavioral anomalies in microservices. This model enables real-time monitoring and sanitization of microservice interactions by detecting deviations from normal behavioral patterns. The system can proactively identify and respond to potential security threats by assessing the trustworthiness of each microservice, thereby strengthening the overall ICPS security. We have extensively simulated the embodiment of our trust model within the orchestration framework, and evaluated its performance using a state-of-the-art platform. Simulation results demonstrated the efficacy of our approach in enhancing security monitoring and threat mitigation within edge computing-enabled ICPS scenarios.

In summary, the key contributions of this paper are:

- Proposed a novel security architecture for microservice orchestration in industrial edge computing which includes critical functions for enhanced security monitoring, threat detection, and system resilience.
- Designed a trust model for microservices that effectively quantifies behavioral anomalies, which enables continuous monitoring to uphold system integrity and security.
- Performed an extensive empirical study to validate our trust model's effectiveness, and demonstrated its practical application in anomaly quantification within our trusted microservice orchestration framework.

The rest of this paper is organized as follows. The section "Proposed Framework" presents the system architecture of our vulnerability-aware microservice orchestration framework. The section "Trust Model" presents our trust model for rapid anomaly detection, and the section "Performance Evaluation" evaluates the performance of our trust model and orchestration framework. Finally, the section "Conclusion and Future Work" concludes the paper.

Proposed Framework

Our proposed trusted microservice orchestration framework addresses the cyber-physical security challenges of ICPS through a cohesive three-layer architecture, as presented in Fig. 1. The *IoT layer* generates vast amounts of data from numerous interconnected devices. When data processing is

required, IoT devices send requests to the *orchestration layer*, signaling the need to launch specific microservices on the edge nodes situated at the *edge layer*. This seamless interaction ensures that data is handled promptly and efficiently, thus maintaining real-time responsiveness which is essential for mission-critical ICPS applications.

The orchestration layer manages the deployment of microservices based on requests from the IoT layer. It assesses the capacity and current load of the available edge nodes to determine the optimal placement of microservices. Once a suitable edge node is identified, the orchestration layer deploys the necessary microservices to process the incoming IoT data. Furthermore, the orchestration layer monitors the entire life-cycle of the deployed microservices, ensuring they operate in a trusted and desired manner. The edge layer is equipped with virtualized physical resources to offer flexible and scalable processing power. Data analytics are executed here as microservices, with the orchestration layer efficiently scheduling these services to ensure optimal performance and resource utilization. This layer reduces the need for extensive data transmission to centralized cloud servers by processing data at the edge, which directly optimizes the network bandwidth consumption and application service delays.

As indictated in Fig. 1, the orchestration layer serves as the "brain" of the system in our proposed framework. The multi-faceted functions of this layer include scheduling and resource management, security enforcement, and continuous monitoring. The details of each function are presented below.

- **Identity and Access Manager (IAM):** IAM is critical for controlling and managing access to microservices and sensitive data. The orchestration layer employs IAM protocols to ensure that both IoT devices and edge nodes have permissions aligned with their designated functions, such as requesting microservice initiation and transmitting data in a compliant format. Specifically, IoT devices must authenticate and authorize their access to edge services to retrieve necessary data, while edge nodes require appropriate permissions to receive and process data from IoT devices. This bi-directional interaction enforces device-level granular control between the edge layer and IoT devices, thus maintaining data confidentiality and integrity by preventing unauthorized access.
- **Zero-Trust Model:** The functions of IAM are enforced by integrating a zero-trust model that enables the orchestration layer to operate under the assumption that no entity (neither a service nor device) is inherently trustworthy. Every access request undergoes rigorous authentication and authorization processes, ensuring that only verified entities can interact with the microservices and data. This stringent security policy can effectively foster a secure industrial operational environment by mitigating the risks of unauthorized access and potential security breaches.
- **Vulnerability-Aware Scheduler:** The vulnerability-aware scheduler is pivotal in maintaining system performance amid the emerging threat landscape. It schedules microservices and data by assessing real-time vulnerability data alongside resource availability. This proactive approach ensures that microservices are deployed on nodes that not only have the necessary computational capacity but also comply with the current security standards, such as the use of firewalls and packet sniffers. The scheduler minimizes the risk of deploying services on compromised or under-resourced nodes by continuously evaluating potential vulnerabilities. It also enables isolated environments for different services during deployment, so that security threats to one microservice do not adversely affect other microservices.
- **Encryption Engine:** Data privacy is safeguarded through an advanced encryption engine that implements end-to-end encryption for data both in transit and at rest. The orchestration layer ensures that sensitive information remains protected against interception by encrypting data across multiple stages, even if the data traverses insecure networks or stored on vulnerable nodes.
- **Real-Time Security Monitoring:** To promptly identify and address security threats, the orchestration layer integrates real-time security monitoring tools such as intrusion and anomaly detection algorithms. These tools continuously analyze system behaviors and network traffic patterns to detect suspicious activities or deviations from normal operations. Upon identifying potential threats, the system can initiate automated responses to mitigate the associated risks and reassess the credibility of the affected services, ensuring the ongoing security of critical industrial infrastructure.
- **Dynamic Rescheduling Mechanism:** In an ever-changing environment, conditions such as fluctuating workloads and on-the-fly cyber-physical security threats necessitate adaptive resource management as part of the security response. The dynamic rescheduling mechanism responds to these changes by relocating microservices as needed. For instance, if a vulnerability is detected in a particular edge node, the mechanism can promptly shift affected microservices to several more secure or less-burdened nodes. This flexibility ensures sustained performance and security, even in the face of unexpected disruptions.
- **Policy Enforcement:** The orchestration layer meticulously enforce the compliance with security policies with varying security priorities of different industrial settings. It applies tenant and industry-specific security policies to dictate access controls, data handling procedures, and operational protocols. By ensuring adherence to these policies, the orchestration layer reduces vulnerabilities and ensures that each service operates within defined industrial security parameters, thereby maintaining overall system compliance and integrity.

Through the aforementioned functions, the orchestration layer ensures robust cyber-physical security when deploying microservices. These

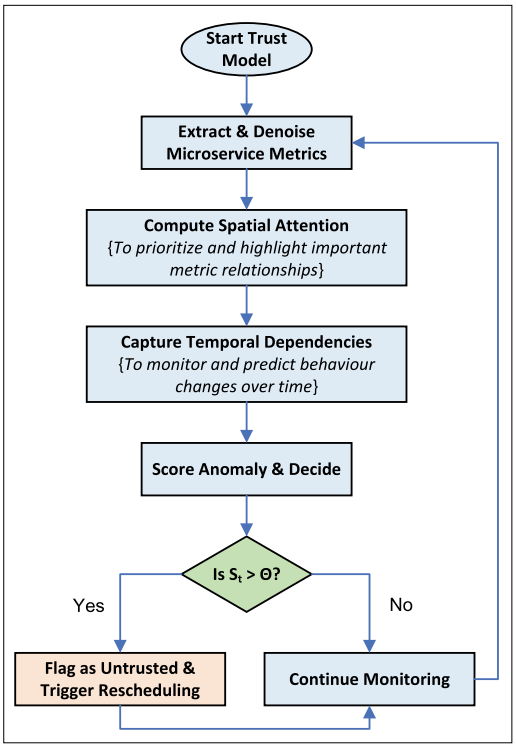

FIGURE 2. Procedures for assessing the trustworthiness of a microservice using our trust model within the proposed orchestration framework.

functions are fundamental to the framework's ability to efficiently and securely manage/process ICPS data. Moreover, these functions are also individually extensible to suit industry-specific needs. In the following section, we present an illustrative example of extending the *Real-time Security Monitoring* function, and propose a trust assessment model specifically designed for anomaly detection at the orchestration layer.

Trust Model

Anomalous behaviors in microservices can lead to severe consequences, such as software bugs, configuration errors, and security breaches; which then pose significant security risks, such as data integrity violations, extended application execution times, and functional safety issues. Therefore, microservices deployed within ICPS require robust mechanisms for continuous trust evaluation to quantify and mitigate these anomalies, and ensuring reliable operation. However, accurate anomaly quantification requires a comprehensive understanding of each microservice's behavioral patterns, which can vary widely depending on the industrial setting and application use case. Hence, an orchestration layer that captures these nuances through a trust assessment model is required. Below we introduce and develop a trust model that address these requirements. We leverage a long-short-term memory (LSTM) network and attention mechanisms to quantify behavioral anomalies in microservices.

Extraction of Microservice-Specific Metrics

As depicted in Fig. 2, our trust assessment model initiates quantifying anomalous microservice behaviors by investigating microservice-specific metrics such as CPU and memory usage, network traffic patterns, and system calls. It involves updating the feature extraction process to incorporate these relevant data points, ensuring that the analysis is aligned with the unique behaviors of microservices. The feature vector $\mathbf{X_t}$ at time t, encapsulates all the essential metrics that characterize microservice behaviors and provide a comprehensive input for subsequent analysis. This vector can be represented as:

$$\mathbf{X_t} = \begin{bmatrix} \text{CPU\_Usage} \\ \text{Memory\_Usage} \\ \text{Network\_Traffic} \\ \text{System\_Calls} \end{bmatrix}$$

It is important to note that given the dynamic nature of ICPS, it is crucial to filter out noise within $\mathbf{X_t}$ before using it as an input to the trust assessment model. Techniques such as denoising autoencoders or noise-robust loss functions (e.g., *correntropy* [11]), are effective in mitigating the impact of sporadic loads on $\mathbf{X_t}$, transforming $\mathbf{X_t}$ into a denoised feature vector $\tilde{\mathbf{X}}_\mathbf{t}$.

Integration of Spatial Attention

The incorporation of a spatial attention mechanism is required to effectively manage the interactions between different microservice metrics, while determining their behavioral anomaly-driven trustworthiness. By adjusting the attention unit to evaluate both *temporal* (time-based) and *spatial* (feature-based) relationships, the model gains a more comprehensive understanding of how various metrics influence each other over time and across different services (attention score) [12]. This dual evaluation enhances the model's ability to identify complex misbehaviors that arise from the interplay between multiple microservice metrics, which are often overlooked if only one type of relationship is considered. The relationship is represented through a Spatial Attention Matrix $\mathbf{A_t}$, which can be mathematically expressed as:

$$\mathbf{A_t} = \text{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{d_k}}\right)\mathbf{V} \quad (1)$$

Here, $\mathbf{Q}$ (Query), $\mathbf{K}$ (Key), and $\mathbf{V}$ (Value) are matrices derived from the denoised features $\tilde{\mathbf{X}}_\mathbf{t}$, representing different projections of the input data that the model uses to compute attention scores. Specifically, $\mathbf{Q}$ is derived from the CPU_Usage and Memory_Usage metrics, providing insights into the processing and memory demands of each microservice and serving as queries that seek relevant information from other metrics. $\mathbf{K}$ is constructed using Network_Traffic and System_Calls data, representing operational interactions and system-level activities that serve as keys against which the queries match to determine relevance. Similarly, $\mathbf{V}$ also stems from Network_Traffic and System_Calls, containing the actual information that will be aggregated based on the attention scores derived from $\mathbf{Q}$ and $\mathbf{K}$. The dimensionality of the key vectors, denoted by d_k, is used to scale the dot product between $\mathbf{Q}$ and $\mathbf{K}$ by dividing by the square root of d_k, which prevents the softmax function from becoming too flat when dealing with

high-dimensional data and maintains the gradients within a manageable range for a more effective learning process. After scaling, the softmax function transforms these attention scores into a probability distribution that sums to one, effectively highlighting the most relevant feature interactions. This allows the model to assign varying levels of importance to CPU_Usage, Memory_Usage, Network_Traffic, and System_Calls, thereby focusing on the interactions that are most indicative of system behavior and potential anomalies.

Capturing of Temporal Dependencies

In ICPS environments, microservices often exhibit repetitive and interdependent behaviors, making it crucial to monitor and predict these patterns for effective anomaly detection. Therefore, our proposed trust assessment model introduces a Temporal Dependencies Network (TDN) specifically designed to capture and analyze how these behaviors evolve over time [13]. When combined with a Spatial Attention mechanism that identifies the key features contributing to potential anomalies, this capability plays a crucial role in detecting deviations from expected behavioral patterns. Advanced techniques such as cyclical LSTM layers and cyclic learning rates are incorporated to further refine the detection capabilities of TDN. Cyclical LSTM layers are adept at retaining information over extended periods, which is vital for recognizing long-duration patterns and dependencies in microservice behaviors. Cyclic learning rates optimize the training process by varying the learning speed in a cyclical pattern, allowing the network to adapt more effectively to new data and avoid the problem of overfitting. In our proposed trust assessment model, the refined Temporal Dependency is expressed using the following equation for a simple LSTM cell:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{A}_t \cdot \tilde{\mathbf{X}}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \quad (2)$$

Here, $\mathbf{h}_t$ represents the learned temporal dependencies up to time t, captured based on the attention-weighted features $\mathbf{A}_t \cdot \tilde{\mathbf{X}}_t$ and the previous short-term $\mathbf{h}_{t-1}$ and long-term $\mathbf{c}_{t-1}$ dependencies. This formulation enables our model to effectively track and analyze temporal patterns, thus enhancing the reliability of the anomaly detection process.

Anomaly Scoring for Microservices

The anomaly scoring mechanism in our proposed trust model is specifically designed to capture the dynamic nature of microservice in ICPS environments. It evaluates anomalies by measuring deviations from typical microservice behavioral profiles, which are continuously learned and updated to reflect changes in deployment and usage patterns. The anomaly score S_t is calculated as a reconstruction error, as follows:

$$S_t = \|\mathbf{h}_t - \mathbf{h}_{\text{typical}}\|_2^2 \quad (3)$$

Here, $\mathbf{h}_{\text{typical}}$ represents the baseline or expected behavior of the microservice, while $\mathbf{h}_t$ captures the current learned temporal dependencies of the microservice. The Euclidean norm $\|\cdot\|_2$ quantifies the distance between these two vectors. A higher anomaly score S_t indicates a greater deviation from normal behavior, signaling a potential anomaly.

A proactive assessment of microservice trustworthiness with anomaly scores is important in ICPS.

The orchestration layer uses this anomaly score S_t to quantify the trustworthiness of individual microservices within the system. The orchestration layer determines whether a microservice is operating within its expected behavioral parameters by continuously monitoring its value. Specifically, if the anomaly score S_t exceeds a pre-defined threshold θ, the microservice is flagged as potentially untrusted. In response to such detection, the orchestration layer triggers its *Dynamic Rescheduling Mechanism* to immediately apply the corresponding mitigation strategies.

The orchestration layer continuously assesses the trustworthiness of each microservice by monitoring its anomaly score S_t, which reflects behavioral deviations such as unexpected resource usage, latency, or error rates. When S_t remains below a pre-defined threshold θ, the microservice is deemed trustworthy and operates within normal parameters. However, if S_t exceeds θ, the orchestration layer flags the microservice as potentially untrusted, signaling potential issues such as software bugs, security breaches, or performance anomalies. For example, a sudden spike in CPU usage could indicate that the microservice is compromised - perhaps due to a memory leak, malicious code execution, or an attacker exploiting vulnerabilities to consume resources, leading to system corruption or increased vulnerability.

Upon detecting an untrusted microservice, the trust model activates the *Dynamic Rescheduler* to implement immediate mitigation strategies. Consider a distributed application with billing, recommendations, and user management services. If the billing service suddenly experiences a spike in CPU usage, potentially indicating corruption or vulnerability, its anomaly score is anticipated to surpass the threshold. In response, the orchestration layer could migrate the billing workload to a backup instance, sandboxing the service to prevent cascading failures and initiate security protocols if a breach is suspected. This rescheduling ensures the system remains resilient by quickly addressing potential threats or performance issues. To ensure practicality, our orchestration layer further employs continuous inspection through *Real-time Security Monitor,* which reassesses microservices after rescheduling. In cases of false positives, this ongoing oversight allows for the rapid reintegration of microservices once they stabilize, maintaining overall system reliability.

A proactive assessment of microservice trustworthiness with anomaly scores is important in ICPS. It detects deviations from expected behaviors, which is crucial in preventing subtle failures in distributed systems that can cause significant problems. The anomaly score triggers deeper investigations or automated safeguards and creates a more balanced and comprehensive assessment when combined with factors like performance overhead, latency, and reliability. Furthermore, leveraging anomaly-based trust assessments for dynamic rescheduling mitigates risks and provides scope for service continuity

Component	Hyperparameters
Denoising Autoencoder	Encoder Layers: 3, Decoder Layers: 3, Activation: ReLU, LR: 0.001, Batch: 64, Epochs: 50
LSTM Attention Model	LSTM Layers: 2, Hidden Units: 128, Attention Heads: 4, Dropout: 0.3, LR: 0.0005, Batch: 32, Epochs: 80
Traditional LSTM Model	LSTM Layers: 2, Hidden Units: 128, Dropout: 0.3, LR: 0.001, Batch: 32, Epochs: 100
Isolation Forest	Trees: 100, Contamination: 0.05
Training Parameters	Optimizer: Adam, Validation Split: 0.2

TABLE 1. Experiment Hyperparameters.

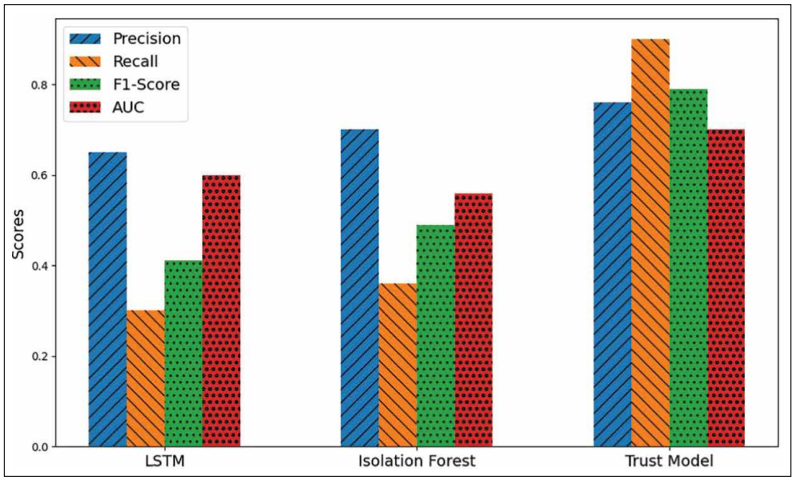

FIGURE 3. Performance comparison of our trust model with LSTM and isolation forest for anomaly detection in microservices. Our model outperforms LSTM and isolation forest across all four metrics (precision, recall, F1-score, AUC).

Category	Parameter configuration
Edge nodes	Count: 10, CPU: 1000-2000 MIPS, Memory: 4-8 GB, Bandwidth: 100-200 Mbps
IoT device	Count: 100 devices, Data Rate: 2 Mbps
Microservice A	Count: 10 , Instructions: 100-200 million, Size: 0.5-1 GB, Function: Data preprocessing
Microservice B	Count: 6 , Instructions: 500-800 million, Size: 2-4 GB, Function: Data aggregation
Microservice C	Count: 4, Instructions: 1000-1200 million, Size: 6-8 GB, Function: Data analysis

TABLE 2. Simulation Parameter Configurations.

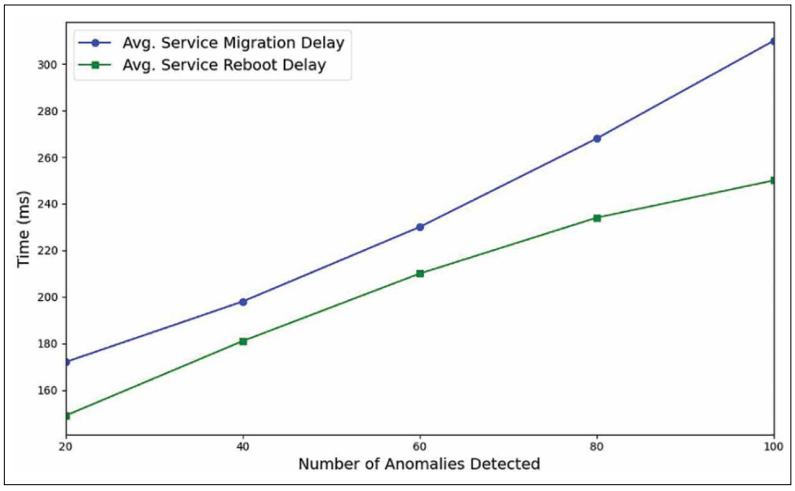

FIGURE 4. The applicability of our proposed trust model in dynamic rescheduling, comparing the average service delays required for migration and reboot when detecting the same number of anomalies.

through reassessment. This integration boosts system resilience and minimizes unnecessary downtime, providing a robust framework for maintaining and enhancing the trustworthiness of microservice-based ICPS.

Performance Evaluation

To rigorously assess the performance of our proposed microservice orchestration framework, we designed and executed two distinct experimental scenarios. Firstly, we evaluated the effectiveness of our proposed trust model in accurately detecting and quantifying anomalies within microservices. Secondly, we explored the efficiency of our trust model in initiating and executing mitigation responses to these anomalies. Below we present the detailed discussions of both experimental setups and their performance evaluation.

Benchmarking Anomaly Detection Performance

In our first experimental scenario, we evaluate the accuracy and effectiveness of our trust model, which uses a LSTM network enhanced with an attention mechanism for detecting anomalies in microservice orchestration. The LSTM network captures long-term temporal dependencies, while the attention mechanism focuses on the most relevant inputs, improving both interpretability and performance. We compare our approach against two baselines: (i) a traditional LSTM model (without attention); and (ii) Isolation Forest, which is a statistical method that partitions the feature space to isolate anomalies. We articulate our comparisons against these two baselines for clarity, and highlight the importance of the attention mechanism. Traditional LSTM represents a well-established neural baseline, while Isolation Forest offers a contrasting, non-neural anomaly detection reference point. This in-depth comparison directly assesses the value of incorporating an attention mechanism in our trust model.

Additionally, we utilize the "Anomaly Detection in Microservices-based Systems" dataset in our experiments [14]. This dataset comprises both normal and anomalous microservice data related to shipment tracking, payment processing, and order latency in a smart industrial setting. The recorded metrics span a 1-hour window over the course of 15 days, and include critical performance indicators such as response times, spikes in CPU usage and abnormal memory consumption. Each data entry also includes an indicator signaling the presence of anomalies. During training, about 70% of the data represents normal operations to help the model learn standard behavior, while the remaining 30% consists of anomalies to ensure it can detect deviations. This balance aids in developing a model that accurately identifies both normal and abnormal conditions, enhancing the security and efficiency of the cyber-physical system. Nevertheless, prior to feeding the data into the models, denoising autoencoders are applied to clean the data and ensure it aligns with the input requirements of the trust model. We have further ensured that each model is trained on a specific portion of the dataset, which includes normal and anomalous behaviors. A separate validation set is used for hyperparameter tuning

to avoid overfitting. Table 1 presents the experiment hyperparameters.

As depicted in Fig. 3, the evaluation of model performance is based on several metrics, including precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). *Precision* measures the proportion of correctly identified anomalies among all detected anomalies, while *recall* assesses the model's ability to identify all actual anomalies. The F1-score balances precision and recall, and ROC-AUC evaluates the trade-off between true positive and false positive rates. Detailed result analysis indicates that our proposed trust model outperforms baseline models across all metrics, particularly in precision and recall. The incorporation of spatial attention and temporal dependencies consideration significantly enhance the model's ability to detect complex and interdependent anomalies more effectively than simpler baseline models, leading to a more accurate assessment of microservices trustworthiness.

Simulated Environment Deployment and Impact Analysis

In our second experimental scenario, we assessed the effectiveness of our proposed trust model within a simulated microservice orchestration environment using the *iFogSim2* simulator [15]. The strategic decision to utilize iFogSim2 in our research was driven by its built-in orchestration model that adeptly manages microservices deployment across multi-tier computing infrastructures. This model enables placement policies to dynamically scale microservices among federated Edge/Fog nodes dynamically, thereby improving resource utilization. Additionally, the simulator's extensible modular architecture allows the integration of customized service discovery and load-balancing policies to simulate dynamic microservice behaviors, which is not available in other simulators. Leveraging this feature of iFogSim 2, we embedded our trust model into the simulator's service management framework, enabling dynamic responses to realistic operational anomalies for ICPS environments. The simulation parameters are presented in Table 2. Furthermore, we introduced anomalous scenarios during the simulations by mimicking resource exhaustion, packet losses, abrupt service terminations, and unauthorized system calls. Our trust model is equipped to calculate anomaly scores for each microservice by processing these metrics. When the anomaly scores exceed pre-defined thresholds, it can also prompt the service management framework to activate dynamic rescheduling mechanisms, such as relocating or restarting affected services.

The performance of the orchestration layer is evaluated by measuring the delay in system restoration under varying mitigation strategies. Fig. 4 shows that as the frequency of anomalous incidents increases, the average service migration time also increases. This increase is attributed to the need to select a trusted node and migrate the service from its last checkpoint, and it is heavily dependent on the underlying network conditions. Nonetheless, this mitigation strategy ensures that intermediate results are preserved despite cyber-physical security threats. However, when compared with service migration, restarting services as a response to anomalies results in quicker system restoration since it reboots services directly on a trusted node, albeit at the cost of losing intermediate data. Therefore, it is crucial to establish policies tailored to specific ICPS needs to determine the appropriate mitigation measures.

Extensive simulations using realistic parameters and real-world datasets demonstrate significant improvements in system security and reliability, underscoring our framework's potential to strengthen edge-based microservices in ICPS.

Conclusion and Future Work

The increasing popularity of deploying microservices in cloud/fog-enabled ICPS, such as in the new Cloud-Fog Automation architecture, has prompted our research efforts into investigating the corresponding cyber-physical security implications and mitigation strategies. In this paper, we addressed the security challenges inherent to edge-based microservices by proposing a novel trusted vulnerability-aware microservice orchestration framework. Our proposed framework incorporates a trust model to dynamically detect and respond to behavioral anomalies, which directly enhances the system resilience of decentralized operations in industrial edge computing. Extensive simulations using realistic parameters and real-world datasets demonstrate significant improvements in system security and reliability, underscoring our framework's potential to strengthen edge-based microservices in ICPS.

Future work will focus on testing and deploying our microservice orchestration framework in real-world production ICPS, devising robust identity verification mechanisms, and incorporating an adaptive zero-trust model that responsively adjusts to emerging threats to ensure sustained protection. In anticipation of advancements in quantum computing, it is also crucial to investigate and incorporate quantum-secure encryption techniques within our microservice orchestration layer to safeguard against potential post-quantum/quantum-enabled security attacks in future ICPS infrastructures.

Furthermore, deploying real-time security monitoring tools at the orchestration layer presents several key challenges. For example, rule-based tools require constant updates and are more likely to miss sophisticated attacks, while AI models demand significant computational resources and expertise, leading to scalability and latency issues. Embedding these solutions as stand-alone edge services further complicates operations, as they necessitate consistent performance across distributed nodes, efficient resource management, and strong data privacy and security. Therefore, future research should investigate streamlined update synchronization with reliable management across edge devices to enhance the feasibility and effectiveness of edge-based security monitoring in ICPS.

References

[1] J. Jin et al., "Cloud-fog automation: Vision, enabling technologies, and future research directions," *IEEE Trans. Ind. Informat.*, vol. 20, no. 2, pp. 1039–1054, Feb. 2024.

[2] M. Alam et al., "SDN-based reconfigurable edge network architecture for industrial Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 18, pp. 16494–16503, Sep. 2023.
[3] M. Afrin et al., "Dynamic task allocation for robotic edge system resilience using deep reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 3, pp. 1438–1450, Mar. 2024.
[4] J. Oliveira et al., "Microservices in edge and cloud computing for safety in intelligent transportation systems," in *Proc. IEEE Netw. Oper. Manage. Symp.*, May 2024, pp. 1–7.
[5] P. Dakić, "Software compliance in various industries using CI/CD, dynamic microservices, and containers," *Open Comput. Sci.*, vol. 14, no. 1, Jul. 2024, Art. no. 20240013.
[6] J. Hurley, "Zero trust is not enough: Mitigating data repository breaches," in *Proc. 18th Int. Conf. Cyber Warfare Secur. (ICCWS)*, 2023, pp. 1–8.
[7] M. You et al., "Hyperion: Hardware-based high-performance and secure system for container networks," *IEEE Trans. Cloud Comput.*, vol. 12, no. 3, pp. 844–858, Jul. 2024.
[8] A. Alamoush and H. Eichelberger, "Open source container orchestration for industry 4.0—Requirements and systematic feature analysis," *Int. J. Softw. Tools Technol. Transf.*, vol. 26, pp. 1–24, Sep. 2024.
[9] L. M. A. Qassem et al., "Containerized microservices: A survey of resource management frameworks," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 4, pp. 3775–3796, Aug. 2024.
[10] Y. Chen et al., "On practitioners' concerns when adopting service mesh frameworks," *Empirical Softw. Eng.*, vol. 28, no. 5, p. 113, Sep. 2023.
[11] Y. Zhang, Z. Fang, and J. Fan, "Generalization analysis of deep CNNs under maximum correntropy criterion," *Neural Netw.*, vol. 174, Jun. 2024, Art. no. 106226.
[12] Z. Li et al., "STADE-CDNet: Spatial–temporal attention with difference enhancement-based network for remote sensing image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5611617.
[13] B. Jiang et al., "Dynamic temporal dependency model for multiple steps ahead short-term load forecasting of power system," *IEEE Trans. Ind. Appl.*, vol. 60, no. 4, pp. 5244–5254, Jul. 2024.
[14] J. Nobre, E. S. Pires, and A. Reis, 2023, "Anomaly detection in microservices-based systems," doi: 10.6084/m9.figshare.22726298.v1
[15] R. Mahmud et al., "IFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *J. Syst. Softw.*, vol. 190, Aug. 2022, Art. no. 111351.

Biographies

Redowan Mahmud (Member, IEEE) (mdredowan.mahmud@curtin.edu.au) received Ph.D. degree from the School of Computing and Information Systems, The University of Melbourne, in 2020. He is currently a Senior Lecturer in computing discipline with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University. His primary research areas are fog/edge computing, the secure Internet of Things, software-defined networking, and mobile cloud computing. He is one of the main contributors to the iFogSim simulator, FogBus framework, and Con-Pi software systems, which are used extensively for resource management research in fog/edge computing.

Jiong Jin (Member, IEEE) (jiongjin@swin.edu.au) received the B.E. degree (Hons.) in computer engineering from Nanyang Technological University, Singapore, in 2006, and the Ph.D. degree in electrical and electronic engineering from The University of Melbourne, Australia, in 2011. He is currently a Full Professor with the School of Engineering, Swinburne University of Technology, Melbourne, Australia. His research interests include network design and optimization, edge computing and intelligence, robotics and automation, and cyber-physical systems and the Internet of Things as well as their applications in smart manufacturing, smart transportation, and smart cities. He was recognized as an Honourable Mention in the AI 2000 Most Influential Scholars List in IoT in 2021 and 2022. He is an Associate Editor of IEEE Transactions on Industrial Informatics and IEEE Transactions on Network Science and Engineering.

Jonathan Kua (Member, IEEE) received the B.Eng. degree (Hons.) in telecommunications and network engineering and the Ph.D. degree in telecommunications engineering from the Swinburne University of Technology, Australia, in 2014 and 2019, respectively. He is currently a Senior Lecturer in Internet of Things with the School of Information Technology, Deakin University, Australia. His research interests include low-latency data transport protocols, adaptive streaming, content delivery, the Internet of Things, and industrial cyber-physical systems, with an overarching focus on improving their network performance and quality of service.

Mahbuba Afrin (Member, IEEE) received the Ph.D. degree from the Swinburne University of Technology. She was a Vice-Chancellor's Post-Doctoral Research Fellow at the University of Southern Queensland. She is currently a Lecturer with the School of Electrical Engineering, Computing and Mathematical Sciences (EECMS), Curtin University, Australia. With expertise in networked robotics and the Internet of Things-enabled Cyber-Physical Systems, her research focuses on developing efficient learning models for intelligent systems.

Sajib Mistry (Member, IEEE) received the B.S. and M.S. degrees in computer science from the University of Dhaka, Bangladesh, and the Ph.D. degree from RMIT University. He was a Post-Doctoral Fellow at The University of Sydney. He is currently a Senior Lecturer with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University. His research spans service computing, cloud/edge computing, machine learning, augmented reality, and the IoT. He actively participates in academic committees and is a member of the Sydney IoT Hub. He has been recognized with awards, such as the Best Research Paper at ICSOC 2016 and has made significant contributions to top journals and conferences.

Aneesh Krishna (Member, IEEE) received the Ph.D. degree in computer science from the University of Wollongong. He was a Lecturer from 2006 to 2009. He is currently a Professor with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University. He has published over 130 articles. His research is supported by the ARC and other Australian entities, such as NSW State Emergency Service and companies, including Woodside Energy. His research spans AI in software engineering, model-driven development, data mining, and renewable energy systems. He also serves as an ARC assessor and participates in various technical committees for international conferences.