Intelligent Intrusion Detection and Service Recovery Framework for Java Microservice Architectures

1. Mohammad Jaweed Yaser
PG Scholar, Department of Information Technology , Shadan College of Engineering and Technology , Hyderabad , Telangana, India - 500086.
Email :- mdjaweedyaser@gmail.com

2. Md. Ateeq Ur Rahman
Professor, Department of Computer Science and Engineering, Shadan College of Engineering and Technology, Hyderabad, Telangana, India - 500086
Email:- mail_to_ateeq@yahoo.com

3. Shaik Shakeer Basha
Professor, Department of Information Technology, Shadan College of Engineering and Technology, Hyderabad, Telangana, India - 500086
Email:- shakir.qa@gmail.com

Abstract: The IDS with Automated Micro-service Recovery seems to be a promising approach for protecting a Java-based web system and ensuring the availability of services. The KNN method examines network data and detects any unusual patterns by comparing patterns of test inputs with patterns of known normal inputs, patterns of unknown inputs and patterns of attack. If something is discovered to be "abnormal", what? The system will automatically call other microservice lines whose scores are close to that of the first one to ensure that there are no issues. MySQL is used as the database management system and Tomcat7 is used as the web server platform. This allows easy integration and enterprise deployment of Java. The accuracy of the IDS model is very high, both during normal time and during attack. The automatic service recovery feature can make self-repair, making the system more reliable, reduce downtime. With real-time tracking and risk assessment, administrators can thwart network threats before they come to harm. Through experiments, it has been demonstrated that the system can maintain a very high detection rate and intelligent rerouting rate, and it is a scalable system architecture for building a resilient networked app.

"Keywords: *Intrusion Detection System, K-Nearest Neighbors, Anomaly Detection, Java Web Application, Microservice Recovery, Self-Healing, Risk Scoring, Network Security."*

1. INTRODUCTION

With widespread acceptance of web-based services and microservice designs, designing, deploying, and scaling systems has changed in a different manner. For business solutions, the Java-based Web environments are the most suitable as they support all platforms and are available with an ecosystem. Now that microservices are in the picture, it's easier to control, and an application can be attacked on multiple fronts – invasion, lateral attack and service level anomalies. As microservices communicate with one another through networks, any slight security breach could rapidly cascade to other services that rely on it, leading to greater harm and disruption of key functions [1].

Traditional network security solutions like firewalls, rule-based tracking and signature-based intrusion detection are difficult to work with microservices because they are constantly changing. Most of these methods are reactive and consist of pre-defined attack signatures and criteria. This means that they can't be used to stop zero-day threats or attacks that use moving vectors. As SDN, cloud-native platforms and container management become more prevalent, it is becoming increasingly difficult to see

and control traffic. Hence, real time anomaly detection is not an option, but an imperative [2, 3].

Anomaly-based attack detection by learning from normal activity has emerged as a popular method to detect unusual activity in normal traffic. Anomaly detection techniques vary and involve analyzing the traffic flow and the interactions between services to identify hidden hazards. Meanwhile, the modern distributed systems demand to be observable, strong, and recover from failures automatically. Cloud-based tracking infrastructures and architectures for telemetry give you real-time views of service health, performance and security events across various scenarios [4, 5].

Thanks to recent developments in "AI" and "graph-based learning", microservices communities can easily find issues. When there are many service connections, technologies based on graph attention networks, temporal modeling and functional connectivity analysis can identify and localize the issues at a very granular level [6, 7, 8]. These clever techniques provide proactive threat detection, reduce the number of false alerts and increase the awareness of the system. Moreover, AI-powered risk reduction technologies automatically react and respond to risks, decreasing reaction and action time for human beings [9, 10].

The primary objective is to create a safe and reliable Java Web environment through continuous monitoring of the network, precise identification of unusual activities and automated service recovery. The purpose of the solution is to ensure uninterrupted availability of the service, reduce the time for security to respond, identify cascaded failures and increase reliability. It will also safeguard applications that are developed using microservice architectures from the network threats of today.

2. RELATED WORK

It has become more complex with cloud-native and microservices architectures, and lots of research has been conducted on smart security solutions to detect and prevent issues and ensure the availability of services. It turns out perimeter security is not sufficient to handle distributed microservices, as an attack is more likely to be identified as being smaller changes in behavior rather than as a signature. Consequently, anomaly detection is one of the most important techniques that can be used to enhance the security of cloud-native applications by continuously learning what is normal and identifying any abnormal and unusual events as they happen [11].

There has been lots of effort to attempt to explain the weird factors in microservice ecosystems. Many data sources have been studied extensively, including logs, metrics and traces, all of which can be used together to illustrate system health. If they are able to link multiple sources, they can better detect attacks and problems and provide more details regarding the attack. For instance, recent studies have demonstrated that statistical and ML techniques can be applied to scale and adapt monitoring of service interactions that evolve over time [12].

It is also available to sample datasets that have been extremely useful for the success of anomaly detection studies. The service detection can be compared using microservice API logs and metrics from public sources. Adhere to normal and abnormal operating patterns in these datasets. They can be employed to test supervised and unsupervised learning techniques, and determine whether security studies can be replicated [13]. These tools are particularly beneficial for examining intrusion

detection methods in scenarios that closely simulate real-life scenarios.

There are no known attack data in real life and thus, unsupervised anomaly detection is gaining more attention. The researchers are primarily looking for oddness via clustering, density estimates and reconstruction methods, but they know nothing about attack signatures. These methods are excellent when developing microservices that frequently change their behavior due to updates, scalability and dynamic routing [14]. Unsupervised models are flexible and can help find risks and performance problems that don't make sense.

Although threat identification is still relevant, it's "special sauce" these days.These days people are working on using root cause analysis and common security designs instead. In integrated techniques, abnormality detection and dependency analysis are analyzed simultaneously to identify failures or attacks in systems with dependency relationships. Rather than send one alarm for a problem, these frameworks provide valuable information that you can utilize to accelerate the diagnosis process and/or reduce the operational weight of a large installation of microservices [15].

Security study is also conducted from a wider software engineering point of view on microservices. The use of terms related to security, and how vulnerabilities evolve over time, reveal common errors in communication, configuration, and deployment of microservices. People can use these to find common security issues and make better ways to find them and fix them [16]. Systematic mapping research can be used to identify and categorize what we know about the security of microservices today, including methods, issues, and areas of missing knowledge that require further research in these fields.

The number of microservices security options which are beginning to incorporate recovery and resilience is growing. Research on Kubernetes orchestration, DR and cloud security indicate that automated response options are critical. These approaches can be used to isolate the faulty services, reroute the services and restore their normal operations within a short time, thereby reducing downtimes and making the services available [18].

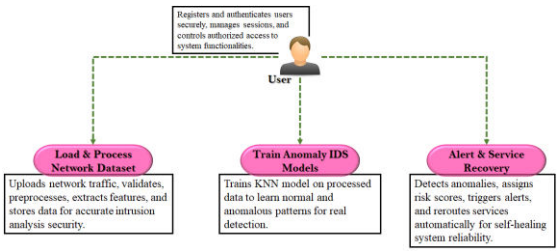
With the addition of AI-based anomaly identification, the idea of a self-healing system has been picking up speed. If they see something out of the ordinary, they respond quickly and automatically, restarting services, directing traffic around, moving resources and more. Self-healing techniques have been demonstrated in actual systems to increase their robustness and reduced dependency on manual handling of security issues [19].

Finally, models for determining what actions might result and what actions might be taken most effectively if an anomaly is detected are added in to the DL-based cybersecurity risk assessment. These approaches can be used to determine the level of risk associated with microservice architectures in order to make proper decisions and take adequate precautions in complex cloud environments [20].

New research has demonstrated that smart systems can be employed for protection and automation. Slamani et al. [26] stress the need for adaptable automation and system efficiency in industrial robots that are focused on people. In their paper [27] Kaushik et al. discuss the potential of combining blockchain with AI to improve the security of IoT. These collectively create safe, smart and sustainable distributed computer environments.

3. MATERIALS AND METHODS

The proposed system employs a smart IDS and automatic microservice recovery to enhance online Java settings safety and guarantee the non-reduction of operation. It analyzes network data using KNN algorithm to identify any unusual behavior and determine if an event is normal or malicious based on the patterns that have already been observed. The system is real-time monitoring of network data and deducing risk scores. This way, potential dangers can be minimised before they occur. The system automatically switches to the backup microservices whenever there is a service problem or attack in the main service path, and maintains the current service to ensure that the system can repair itself. Your data can be efficiently stored using MySQL and your Web operations will be robust and scalable through Tomcat7's scalable setup options. It is intuitive and is suitable for enterprise-level applications [21-23]. It allows you to upload data sets, train an IDS model, and view real time anomaly detection and service rerouting.



“Fig.1 Proposed Architecture of the Integrated IDS System”

The work flow of an user focused IDS is shown in figure 1. 1. The session and user authentication process starts. Then, it breaks down into three major sections: loading and preparing network data for feature extraction; training a KNN model to recognize normal traffic and abnormal traffic patterns; ensuring the self-healing system remains stable by automatically triggering alarms and restoring service.

i) Load & Process Network Dataset:

This is an important component of the system known as "Load & Process Network Dataset. It's where users will input the data of the network traffic for analysis. Once uploaded, the system performs a large-scale check and analysis to ensure that all of the data has been filled in properly, is accurate, and is in the proper format. This reduces the risk of error during further analysis of this data. The data is preprocessed to remove noise, eliminate missing values and normalize significant features. This is done to make sure that the dataset is consistent and good for finding anomalies. The module also processes some aspects of the raw network data and converts them to a structured format suitable for easy processing by the IDS. This data is then securely stored in the database, ensuring data accuracy and enabling seamless access for real-time monitoring and analysis. The module's networking data processing and organization increases the reliability and speed of the IDS and its ability to detect unusual patterns. It also provides proactive security management in a Java environment on the Web.

ii) Train Anomaly IDS Models:

The Train Anomaly IDS Models module is a clever model of identifying and recognizing normal and abnormal network behavior. This cleaned and checked network data is used to identify patterns and correlations in traffic data in the system. It also documents characteristics of normal operations and potential anomalies. This will help the model to get better at identifying such minor variations and, consequently, better at detecting threats in real time or failures in the system. This module also fine-tunes the model parameters to ensure that the network event analysis is reliable and that we reduce the number of false-positive and false-negative results

and increase the accuracy of the classification. The model uses both old and new data to learn more and get better at making predictions so that choices can be made ahead of time. Upon learning, it can accurately classify network events in real time and identify the level of risk associated with each event to enable automatic responses. This ensures that the processes of microservices are persistent, secure and recoverable.

iii) Alert & Service Recovery:

The Alert & Service Recovery module constantly checks network events as they happen and looks for strange things in the arriving data that could be a security risk or cause problems in the network. A risk number is given to each event that shows how bad and likely an attack or failure is to happen. If there is an anomaly with a high risk, the other microservices are quickly switched on to reroute traffic. This ensures a continuity of service and smooth operations. Since the system can heal itself, the important services can continue to function even if the system fails during an attack or during the attack. Meanwhile, alerts are sent to administrators to provide them with live data of any identified threats and to enable them to monitor on demand. The module is supposed to be built in a way that makes it proactive in finding strange behavior and automatically recovering from it. This will render the online Java environment more reliable, safe and resilient. The system will continue to operate and be safe as long as there are still issues. It also provides efficient and stable network management, reduces downtime and ensures private data security.

iv) Algorithm:

KNN is one of the most popular supervised ML algorithms for classification and regression. It does this by determining the similarity between two sets of data points. In this method, the closest neighbors are found by finding the distances between each training example and each test instance, normally using a distance measure like Euclidean distance or Manhattan distance. K is a constant number of neighbors that are very close. Once that is done, the class of the test instance is based on the majority name of these neighbors. KNN has been used successfully in network security to find strange patterns in arriving network traffic by comparing them with known patterns of normal and malicious behavior [24]. This allows the system to assign a risk score to each network event, identify potential risks and aid in proactive risk management. KNN is simple to implement and can be applied to data analysis in real time which can be used for constant monitoring, automatic decision making, and maintaining online services' stability in Java-based applications, making it suitable for environments with rapid changes [25].

4. RESULTS AND DISCUSSIONS

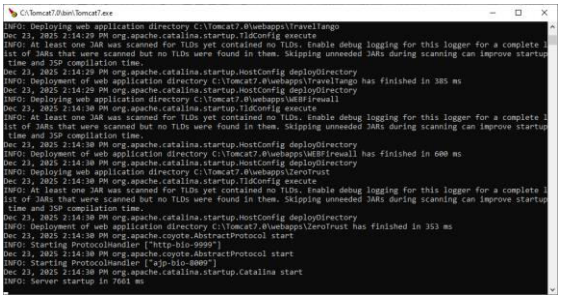


Fig.2 Tomcat7 Server

In Figure 2, it can be observed that Java Web Server is running. Just open your browser and type in http://localhost:8080/MicroserviceIDS/index.jsp. Then press the enter key to go to the next page.

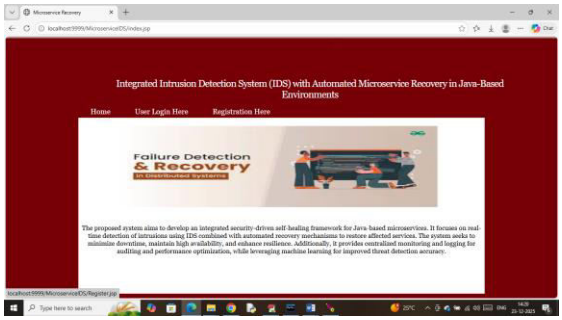


Fig.3 Home Page

To get to the page below, click on the "Registration Here" link above Figure 3.

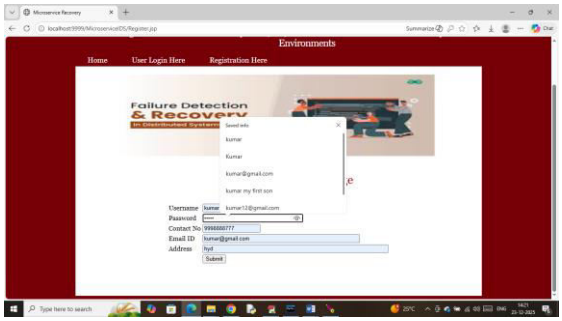


Fig.4 Registration

The user will type in their sign up data and then click the button to proceed to the next page. In the figure 4 the user fills in the information required to register and then clicks on the button to proceed to the next page.

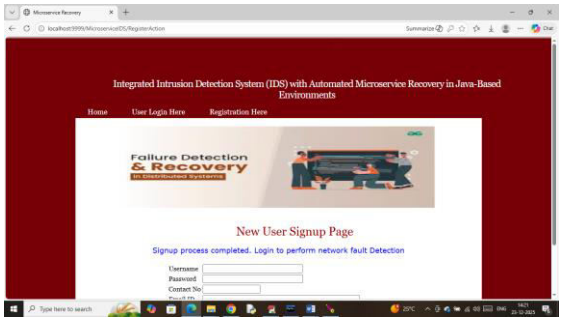


Fig.5 Registration Process Completed

The user has successfully registered as displayed in figure 5. Click on the "User Login" link above to view the following page.

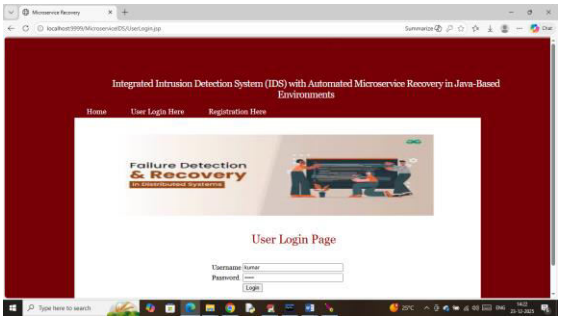


Fig.6 User Login

After logging in you will see the following page in fig. 6.

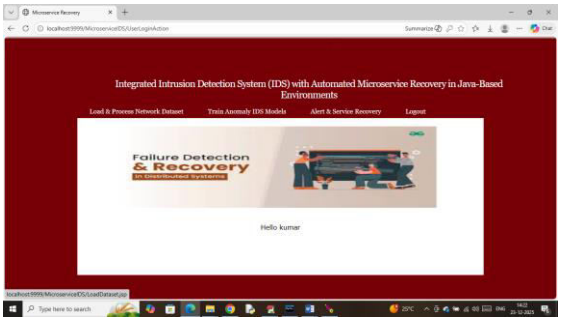


Fig.7 Main Page

On Figure 7, click on the "Load & Process Network Data" link. This will take you to the next page:

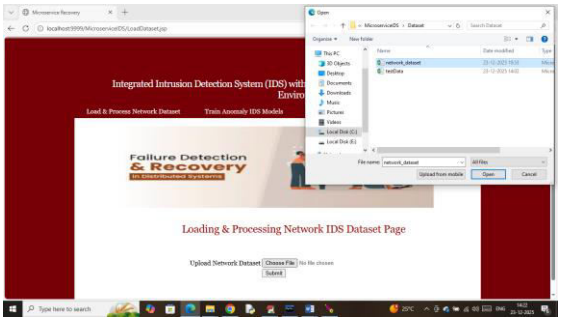


Fig.8 Upload Network Dataset

To share a dataset, follow the instructions in figure 8. Then, click on the buttons below to go to the next page.

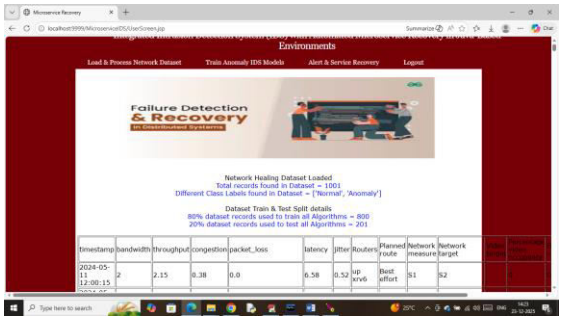


Fig.9 Dataset Loaded - Result

Next, click on the link in Fig 9 labelled "Train Anomaly IDS Model. This will take you to the next page, where the model will be trained.

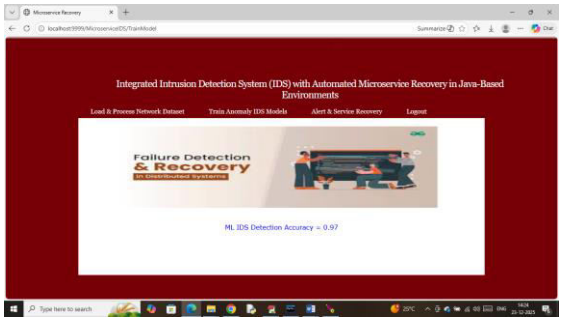


Fig.10 ML IDS Detection Accuracy

From the results, it can be seen that the Java based KNN IDS model has been accurate 97% of the time as seen in figure 10. Click on the "Alert & Service Recover" link to see the next page.

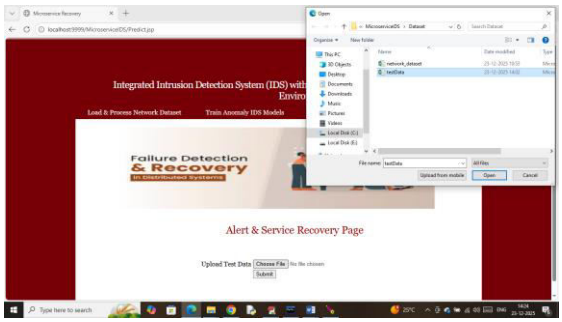


Fig.11 Upload Test Data

Select the data shown in Fig. 11 and distribute it. Then, click on any button to go to Pages 12 and 13.

Fig.12 Test Data Result – Normal or Fault Detected

The network test data are provided in the first column of figure 12, and the status and risk score detected from the network are provided in the second column. If it is "Normal" it will continue to "Primary Route Service", but if it is "Fault" in the "Primary Route", it will be "S2" to indicate the "Self Healing Concept" in the "Alternate Route".

5. CONCLUSION

The Integrated IDS with Automated Micro-service Recovery is a wonderful solution to ensure the safety and reliability of online systems constructed with Java. The KNN algorithm is able to quickly detect unusual network activity by comparing the data received with data that is known as a normal case or attack case. High detection accuracy of 97%, which means there will be less false alarms for events in a network. The automatic service recovery method is always active - whenever it detects a problem in the main service route, it automatically switches to the other microservices. That's how it will heal itself. When used together, MySQL for database management and Tomcat7 for web deployment make a business application platform that is powerful and scalable. Track, risk score and analyze your network in real time for valuable information about your network health. This enables them to react swiftly to dangers. Overall, the system performs well in merging the features of identifying anomalies and smart service management, to

provide a robust and secure architecture, which ensures smooth functioning of operations, reduces downtime and makes Java-based Web services more secure.

Future studies could explore other advanced ML and DL algorithms such as RF, SVM, or LSTM for more precise anomaly detection and for better adapting to evolving attacks. They could be enhanced with automatic threat prioritization and real-time threat information feeds to become more proactive. If microservices were distributed and deployed to the cloud, it would be easier to scale and manage them and their errors. Advanced and zero-day risks can be found with the help of predictive analytics and behavioral profiling. A real-time dash board with graphical representation, indicating network status, problems, and recovery actions for services will also make the system easier to use, smarter and more flexible for network security management.

REFERENCES

[1] Mohottige, T. I., Polyvyanyy, A., Buyya, R., Fidge, C., & Barros, A. (2024). Microservices-based Software Systems Reengineering: State-of-the-Art and Future Directions. arXiv preprint arXiv:2407.13915.

[2] Rahim, J. A., Nordin, R., & Amodu, O. A. (2024). Open-Source Software Defined Networking Controllers: State-of-the-Art, Challenges and Solutions for Future Network Providers. Computers, Materials & Continua, 80(1).

[3] Abdelfattah, A. S., Cerny, T., Yero Salazar, J., Li, X., Taibi, D., & Song, E. (2024). Assessing evolution of microservices using static analysis. Applied Sciences, 14(22), 10725.

[4] Choudhury, S., & Liibaan, L. (2025). Cloud-based Observability in Distributed Systems: Designing a scalable observability architecture in the cloud using Azure, OpenTelemetry and. NET.

[5] Seroukhov, S. (2024). MICROSERVICES DESIGN PATTERNS WITH JAVA. Bpb Publications.

[6] Akmeemana, L., Attanayake, C., Faiz, H., & Wickramanayake, S. (2025). GAL-MAD: Towards explainable anomaly detection in microservice applications using graph attention networks. arXiv.

[7] Zhang, Q., Lyu, N., Liu, L., Wang, Y., Cheng, Z., & Hua, C. (2025). Graph neural AI with temporal dynamics for comprehensive anomaly detection in microservices. arXiv.

[8] Winchester, G., Parisis, G., & Berthouze, L. (2025). FC-ADL: Efficient microservice anomaly detection and localisation through functional connectivity. arXiv.

[9] Singh, S. (2025). Integrating AI-Based anomaly detection with MPK-isolated microservices for proactive security in optical networks. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol, 11(2), 1693–1704.

[10] Ramamoorthi, V. (2024). Anomaly detection and automated mitigation for microservices security with AI. Applied Research in Artificial Intelligence and Cloud Computing, 7(6), 211–222.

[11] Enhancing cloud-native application security using AI-driven microservice anomaly detection. (2025). Journal of Artificial Intelligence for Data-Driven Discovery, 9, 32–41.

[12] Nobre, J., Solteiro Pires, E. J., & Reis, A. (2023). Anomaly detection in microservice-based systems. Applied Sciences, 13(13), 7891.

[13] LO2: Microservice API anomaly dataset of logs and metrics. (2025). arXiv.

[14] Unsupervised anomaly detection in microservices. (2025). Upubscience.

[15] Meyer, O., Johnson, E., & Brown, J. (2025). A unified framework for anomaly detection and root cause analysis in microservice systems. Computer Life.

[16] Automated identification of security discussions in microservices systems. (2021). Journal of Systems and Software, 181, 111046.

[17] Securing microservices and microservice architectures: A systematic mapping study. (2021). Computer Science Review, 41.

[18] Disaster recovery and application security in microservices: exploring Kubernetes and cloud solutions. (2024). Journal of Artificial Intelligence and Big Data.

[19] Self-healing microservices and AI-based anomaly detection system. (2025). International Research Journal of Modernization in Engineering Technology and Science.

[20] Towards deep learning enabled cybersecurity risk assessment for microservice architectures. (2025). Cluster Computing.

[21] NetMoniAI: An agentic AI framework for network security & monitoring. (2025). arXiv.

[22] Anomaly Detection in Microservice-Based Systems. (2023). MDPI Applied Sciences article.

[23] AI-driven intrusion detection and prevention systems to safeguard 6G networks. (2025). Scientific Reports.

[24] Clustered federated learning architecture for network anomaly detection in large scale IoT. (2023). arXiv.

[25] Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. (2022). arXiv.

[26] Slamani, M., Louhichi, B., Arslane, M., & Alawad, M. A. (2026). Smart industrial robots in the transition from automation to human-centric sustainable systems: a comprehensive review of technologies, challenges, and future directions. The International Journal of Advanced Manufacturing Technology, 1–28.

[27] Kaushik, D., Gulia, P., Gill, N. S., Yahya, M., Shukla, P. K., & Shreyas, J. (2026). Synergizing blockchain and AI to fortify IoT security: a comprehensive review. Artificial Intelligence Review, 59(2), 41.