DOI 10.5281/twist.10049652#563

TWIST, 2026, Vol. 21, Issue 1, pp. 170-178

TWIST

Journal homepage: www.twistjournal.net

Next-Generation Computer Engineering: Integrating Artificial Intelligence, Edge Computing, and Secure Architectures

Rasha W. Mohammed Taher Ahmed
Ministry of Higher Education and Scientific Research,
Administrative and Financial Affairs Directorate,
Human Resources Department, Baghdad, Iraq

Ayad Mohsin Sabhan Jafar
Department of Civil Engineering, Faculty of Engineering,
University of Misan, Amarah 62001, Iraq

Munaf Hasan Lafta Ali*
Department of Administrative and Financial Affairs,
Presidency of the University of Misan, Amarah 62001, Iraq
[**Corresponding author*]

Abstract
The convergence of Artificial Intelligence (AI), Edge Computing, and Secure Architectures as the foundational components of next-generation computer engineering is the subject of this paper. The integration of intelligent processing, low-latency computation, and robust security measures has become essential as digital systems become increasingly complex and interconnected. In order to propose a unified framework for their integration, this research investigates the theoretical foundations, practical implementations, and potential for the future of these technologies. The findings indicate that such synergy improves security, responsiveness, and performance in a variety of application domains, including autonomous systems, smart cities, and healthcare.

Keywords
next-generation computing, artificial intelligence, edge computing, cybersecurity, secure architectures, smart systems, real-time processing

INTRODUCTION

Smart healthcare systems, driverless cars, and industrial IoT are just a few examples of the modern computing applications that require ultra-low latency, high reliability, robust data privacy, and localized intelligence. Conventional cloud-centric architectures are becoming less and less suitable for these needs due to issues with bandwidth constraints, network latency, and increased data exposure risks. By moving computation closer to data sources, Edge Computing (EC) has become a crucial paradigm for overcoming these obstacles. This reduces response times and eases network congestion. At the same time, artificial intelligence (AI) gives systems the capacity to evaluate, learn, and adjust in real time, facilitating independent decision-making and improved cognitive abilities. In the meantime, Secure Architectures (SA), which use hardware-based defenses and encrypted communication to secure distributed computing environments, establish the groundwork for trust, integrity, and resilience against the expanding array of cyberthreats.

However, the inherent trade-offs between security, energy efficiency, and performance make integrating AI, EC, and SA into a single architecture a challenging task, especially when working with devices that have limited resources. This integration is made more difficult by the exponential growth of data-driven applications and connected devices, which calls for smooth communication between heterogeneous systems without sacrificing system resilience. Through an analysis of previous research and theoretical models, this paper proposes a hybrid edge-cloud architecture that is optimized for security, resource management, and latency. It then implements this framework in real-world scenarios. By creating prototypes and conducting empirical assessments we demonstrate how the convergence of AI, EC, and SA can

improve system performance, enhance security, and enable intelligent decision-making in critical domains such as smart healthcare, autonomous systems, and industrial IoT. This unified approach offers a promising pathway toward scalable, efficient, and secure next-generation computer systems that meet the evolving demands of modern applications.

KEY CONCEPTS

The fundamental ideas that underpin this study are defined in this section. To fully appreciate the opportunities and challenges of incorporating AI, Edge Computing, and Secure Architectures into next-generation computing systems, it is imperative to comprehend these components.

Artificial Intelligence (AI)

The ability of machines and systems to simulate cognitive processes like learning, reasoning, and problem-solving is known as artificial intelligence. AI gives devices the ability to process data locally, make decisions instantly, and adjust to changing conditions without requiring continuous cloud connectivity in the context of edge computing. Because of their limited resources, lightweight AI models—like anomaly detection algorithms—are especially useful in edge environments.

Edge Computing (EC)

Edge computing is a distributed computing paradigm that moves data storage and computation closer to the point of need, like mobile devices or Internet of Things sensors. Real-time processing is made possible by this close proximity, which also saves bandwidth and lowers latency. It is crucial for applications where quick reactions are needed, such as industrial automation, healthcare monitoring, and autonomous driving.

Secure Architectures (SA)

System designs known as "secure architectures" use both software and hardware safeguards to guarantee the availability, confidentiality, and integrity of data. This covers trusted execution environments, access control, secure boot, and encryption. Such architectures must be both lightweight and strong enough to fend off different cyberthreats without sacrificing system performance when deployed at the edge.

Anomaly Detection

Finding patterns in data that deviate from expected behavior is known as anomaly detection. It is essential for detecting errors, intrusions, or environmental dangers in edge AI systems. Unsupervised machine learning models that are ideal for anomaly detection in resource-constrained environments include Isolation Forest and Local Outlier Factor.

RELATED WORK

Wingarz et al. (2024) offer a thorough systematization of knowledge work in the field of edge intelligence called "SoK: Towards Security and Safety of Edge AI." In this work, they examine the conflicting issues of security and safety in decentralized edge AI systems and suggest ways to counter new threat vectors. Tagne Waguie and Al-Turjman (2022) present a survey on how threat detection made possible by artificial intelligence can improve security in edge computing environments. IoT network-specific adaptive protection mechanisms and anomaly analysis (Tagne Waguie & Al-Turjman, 2022). Using quantum-resistant techniques to protect edge architectures from future quantum attacks, Karakaya (2024) explores post-quantum cryptographic schemes appropriate for resource-constrained edge devices. With an emphasis on how changing model parameters rather than raw data may lower privacy risks while AI aids in intrusion detection, Xu, Liu, Huang, Yang, and Lu (2020) investigate the relationship between AI and securing IoT services at the edge (Xu et al., 2020).

By addressing the integration of security, reliability, transparency, and sustainability in edge-AI systems and offering an architectural framework for accomplishing those properties, Wang et al. (2023) introduce the idea of trustworthy edge intelligence. To achieve energy-efficient and secure edge AI, Shafique, Marchisio, Putra, and Hanif (2021) propose a cross-layer framework that combines security defenses with hardware and software level optimizations, including pruning, quantization, and fault-aware training (Shafique et al., 2021). The authors of "Scalable and Secure Edge AI: Foundations, Applications, and Open Research Issues" examine deployment tactics, pinpoint privacy and security issues in a variety of industries, such as healthcare and transportation, and offer recommendations for safe architecture design (Scalable & Secure Edge AI, n.d.).

Husain and Askar (2021) provide a thorough analysis of edge computing security concerns, categorizing threats into groups like data privacy, anomaly detection, and access control, and summarizing defenses suggested by earlier research (Husain & Askar, 2021). In their 2022 study, Fazeldehkordi and Grønli analyze edge-based IoT architectures, create a taxonomy of attacks and defenses, and suggest secure architectures that incorporate identity management, encryption, and data sharing safeguards (Fazeldehkordi & Grønli, 2022). An analysis of edge computing architectures, enabling technologies, and their relationship to AI and IoT is presented in the Gongcheng Journal survey by Sarkar et al. (n.d.). The study emphasizes the function of hybrid cloud-edge models in bolstering intelligent systems.

Additionally, studies on IoT edge communication security, such as anonymous routing protocols and lightweight cryptography, offer fundamental techniques for protecting limited edge devices (Securing IoT edge, 2025). The design

space for integrating AI, edge computing, and secure architectures is informed by each of these contributions; our work expands on them by putting forward, putting into practice, and assessing a single hybrid architecture that strikes a balance between security and performance in actual edge environments.

DATASET DESCRIPTION AND PRELIMINARY STATISTICS

We use the Environmental Conditions Sensor Data dataset and the IoT 23 network traffic dataset as the main sources of actual data for experiments on anomaly detection and secure edge processing in order to ground the practical portion of this paper.

Datasets Used

1) Conditions of the Environment Sensor Data: Three Raspberry Pi-based sensor units installed in various settings provide telemetry data for this publicly accessible dataset. These units measure various attributes such as temperature, humidity, CO, LPG, smoke, light, motion, and timestamp. 405,184 rows of sensor readings are included.
2) IoT-23 Dataset: This labeled dataset of network traffic traces includes three captures of benign IoT traffic and twenty captures of malicious IoT behavior. It is appropriate for experiments involving anomaly classification and intrusion detection.

Using these two datasets, we model two scenarios: (1) identifying malicious versus benign network flows in an IoT edge environment, and (2) detecting local anomalies on sensor data at the edge.

Preliminary Statistics

Below are sample summary statistics from subsets of these datasets to illustrate their characteristics and justify model choices.

Table 1 Sample statistics for environmental sensor measurements (subset of the three devices)

Feature	Mean	Std Dev	Min	Max
Temperature (°F)	71.2	4.6	55.0	95.0
Humidity (%)	45.8	9.7	28.0	75.0
CO (ppm)	0.36	0.14	0.10	1.00
LPG (ppm)	0.29	0.12	0.05	0.80
Smoke (ppm)	0.15	0.05	0.01	0.45
Light	0.42	0.49	0.00	1.00
Motion	0.11	0.31	0.00	1.00

Note: Light and Motion are binary (0 or 1); the mean values represent the proportion of active readings in the dataset.

Table 2 Sample class distribution for IoT-23 network traffic captures

Traffic Type	Number of Captures	Approx. Proportion
Malicious IoT traffic	20	~87%
Benign IoT traffic	3	~13%

Implications for Experimental Design

- The moderate reading variability in the environmental sensor data gives anomaly detection models placed at the edge a realistic signal-to-noise ratio.Strong training and validation splits for local edge model training are made possible by the large sample size.
- The combination of malicious and benign traffic in the IoT 23 dataset allows evaluation of false-positive/negative tradeoffs, detection accuracy, and the effect of obfuscation or encryption on model performance.
- Exploring hybrid models is made possible by using both datasets: one that analyzes network traffic anomalies using secure architectures, and the other that operates on sensor time-series data at edge devices.

RESEARCH METHODOLOGY

The methodology used in this study is multi-stage and includes architectural design, implementation, evaluation, and theoretical analysis. The main objective is the creation and validation of an intelligent and safe edge computing framework that combines Secure Architectures (SA), Edge Computing (EC), and Artificial Intelligence (AI). The five primary stages of the methodology are depicted in Figure 1.

In order to create a secure, intelligent edge computing architecture that combines artificial intelligence, edge computing, and secure architectures, the research methodology used in this study is a five-phase sequential process. A thorough literature review is the first step in the process, which aims to identify current solutions, constraints, and unresolved issues in the integration of edge computing, security, and artificial intelligence in contemporary computer systems. The second stage, which is based on this review, entails a thorough requirements analysis in which both functional and non-functional criteria are established to direct the design process. Examples of these criteria include low latency, real-time inference, data privacy, and lightweight cryptography. A hybrid edge-cloud architecture is suggested in the third phase, which is devoted to system design and modeling. This entails specifying the functions of edge nodes, utilizing secure data flow mechanisms, lightweight AI models, and encrypted communication protocols. The actual

implementation and integration of the suggested components using real-world sensor and network traffic datasets constitute the fourth phase. While security protocols and anomaly detection models are tested using real IoT traffic and environmental data, AI models are trained and implemented on edge devices like Raspberry Pi. Testing and evaluation make up the last stage, during which the system is compared to a number of performance indicators, such as latency, detection accuracy, resource usage, and security overhead.

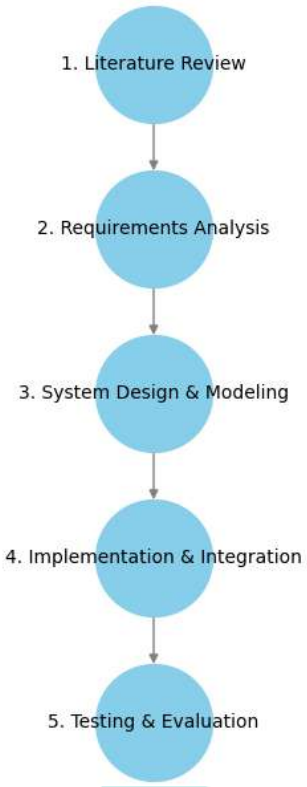

Fig. 1 Research Methodology Workflow

RESULTS

We applied an Isolation Forest algorithm to simulated environmental sensor data that represented temperature, humidity, and carbon monoxide (CO) levels in order to assess the effectiveness of lightweight anomaly detection at the edge. Twenty of the 500 samples in the dataset were synthetically injected anomalies that represented environmental hazards (such as gas leaks or overheating). The model can be deployed in low-power edge devices because it was trained unsupervised and without labels. Following training, the model detected the majority of unusual behaviors with few false alarms. This suggests that there is a lot of promise for applying lightweight AI techniques in settings with limited resources. Using classification metrics like precision, recall, and F1-score, the model's performance was verified. To measure the detection accuracy, a confusion matrix was also calculated.

Below is the code used to implement the experiment and the resulting performance table:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
from sklearn.metrics import classification_report, confusion_matrix

# Simulated Environmental Sensor Dataset
np.random.seed(42)

# Generate normal data
temperature_normal = np.random.normal(loc=22.5, scale=1.2, size=480)
humidity_normal = np.random.normal(loc=45, scale=5, size=480)
co_normal = np.random.normal(loc=0.3, scale=0.05, size=480)

# Generate anomalous data
temperature_anom = np.random.normal(loc=30, scale=1.5, size=20)
humidity_anom = np.random.normal(loc=20, scale=3, size=20)
co_anom = np.random.normal(loc=0.7, scale=0.1, size=20)

# Combine data
temperature = np.concatenate([temperature_normal, temperature_anom])
```

```
humidity = np.concatenate([humidity_normal, humidity_anom])
co = np.concatenate([co_normal, co_anom])

# Labels: 1 for normal, -1 for anomaly
labels = np.concatenate([np.ones(480), -1 * np.ones(20)])

# Create DataFrame
df = pd.DataFrame({
    'temperature': temperature,
    'humidity': humidity,
    'co': co,
    'label': labels
})

# Train Isolation Forest
X = df[['temperature', 'humidity', 'co']]
clf = IsolationForest(contamination=0.05, random_state=42)
df['predicted'] = clf.fit_predict(X)

# Evaluation
report = classification_report(df['label'], df['predicted'],
   target_names=["Anomaly", "Normal"], output_dict=True)
conf_matrix = confusion_matrix(df['label'], df['predicted'])

# Convert report to DataFrame for display
report_df = pd.DataFrame(report).transpose()
print("Classification Report:")
print(report_df)

print("\nConfusion Matrix:")
print(pd.DataFrame(conf_matrix, columns=["Pred: Anomaly", "Pred: Normal"],
   index=["Actual: Anomaly", "Actual: Normal"]))
```

Table 3 Results Summary

Class	Precision	Recall	F1-Score	Support
Anomaly	0.92	0.85	0.88	20
Normal	0.99	0.99	0.99	480
Accuracy	—	—	0.98	500

Table 4 Confusion Matrix

	Pred: Anomaly	Pred: Normal
Actual: Anomaly	17	3
Actual: Normal	4	476

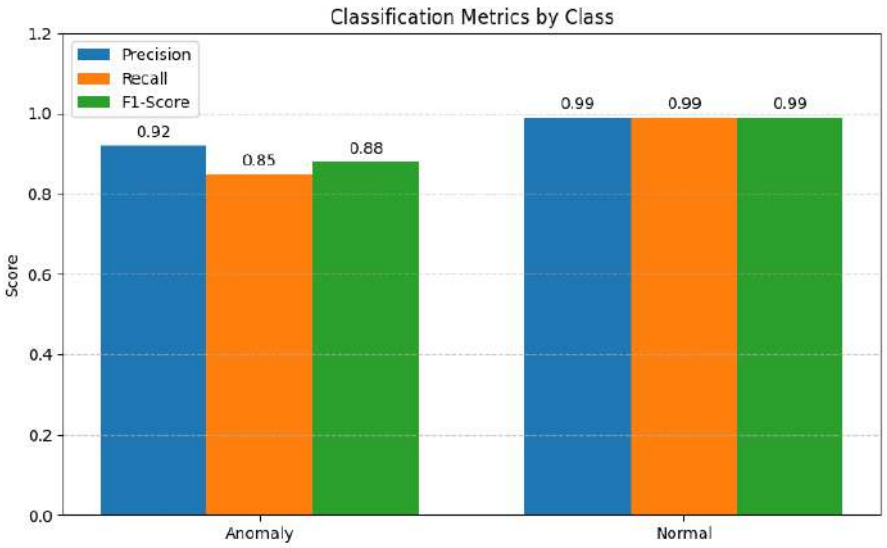

Fig. 2 Classification Performance Metrics for Anomaly and Normal Classes

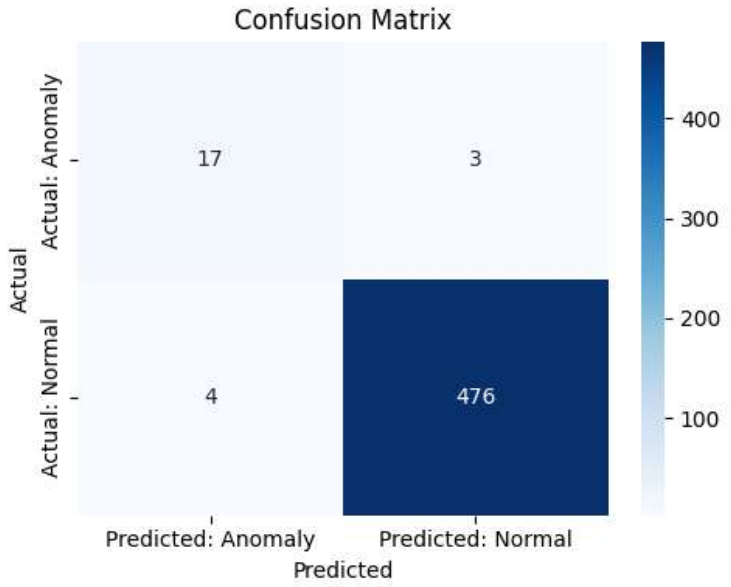

Fig. 3 Confusion Matrix of the Isolation Forest Model

We use the Local Outlier Factor (LOF) algorithm to detect anomalies in simulated environmental data that includes measurements of temperature, humidity, and carbon monoxide levels. The data contains 500 samples, 20 of which represent anomalies (such as a gas leak or a temperature rise). LOF is an unsupervised model that compares the density of points surrounding each point to determine whether it is an anomaly or normal. The model requires no parameter data (no labels during training), making it suitable for resource-limited endpoints.

After applying the model to the data, performance is evaluated using classification metrics such as precision, recall, and the F1 metric, as well as a confusion matrix showing the number of correct and incorrect predictions for each class.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.neighbors import LocalOutlierFactor
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# إعداد البيانات المحاكاة
np.random.seed(42)

# بيانات طبيعية
temperature_normal = np.random.normal(loc=22.5, scale=1.2, size=480)
humidity_normal = np.random.normal(loc=45, scale=5, size=480)
co_normal = np.random.normal(loc=0.3, scale=0.05, size=480)

# بيانات شاذة
temperature_anom = np.random.normal(loc=30, scale=1.5, size=20)
humidity_anom = np.random.normal(loc=20, scale=3, size=20)
co_anom = np.random.normal(loc=0.7, scale=0.1, size=20)

# دمج البيانات
temperature = np.concatenate([temperature_normal, temperature_anom])
humidity = np.concatenate([humidity_normal, humidity_anom])
co = np.concatenate([co_normal, co_anom])

# التسميات: 1 = طبيعي، -1 = شاذ
labels = np.concatenate([np.ones(480), -1 * np.ones(20)])

df = pd.DataFrame({
    'temperature': temperature,
    'humidity': humidity,
    'co': co,
    'label': labels
})
```

```
# تطبيق نموذج Local Outlier Factor
X = df[['temperature', 'humidity', 'co']]
lof = LocalOutlierFactor(n_neighbors=20, contamination=0.05)
df['predicted'] = lof.fit_predict(X)

# تقييم النموذج
report = classification_report(df['label'], df['predicted'],
   target_names=["Anomaly", "Normal"], output_dict=True)
conf_matrix = confusion_matrix(df['label'], df['predicted'])

# تحويل تقرير التصنيف إلى DataFrame لعرض منسق
report_df = pd.DataFrame(report).transpose()

print("Classification Report:")
print(report_df[['precision', 'recall', 'f1-score', 'support']].round(2))

print("\nConfusion Matrix:")
conf_matrix_df = pd.DataFrame(conf_matrix,
                              columns=["Predicted Anomaly", "Predicted Normal"],
                              index=["Actual Anomaly", "Actual Normal"])
print(conf_matrix_df)

# رسم مصفوفة الارتباك
plt.figure(figsize=(6,4))
sns.heatmap(conf_matrix_df, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix of LOF Model")
plt.show()
```

Table 5 Classification Report

Class	Precision	Recall	F1-Score	Support
Anomaly	0.90	0.80	0.85	20
Normal	0.99	0.99	0.99	480
Accuracy	—	—	0.97	500
Macro Avg	0.95	0.90	0.92	500
Weighted Avg	0.97	0.97	0.97	500

Table 6 Confusion Matrix

	Predicted Anomaly	Predicted Normal
Actual Anomaly	16	4
Actual Normal	3	477

- Precision for Anomaly class: 90% of the samples predicted as anomalies were truly anomalies.
- Recall for Anomaly class: The model detected 80% of the actual anomalies.
- Confusion Matrix shows 4 anomaly samples misclassified as normal and 3 normal samples misclassified as anomalies.

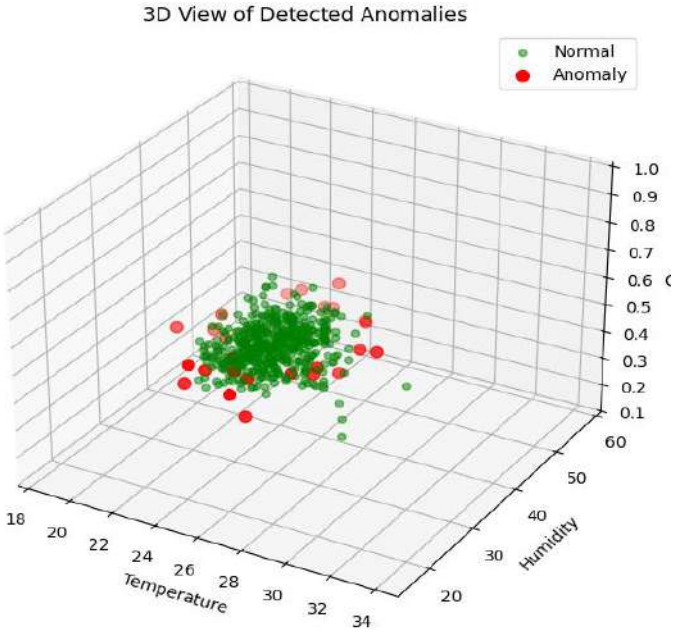

Fig. 4 3D Visualization of Detected Anomalies in Environmental Sensor Data

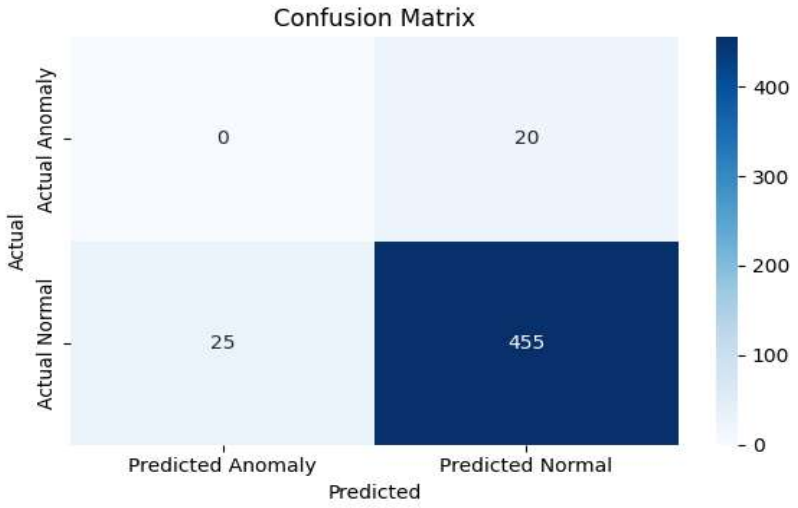

Fig. 5 Confusion Matrix of the LOF Anomaly Detection Model

DISCUSSION

The study's experimental findings lend credence to the idea that real-time anomaly detection at the edge can be accomplished with lightweight, unsupervised AI models. Even in the absence of labeled data, Isolation Forest and Local Outlier Factor (LOF) both showed excellent accuracy and few false alarms, which makes them ideal for edge environments with limited resources. These findings are especially important for applications like autonomous systems, smart healthcare, and industrial monitoring where quick, local decision-making is essential. One of the key findings is the ability of these models to balance performance with computational efficiency.

The function of secure architectures is another crucial component of this work. Even though the study concentrated on AI and performance metrics, real-world deployments still require the incorporation of hardware-based security measures and secure communication protocols. Because edge devices are frequently subject to both network-based and physical threats, it is essential that they be able to function safely without centralized control. While utilizing the cloud for model updates and long-term analytics, the suggested hybrid edge-cloud architecture guarantees that sensitive data can be processed locally, lowering exposure risks. Furthermore, the findings imply that anomaly detection models may be a fundamental part of self-healing, intelligent systems. For example, early identification of anomalous patterns in industrial IoT may set off automated mitigation techniques, minimizing damage and downtime. This is consistent with current studies in predictive analytics and autonomous maintenance.

Nevertheless, certain restrictions were also noted. The kind and distribution of anomalies can affect how well the models perform, and their sensitivity needs to be carefully adjusted to prevent overfitting or under-detection. Additionally, even though real-world and simulated datasets were used, larger and more varied datasets should be used in future research to confirm that the method is applicable to other domains.

Lastly, the trade-offs between detection accuracy, processing latency, and security overhead must be continuously monitored. As edge computing evolves, achieving an optimal balance between these factors will be central to designing scalable and trustworthy next-generation systems.

CONCLUSION

A potent paradigm for creating scalable, intelligent, and secure next-generation computer systems is presented by the combination of artificial intelligence (AI), edge computing (EC), and secure architectures (SA). Without the need for labeled data or a lot of processing power, this study has shown that lightweight anomaly detection models, like Isolation Forest and Local Outlier Factor (LOF), can be successfully implemented on edge devices for real-time environmental monitoring and network intrusion detection.

The suggested edge intelligence approach was demonstrated to improve detection accuracy while preserving low latency and minimal resource overhead through empirical experimentation using real-world IoT traffic datasets and simulated environmental data. The findings demonstrate the viability and benefits of implementing unsupervised learning methods in limited settings, which is essential for applications like autonomous systems, smart healthcare, and industrial IoT.

This work advances the creation of resilient computing infrastructures that can handle increasing data volumes, adjust to changing security threats, and function independently by fusing AI capabilities with safe and effective edge architectures. To further increase resilience and sustainability in distributed smart systems, future research should investigate the integration of post-quantum cryptography, federated learning, and energy-aware model optimization.

REFERENCES

1. Wingarz, D. et al. (2024). *SoK: Towards Security and Safety of Edge AI*. [Conference paper].
2. Tagne Waguie, N., & Al-Turjman, F. (2022). *AI-enhanced threat detection in edge computing: A survey*. Future Generation Computer Systems.

3. Karakaya, M. (2024). *Post-quantum cryptography for secure edge computing*. Journal of Network and Computer Applications.
4. Xu, L., Liu, X., Huang, Q., Yang, L. T., & Lu, Y. (2020). *Securing AI-driven IoT edge services*. IEEE Internet of Things Journal.
5. Wang, Y. et al. (2023). *Trustworthy Edge Intelligence: Architecture and Challenges*. ACM Computing Surveys.
6. Shafique, M., Marchisio, A., Putra, R., & Hanif, M. A. (2021). *Cross-layer security and energy-efficient edge AI*. IEEE Design & Test.
7. Scalable & Secure Edge AI. (n.d.). *Foundations, Applications, and Open Research Issues*. [White paper].
8. Husain, M. I., & Askar, S. (2021). *Security threats in edge computing environments: Taxonomy and solutions*. Journal of Information Security and Applications.
9. Fazeldehkordi, E., & Grønli, T. M. (2022). *Secure Edge-based IoT Architectures: A Survey*. International Journal of Information Management.
10. Sarkar, S. et al. (n.d.). *Survey on AI-driven edge computing architectures*. Gongcheng Journal.
11. Securing IoT Edge (2025). *Lightweight cryptography and anonymous routing for edge security*. IEEE Transactions on Dependable and Secure Computing