International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

Machine Learning-Based Anomaly Detection Model for Resilient Cloud and Edge Computing Systems

Alan Zaoxing Liu
Assistant Professor
Department of Computer Science
D.Y.A. International University, USA

Abstract

The rapid adoption of cloud computing, edge computing and distributed digital infrastructure has created increasingly complex environments in which conventional security and reliability mechanisms may struggle to identify abnormal behaviour in real time. Cloud–edge systems involve heterogeneous devices, dynamic workloads, distributed services, virtualised resources and continuously changing communication patterns, creating new opportunities for system failures, cyberattacks and performance degradation. Machine learning (ML)-based anomaly detection provides a promising approach for identifying deviations from normal system behaviour and supporting resilient infrastructure. This paper proposes a machine learning-based anomaly detection model for cloud and edge computing systems that integrates data collection, feature engineering, behavioural modelling, anomaly classification and adaptive response. The proposed architecture combines network traffic, resource utilisation, application performance and system-log information to construct a multidimensional representation of system behaviour. Supervised, unsupervised and hybrid ML approaches are considered for detecting both known and previously unseen anomalies. The framework incorporates adaptive thresholds, distributed inference and continuous model updating to address the dynamic nature of cloud–edge environments. Key performance indicators include detection accuracy, precision, recall, F1-score, false-positive rate, detection latency and computational overhead. The paper further examines challenges associated with data imbalance, concept drift, heterogeneous infrastructure, limited edge resources, adversarial manipulation and privacy-preserving analytics. The proposed model aims to strengthen resilience by enabling early anomaly identification, prioritised risk assessment and intelligent response while minimising disruption to legitimate workloads. The study concludes that integrating lightweight machine learning with distributed cloud–edge

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

monitoring can provide an effective foundation for proactive, scalable and adaptive infrastructure resilience.

Keywords: Machine Learning, Anomaly Detection, Cloud Computing, Edge Computing, Cybersecurity, Network Security, System Resilience, Intrusion Detection, Distributed Computing, Intelligent Infrastructure.

1. Introduction

Cloud computing and edge computing have become fundamental components of modern digital infrastructure.

Cloud platforms provide:

- Elastic computing resources.
- Centralised data processing.
- Scalable storage.
- Virtualised services.
- On-demand infrastructure.

Edge computing extends computational capabilities closer to data sources, enabling:

- Lower latency.
- Faster decision-making.
- Reduced bandwidth consumption.
- Localised processing.
- Improved responsiveness.

However, the distributed nature of cloud–edge environments also increases system complexity.

Anomalies may arise from:

- Cyberattacks.
- Hardware failures.
- Software defects.
- Network congestion.
- Resource exhaustion.
- Configuration errors.
- Malicious insider behaviour.
- Unexpected workload changes.

Traditional rule-based monitoring systems may fail to recognise sophisticated or previously unseen deviations.

Machine learning offers an alternative by learning patterns of normal and abnormal behaviour from data.

2. Cloud–Edge Computing Environment

A typical cloud–edge infrastructure consists of several layers.

Layer 1 — IoT and End Devices

Examples include:

- Sensors.
- Cameras.
- Industrial controllers.
- Mobile devices.
- Smart appliances.

Layer 2 — Edge Infrastructure

Includes:

- Edge servers.
- Gateways.
- Micro-data centres.
- Local computing nodes.

Layer 3 — Cloud Infrastructure

Includes:

- Virtual machines.
- Containers.
- Distributed databases.
- Cloud applications.
- Centralised analytics.

Layer 4 — Management and Security

Responsible for:

- Monitoring.
- Authentication.
- Resource allocation.
- Security policies.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

- Incident response.

3. The Need for Anomaly Detection

Conventional security systems frequently depend on predefined signatures or rules.

For example:

Known Attack Pattern → Signature → Detection

This approach can be effective for previously identified threats.

However:

Unknown Attack → No Existing Signature → Potential Missed Detection

Machine learning can instead model behavioural patterns.

Normal Behaviour → Learned Model → Deviation → Anomaly Alert

4. Definition of Anomaly

An anomaly is an observation that differs significantly from expected system behaviour.

Examples include:

- Unusually high CPU usage.
- Sudden network traffic increases.
- Unexpected port activity.
- Abnormal memory consumption.
- Repeated authentication failures.
- Unusual application latency.
- Unexpected communication between nodes.

Not every anomaly is necessarily malicious.

Therefore, anomaly detection should distinguish between:

Security Anomalies

and

Operational Anomalies.

5. Machine Learning for Anomaly Detection

ML models can learn relationships between system variables and identify deviations.

Potential approaches include:

Supervised Learning

Uses labelled normal and malicious samples.

Unsupervised Learning

Discovers unusual patterns without requiring complete labels.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

Semi-Supervised Learning

Uses primarily normal data and identifies deviations from learned normal behaviour.

Deep Learning

Learns complex representations from high-dimensional data.

6. Proposed Anomaly Detection Model

The proposed framework consists of eight stages:

1. Data acquisition.
2. Data preprocessing.
3. Feature extraction.
4. Behavioural modelling.
5. Anomaly scoring.
6. Threat classification.
7. Intelligent response.
8. Continuous learning.

The overall workflow is:

Cloud + Edge Data

↓

Preprocessing

↓

Feature Engineering

↓

ML Model

↓

Anomaly Score

↓

Risk Classification

↓

Response

↓

Model Update

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

7. Data Collection

The framework collects information from multiple sources.

Network Data

- Packet rates.
- Protocol information.
- Source and destination information.
- Connection duration.
- Traffic volume.

System Data

- CPU usage.
- Memory usage.
- Disk utilisation.
- Process activity.

Application Data

- Response time.
- Request frequency.
- Error rates.
- Service availability.

Security Data

- Login attempts.
- Authentication failures.
- Access events.
- Firewall alerts.

8. Data Preprocessing

Raw data may contain:

- Missing values.
- Noise.
- Duplicates.
- Inconsistent formats.

Preprocessing may include:

1. Data cleaning.
2. Missing-value treatment.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

3. Normalisation.
4. Encoding.
5. Outlier filtering.
6. Temporal aggregation.

High-quality preprocessing improves model performance.

9. Feature Engineering

Relevant features can be divided into four categories.

Feature Category	Examples
Network	Packet rate, flow duration, bandwidth
System	CPU, memory, disk utilisation
Application	Latency, request rate, error rate
Security	Failed logins, access frequency

Feature selection is particularly important at the edge because computational resources can be limited.

10. Temporal Behaviour Modelling

Cloud and edge systems change over time.

For example:

Morning → Low Load

Working Hours → High Load

Night → Reduced Load

A static anomaly threshold could therefore produce false alarms.

ML models can instead learn temporal patterns.

11. Unsupervised Anomaly Detection

Unsupervised methods are useful when labelled attack datasets are limited.

Potential techniques include:

- K-means clustering.
- DBSCAN.
- Isolation Forest.
- Autoencoders.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

- Principal Component Analysis.

The model learns the structure of normal observations and identifies observations that deviate substantially.

12. Isolation Forest

Isolation Forest can identify unusual observations by examining how easily they can be isolated from other observations.

Anomalous observations generally require fewer partitioning steps.

This approach can be attractive for distributed environments because it can operate without requiring extensive labelled attack data.

13. Autoencoder-Based Detection

An autoencoder learns to reconstruct input data.

The basic process is:

Input Data

↓

Encoder

↓

Latent Representation

↓

Decoder

↓

Reconstructed Data

If reconstruction error becomes unusually high, the observation may represent an anomaly.

14. Supervised Classification

When labelled security datasets are available, classification algorithms can distinguish between normal and malicious activity.

Potential models include:

- Random Forest.
- Support Vector Machine.
- Gradient Boosting.
- Neural Networks.

Supervised learning can achieve strong performance for known attack categories.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

15. Hybrid Detection

A hybrid architecture combines:

Unsupervised Detection

for unknown anomalies

with

Supervised Classification

for known threats.

This can provide broader coverage.

16. Anomaly Score

Each observation can be assigned an anomaly score.

Conceptually:

Low Score → Normal

Moderate Score → Suspicious

High Score → Critical Anomaly

The score can be calculated from:

- Model confidence.
- Behavioural deviation.
- Threat indicators.
- Resource impact.

17. Adaptive Thresholding

Fixed thresholds can be ineffective in dynamic cloud–edge systems.

For example, 90% CPU usage may be:

- Normal during peak processing.
- Suspicious during an idle period.

Adaptive thresholds can account for:

- Time.
- Workload.
- Location.
- Service type.
- Historical behaviour.

18. Edge-Level Detection

Detection should occur close to data sources when low latency is important.

Edge-based detection can provide:

- Faster response.
- Reduced cloud traffic.
- Local decision-making.
- Greater operational resilience.

However, edge nodes have limited resources.

Therefore, lightweight models are often desirable.

19. Cloud-Level Analytics

Cloud infrastructure can perform more computationally intensive analysis.

Cloud-level functions include:

- Global anomaly correlation.
- Historical analysis.
- Model training.
- Cross-node threat detection.
- Long-term behavioural modelling.

20. Distributed Detection Architecture

The proposed architecture distributes intelligence across two major locations:

Edge

Rapid Detection

Cloud

Deep Analysis

This creates:

Edge Detection → Cloud Correlation → Coordinated Response

21. Collaborative Detection

An anomaly observed at one edge node may appear insignificant in isolation.

However, if several edge nodes report similar behaviour, the cloud may identify a coordinated event.

For example:

Edge A → Suspicious Traffic

Edge B → Suspicious Traffic

Edge C → Suspicious Traffic

↓

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

Cloud Correlation

↓

Potential Distributed Attack

22. Concept Drift

Cloud–edge systems evolve continuously.

Changes may occur because of:

- Software updates.
- New devices.
- Changing workloads.
- Network restructuring.
- User behaviour.

This phenomenon is known as concept drift.

Anomaly-detection models should therefore be periodically updated.

23. Continuous Learning

A continuous learning architecture can follow:

New Data

↓

Model Evaluation

↓

Drift Detection

↓

Model Update

↓

Validation

↓

Deployment

This allows the model to remain relevant as system behaviour changes.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

24. Threat Classification

Detected anomalies can be categorised into risk levels.

Risk Level	Example
Low	Minor deviation
Medium	Suspicious behaviour
High	Potential attack
Critical	Coordinated or severe attack

Risk classification helps prioritise response.

25. Intelligent Response

Once an anomaly is identified, the system can recommend or initiate appropriate actions.

Potential responses include:

- Blocking traffic.
- Isolating an edge node.
- Limiting resource access.
- Restarting compromised services.
- Increasing monitoring.
- Alerting administrators.

Automated actions should be carefully controlled for high-impact events.

26. Resilience-Oriented Architecture

Security detection should not simply identify threats.

It should also support continued operation.

The resilience cycle can be expressed as:

Detect → Assess → Respond → Recover → Learn

This makes anomaly detection part of a broader resilience strategy.

27. Fault Detection

Not all anomalies are cyberattacks.

ML can also identify:

- Hardware failures.
- Memory leaks.

- Network failures.
- Service degradation.
- Overloaded resources.

This enables unified security and reliability monitoring.

28. Predictive Failure Detection

Historical data can be used to predict potential failures.

For example:

Increasing Temperature

-

Increasing CPU Load

-

Increasing Error Rate

↓

Predicted Infrastructure Failure

Early warning allows preventive intervention.

29. Network Traffic Analysis

Network behaviour can be represented through:

- Flow frequency.
- Packet distribution.
- Connection patterns.
- Protocol usage.
- Communication relationships.

ML can detect deviations from established communication behaviour.

30. Lateral Movement Detection

Attackers may attempt to move from one compromised system to another.

Anomaly detection can identify:

- Unexpected node-to-node connections.
- Unusual authentication patterns.
- New communication pathways.

This can help limit attack propagation.

31. Distributed Denial-of-Service Detection

Sudden increases in:

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

- Requests.
- Packet rates.
- Connection attempts.

may indicate denial-of-service activity.

ML can distinguish between legitimate traffic spikes and abnormal behaviour by analysing historical patterns.

32. Resource Exhaustion Detection

Cloud and edge systems may experience resource exhaustion through:

- CPU abuse.
- Memory exhaustion.
- Storage saturation.
- Bandwidth consumption.

ML-based monitoring can detect unusual resource trajectories before complete service failure.

33. Container and Virtual Machine Monitoring

Modern cloud–edge environments commonly use:

- Containers.
- Virtual machines.
- Microservices.

Anomaly detection can monitor:

- Container behaviour.
- Resource consumption.
- Inter-service communication.
- Process activity.

34. Microservice Anomaly Detection

A failure in one microservice can propagate through dependencies.

AI can analyse:

Service A → Service B → Service C

and identify abnormal propagation patterns.

This supports early intervention.

35. Explainable Anomaly Detection

Security administrators need to understand why an alert was generated.

Instead of:

"Anomaly Detected."

the system should ideally report:

"Anomaly detected because network traffic increased 4× above the learned baseline while authentication failures increased significantly."

This improves trust and investigation efficiency.

36. False Positives

False positives are a major problem.

If legitimate activity is repeatedly classified as malicious, administrators may experience:

Alert Fatigue

This can result in genuine threats being overlooked.

Therefore, reducing false-positive rates is an important design objective.

37. Model Evaluation

Important performance metrics include:

Accuracy

Percentage of correctly classified observations.

Precision

Proportion of detected anomalies that are genuinely anomalous.

Recall

Proportion of actual anomalies successfully detected.

F1-score

Balance between precision and recall.

Detection Latency

Time required to identify an anomaly.

False-Positive Rate

Frequency of normal events incorrectly classified as anomalous.

38. Proposed Experimental Design

A potential evaluation can compare:

1. Rule-based detection.
2. Traditional ML.
3. Deep-learning detection.
4. Proposed hybrid cloud–edge model.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

Each approach can be evaluated under:

- Normal workload.
- High workload.
- Known attacks.
- Unknown anomalies.
- Network failures.
- Resource exhaustion.

39. Illustrative Evaluation Table

Model	Accuracy	Precision	Recall	F1-score	Detection Latency
Rule-Based	—	—	—	—	High
Random Forest	—	—	—	—	Medium
Autoencoder	—	—	—	—	Medium
Hybrid ML	—	—	—	—	Low

Note: Values should be populated using experimentally obtained results rather than fabricated data.

40. Edge Resource Constraints

Edge devices may have:

- Limited CPU.
- Limited memory.
- Limited storage.
- Restricted battery capacity.

Large deep-learning models may therefore be impractical.

Potential solutions include:

- Model compression.
- Quantisation.
- Knowledge distillation.
- Lightweight architectures.

41. Privacy-Preserving Learning

Sending all raw data to the cloud can create privacy risks.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

Federated learning provides an alternative.

The general approach is:

Local Data

↓

Local Model Training

↓

Model Update

↓

Central Aggregation

↓

Updated Global Model

This reduces the need to centralise raw data.

42. Federated Anomaly Detection

Federated learning can allow multiple edge nodes to collaboratively improve a model without directly sharing all local observations.

This is particularly useful for distributed infrastructure.

43. Adversarial Machine Learning

Attackers may deliberately manipulate data to deceive ML models.

Potential attacks include:

- Data poisoning.
- Evasion attacks.
- Model manipulation.

Therefore, ML-based security systems themselves require protection.

44. Model Security

A resilient anomaly-detection system should incorporate:

- Secure model storage.
- Authentication.
- Integrity verification.
- Secure model updates.
- Access control.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

45. Scalability

A cloud–edge environment may contain thousands or millions of devices.

The proposed framework should therefore support:

- Distributed inference.
- Hierarchical monitoring.
- Efficient data aggregation.
- Automated model deployment.

46. Proposed Research Framework

The proposed framework contains five major layers:

Layer 1 — Data Acquisition

Collect network, system and application data.

Layer 2 — Edge Intelligence

Perform rapid anomaly detection locally.

Layer 3 — Cloud Intelligence

Conduct global correlation and advanced analytics.

Layer 4 — Response

Prioritise and mitigate detected events.

Layer 5 — Learning

Continuously update models using new observations.

47. Conceptual Architecture

IoT / Edge Devices

↓

Edge Monitoring Agents

↓

Feature Extraction

↓

Lightweight ML Detection

↓

Local Response

↓

Cloud Aggregation

↓

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

Advanced ML Analytics

↓

Cross-Node Correlation

↓

Threat Intelligence

↓

Coordinated Response

↓

Continuous Learning

48. Research Objectives

The study aims to:

1. Develop a machine-learning model for cloud–edge anomaly detection.
2. Integrate network, system and application telemetry.
3. Reduce anomaly-detection latency.
4. Minimise false-positive rates.
5. Detect previously unseen behavioural deviations.
6. Support distributed edge-level inference.
7. Integrate cloud-based global correlation.
8. Improve infrastructure resilience.
9. Investigate adaptive learning under concept drift.
10. Evaluate the computational overhead of the proposed approach.

49. Research Questions

1. How effectively can ML identify anomalies across heterogeneous cloud–edge environments?
2. Which features provide the greatest contribution to anomaly detection?
3. Can distributed edge inference reduce detection latency?
4. How effectively can hybrid models detect unknown anomalies?
5. How can adaptive learning address concept drift?
6. What trade-offs exist between detection accuracy and edge computational overhead?
7. Can federated learning improve privacy while maintaining detection performance?
8. How can anomaly detection contribute to infrastructure resilience?
9. How can false positives be reduced?

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

10. What level of automation is appropriate for anomaly response?

50. Expected Contributions

The proposed study contributes:

Contribution 1

A unified cloud–edge anomaly-detection architecture.

Contribution 2

A hybrid ML strategy for known and unknown anomalies.

Contribution 3

Adaptive anomaly scoring for changing workloads.

Contribution 4

Distributed edge inference for low-latency detection.

Contribution 5

Cloud-based correlation for system-wide threat analysis.

Contribution 6

A resilience-oriented response mechanism.

51. Practical Applications

Potential applications include:

- Smart cities.
- Industrial IoT.
- Healthcare infrastructure.
- Autonomous systems.
- Financial platforms.
- Telecommunications.
- Smart manufacturing.
- Connected transportation.

52. Industrial IoT

Industrial systems require high availability.

Anomaly detection can monitor:

- Machines.
- Controllers.
- Industrial networks.
- Edge gateways.

Early detection can reduce both cyber and operational risks.

53. Healthcare Edge Computing

Healthcare systems may process sensitive data near patients.

AI-based anomaly detection can help monitor:

- Medical devices.
- Edge gateways.
- Clinical applications.
- Network traffic.

Privacy-preserving approaches are particularly important in this environment.

54. Smart Cities

Smart-city infrastructures involve:

- Traffic systems.
- Surveillance.
- Energy networks.
- Environmental sensors.

A distributed anomaly-detection system can identify abnormal behaviour across multiple urban subsystems.

55. Telecommunications

Telecommunication networks require continuous monitoring.

ML can identify:

- Traffic anomalies.
- Network congestion.
- Service degradation.
- Suspicious communication.

Edge intelligence can support rapid detection.

56. Limitations

The proposed model has several limitations.

First, ML performance depends heavily on data quality.

Second, rare attacks may be poorly represented.

Third, concept drift can reduce long-term accuracy.

Fourth, edge resources can restrict model complexity.

Fifth, automated response may create operational risks if anomalies are incorrectly classified.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

57. Future Research

Future work should investigate:

- Explainable AI.
- Federated anomaly detection.
- TinyML.
- Graph neural networks.
- Transformer-based time-series analysis.
- Digital twins.
- Adversarially robust ML.
- Self-adaptive detection.
- Autonomous incident response.

58. Discussion

Cloud–edge computing changes the nature of anomaly detection because infrastructure is increasingly distributed, dynamic and heterogeneous.

A centralised detection model may experience:

- High data-transfer requirements.
- Increased latency.
- Network dependency.
- Central bottlenecks.

A purely edge-based model, meanwhile, may lack sufficient computational power and global visibility.

The proposed hybrid architecture addresses this trade-off by dividing intelligence between edge and cloud.

Edge = Speed and Local Awareness

Cloud = Scale and Global Correlation

This combination can support a more resilient security architecture.

59. Toward Self-Adaptive Infrastructure

The ultimate objective is to develop systems capable of:

Observe → Detect → Understand → Respond → Recover → Learn

Such systems could continuously adapt to new operational conditions and emerging threats.

However, complete autonomy should be introduced gradually, particularly for critical infrastructure.

International Research Journal of Studies in Computer Science and Engineering
Volume: 01, Issue: 01, August 2026 | ISSN: XXXX.XXXX

60. Conclusion

Machine learning-based anomaly detection offers a promising approach for improving the resilience of cloud and edge computing systems. The distributed nature of modern computing environments creates complex behavioural patterns that cannot always be captured by static rules or traditional signature-based detection. ML can provide adaptive behavioural modelling capable of identifying both known and previously unseen deviations.

The proposed framework combines edge-level lightweight detection with cloud-level advanced analytics and cross-node correlation. Network, system, application and security telemetry are transformed into meaningful features, analysed through machine-learning models and converted into anomaly scores and risk classifications. Adaptive thresholds and continuous learning allow the framework to respond to changing workloads and concept drift. A major strength of the architecture is its resilience-oriented design. Rather than treating anomaly detection solely as a cybersecurity function, the proposed model integrates security monitoring with operational fault detection, predictive failure analysis and intelligent response. This enables the infrastructure to move from reactive monitoring toward proactive detection and recovery.

Nevertheless, challenges remain in data quality, model robustness, computational limitations, false positives, privacy, adversarial attacks and scale-up. Future systems should therefore integrate lightweight AI, federated learning, explainable models and secure adaptive mechanisms.

Overall, the integration of machine learning with distributed cloud–edge infrastructure provides a strong foundation for intelligent, low-latency and adaptive anomaly detection, supporting the development of more secure and resilient digital ecosystems.

References

1. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58.
2. Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 51(12), 3448–3470.
3. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.

4. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy, 305–316.
5. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. Military Communications and Information Systems Conference, 1–6.
6. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. IEEE International Conference on Data Mining, 413–422.
7. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507.
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
9. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646.
10. Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30–39.
11. Zhou, Z., Chen, X., Li, E., et al. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738–1762.
12. Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450–465.
13. Mothukuri, V., Khare, P., Parizi, R. M., et al. (2021). Federated learning-based cyber intrusion detection for IoT networks. IEEE Transactions on Network Science and Engineering, 8(4), 3173–3187.
14. McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralised data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 1273–1282.
15. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?” Explaining the predictions of any classifier. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144.