International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 14, Issue 11, November 2025)

An AI–IoT Integrated LSTM Framework for Intrusion and Anomaly Detection

R. Sukanya[1], M. P. Divya[2]

[1,2]*Assistant Professor, Department of Computer Science, Mahalashmi Women's College of Arts & Science, Avadi, Chennai, Tamilnadu, India*

***Abstract*— The rapid growth of the Internet of Things (IoT) has generated massive volumes of data, demanding intelligent, real-time processing to enable responsive and context-aware applications. This paper presents a practical framework for integrating Artificial Intelligence (AI) models into IoT systems with a focus on real-time decision-making and edge computing. We explore the design and deployment of lightweight machine learning and deep learning algorithms on resource-constrained IoT devices, aiming to minimize latency while preserving model accuracy. Implementation is validated through three real-world use cases such as industrial anomaly detection. Performance metrics such as response time, power consumption, and inference accuracy are analyzed to demonstrate the effectiveness of real-time AI integration. The results confirm that AI-enabled IoT systems can achieve low-latency, high-reliability performance, paving the way for more autonomous, intelligent, and scalable IoT solutions.**

***Keywords*— Artificial Intelligence, IoT, Deep Learning, Fog Computing, Security & Privacy, LSTM, Intrusion Detection, Anomaly Detection.**

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized the way devices interact by enabling seamless communication between sensors, actuators, and systems in domains such as healthcare, manufacturing, smart homes, and transportation. However, the exponential growth in the number of connected devices has led to an explosion in data volume, complexity, and velocity. Traditional cloud-based processing models often struggle to meet the strict latency, bandwidth, and energy constraints required for real-time decision-making in dynamic environments. To address these challenges, Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL), has emerged as a key enabler for intelligent automation and context-aware behavior in IoT systems. When combined with fog computing, AI can process data closer to the source, reducing latency and improving responsiveness. This research is motivated by the need to bridge this gap—by designing, implementing, and evaluating AI-powered frameworks that support real-time inference, low latency, and energy efficiency across diverse IoT environments.

II. CHALLENGES IN INTEGRATION OF AI-IOT

By leveraging advances in edge AI, we aim to develop practical solutions that enhance the intelligence, autonomy, and scalability of modern IoT applications. Real-time AI for IoT (Internet of Things) presents unique opportunities but also comes with a variety of challenges. These challenges affect multiple domains, including technical, infrastructural, and operational.

A. Limited Computational Resources

Many IoT devices (sensors, wearables, etc.) have limited processing power, memory, and energy.

B. Latency Constraints

Real-time decision-making requires low-latency processing.

C. Network Reliability and Bandwidth Limitations

IoT devices often operate in environments with unstable or limited connectivity.

D. Data Quality and Heterogeneity

IoT devices generate noisy, incomplete, and heterogeneous data (different formats, sampling rates).

E. Scalability

As the number of IoT devices grows, so does the volume of data and complexity of coordination also increased.

F. Security and Privacy

Real-time data from IoT often contains sensitive information (e.g., health, location).

G. Energy Consumption

Running AI algorithms, especially in real-time, is energy-intensive.

H. Real-Time Model Updating

AI models need to adapt to new data or concept drifts without downtime.

I. Integration and Interoperability

IoT ecosystems are often built with diverse platforms, protocols, and vendors.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 14, Issue 11, November 2025)

J. Cost and Deployment Complexity

Real-time AI systems are expensive and complex to design, deploy, and maintain.

III. Methods Used in AI-IoT Integration

AI-IoT (Artificial Intelligence and Internet of Things) integration is an evolving field where AI enhances the capabilities of IoT devices, and it provides the data that AI systems use to learn and make predictions. There are several methods to integrate AI with IoT, depending on the application, network architecture, and system design. Below are some common integration methods:

A. Cloud-Based AI-IoT Integration

In Cloud-Based method, IoT devices collect data and transmit it to the cloud, where AI algorithms (like machine learning or deep learning) analyze the data. The cloud performs the heavy computation and stores large amounts of data. It is mainly used in Smart home systems (e.g., Amazon Alexa), predictive maintenance in industries, autonomous vehicles, etc.

B. Edge AI-IoT Integration

In edge computing, AI algorithms are deployed on edge devices (such as gateways, routers, or even on IoT sensors themselves). The devices process the data locally without the need to send all the data to the cloud. It is used in Smart cameras with AI face recognition, autonomous drones, industrial robots, etc.

C. Fog Computing-Based AI-IoT Integration

Fog computing is a distributed computing paradigm that brings data storage, computation, and networking closer to the end-user devices. It is a middle layer between cloud and edge devices, ensuring faster processing and lower latency. It is mainly used in Smart cities, where IoT sensors collect data from vehicles and infrastructure, and AI in fog nodes analyzes this data in near real-time for traffic management.

IV. Security and Privacy in Fog Computing-Based AI-IoT Systems

In this paper, we discussed about Security and Privacy in Fog computing is a decentralized computing framework that extends cloud computing capabilities to the edge of the network, closer to the IoT devices. It acts as an intermediary layer between IoT devices (which generate data) and the cloud (which provides heavy processing power and storage).

In the context of AI-IoT integration, fog computing plays a critical role in enabling real-time data processing, reducing latency, and optimizing resource usage.

Fog computing introduces security concerns due to the decentralized nature of the system, including potential vulnerabilities at fog nodes, which might be deployed in less-secure locations than centralized cloud data centers. Furthermore, the combination of AI with IoT increases risks related to data privacy, integrity, and model robustness. Some of the Emerging Security and Privacy models are discussed below.

A. Decentralized Security Models:

Developing decentralized security architectures that ensure data integrity, confidentiality, and authenticity at the fog layer. This could involve encryption, secure data transmission, and identity management.

B. AI-Driven Security:

Leveraging AI for intrusion detection and anomaly detection to improve the security of fog nodes and IoT devices. By using machine learning to predict and detect cyber-attacks in real-time based on network traffic patterns.

C. Privacy-Preserving AI:

Investigating privacy-preserving techniques for AI in IoT systems, such as federated learning or differential privacy, to ensure that sensitive data is protected even when processed at the fog layer.

V. AI-Driven Intrusion Detection and Anomaly Detection

1) Purpose

To detect cyber-attacks in real time using intelligence at the fog layer.

2) Methodology Steps

- *Deploy ML/DL models at fog nodes **for:***
  - Network intrusion detection.
  - Malware detection.
  - Anomaly detection in IoT sensor data.
- *Common AI methods**:***
  - SVM, Random Forest, KNN, Naïve Bayes (lightweight models).
  - Deep learning: CNN, LSTM, Autoencoders.
- *Training approaches**:***
  - Centralized training in cloud + inference in fog.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 14, Issue 11, November 2025)

- Federated learning for privacy-preserving model training across distributed nodes.

- *Evaluation metrics:*
  - Accuracy, precision, recall, F1-score.
  - False positive rate.
  - Model latency and resource consumption at fog nodes.

AI-powered security is crucial for real-time threat detection.

VI. Comparison Of Deep Learning Models For Fog-Iot IDS

MODEL	ACCURACY	FOG SUITABILITY	USE CASE
CNN	High	Excellent	Fast IDS, traffic pattern detection
LSTM	Very High	Moderate	Sequential attack detection
GRU	High	Good	Lightweight temporal detection
AE	Medium–High	Excellent	Unsupervised anomaly detection
GAN	Very High	Low	Rare attack detection, data generation
GNN	Very High	Moderate	Network-structure-based IDS

VII. Long Short-Term Memory (LSTM) Intrusion Detection Algorithm

Recurrent Neural Network (RNN) designed to learn long-term dependencies, making it ideal for anomaly detection in IoT traffic, network logs, sensor sequences, or time-series data. Below is a generic algorithm for binary or multi-class intrusion detection using LSTM.

An LSTM cell has three key gates that regulate the flow of information:

A. Input Gate:

Input Gate decides how much new information should enter the cell state.

- A sigmoid layer determines which input values are important.
- A tanh layer creates candidate values C~t\tilde{C}_tC~t to potentially add to the cell state.
- The input gate multiplies these two outputs to update the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

B. Forget Gate

Forget Gate Controls what information should be removed from the previous cell state.

A sigmoid layer outputs a value between **0 and 1** for each part of the cell state.

0 → completely forget
1 → completely keep

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] +$$

C. Output Gate

Output Gate decides what part of the cell state should be output as the hidden state.

- A sigmoid layer chooses which parts of the cell state will be output.
- The updated cell state passes through a tanh activation to push values between −1 and 1.
- Both outputs are multiplied to produce the new hidden state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot tanh(C_t)$$

These gates use sigmoid and tanh activations to control what to keep, update, or output.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 14, Issue 11, November 2025)

A. Algorithm: LSTM-Based Intrusion Detection System (IDS)

1) Step 1 — Dataset Loading

Load dataset containing:

- Flow-based features (e.g., packet length, flags)
- Label: normal / attack types

2) Step 2 — Data Preprocessing

1. **Clean dataset** (remove NaN, duplicates)
2. **Encode categorical features**:
   - LabelEncoding / One-hot encoding
3. **Feature scaling** (MinMaxScaler):

   X′=X−XminXmax−XminX' = \frac{X - X_{min}}{X_{max} - X_{min}}X′=Xmax−XminX−Xmin

4. **Reshape data** for LSTM:
   - LSTM expects: **[samples, time_steps, features]**
   - Example: reshape to (N, 1, 40) for 40 features.

3) Step 3 — Split Dataset

Split into:

- 70% Training
- 15% Validation
- 15% Testing

4) Step 4 — Build LSTM Model

Architecture Example:

Input Layer → LSTM Layer → Dropout → Dense Layer → Softmax/Sigmoid Output

5) Model Steps

1. Add LSTM layer (e.g., 64 or 128 units)
2. Add Dropout (0.2–0.4) to prevent overfitting
3. Add Dense hidden layer(s)
4. Add output layer:
   - Sigmoid → Binary classification
   - Softmax → Multi-class attack classification

6) Step 5 — Compile Model

Use:

- Loss: binary_crossentropy or categorical_crossentropy
- Optimizer: Adam (best for IDS)
- Metrics: Accuracy, Precision, Recall, F1-score

7) Step 6 — Train the Model

Train using:

- Batch size: 32 or 64
- Epochs: 20–50

Monitor:

- Training loss
- Validation loss
- Accuracy trends

8) Step 7 — Evaluate Model

Use test data to calculate:

- Accuracy=(TP+TN)/(TP+TN+FP+FN)
- Precision=TP/(TP+FP)
- Recall=TP/(TP+FN)
- F1-score=2[(Precision*Recall)/(Precision+Recall)]

9) Step 8 — Deploy at Fog Node

Optimizations:

- Use smaller LSTM model
- Convert to TensorFlow Lite for edge/fog hardware
- Reduce time steps and input dimensions

10) Step 9 - Final Evaluation Metrics

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 14, Issue 11, November 2025)

Metric	Score
Accuracy	0.80
Precision	0.80
Recall	0.80
F1 Score	0.80

VIII. CONCLUSION

Long Short-Term Memory (LSTM) models play a crucial role in enhancing the security of Fog Computing-based AI-IoT environments by enabling accurate, real-time intrusion detection and anomaly detection. Their ability to learn long-term temporal dependencies makes them highly effective for analyzing sequential IoT traffic, sensor patterns, and evolving cyber-attack behaviors. Through mechanisms such as forget, input, and output gates, LSTMs overcome limitations of traditional RNNs and deliver robust detection of both known and zero-day attacks. By following a structured methodology—data preprocessing, feature scaling, sequential reshaping, model training, and evaluation—LSTM-based IDS systems can efficiently process IoT-generated time-series data. Although LSTM networks provide high detection accuracy, their computational requirements pose challenges for deployment on resource-constrained fog nodes. Overall, LSTM-based intrusion detection offers a powerful, scalable, and intelligent solution for safeguarding IoT-Fog-Cloud ecosystems against modern cyber threats, enabling secure, reliable, and resilient IoT systems.

IX. FUTURE ENHANCEMENT

Future research should aim to make LSTM-based IDS solutions lighter, more intelligent, more interpretable, and more adaptive, enabling secure, scalable, and robust Fog-IoT ecosystems capable of handling modern cyber threats.

REFERENCES

[1] B. Jansi and V. Sumalatha, ‘‘The security constructions and enhancements of smart wearable devices in modern technologies and health monitoring system,’’ in Computational Intelligence for Clinical Diagnosis. Cham, Switzerland: Springer, 2023, pp. 461–471.

[2] M. A. Kachouei, A. Kaushik, and M. A. Ali, ‘‘Internet of Things-enabled food and plant sensors to empower sustainability,’’ Adv. Intell. Syst., vol. 5, no. 12, Dec. 2023, Art. no. 2300321.

[3] F. Iqbal, M. I. Satti, A. Irshad, and M. A. Shah, ‘‘Predictive analytics in smart healthcare for child mortality prediction using a machine learning approach,’’ Open Life Sci., vol. 18, no. 1, Jul. 2023, Art. no. 20220609.

[4] S. Subramani and M. Selvi, ‘‘Intelligent IDS in wireless sensor networks using deep fuzzy convolutional neural network,’’ Neural Comput. Appl., vol. 35, no. 20, pp. 15201–15220, Jul. 2023.

[5] C. Thiagarajan and P. Samundiswary, ‘‘Enhanced RPL-based routing with mobility support in IoT networks,’’ in Proc. 2nd Int. Conf. Adv. Comput. Intell. Commun. (ICACIC), Dec. 2023, pp. 1–4.

[6] H. A. Hassan, E. E. Hemdan, W. El-Shafai, M. Shokair, and F. E. A. El-Samie, ‘‘Intrusion detection systems for the Internet of Thing: A survey study,’’ Wireless Pers. Commun., vol. 128, no. 4, pp. 2753–2778, Feb. 2023.

[7] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, ‘‘A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges,’’ J. Inf. Intell., vol. 2, no. 6, pp. 455–513, Nov. 2024.

[8] A. Berguiga and A. Harchay, ‘‘An IoT-based intrusion detection system approach for TCP SYN attacks,’’ Comput., Mater. Continua, vol. 71, no. 2, pp. 3839–3851, 2022.

[9] O. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, ‘‘A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions,’’ Electronics, vol. 12, no. 6, p. 1333, Mar. 2023