ITM Web of Conferences **79**, 01042 (2025) https://doi.org/10.1051/itmconf/20257901042
KEIS-2025

IntruDet-LSTM: A Knowledge-Driven Hybrid Intrusion Detection System for IoT Cybersecurity

Baker Karim[1], *Aruna*. M[2*], *K.* Deepika[3], *G.* Srinivasalu[4], and *Sandeep* Kumar[5]

[1]Department of Computers Techniques Engineering, College of Technical Engineering, The Islamic University, Najaf, Iraq
[2]Department of Electrical and Electronics Engineering, Nitte Meenakshi Institute of Technology, Nitte (Deemed -to-be University), Bengaluru, India
[3]Kakatiya Institute of Technology and Science, Warangal, India
[4]Department of Electronics and Communication Engineering, Cambridge Institute of Technology, Bengaluru, India
[5]Department of Computer Science and Engineering, Cambridge Institute of Technology, Bengaluru, India

Abstract. In recent times, Internet of Things (IoT) ecosystem is rapidly expanding, with a flow in various devices being integrated to allow continuous and efficient communication. Most IoT devices are resource-constrained, and without clearly defined security standards, their communications remain exposed to potential risks. As a result, quickly identifying threats within IoT networks is critical, making Intrusion Detection Systems (IDS) an essential component of modern cybersecurity strategies. The unpredictable behavior of IoT traffic demands dynamic and context-sensitive rule configurations. Software Defined Networks (SDN's) is programmable architecture enables real-time threat justification across heterogeneous IoT environments. The proposed IntruDet-LSTM which is Intrusion Detection with Long Short-Term Memory method introduces a hybrid system for intrusion detection and dynamic rule-based configuration, combining a signature-based SNORT method with a data-driven ensemble model built on LSTM. Fault tolerance is achieved through a dual-layer design, where the intrusion detection and rule configuration models are dissociated, enabling uninterrupted performance even when one layer is compromised. IntruDet-LSTM method effectively reduces false alarms, allowing true IoT traffic to flow continuous and still delivering high detection accuracy. The proposed IntruDet-LSTM achieves accuracy of 99.8%, which is better than existing Deep Integrated Stacking for the IoT (DIS-IoT).

1 Introduction

The Internet of Things (IoT) introduces a new computing concept and actively supports applications in transportation, smart homes, healthcare, industrial automation, security, supply chain management, and blockchain monitoring [1]. As cyber and physical systems combine, especially in critical infrastructure, the complexity and possibility of threats increase, leading to greater risks to those responsible for their safe and reliable functioning [2]. IoT devices remain vulnerable to cyber threats, mainly because manufacturers stress low cost and user convenience while dominating robust security measures [3]. Data security focuses on building models that protect the confidentiality, integrity, and availability of networks [4]. Intrusion detection systems (IDSs) are tools for detecting and classifying cyber-attacks within hosts or network environments by applying algorithmic techniques [5]. In IoT environments, traditional intrusion detection systems often employ machine learning algorithms such as random trees, k-nearest neighbors (KNN), and Naïve Bayes, which are limited when dealing with high-dimensional datasets [6]. Despite their computational complexity, Deep Learning (DL) methods are commonly used in machine-learning techniques owing to their superior performance [7]. The growing reach of the Internet and mobile connectivity to increased online activity and greater reliance on cloud-based data storage [8]. The integration combines various technologies, including network communication, IoT, Artificial Intelligence (AI), extended and virtual reality, and blockchain, to enable advanced digital environments [9]. The low frequency of attack traffic compared to normal traffic makes it challenging for IDS systems to precisely differentiate between malicious and benign activities [10].

The contribution of the proposed IntruDet-LSTM is as follows:

- The hybrid IntruDet-Short-Term Memory (LSTM)architecture enhances the fault tolerance by combining deterministic and sequential anomaly detection.
- The Host Status Rating mechanism improves decision reliability by resolving conflicts using historical packet behavior.
- SDN-based rule verification and impact scoring enable dynamic quality-driven rule deployment with minimal false positives.

* Corresponding author: aruna.m@nmit.ac.in

In following Section 2 discuss the literature survey of different authors, Section 3 describes the problem statement solved in the proposed method in Section 4. Section 5 and 6 discuss the experimental results and conclusions of this paper.

2 Related Works

Amritpal Singh et al. [11] developed SecureFlow, a hybrid IDS and dynamic rule configuration method designed for Software Defined Network (SDNs) IoT environments. The developed SecureFlow method integrates both the knowledge-based Snort method and data-driven Support Vector Machine (SVM) systems to identify anomalies and attacks in network traffic. Once threats are detected, SecureFlow dynamically generates, verifies, prioritizes, and deploys flow rules using an SDN controller to mitigate incidents and restore the network performance. The developed method also includes a host rating mechanism and an impact calculator for rules to measure the efficiency of deployed rules and improve the pending responses. The main purpose of SecureFlow is to propose a fault-tolerant adaptive security solution capable of handling both known and unknown threats in heterogeneous IoT networks. However, SecureFlow was tested in limited attack situations and rule sets, which did not fully signify the range and complexity of real-world IoT threats.

Riccardo Lazzarini et al. [12] introduced Deep Integrated Stacking for IoT (DIS-IoT), a stacking-based ensemble method designed to enhance intrusion detection in IoT environments. The introduced DIS-IoT model integrates four distinct DL models: Multilayer Perceptron (MLP), Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) into a combined architecture, where each base learner independently classifies network traffic, and their outputs are fused through a fully connected meta-learner layer. Each model was trained independently on the same input data to capture the different aspects of network traffic. Once trained, their predictions were passed to a fully connected meta-learner layer, which integrated the outputs and performed the final classification. However, the introduced DIS-IoT lacks adaptability to real-time updates, making it less suitable for dynamic or evolving threat environments.

Alireza Zohourian et al. [13] employed an IoT-PRIDS (Packet Representation Intrusion Detection System for IoT), which is a lightweight host-based anomaly detection system that outlines IoT devices by converting network packets into simplified representations using communication patterns, service types, and header fields. During the profiling phase, IoT-PRIDS builds a baseline for normal packet representations using only mild traffic. In the monitoring phase, incoming packets are compared with the baseline using the Hamming distance to detect anomalies. However, IoT-PRIDS is sensitive to unseen devices or traffic patterns, and if a device's behavior was not effectively profiled, the system generates a high number of false positives.

Afrah Gueriani et al. [14] developed a CNN-LSTM IDS to enhance IoT security by combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. The process begins with data preprocessing and binary label encoding, followed by training on the CICIoT2023 dataset using a layered architecture such as CNN layers to extract spatial features from network traffic, while LSTM layers capture temporal dependencies. The results are combined using dense layers and a final softmax classifier to differentiate between benign and malicious traffic. However, the developed CNN-LSTM provides more false positive rates, indicating occasional misclassification of benign traffic as malicious, potentially leading to unnecessary alerts in real-world deployments.

Mansi Bhavsar et al. [15] introduced Pearson-Correlation Coefficient-Convolutional Neural Networks (PCC-CNN), an IDS for network anomalies in Internet of Things (IoT) applications. The process begins by extracting important features using linear-based techniques and then feeding these features into a convolutional neural network. The introduced PCC-CNN allows for both binary classification, which simply detects whether an anomaly exists, and multiclass classification, which identifies various types of attacks. The introduced PCC-CNN method addresses the shortcomings of traditional security measures, such as signature- and rule-based methods, which are inefficient for securing IoT devices owing to their limited bandwidth and global connectivity. However, PCC-CNN is heavily dependent on the effectiveness of this initial linear feature extraction, which was not sufficient to capture all nuanced attack patterns in modern, highly complex IoT networks.

3 Problem Statement

Existing IDS in IoT environments struggle with high false positive rates, limited adaptability to evolving threats, and lack fault tolerance, especially when handling dynamic, heterogeneous traffic. The proposed IntruDet-LSTM developed a hybrid SDN-enabled IDS that integrates signature-based and temporal anomaly detection to ensure accurate real-time threat identification and resilient rule deployment.

4 Proposed Method

IntruDet-LSTM introduces a two-tier system for intrusion detection and dynamic rule configuration adapted to SDN-enabled IoT environments. The proposed IntruDet-LSTM enhances the SecureFlow architecture by embedding an (LSTM) model within the data-driven intrusion detection layer. Temporal modeling allows for a more accurate identification of sequential anomalies and supports adaptive responses to evolving network threats, as shown in Figure 1.

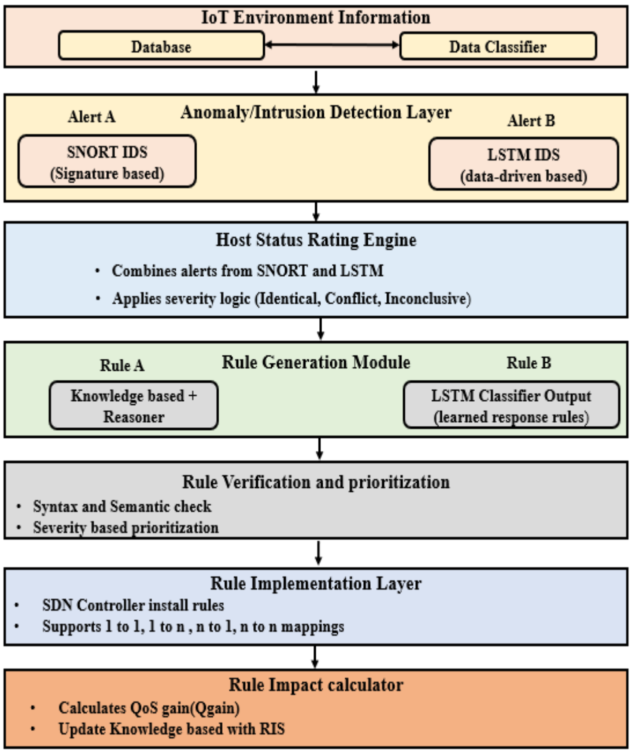

Fig. 1. The overall architecture of proposed IntruDet-LSTM method.

4.1 IoT Environment Information

Performance metrics and device statistics from the IoT ecosystem are collected and organized using a particular element. Core elements include:

4.1.1 Database

Maintains structured records of both device-level and network-level data.

4.1.2 Data Classifier

Categories incoming information to support intrusion detection, rule generation, and verification processes

4.2 Anomaly/Intrusion Detection layer

The core of IntruDet-LSTM is placed on a dual-layer IDS structured as follows:

4.2.1 Knowledge-based IDS (SNORT)

Algorithm 1: SNORT-based IDS

- Input: Packets in traffic (P), Header packet (H_T), Rule:A_S
- Output: Alerts:(B, B_H), Memory of buffer:(KN_R^{LE}), $Label\ of\ traffic$: (P_M^{LE})

while (P≠null) do
    DECODE ⟶ P
    CHECK ⟶ H_T
    if H_T ==ANOMALOUS then
        GENERATE ALERT ⟶ H_B
        CALL SDN Controller
        EXECUTE ⟶ Configuration of Dynamic Systems
        STORE MAC Host and P_M^{LE} ⟶ KN_R^{LE}
    else
        PROCEED to next step
    end if
    PRE-PROCESS ⟶ P
    MATCH P ⟶ A_S
    if (TRUE) then
        GENERATE ALERT ⟶ B_P
        CALL SDN Controller

EXECUTE $\longrightarrow$ Dynamic Configuration System
STORE MAC Host and $P_M^{LE} \longrightarrow KN_R^{LE}$
else
FOLLOW Flow Table; FORWARD to next hop
end if
end while

Signature-based detection operates through predefined rules that identify known threats by matching specific patterns within network traffic. The process begins with a Packet Decoder, which interprets incoming data streams, followed by preprocessors that normalize and prepare the data for analysis. The Detection Engine then scans the packet headers and loads for signatures associated with malicious activities. When a match was found, the Alert Generator activated and flagged the event for further investigation. This approach ensures rapid identification of familiar attack vectors within SDN-enabled IoT environments. The Knowledge-based IDS process, as presented in Algorithm 1, begins when a packet of traffic (P) is received from a designated host. This packet is directed to a snort-based environment to determine whether it is benign or malicious. The packet decoder initiates the process by generating a detailed profile of P and extracting its header (H_T) for anomaly inspection. If H_T is flagged as anomalous, an alert (B_H) is generated and the SDN controller is notified to execute the dynamic system configuration. The MAC address of the host and the corresponding label of traffic (P_M^{LE}) are stored in buffer memory (KN_R^{LE}).

4.2.2 Data-driven IDS (LSTM)

LSTM is used to capture the temporal dependencies in IoT traffic. Network traffic is first transformed into time-series sequences to preserve temporal patterns and dependencies. Labeled datasets that distinguished between normal and malicious behaviors are then used to train the LSTM model.

- Learning Phase: The model begins by identifying the input and output domains and mapping them using a function to extract meaningful outputs. The input data are represented as feature vectors and transformed into a higher-dimensional space for better separation. The model iteratively adjusts to minimize classification errors and organizes the data points into distinct categories.
- Classification Phase: The model trained and evaluated to measure its classification accuracy. During this process, a prediction file s produced, which is later utilized for further analysis and postprocessing activities.
- Post-processing phase: The prediction result is measured to determine how many instances were classified accurately and how many misclassified. Based on these results, the precision and recall values re computed to evaluate how effectively the model identified and distinguished between normal and malicious traffic.
- After completing the training phase, the proposed model actively monitors the network traffic, detects unusual patterns in real time, and issues alerts when suspicious behavior is observed. Using temporal analysis, the data-driven method successfully identified long-term trends, reduced incorrect alerts, and adjusted to evolving traffic conditions. As outlined in Algorithm 2, the LSTM-based intrusion detection system begins by receiving traffic (F) from multiple hosts. Each packet is then analyzed by the trained model, which classifies it as either normal (N) or malicious (P), thereby ensuring accurate and timely threat detection. The traffic is identified as malicious, and an alert (R_F) is generated, prompting the SDN controller to initiate a Dynamic Configuration System. The traffic label (L_T^{CC}) and MAC address of the host and the corresponding are then stored in buffer memory (MB_R^{CC}) for future reference and analysis.

Algorithm 2: Data-driven (LSTM)-based IDS

- Input: Traffic: F
- Output: Alert: (R), Memory of a buffer: (MB_R^{CC}), Traffic Label (L_T^{CC}).

while (F$\neq$ null) do
DECODE $\longrightarrow$ F
PRE-PROCESS $\longrightarrow$ F
MAPPING F $\longleftrightarrow$ Trained model
FETCH Class of F: {P, Q}
if (F==N) then
Generate ALERT $\longrightarrow R_F$
CALL SDN Controller
EXECUTE $\longrightarrow$ Dynamic Configuration System
STORE Host MAC and $L_T^{CC} \longrightarrow MB_R^{CC}$

```
        else
            FOLLOW Table Flow; FORWARD to next hop
        end if
end while
```

4.3 Host Status Rating Mechanism

Host reliability is measured by integrating decisions from both Snort and LSTM-based detection systems. When both systems produced identical classifications, either malicious or normal, the results is treated with high confidence. In cases of conflicting results, historical packet behavior and a predefined host rating formula are applied to resolve ambiguity. If neither system provides a clear decision, the final decision defaults on the available IDS output, confirming continuity in threat assessment.

4.3.1 Identical Decisions

After examining the incoming traffic, the signature-based and anomaly based IDS components independently arrived at the same classification as either malicious or normal. This consistent result in both layers reinforces the reliability of the ensemble IDS and validates its unified decision making process. The host MAC (M_c) and labels produced L_T^{CC} and L_T^{BC} are placed in MB_R^{CC} and MB_R^{BC} respectively.

4.3.2 Incompatible Decisions

Here, the two IDS components provide conflicting calculations; one identifies the traffic as malicious, whereas the other considers it normal. To resolve this difference, the system reviews a set of packets previously transmitted from the host. The final verdict is then based on the behavior and classification of the earlier packets. The host rating (R_{MAC}^H) formula is defined by Equation (1): If $R_{MAC}^H > 0$, the chosen H_{MAC} is suspicious and is blocked and referred for further inspection by a security expert.

$$R_{MAC}^H = \frac{L_{BC} - L_{CC}}{2} \quad (1)$$

4.4 Rule Generation System

Flow rule generation is initiated in response to intrusion detection alerts, leveraging a hybrid reasoning mechanism to determine the appropriate network actions.

4.4.1 Knowledge-based rule generator

Predefined actions are selected based on domain-specific insights and labeled traffic characteristics. When a suspicious payload is detected, the system redirects the associated traffic to a honeypot host for containment and further analysis. The knowledge-based rule generator method confirms that responses are contextually informed and aligned with established threat-handling strategies, thereby enhancing both precision and operational security within SDN-enabled IoT environments. When a PACKET_IN matches a predefined rule, the PACKET_IN signal is activated and is sent to the controller. Based on the traffic label, the controller determines an appropriate action and generates a new rule. This rule is then sent back to for the relevant switches in the network, PACKET_OUT. If the analyzer of traffic detects an irregular packet size, the traffic is redirected to a honeypot for closer inspection. If multiple messages are in excess from the same host, they are queued for further analysis, and the packet payloads are examined. Based on this, the system takes preventive actions to moderate potential denial-of-service attacks. Similarly, if identical messages are multicast to multiple destinations, packets are temporarily held. The system then checks the priority of the queued messages and forwards the traffic accordingly to maintain network integrity.

4.4.2 Data-Driven Rule Generator (LSTM)

LSTM-based predictions guide the dynamic generation of flow rules, with continuous learning from past incidents, enhancing the precision and adaptability of rule selection over time.

4.5 Rule Deployment System

Effective implementation of generated flow rules involves a multistep process that begins with verification, where syntax, semantics, and header-action mappings are systematically checked for consistency and correctness. Once validated, rules are prioritized according to threat severity, with critical alerts assigned the highest execution priority. The final deployment is handled by the SDN controller, which applies rules across network paths using appropriate configurations such as one-to-one or one-to-many mappings.

4.6 Rule Verification

In the proposed method, all rules are defined and validated prior to deployment and each generated rule undergoes a syntax and semantic check to ensure that it meets the required standards and adheres to the correct structural format. Only rules that passed this verification process are considered for implementation. The traffic labels (TL) and corresponding actions (Ac) are categorized as L_1, L_2, L_3, L_4 and A_1, A_2, A_3, A_4, respectively. Each label is mapped to a specific action. If the mapping is incorrect, the rule fails in the verification process. An incorrect mapping between the traffic label and action resulted in a failed verification. First, the rule header flow (RH_F) was retrieved to confirm that all required header fields present and valid. If the header passes this check, TL is matched to its corresponding action label. A successful match marks the rule as verified successful (VS), while a mismatch leads to a Verification Failed (VF) status. In cases of failure, the SDN controller is alerted and the rule is discarded. If the rule passes verification, it is executed by the controller with its implementation.

4.7 Rule Impact Calculator

The effectiveness of the deployed flow rules is assessed using Quality of Service (QoS) metrics, with an Impact Score derived from the observed performance improvements following rule implementation. A continuous feedback loop updates the system knowledge base, enabling more informed and adaptive rule selection in upcoming deployments. If the Quality of Service (QoS) improves relative to the level of the previous incident, it means that the rule generated is effective. The performance gain (Q_{gain}) is calculated using Equation (2), where Q_{AS}, Q_{CD}, Q_{RS} are the superior metric values in % through the incident, later ruler reply, and earlier incident, respectively.

$$Q_{gain} = \frac{Q_{AS} - Q_{CD}}{Q_{RS}} \quad (2)$$

5 Experimental Results

The proposed IntruDet-LSTM method works well with a simple and efficient setup that runs smoothly on a Linux-based machine such as Ubuntu 18.04, with basic tools such as Java 1.8, Apache Ant, Maven, Python 3.9, and Snort 2.9.7.0. The SDN controller used is Floodlight, which helps manage the network traffic. For hardware, a system with an Intel Core i7 processor, 16–32 GB RAM, and a mid-range GPU, such as NVIDIA RTX 2060, is suitable. This setup supports both signature-based detection using Snort and ML-based classification using LSTM. IntruDet-LSTM uses an SDN simulation platform to emulate and efficiently handle IoT traffic. It detects anomalies, generates rules, and restores the network performance without requiring high-end resources.

5.1 Evaluation Metrics

Evaluating the proposed IntruDet-LSTM model is essential to understand its performance, and the standards differ depending on the model design and planned use, which are used to evaluate the model's performance as shown in Equations (3) to (6).

- Accuracy: The accuracy reflects the overall correctness of the model, which is the ratio of correctly classified instances (both normal and attack) to the total number of instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

- Precision: Precision is defined as the number of instances projected as an attack. High precision in IntruDet-LSTM implies that fewer false alarms are important for reducing unnecessary actions.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

- Recall: Recall indicates the number of actual attacks that are correctly identified. The IntruDet-LSTM objective is for high recall to ensure that it does not miss real threats, which is especially critical in security-sensitive IoT environments.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

- F1-Score: The F1 Score balances precision and recall and offers a single metric that considers both false positives and false negatives. IntruDet-LSTM uses the F1 Score to evaluate its overall robustness, particularly when attack classes are rare or unevenly distributed.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

5.2 Performance Evaluation

To evaluate the efficiency of the proposed IntruDet-LSTM model, a comprehensive performance evaluation for comparative analysis with state-of-the-art methods, as shown in Figure 2. The evaluation highlights critical performance metrics, such as accuracy, precision, recall, and F1-score, to provide a balanced understanding of detection capability, reliability, and robustness against diverse attack patterns.

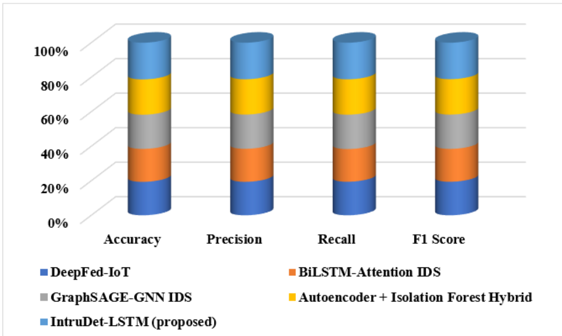

Fig. 2. Quantitative analysis of state of art methods.

5.3 Time Complexity

To prove that the proposed IntruDet-LSTM method is not only accurate but also computationally feasible for real-world deployment, the time complexity of each core component is analyzed. Table 1 shows the algorithmic efficiency for both signature-based and sequential anomaly detection layers, rule generation, and SDN controller operations. By measuring the per-packet processing cost and overall system overhead, the results show that IntruDet-LSTM maintains high detection performance without compromising scalability or responsiveness in resource-constrained IoT environments.

Table 1. Time complexity of each component in IntruDet-LSTM.

Stages	Variables	Time Complexity
Packet decodes (Snort)	Per packet	O (1)
Pre-processing	Per packet	O (f)
Signature match (Snort)	Per packet, r rules	O(r)
Alert + Buffer	Per alert	O (1)
LSTM inference (per sequence)	T, d, h	$O(T(dh+\boldsymbol{h}^2)$
SDN rule generation	Per rule	O (1)
SDN rule verification	Per rule	O (1)
Controller deployment	Network level	O (u)
Overall, per packet	n, r, f	O (n (f + r))

5.4 Ablation Study

To validate the individual contributions of each architectural component within the IntruDet-LSTM, a complete ablation study is conducted. The ablation analysis separates and evaluates the performance of key modules, including SNORT-based detection, LSTM-driven anomaly classification, Host Status Rating, SDN rule verification, and QoS-based rule impact scoring. By comparing these components against the fully integrated system, Figure 3 shows how each element enhances the detection accuracy, reduces false positives, and contributes to the overall robustness of the proposed method.

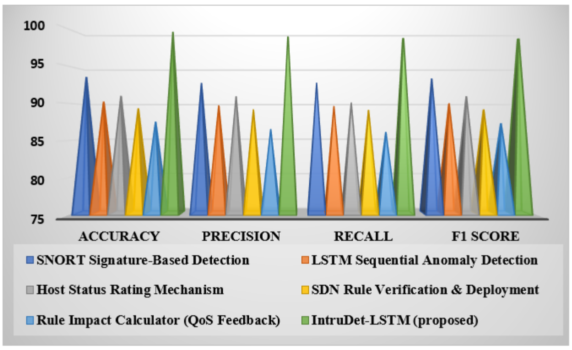

Fig. 3. Evaluation of each component of proposed IntruDet-LSTM.

5.5 Comparison Study

To support the effectiveness of the proposed IntruDet-LSTM framework, a comparison with existing intrusion detection methods. This includes models such as SecureFlow [11], DIS-IoT [12], IoT-PRIDS [13], and CNN-LSTM [14]. The evaluation presented in Table 2 is based on the commonly used metrics of accuracy, precision, recall, and F1-score to highlight differences in detection performance and reliability.

Table 2. Comparison evaluation with existing methods.

Model	Accuracy	Precision	Recall	F1 Score
SecureFlow [11]	99.5	94	98.6	96
DIS-IoT [12]	99.7	99.7	99.7	99.7
IoT-PRIDS [13]	98.74	93.84	99.71	95.29
CNN-LSTM [14]	97.46	97.17	97.15	97.05
IntruDet-LSTM (proposed)	99.8	99.10	98.88	98.81

5.6 Discussion

The experimental results show that the proposed IntruDet-LSTM method performs significantly better than the existing IDS models. The superior performance is attributed to its hybrid architecture, which integrates SNORT-based signature detection with LSTM-driven anomaly classification. The dual-layer design confirms fault tolerance and minimizes false positives, whereas the SDN-enabled rule deployment system enhances adaptability to dynamic IoT traffic. Compared to models such as DIS-IoT and SecureFlow, IntruDet-LSTM exhibits improved robustness and real-time responsiveness. The ablation study further confirmed that each component's host rating, rule verification, and QoS-based impact scoring contribute meaningfully to the overall consistency of the systems. These findings validate IntruDet-LSTM as a scalable high-performance IDS for heterogeneous IoT environments. However, certain challenges remain that open paths for future improvement, where IntruDet-LSTM faces challenges when scaling to extremely large or encrypted IoT networks, where traffic patterns are more complex and difficult to classify.

6 Conclusion

The IntruDet-LSTM method provides a reliable and scalable solution for intrusion detection in IoT networks. By combining SNORT-based signature analysis with LSTM-driven anomaly classification, the proposed method achieves high accuracy and low false-alarm rates for various traffic conditions. Each component's host rating, rule verification, and QoS-based impact scoring improve the value of the detection process, as confirmed by the ablation study. Compared to existing models, such as DIS-IoT and SecureFlow, the proposed system shows improved responsiveness, fault tolerance, and adaptability to dynamic environments. The SDN-enabled rule deployment mechanism supports real-time updates, making IntruDet-LSTM suitable for resource-constrained and privacy-sensitive settings. With its modular design and interpretable outputs, the system aligns well with the practical needs for secure IoT deployments. The results highlight its potential for real-world applications and future research in privacy-preserving AI-driven security systems in terms of accuracy, precision, recall, and F1-score, achieving

99.8%, 99.1%, 98.88%, and 98.81%, respectively. Overall, IntruDet-LSTM is a robust method for next-generation IDS systems, presenting both technical depth and practical importance for evolving network landscapes. Future work can be extended by integrating federated learning to enhance privacy across distributed devices, and transformer-based architectures to better capture long-range dependencies in traffic sequences.

References

1. A. Awajan, A novel deep learning-based intrusion detection system for IoT networks. Computers **12**, 34 (2023). https://doi.org/10.3390/computers12020034
2. S.C. Phillips, S. Taylor, M. Boniface, S. Modafferi, M. Surridge, Automated knowledge-based cybersecurity risk assessment of cyber-physical systems. IEEE Access **12**, 82482-82505 (2024). https://doi.org/10.1109/ACCESS.2024.3404264
3. H. Yi, S. Zhang, D. An, Z. Liu, PatchesNet: PatchTST-based multi-scale network security situation prediction. Knowl.-Based Syst. **299**, 112037 (2024). https://doi.org/10.1016/j.knosys.2024.112037
4. J.B. Awotunde, F.E. Ayo, R. Panigrahi, A. Garg, A.K. Bhoi, P. Barsocchi, A multi-level random forest model-based intrusion detection using fuzzy inference system for Internet of Things networks. Int. J. Comput. Intell. Syst. **16**, 31 (2023). https://doi.org/10.1007/s44196-023-00205-w
5. A. Henry, S. Gautam, S. Khanna, K. Rabie, T. Shongwe, P. Bhattacharya, B. Sharma, S. Chowdhury, Composition of hybrid deep learning model and feature optimization for intrusion detection system. Sensors **23**, 890 (2023). https://doi.org/10.3390/s23020890
6. T. Yang, J. Chen, H. Deng, B. He, A lightweight intrusion detection algorithm for IoT based on data purification and a separable convolution improved CNN. Knowl.-Based Syst. **304**, 112473 (2024). https://doi.org/10.1016/j.knosys.2024.112473
7. S.E. Maoudj, A. Belghiat, A deep learning-based approach with two-step minority classes prediction for intrusion detection in Internet of Things networks. Knowl.-Based Syst. **312**, 113143 (2025). https://doi.org/10.1016/j.knosys.2025.113143
8. Q. Liu, D. Wang, Y. Jia, S. Luo, C. Wang, A multi-task based deep learning approach for intrusion detection. Knowl.-Based Syst. **238**, 107852 (2022). https://doi.org/10.1016/j.knosys.2021.107852
9. T. Gaber, J.B. Awotunde, M. Torky, S.A. Ajagbe, M. Hammoudeh, W. Li, Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks. Internet Things **24**, 100977 (2023). https://doi.org/10.1016/j.iot.2023.100977
10. J. Zhang, R. Chen, Y. Zhang, W. Han, Z. Gu, S. Yang, Y. Fu, MF2POSE: Multi-task feature fusion pseudo-siamese network for intrusion detection using category-distance promotion loss. Knowl.-Based Syst. **283**, 111110 (2024). https://doi.org/10.1016/j.knosys.2023.111110
11. A. Singh, P.K. Chouhan, G.S. Aujla, SecureFlow: Knowledge and data-driven ensemble for intrusion detection and dynamic rule configuration in software-defined IoT environment. Ad Hoc Netw. **156**, 103404 (2024). https://doi.org/10.1016/j.adhoc.2024.103404
12. R. Lazzarini, H. Tianfield, V. Charissis, A stacking ensemble of deep learning models for IoT intrusion detection. Knowl.-Based Syst. **279**, 110941 (2023). https://doi.org/10.1016/j.knosys.2023.110941
13. A. Zohourian, S. Dadkhah, H. Molyneaux, E.C.P. Neto, A.A. Ghorbani, IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks. Comput. Secur. **146**, 104034 (2024). https://doi.org/10.1016/j.cose.2024.104034
14. A. Gueriani, H. Kheddar, A.C. Mazari, Enhancing IoT security with CNN- and LSTM-based intrusion detection systems. In 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), IEEE, EL OUED, Algeria, April 24-25 (2024), 1-7
15. M. Bhavsar, K. Roy, J. Kelly, O. Olusola, Anomaly-based intrusion detection system for IoT application. Discover Internet Things **3**, 5 (2023). https://doi.org/10.1007/s43926-023-00034-5