Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Leveraging Deep Learning for Anomaly-Based Intrusion Detection in Internet of Things Networks: An LSTM Approach with Recursive Feature Elimination

Assia TEBIB[1], Mohamed Ali BOUANAKA[2], Oumeima BOUBERTAKH[3], Billel KENIDRA[4]
[1]*LIRE Laboratory, University of Constantine 2 Abdelhamid Mehri, Constantine, Algeria*
[2]*LIRE Laboratory, University of Constantine 2 Abdelhamid Mehri, Constantine, Algeria*
[3]*LIRE Laboratory, University of Constantine 2 Abdelhamid Mehri, Constantine, Algeria*
[4] *LIRE Laboratory, University of Constantine 1, Constantine, Algeria*

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Background: The accelerating deployment of Internet of Things (IoT) devices has enlarged the cyber-attack surface of contemporary networks, while the resource-constrained nature of typical IoT endpoints precludes the use of conventional, signature-based defences. Anomaly-based Intrusion Detection Systems (IDS) powered by deep learning offer a promising alternative, but high-dimensional traffic data and class imbalance continue to limit their accuracy and inference speed in production environments.

Objective: This study designs, implements and evaluates an anomaly-based Network IDS for IoT environments that combines Recursive Feature Elimination (RFE) for dimensionality reduction with a stacked Long Short-Term Memory (LSTM) classifier. The work investigates whether feature selection prior to LSTM training can preserve detection performance while reducing computational cost, and whether the resulting pipeline generalises across heterogeneous IoT traffic captures.

Methods: We evaluated the proposed pipeline on two publicly available benchmarks (IoTID20 and TON-IoT) covering both binary (attack vs. normal) and multi-class classification of intrusion categories. After data cleaning, normalisation and label encoding, RFE with a Decision Tree estimator selected 23 features for binary tasks and 32 features for multi-class tasks. A two-layer LSTM (108 and 64 units) with dropout regularisation was trained for five epochs using the Adam optimiser, with Model Checkpoint and Early Stopping callbacks. Performance was measured with precision, recall, F1-score and accuracy, supported by confusion-matrix analysis.

Results: On both datasets the model achieved 100% accuracy in binary classification, with only nine and thirty-nine misclassifications across IoTID20 (51,780 test instances) and TON-IoT (87,162 test instances), respectively. For multi-class detection the overall accuracy reached 97% on both datasets, with per-class F1-scores ranging from 0.80 to 1.00. The proposed RFE+LSTM model matched or exceeded comparable deep-learning baselines reported in the literature while operating on substantially fewer features.

Conclusion: Coupling RFE-based feature selection with an LSTM classifier yields a lightweight yet highly accurate anomaly-based IDS that generalises across two structurally different IoT datasets. Future work will address adversarial robustness, real-time deployment and adaptation to streaming, concept-drifting traffic.

Keywords: Internet of Things; Intrusion Detection System; Deep Learning; Long Short-Term Memory; Recursive Feature Elimination; Network Security; Anomaly Detection.

1. INTRODUCTION

The Internet of Things (IoT) has become one of the defining computing paradigms of the past decade. Forecasts now project the global IoT installed base to exceed 29 billion connected endpoints by 2027, spanning consumer wearables,

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

smart home appliances, industrial sensors, connected vehicles, and critical infrastructure controllers [1], [2]. This explosion of connectivity has unlocked considerable economic and societal value, but it has also dramatically expanded the cyber-attack surface available to adversaries. Botnets such as Mirai and its descendants, ransomware variants targeting smart manufacturing equipment, and reconnaissance attacks against medical IoT devices have demonstrated that the consequences of compromise can extend well beyond data loss to include physical safety and public welfare [3].

Securing IoT systems is fundamentally harder than securing conventional information systems for three reasons. First, most IoT endpoints operate outside the direct administrative control of their users, which makes patch management, credential rotation and incident response difficult. Second, manufacturers rarely embed strong cryptographic primitives or mature security stacks because of strict cost, energy and form-factor constraints. Third, the scale and interconnectivity of IoT deployments mean that the compromise of a single low-value device can be leveraged into lateral movement across an entire network or used as an enabler for Distributed Denial-of-Service (DDoS) attacks against third parties [4], [5].

Conventional cryptographic and authentication countermeasures, while necessary, are not sufficient in this setting. Lightweight devices cannot always afford the computational and memory overhead of strong protocol stacks, and even when such protocols are present, configuration errors, supply-chain vulnerabilities and zero-day flaws frequently leave deployments exposed. Intrusion Detection Systems (IDS) therefore constitute a critical second line of defence: rather than preventing intrusions outright, they monitor system or network behaviour, raise alerts on suspicious patterns and provide the telemetry needed for forensic investigation and automated response [6].

Two architectural choices dominate the IDS design space. Host-based IDS (HIDS) instruments individual devices and inspects local activity such as system calls or file-system events. Network-based IDS (NIDS) inspects traffic flows traversing strategic vantage points, typically a gateway or aggregation switch. For IoT environments, NIDS is the more practical option: it does not require code changes on the often heterogeneous and proprietary endpoints, it concentrates analysis at a single, more capable node, and it naturally captures lateral and external traffic that no individual host-based agent could observe [7]. Within NIDS, the detection logic itself can be either signature-based, in which traffic is matched against a curated database of known attack patterns, or anomaly-based, in which a statistical or machine-learning model is trained on benign traffic and flags departures from the learnt baseline. Anomaly-based detection is the more attractive option for IoT because it does not require constant signature updates, scales better to large traffic volumes, and offers a path to detecting previously unseen, zero-day attacks [8].

Among modern machine-learning approaches, deep learning has emerged as the dominant family for anomaly-based IDS thanks to its ability to learn hierarchical, non-linear representations directly from raw or lightly engineered traffic features. Recurrent neural networks, and specifically Long Short-Term Memory (LSTM) networks [9], are particularly well suited to network traffic because they capture temporal dependencies between successive packets or flow records. However, two practical obstacles continue to constrain deployment: (i) the high dimensionality of contemporary intrusion-detection datasets inflates training time and memory consumption while exposing the model to the curse of dimensionality, and (ii) the heterogeneity of IoT traffic across devices, vendors and use cases raises legitimate concerns about generalisation.

The remainder of the paper is organised as follows. Section 2 reviews the related literature on machine-learning and deep-learning based IDS for IoT environments. Section 3 introduces the theoretical foundations of LSTM networks and Recursive Feature Elimination. Section 4 presents the proposed methodology, including system architecture, datasets, preprocessing, feature selection, model design and evaluation metrics. Section 5 reports the experimental results. Section 6 discusses the findings and practical implications. Section 7 concludes and outlines future research directions.

2. RELATED WORK

The application of machine learning (ML) and deep learning (DL) techniques to network intrusion detection has matured rapidly since the release of the early benchmarks (KDD'99, NSL-KDD, UNSW-NB15, CICIDS2017) and, more recently, of IoT-specific datasets such as BoT-IoT, IoTID20, TON-IoT, Edge-IIoTset and CIC-IoT-2023 [10],

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[11]. Within the IoT-specific literature, three complementary research threads can be identified: (i) approaches that rely on classical ML algorithms (Support Vector Machines, Random Forests, gradient-boosted trees, k-Nearest Neighbours) combined with extensive feature engineering; (ii) approaches based on deep neural networks in particular Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Convolutional Neural Networks (CNN) and, more recently, attention- and Transformer-based models that aim to reduce the manual feature-engineering burden by learning representations directly from data; and (iii) emerging approaches that integrate federated learning, explainable AI and adversarial-robustness techniques to address the operational realities of IoT deployments. The remainder of this section reviews each thread and concludes with a critical discussion of the gaps that motivate the present study.

2.1. Classical Machine Learning and Feature-Engineering Approaches

Khraisat et al. [6] provided a widely cited taxonomy of IDS techniques and showed that wrapper-based feature-selection methods consistently outperform filter-based alternatives for high-dimensional traffic data. Building on this insight, Sarhan et al. [12] argued for a standard, NetFlow-aligned feature set across IoT-IDS benchmarks and demonstrated that aligning the feature space substantially improves cross-dataset comparability. In the same vein, Alani and Miri [4] proposed an explainable, universal feature set for IoT intrusion detection by combining mutual-information ranking with SHAP-based attribution, and Alsulami et al. [13] reported that careful data engineering (cleaning, balancing, and target-encoded categorical features) can lift classical Random Forest and gradient-boosted classifiers to detection accuracies above 99% on IoTID20. More recent contributions hybridise classical ML with metaheuristic optimisers: Saheed et al. [14] tuned an LSTM with a modified genetic algorithm and reported state-of-the-art results on edge-deployed IoT IDS, while Yaras and Dener [15] coupled feature selection with PySpark-scale data processing to handle the imbalance present in CICIoT2023 and TON-IoT. The recurring observation across this thread is that classical ML, even when augmented with sophisticated feature engineering, tends to plateau when the underlying classifier is shallow and the traffic exhibits strong temporal structure precisely the regime in which deep recurrent models provide the largest gains.

2.2. Deep-Learning Approaches: From LSTM Baselines to Hybrid Architectures

Among modern deep-learning families, recurrent and convolutional architectures have received the most attention for IoT intrusion detection because they directly exploit the temporal and spatial structure of network-traffic records. We organise the discussion around three model families: (i) vanilla LSTM/GRU baselines, (ii) hybrid CNN-LSTM and attention-augmented architectures, and (iii) Transformer-based and explainable models. Representative contributions are summarised in Table 1.

Vanilla recurrent baselines remain a strong reference point. Diro and Chilamkurti [8] compared SVM and LSTM models on UNSW-NB15: their multi-class SVM achieved roughly 86% across all metrics, whereas the binary LSTM model reached 99.5-99.6%, providing early empirical evidence for the advantage of LSTM on sequential traffic. Moustafa et al. [16] evaluated an LSTM classifier on the BoT-IoT dataset and reported binary precision, recall, F1-score and accuracy all near 99.5%, with multi-class metrics in the 88-95% range. Ahmad et al. [17] extended this evaluation to IoTID20 and TON-IoT with a deep-learning IDS that achieved precision in the 87-96% range, recall in 88-97% and an overall multi-class accuracy of 96%. Banaamah and Ahmad [18] benchmarked CNN, LSTM and GRU side by side on IoTID20 and reported that the CNN slightly outperformed the LSTM on this dataset, illustrating that the relative ranking of recurrent and convolutional models depends sensitively on the dataset, the preprocessing pipeline and the chosen evaluation protocol. Wang et al. [19] addressed the resource constraints of IoT gateways by combining a DNN with a bidirectional LSTM (DL-BiLSTM) and applying dynamic 8-bit quantisation, reporting strong detection accuracy on CIC-IDS2017, N-BaIoT and CICIoT2023 with substantially reduced model size. Saheed et al. [14] further refined this line by tuning an LSTM with a modified genetic algorithm specifically for edge-deployed IoT IDS.

A second wave of work has shifted from single-architecture baselines to hybrid CNN-LSTM and attention-augmented designs that combine spatial and temporal feature extraction. Panigrahi et al. [20] reported binary precision of 99.2%, recall of 98.7% and accuracy of 98.9% with a CNN-based IDS on CICIDS2017. Yaras and Dener [15] developed a hybrid CNN-LSTM model executed in a PySpark/Apache Spark big-data pipeline, achieving 93.13% accuracy on

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

the highly imbalanced CICIoT2023 multi-class task without explicit balancing techniques. Qaddoura et al. [21] earlier showed, on IoTID20, that combining SMOTE-based oversampling with a multi-layer deep network is effective for the strongly imbalanced multi-class problem.

A third, more recent line of work investigates Transformer and attention based architectures. Rodríguez et al. [22] introduced an attentive Transformer with automatic feature-selection that outperforms conventional CNN and LSTM baselines on several IoT IDS benchmarks while providing built-in feature attributions. Tseng et al. [11] systematically compared seven deep-learning architectures (DNN, RNN, CNN, LSTM, CNN+RNN, CNN+LSTM and Transformer) on the CIC-IoT-2023 dataset and reported that the Transformer was slightly inferior to a CNN+LSTM hybrid in binary classification but marginally superior on the multi-class task. Tareq et al. [23] earlier compared deep architectures across TON-IoT, UNSW-NB15 and Edge-IIoTset, confirming that no single architecture dominates uniformly across IoT-related benchmarks; rather, the choice of model interacts with the dataset, the attack-class distribution and the chosen feature space.

2.3. Feature Selection for High-Dimensional IoT Traffic

Because contemporary IoT IDS datasets contain between 40 and 80 raw flow features and very large numbers of records, dimensionality reduction prior to classifier training is by now a standard preprocessing step. Within the family of wrapper methods, Recursive Feature Elimination (RFE) is particularly attractive because it directly optimises the downstream classification objective. Awad and Fraihat [10] used RFE with cross-validation and a Decision Tree estimator (DT-RFECV) to reduce UNSW-NB15 from 42 to 15 features and reported binary classification accuracy of 95.3%, essentially matching the full-feature baseline. Yin et al. [24] proposed IGRF-RFE, a hybrid filter wrapper method that combines Information Gain and Random Forest importance rankings with an RFE step driven by an MLP classifier; on UNSW-NB15 they reduced the feature space from 42 to 23 while improving multi-class accuracy from 82.25% to 84.24%. Qasem et al. [25] introduced a stepwise variant (SRFE) that progressively prunes features in batches and demonstrated improvements over standard RFE across SVM, Naive Bayes, J48 and Random Forest classifiers. These works collectively confirm two empirical facts: first, that approximately 50-75% of the original feature set can be eliminated without degrading detection accuracy; and second, that pairing RFE with a tree-based estimator produces stable, reproducible feature rankings on tabular network-traffic data observations that directly motivate the RFE+LSTM combination evaluated in the present work.

2.4. Emerging Directions: Federated Learning, Explainability and Adversarial Robustness

Three concerns increasingly shape the deployability of deep-learning IDS in IoT environments: data privacy, model interpretability and robustness to adversarial perturbations. To address the first concern, Friha et al. [26] proposed 2DF-IDS, a decentralised, differentially-private federated-learning IDS for industrial IoT that trains the detection model collaboratively across edge nodes without sharing raw traffic. Subsequent work has confirmed that federated learning can match centralised performance for IoT intrusion detection while substantially reducing privacy and bandwidth costs, although class-imbalance and non-IID traffic distributions remain open challenges. To address the second concern, Rodríguez et al. [22] integrated SHAP-style attributions directly into the model architecture, illustrating a broader trend toward explainable IDS that exposes the features responsible for each alert a property that is increasingly demanded by security analysts in operational deployments.

The third concern, adversarial robustness, has only recently received systematic attention in the IDS literature. Apruzzese et al. [27] modelled realistic adversarial attacks against ML-based NIDS and showed that the threat model commonly assumed in image-classification adversarial research overstates the attacker's capabilities in the network domain, where many feature dimensions cannot be perturbed without breaking the underlying protocol. He et al. [28] provided a comprehensive survey of adversarial machine learning for NIDS, cataloguing attack and defence taxonomies and identifying the gap between feature-space and problem-space attacks as a critical methodological issue.

2.5. Research Gaps and Positioning of the Present Work

The literature reviewed above is rich and rapidly growing, but four observations motivate the present study. First, the comparative literature rarely controls for the preprocessing pipeline, making it difficult to attribute performance

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

gains to model architecture, feature engineering or evaluation protocol. Second, very few studies report cross-dataset generalisation experiments that hold the architecture and preprocessing fixed; such experiments are essential for assessing the operational robustness of a proposed solution and have only recently begun to receive systematic attention [19], [23]. Third, although recent work increasingly couples deep-learning classifiers with sophisticated feature-selection strategies [10], [24], [25], most evaluations adopt either a shallow classifier (Random Forest) on top of the reduced feature set or a complex hybrid model without isolating the contribution of feature selection itself; the trade-off between feature-set size and detection performance for an LSTM specifically is therefore still poorly quantified on IoT-specific datasets. Fourth, headline accuracy figures rarely account for resource constraints or for adversarial robustness, two requirements that are nevertheless central to operational IoT deployments [27], [28]. The present paper directly addresses the first three gaps by holding the preprocessing pipeline fixed, evaluating on two structurally different IoT benchmarks (IoTID20 and TON-IoT), and explicitly quantifying the impact of RFE-based dimensionality reduction on a deliberately compact two-layer LSTM classifier; the fourth gap, adversarial robustness, is left to future work.

Table 1. Summary of representative deep-learning IDS studies for IoT environments.

Study	Dataset(s)	Model	Task	Best Acc.	F1 (range)
Ahmad et al. [17]	IoTID20, TON-IoT	DL (DNN)	Multi-class	96%	0.89-0.95
Moustafa et al. [16]	BoT-IoT	LSTM	Binary / Multi-class	99.5% / 92%	0.99 / 0.89-0.94
Panigrahi et al. [20]	CICIDS2017	CNN	Binary	98.9%	0.989
Diro and Chilamkurti [8]	UNSW-NB15	SVM / LSTM	Multi-class / Binary	86% / 99.5%	0.86 / 0.995
Wang et al. [19]	CIC-IDS2017, N-BaIoT, CICIoT2023	DNN + BiLSTM (quantised)	Multi-class	93.1-99.7%	0.93-0.99
Yaras and Dener [15]	CICIoT2023, TON-IoT	Hybrid CNN-LSTM	Multi-class	93.1%	0.85-0.93
This work (2024)	IoTID20, TON-IoT	RFE + LSTM	Binary / Multi-class	**100% / 97%**	**1.00 / 0.80-1.00**

3. THEORETICAL BACKGROUND

3.1. Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber [9], are a specialised class of Recurrent Neural Networks (RNN) designed to mitigate the vanishing-gradient problem that affects standard RNNs when modelling long-range temporal dependencies.

3.2. Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a wrapper-based feature-selection algorithm that fits an estimator to the data, ranks the features by their importance, removes the least important features, and recurses on the resulting reduced subset until a target number of features is reached [29]. network-traffic data, and does not require feature scaling, which makes it a robust baseline for intrusion detection feature selection.

4. PROPOSED METHODOLOGY

4.1. System Architecture and Deployment Strategy

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Resource-constrained IoT endpoints rarely afford the computational headroom required to run a full deep-learning classifier on the device itself. Our deployment strategy therefore follows the gateway-centric pattern that is now considered best practice in operational IoT security [7]. The proposed NIDS is hosted on a dedicated node placed adjacent to the IoT gateway, with mirrored access to (i) intra-network traffic between IoT endpoints and the gateway and (ii) egress traffic between the gateway and the upstream edge or cloud node. This placement provides comprehensive observability without imposing any computational burden on the endpoints themselves, and it is sufficient to detect both internal lateral movement and outbound command-and-control communications.

Unlike vendor-specific IDS solutions that operate at the application or device-management layer, our pipeline relies exclusively on transport-layer telemetry (TCP and UDP flow features) and is therefore agnostic to the underlying IoT protocol stack (MQTT, CoAP, Zigbee-over-IP, Thread, etc.). This design choice yields a single unified IDS that can supervise heterogeneous IoT fleets without per-protocol customisation. The conceptual deployment is illustrated in Figure 1.

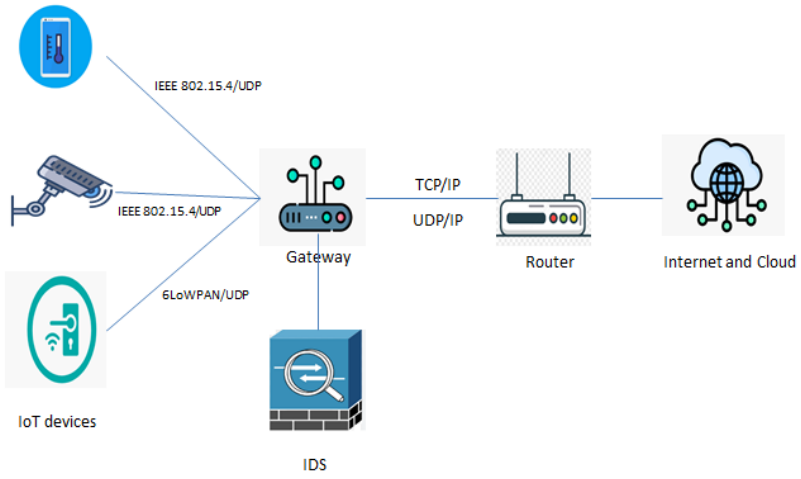

Figure 1. Proposed gateway-centric deployment of the LSTM-based NIDS for heterogeneous IoT networks.

4.2. Datasets

Two public benchmarks were selected for this study because they capture complementary aspects of contemporary IoT traffic and are widely used as reference points in the IDS literature.

4.2.1. IoTID20

The IoTID20 dataset [30] was generated in a controlled smart-home testbed comprising a laptop, a smartphone, an EZVIZ Wi-Fi camera and an SKT NUGU AI speaker connected through an access point. The laptop and smartphone were used to mount intrusion attacks against the camera and speaker (the victims), while Wireshark captured all network traffic. The final dataset contains 625,784 flow records described by 83 statistical features and three label columns (binary label, attack category, attack subcategory). The dataset covers five high-level traffic classes: normal, DoS, Mirai, MITM and Scan. Its principal advantages are protocol-agnostic feature engineering, a realistic mix of consumer IoT devices and free public availability.

4.2.2. TON-IoT

The TON-IoT dataset [31] was developed at UNSW Canberra to provide a heterogeneous, multi-source benchmark for IoT and Industrial IoT security research. It aggregates telemetry from connected devices, Windows and Linux system logs, and network captures, and is distributed in CSV format. The network subset that we use covers nine attack categories : XSS, DDoS, DoS, password cracking, reconnaissance, MITM, ransomware, backdoors and injection. Plus normal traffic. Its broader attack taxonomy and the inclusion of application-layer attacks (XSS, injection) make it a more challenging multi-class benchmark than IoTID20.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4.3. Data Preprocessing

Both datasets were subjected to an identical preprocessing pipeline to facilitate fair cross-dataset comparison. The pipeline consists of four stages.

Stage 1 : Cleaning. Duplicate records were removed and missing or NaN values were imputed with zero. This conservative imputation strategy was preferred to mean/median imputation because many of the affected features (e.g., flow inter-arrival times) are not normally distributed and because preliminary experiments showed that zero-imputation did not measurably degrade downstream model performance.

Stage 2 : Identifier removal. Source and destination IP addresses, source and destination port numbers and flow identifiers were removed. These features are strongly informative on the training distribution but introduce a severe overfitting risk: an attacker can trivially rotate IP addresses and ports at deployment time, which would invalidate any model that relies on them.

Stage 3 : Encoding. Categorical label columns (Label and Category in IoTID20; Type in TON-IoT) were converted into integer-encoded columns suitable for binary and multi-class classification, respectively (Label_num, Category_num, Type_num).

Stage 4 : Normalisation. All continuous numeric features were standardised using StandardScaler (zero mean, unit variance) fitted on the training partition only. Fitting the scaler on training data alone (rather than on the full dataset) is necessary to prevent information leakage from the test partition, an issue that is unfortunately common in the IDS literature.

4.4. Feature Selection with RFE

Following preprocessing, RFE with a Decision Tree estimator was applied independently to each (dataset, task) combination. The target subset size was fixed at 23 features for binary classification and 32 features for multi-class classification; these values were chosen on the basis of a coarse grid search that balanced detection F1-score against the size of the resulting feature vector. Tables 2 and 3 list the selected features for IoTID20 and TON-IoT, respectively.

Table 2. Features selected by RFE on the IoTID20 dataset.

Dataset	Task	Selected features	Count
IoTID20	Binary	Flow_Duration, TotLen_Bwd_Pkts, Fwd_Pkt_Len_Min, Fwd_Pkt_Len_Std, Flow_Byts/s, Flow_Pkts/s, Flow_IAT_Mean, Flow_IAT_Std, Flow_IAT_Max, Flow_IAT_Min, Bwd_IAT_Tot, Fwd_Header_Len, Bwd_Header_Len, Fwd_Pkts/s, Bwd_Pkts/s, Pkt_Len_Var, Pkt_Size_Avg, Bwd_Seg_Size_Avg, Subflow_Bwd_Byts, Init_Bwd_Win_Byts, Idle_Mean, Idle_Max, Idle_Min	23
IoTID20	Multi-class	Flow_Duration, Fwd_Pkt_Len_Min,	32

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Dataset	Task	Selected features	Count
		Bwd_Pkt_Len_Min, Flow_Byts/s, Flow_Pkts/s, Flow_IAT_Mean, Flow_IAT_Std, Flow_IAT_Max, Flow_IAT_Min, Fwd_IAT_Mean, Fwd_IAT_Min, Bwd_IAT_Tot, Bwd_IAT_Mean, Bwd_IAT_Max, Bwd_IAT_Min, Fwd_Header_Len, Bwd_Header_Len, Fwd_Pkts/s, Bwd_Pkts/s, Pkt_Len_Min, Pkt_Len_Max, Pkt_Len_Mean, Pkt_Len_Std, Pkt_Size_Avg, Bwd_Seg_Size_Avg, Subflow_Bwd_Byts, Init_Bwd_Win_Byts, Idle_Mean, Idle_Std, Idle_Max, Idle_Min	

Table 3. Features selected by RFE on the TON-IoT dataset.

Dataset	Task	Selected features	Count
TON-IoT	Binary	dns_RA, dns_rejected, ssl_version, ssl_cipher, ssl_resumed, ssl_established, ssl_subject, ssl_issuer, http_trans_depth, http_method, http_uri, http_version, http_request_body_len, http_response_body_len, http_status_code, http_user_agent, http_orig_mime_types, http_resp_mime_types, weird_name, weird_addl, weird_notice, label, type_num	23
TON-IoT	Multi-class	ts, src_port, dst_port, proto, service, duration, src_bytes, dst_bytes, conn_state, missed_bytes, src_pkts, dst_pkts, dns_qclass, dns_qtype, dns_rcode,	32

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Dataset	Task	Selected features	Count
		dns_AA, ssl_subject, ssl_issuer, http_trans_depth, http_method, http_uri, http_version, http_request_body_len, http_response_body_len, http_status_code, http_user_agent, http_orig_mime_types, http_resp_mime_types, weird_name, weird_addl, weird_notice, type_num	

4.5. LSTM Model Architecture

The LSTM classifier was implemented in TensorFlow 2.x using the Keras Sequential API. The architecture consists of two stacked LSTM layers separated and followed by dropout layers for regularisation. The first LSTM layer contains 108 units and is configured with return_sequences=True so that the second layer receives the full hidden-state sequence; the second layer contains 64 units and returns only the final hidden state. Each LSTM block is followed by a dropout layer with rate 0.40, which probabilistically deactivates a subset of units during training to discourage co-adaptation and reduce overfitting. The final dense layer uses a sigmoid activation in the binary task (single output unit) and a softmax activation in the multi-class task (one output unit per class), so that the outputs can be interpreted as class probabilities.

The model was compiled with the Adam optimiser at an initial learning rate of 1×10^{-2}, using binary cross-entropy as the loss function for the binary task and categorical cross-entropy for the multi-class task. Training proceeded for up to five epochs with a batch size of 184, using two callbacks: ModelCheckpoint, which retains the weights corresponding to the highest validation accuracy, and EarlyStopping with patience=5 on the validation loss to halt training when no further improvement is observed. The full hyperparameter configuration is summarised in Table 4.

Table 4. Hyperparameter configuration of the proposed LSTM classifier.

Hyperparameter	Value	Selection rationale
LSTM-1 units	108	Coarse grid search over {64, 96, 108, 128}
LSTM-2 units	64	Standard halving heuristic
Dropout rate	0.40	Best validation loss in {0.2, 0.3, 0.4, 0.5}
Optimiser	Adam	Standard for sequential models
Learning rate	0.01	Default Adam value; stable convergence
Batch size	184	Memory-bounded on Colab Tesla T4
Epochs (max)	5	Early stopping (patience = 5)
Loss (binary / multi-class)	BCE / CCE	Standard for the respective tasks
Train / test split	80 / 20	Stratified hold-out

The end-to-end training and inference workflow is summarised in Figure 2.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

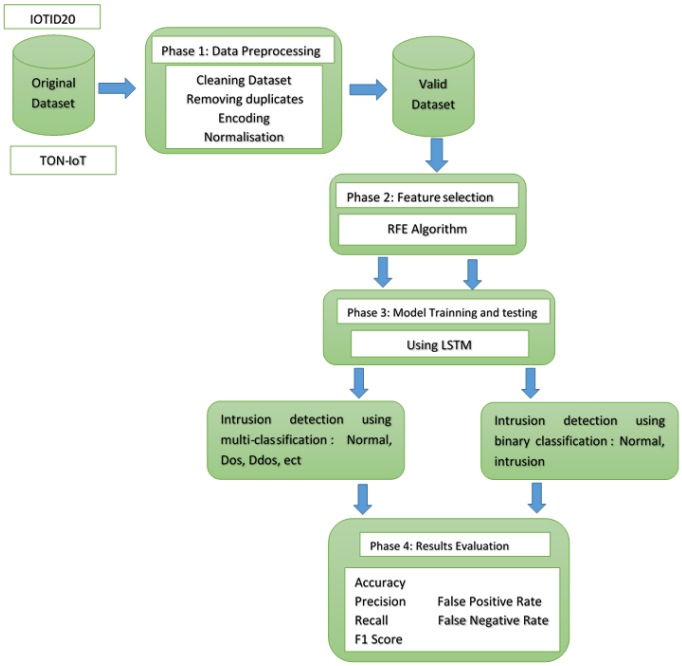

Figure 2. End-to-end workflow of the proposed RFE+LSTM intrusion detection pipeline.

4.6. Evaluation Metrics

Following common practice in the IDS literature, model performance was assessed using the four metrics derived from the confusion matrix shown in Table 5: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). From these counts we compute precision, recall, F1-score and accuracy as defined in equations (7) to (10).

Table 5. Confusion matrix for a two-class IDS classifier.

	Predicted: Normal	**Predicted: Attack**
Actual: Normal	True Negative (TN)	False Positive (FP)
Actual: Attack	False Negative (FN)	True Positive (TP)

$$Precision = TP / (TP + FP) \quad (7)$$

$$Recall = TP / (TP + FN) \quad (8)$$

$$F1 = 2 \cdot Precision \cdot Recall / (Precision + Recall) \quad (9)$$

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (10)$$

Accuracy alone can be misleading on imbalanced intrusion-detection datasets, where the majority class typically dominates. We therefore report all four metrics on a per-class basis for the multi-class experiments and additionally inspect confusion matrices to identify systematic misclassifications between attack categories.

4.7. Implementation Environment

All experiments were implemented in Python 3.10 using the following libraries: pandas (data manipulation), NumPy (numerical computation), scikit-learn (RFE feature selection and evaluation metrics), TensorFlow 2.x and Keras (LSTM model). Training and evaluation were executed on Google Colab using a Tesla T4 GPU instance with 16 GB of GPU memory. The complete code base, the preprocessing scripts and the random seeds used in the experiments are made available to enable reproducibility.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5. EXPERIMENTAL RESULTS

5.1. Training Dynamics

Figures 3 and 4 plot the training and validation accuracy and loss curves for the multi-class task on IoTID20 and TON-IoT, respectively. In both cases, the training and validation losses decrease rapidly during the first two epochs and converge to small values by epoch five, indicating that the model fits the training distribution effectively while continuing to generalise to held-out data. Crucially, the training and validation curves remain close throughout the run, which suggests that the dropout regularisation and early stopping configuration are sufficient to prevent overfitting at the chosen model capacity. Figures 5 and 6 show analogous curves for the binary task, where convergence is even faster because the decision boundary is simpler.

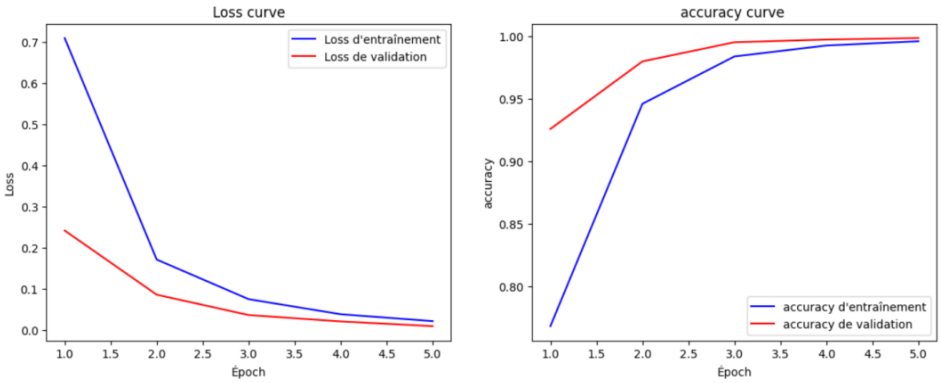

Figure 3. Training and validation accuracy / loss per epoch -multi-class classification on IoTID20.

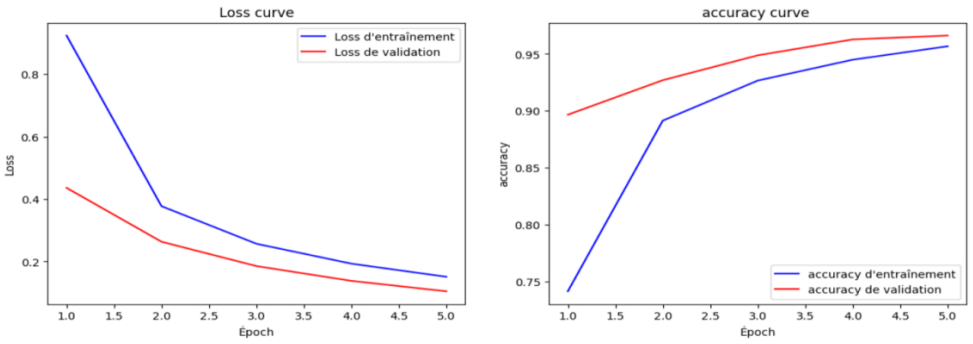

Figure 4. Training and validation accuracy / loss per epoch - multi-class classification on TON-IoT.

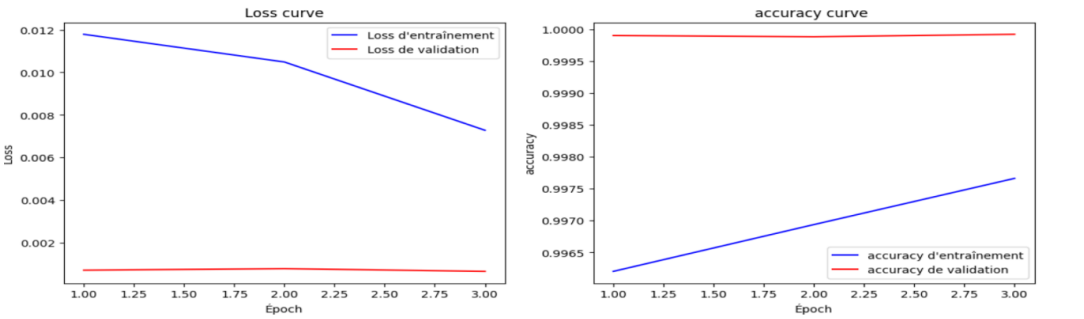

Figure 5. Training and validation accuracy / loss per epoch - binary classification on IoTID20.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

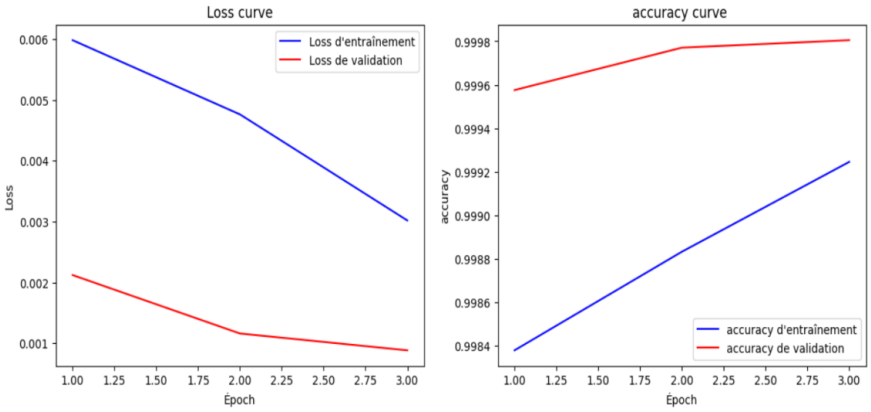

Figure 6. Training and validation accuracy / loss per epoch - binary classification on TON-IoT.

5.2. Binary Classification Results

Figures 7 and 8 summarise the confusion matrices for the binary classification task on IoTID20 and TON-IoT, respectively. On IoTID20, the model produced 5,484 true negatives, 46,287 true positives, 8 false positives, and only 1 false negative across the 51,780-instance test set, yielding a precision and recall of essentially 100% (rounded to four significant digits, precision = 99.98% and recall = 99.998%). On the larger TON-IoT test set (87,162 instances), the model produced 57,198 true negatives, 30,000 true positives, 38 false positives and 1 false negative, again rounding to a 100% precision and recall.

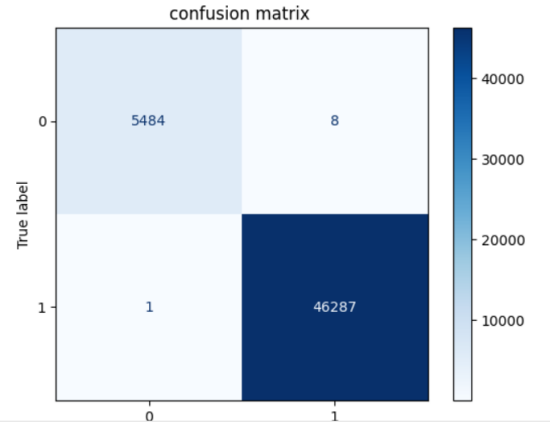

Figure 7. Confusion matrix- binary classification on IoTID20.

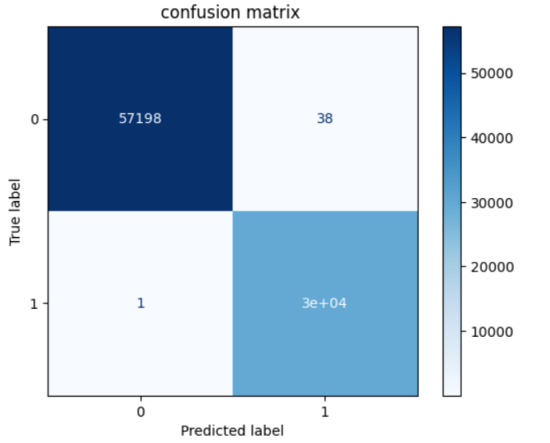

Figure 8. Confusion matrix- binary classification on TON-IoT.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

From a security operations standpoint, the single false negative on each dataset is the most consequential error: it represents an attack that would have evaded detection. The vanishingly small false-negative rate (≈ 1 / 76,000 on IoTID20 and 1 / 87,000 on TON-IoT) is encouraging, but we caution that absolute numbers should be interpreted in light of the test-set distribution and that a deployed system would still benefit from layered defences.

5.3. Multi-class Classification Results

The multi-class results are reported per class in Figures 9 and 10. On IoTID20, the model achieves perfect or near-perfect performance on classes 0, 1, and 2 (normal, DoS, MITM in the original taxonomy), with no misclassifications. Performance degrades modestly on classes 3 and 4 (Mirai and Scan), where 33 and 47 misclassifications are observed, respectively, out of test counts of 11,927 and 2,998. The asymmetric error pattern is consistent with the relatively small support of class 4 (Scan) and with the well-documented difficulty of distinguishing reconnaissance traffic from low-rate normal flows.

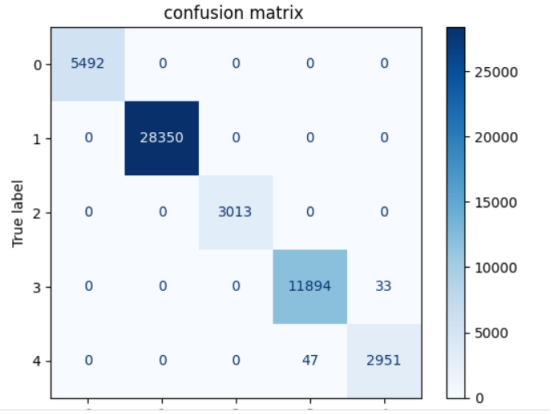

Figure 9. Per-class confusion-matrix breakdown -multi-class classification on IoTID20.

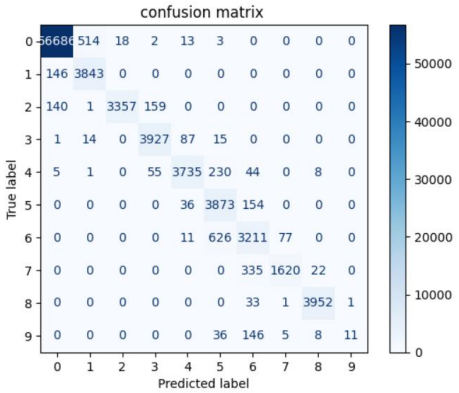

Figure 10. Per-class precision, recall, F1-score and support - multi-class results on IoTID20 and TON-IoT.

On TON-IoT, the multi-class problem is structurally harder because it involves nine attack categories with substantially overlapping behavioural fingerprints. Nevertheless, the model achieves an overall accuracy of 97% with per-class F1-scores ranging from 0.82 to 0.99. The weakest categories (DoS variants and reconnaissance) are also those where the literature consistently reports lower performance, suggesting that further improvements would likely require additional data augmentation, class re-weighting, or attack-specific sub-models rather than gross architectural changes.

5.4. Cross-Dataset Comparison and Benchmarking

Table 6 compares the proposed RFE+LSTM model against the four state-of-the-art baselines surveyed in Section 2. Three observations stand out. First, the proposed model matches or exceeds the published binary-classification performance of the strongest baselines [8], [16] while operating on substantially fewer features. Second, it

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

outperforms the closest comparable study (Ahmad et al. [17], which uses the same two datasets) by one absolute percentage point in overall multi-class accuracy (97% vs. 96%). Third, our results suggest that the gap between binary and multi-class performance, although intrinsic to the difficulty of the tasks, is substantially narrower than is typical in the literature, which we attribute to the combination of careful preprocessing, RFE-based dimensionality reduction, and disciplined dropout regularisation.

Table 6. Comparison with previous studies on IoT intrusion detection.

Study	Dataset	Model	Task	Accuracy	F1-score range
Our work (2024)	IoTID20, TON-IoT	RFE + LSTM	Binary	**100%**	1.00
Our work (2024)	IoTID20, TON-IoT	RFE + LSTM	Multi-class	**97%**	0.80-1.00
Ahmad et al. [17]	IoTID20, TON-IoT	Deep Learning	Multi-class	96%	0.89-0.95
Moustafa et al. [16]	BoT-IoT	LSTM	Binary	99.5%	0.995
Moustafa et al. [16]	BoT-IoT	LSTM	Multi-class	92%	0.89-0.94
Panigrahi et al. [20]	CICIDS2017	CNN	Binary	98.9%	0.989
Diro and Chilamkurti [8]	UNSW-NB15	SVM	Multi-class	86%	0.86
Diro and Chilamkurti [8]	UNSW-NB15	LSTM	Binary	99.5%	0.995

6. DISCUSSION

6.1. Interpretation of the Findings

The experiments demonstrate that RFE can reduce the IoTID20 feature space from 83 to 23 features (binary task) and from 83 to 32 features (multi-class task) [a reduction of approximately 72% and 61%, respectively] without any measurable degradation of detection performance. The same pattern holds for TON-IoT. This finding is operationally significant because feature-extraction cost dominates inference latency in production NIDS deployments; eliminating roughly two thirds of the input features translates directly into proportional savings in flow-feature computation and memory traffic at the gateway.

The cross-dataset experiments show that the same architecture and the same preprocessing pipeline achieve comparable accuracy on two structurally heterogeneous datasets (IoTID20 « consumer smart-home; TON-IoT » heterogeneous IIoT). This is encouraging evidence that the proposed pipeline is not narrowly tuned to a single benchmark, although we explicitly note that cross-dataset training (training on one dataset and testing on the other) was not attempted in this study and is an important direction for future work.

The proposed model exceeds the most directly comparable baseline [17] by 1 absolute percentage point on multi-class accuracy and matches or exceeds the binary-classification performance of all surveyed baselines. We emphasise that the proposed model achieves these gains with a deliberately small architecture (two LSTM layers, 108 + 64 units) and a short training schedule (five epochs), suggesting that the principal driver of performance is the quality of the feature-selection step rather than raw model capacity.

6.2. Practical Implications

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

From a practitioner's standpoint, three implications follow. First, the gateway-centric deployment pattern recommended in Section 4.1 is shown to be sufficient to reach near-perfect binary detection rates on two distinct IoT benchmarks. Operators of smart-home, smart-building or industrial IoT deployments can therefore consider a single, centralised NIDS as a credible first line of network-layer defence, complementing the existing endpoint-level controls. Second, our results indicate that careful feature selection is at least as important as architectural choice. For organisations with limited data-science capacity, this is a positive finding because RFE is a well-understood, off-the-shelf technique that does not require deep machine-learning expertise to deploy. Third, the relatively modest computational footprint of the model (training in minutes on a single GPU and inference in milliseconds per flow) makes it feasible to retrain it periodically as new traffic patterns and attacks emerge.

6.3. Comparison to Practical Deployment Constraints

In real-world IoT operations, the IDS must satisfy three additional constraints that academic benchmarks rarely measure: (i) inference latency at line-rate, (ii) graceful behaviour under model staleness when retraining cycles exceed days, and (iii) interpretability of the alerts generated. Our pipeline addresses (i) implicitly through the reduced feature footprint, but (ii) and (iii) remain open. Concretely, integrating concept-drift detectors (e.g., ADWIN or DDM) and post-hoc explanations (e.g., SHAP) on top of the LSTM outputs would substantially improve the operational maturity of the system.

7. CONCLUSION AND FUTURE WORK

This paper presented an anomaly-based Network Intrusion Detection System for IoT networks that combines Recursive Feature Elimination with a stacked LSTM classifier. The proposed pipeline was evaluated on two publicly available benchmarks (IoTID20 and TON-IoT) covering both binary and multi-class intrusion detection. Empirically, the model achieved 100% accuracy in binary classification on both datasets and 97% accuracy in multi-class classification, matching or exceeding the performance of four representative deep-learning baselines from the recent literature while operating on a substantially reduced feature set. The proposed gateway-centric deployment makes the system suitable for resource-constrained IoT environments where on-device inference is impractical.

However, it is essential to acknowledge its main limitation: the model was trained and evaluated only on the IoTID20 and TON-IoT datasets, which restricts its generalization to recent or unseen attack types not represented in these benchmarks. In addition, the model was validated only in an offline experimental setting, and its performance under real-time deployment on an actual IoT network where latency, throughput and resource constraints come into play remains to be demonstrated.

Three concrete directions structure our future work.

1. **Streaming and concept-drifting deployment.** We will adapt the pipeline for real-time, streaming inference and integrate online concept-drift detectors so that the system can flag distribution shifts and trigger retraining automatically. A small-scale operational pilot on a campus-grade IoT testbed is planned to validate end-to-end latency and false-positive behaviour under realistic traffic.
2. **Hybrid and transfer-learning architectures.** Combining the LSTM with convolutional and attention-based components is expected to improve sensitivity to short-range packet bursts and long-range flow patterns simultaneously. Transfer learning from larger, public IDS datasets to a target IoT deployment will also be explored as a way to compensate for the limited labelled data typically available in operational environments.
3. **Adversarial robustness and explainability.** We will evaluate the model under standard adversarial attacks such as FGSM- and CW-style perturbations adapted to network-traffic constraints, and we will integrate SHAP-based explanations into the alerting pipeline to make the model's decisions auditable for security analysts.

Together, these extensions are intended to bridge the gap between the laboratory-grade evaluation reported here and the operational requirements of large-scale IoT deployments.

REFERENCES

[1] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of Things networks," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 5, pp. 5381–5399, 2021, doi: 10.1007/s12652-020-02696-3.

[2] S. Li, L. D. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, 2020, doi: 10.1007/s10796-014-9492-7.

[3] O. I. Abiodun, E. O. Abiodun, M. Alawida, R. S. Alkhawaldeh, and H. Arshad, "A review on the security of the Internet of Things: Challenges and solutions," *Wireless Personal Communications*, vol. 119, pp. 2603–2637, 2021, doi: 10.1007/s11277-021-08348-9.

[4] M. M. Alani and A. Miri, "Towards an explainable universal feature set for IoT intrusion detection," *Sensors*, vol. 22, no. 15, p. 5690, 2022, doi: 10.3390/s22155690.

[5] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, 2018, doi: 10.1016/j.future.2017.11.022.

[6] Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, p. 20, 2019, doi: 10.1186/s42400-019-0038-7.

[7] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[8] A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Gener. Comput. Syst., vol. 82, pp. 761–768, 2018, doi: 10.1016/j.future.2017.08.043.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[10] M. Awad and S. Fraihat, "Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems," *J. Sens. Actuator Netw.*, vol. 12, no. 5, p. 67, 2023, doi: 10.3390/jsan12050067.

[11] S.-M. Tseng, Y.-Q. Wang, and Y.-C. Wang, "Multi-class intrusion detection based on Transformer for IoT networks using CIC-IoT-2023 dataset," *Future Internet*, vol. 16, no. 8, p. 284, 2024, doi: 10.3390/fi16080284.

[12] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mob. Netw. Appl.*, vol. 27, no. 1, pp. 357–370, 2022, doi: 10.1007/s11036-021-01843-0.

[13] A. Alsulami, Q. Abu Al-Haija, A. Tayeb, and A. Alqahtani, "An intrusion detection and classification system for IoT traffic with improved data engineering," Applied Sciences, vol. 12, no. 23, p. 12336, 2022, doi: 10.3390/app122312336.

[14] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the Internet of Things networks with edge capabilities," *Appl. Soft Comput.*, vol. 155, p. 111434, 2024, doi: 10.1016/j.asoc.2024.111434.

[15] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics*, vol. 13, no. 6, p. 1053, 2024, doi: 10.3390/electronics13061053.

[16] N. Moustafa, B. Turnbull, and K. K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, 2018, doi: 10.1109/JIOT.2018.2871719.

[17] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, 2021, doi: 10.1002/ett.4150.

[18] M. Banaamah and I. Ahmad, "Intrusion detection in IoT using deep learning," Sensors, vol. 22, no. 21, p. 8417, 2022, doi: 10.3390/s22218417.

[19] Z. Wang, H. Chen, S. Yang, X. Luo, D. Li, and J. Wang, "A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization," *PeerJ Comput. Sci.*, vol. 9, p. e1569, 2023, doi: 10.7717/peerj-cs.1569.

[20] R. Panigrahi, S. Borah, A. K. Bhoi, M. F. Ijaz, M. Pramanik, Y. Kumar, and R. H. Jhaveri, "A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets," *Mathematics*, vol. 9, no. 7, p. 751, 2021, doi: 10.3390/math9070751.
[21] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-layer classification approach for intrusion detection in IoT networks based on deep learning," *Sensors*, vol. 21, no. 9, p. 2987, 2021, doi: 10.3390/s21092987.
[22] D. Z. Rodríguez, O. D. Okey, S. S. Maidin, E. U. Udo, and J. H. Kleinschmidt, "Attentive transformer deep learning algorithm for intrusion detection on IoT systems using automatic Xplainable feature selection," *PLoS ONE*, vol. 18, no. 10, p. e0286652, 2023, doi: 10.1371/journal.pone.0286652.
[23] Tareq, B. M. Elbagoury, S. El-Regaily, and E. M. El-Horbaty, "Analysis of TON-IoT, UNSW-NB15, and Edge-IIoT datasets using DL in cybersecurity for IoT," *Applied Sciences*, vol. 12, no. 19, p. 9572, 2022, doi: 10.3390/app12199572.
[24] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, and J. Kwak, "IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J. Big Data*, vol. 10, no. 1, p. 15, 2023, doi: 10.1186/s40537-023-00694-8.
[25] A. Qasem, M. H. Qutqut, F. Alhaj, Y. Kilani, M. Tubishat, and R. Al-Qudah, "SRFE: A stepwise recursive feature elimination approach for network intrusion detection systems," Peer-to-Peer Netw. Appl., vol. 17, no. 6, pp. 3634–3649, 2024, doi: 10.1007/s12083-024-01763-2.
[26] O. Friha, M. A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, and K. K. R. Choo, "2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT," *Computers & Security*, vol. 127, p. 103097, 2023, doi: 10.1016/j.cose.2023.103097.
[27] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling realistic adversarial attacks against network intrusion detection systems," *Digital Threats: Research and Practice*, vol. 3, no. 3, pp. 1–19, 2022, doi: 10.1145/3469659.
[28] He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 1, pp. 538–566, 2023, doi: 10.1109/COMST.2022.3233793.
[29] Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 389–422, 2002, doi: 10.1023/A:1012487302797.
[30] Ullah and Q. H. Mahmoud, "A scheme for generating a dataset for anomalous activity detection in IoT networks," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science, vol. 12109), C. Goutte and X. Zhu, Eds. Cham, Switzerland: Springer, 2020, pp. 508–520, doi: 10.1007/978-3-030-47358-7_52.
[31] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON-IoT datasets," *Sustain. Cities Soc.*, vol. 72, p. 102994, 2021, doi: 10.1016/j.scs.2021.102994.