

RESEARCH PAPER

A Neural Network-Based Approach for Trust-Guided Anomaly Detection in IoT Networks

Vikas Sadashiv More¹, Monaj Varshney², Lomesh Kautik Ahire³

¹ Ph.D. Scholar, Department of Computer Engineering & Applications, Mangalayatan University, Aligarh, Uttar Pradesh, India

Email: 20242570_vikas@mangalayatan.edu.in

² Associate Professor, Department of Computer Engineering & Applications, Mangalayatan University, Aligarh, Uttar Pradesh, India

Email: manoj.varshney_dcea@mangalayatan.edu.in

³ Assistant Professor, Vishwakarma Institute of Technology, Pune, Maharashtra, India

Email: lomesh.ahire@vit.edu

Abstract

The rapid growth of heterogeneous Internet of Things (IoT) networks has expanded the attack surface and exposed the limitations of static, signature-based intrusion detection. This study proposes and evaluates a trust-guided anomaly-detection framework in which a label-free behavioral trust score is computed for each network flow and appended to conventional traffic features. The score combines connection state, missed bytes, packet and byte balance, flow duration, source and destination reputation, and interaction confidence. Experiments were conducted on the TON_IoT network dataset containing 211,043 flows (50,000 normal and 161,043 attack records). Four binary detectors were evaluated on identical splits: a trust-agnostic Random Forest (RF), a trust-guided RF, an LSTM autoencoder, and a trust-guided LSTM autoencoder; a separate trust-guided RF performed ten-class attack classification. On the stratified test split, the trust-guided RF achieved 99.93% accuracy, an F1 score of 0.9995, and a 0.24% false-positive rate, compared with 99.87%, 0.9992, and 0.33%, respectively, for the baseline RF. This corresponds to a 28% relative reduction in false positives. The LSTM autoencoders achieved F1 scores near 0.81 but produced false-positive rates above 0.92. A grouped source-IP validation yielded an F1 score of 0.9627 with the full feature set; performance declined substantially after removing port, DNS, and HTTP status features, revealing dataset-specific dependence. The findings show that behavior-derived trust features can improve supervised IoT anomaly detection, while also demonstrating the need for grouped and cross-dataset validation before deployment.

Keywords: Internet of Things; anomaly detection; behavioral trust; Random Forest; LSTM autoencoder; TON_IoT; intrusion detection; leakage analysis.

How to cite this article: Vikas Sadashiv More, Monaj Varshney, Lomesh Kautik Ahire, “A Neural Network-Based Approach for Trust-Guided Anomaly Detection in IoT Networks”. Int J Drug Deliv Technol. 2026;16(86s): 1786-1803; DOI: 10.25258/ijddt.16.86s.162

1. INTRODUCTION

The Internet of Things (IoT) connects sensors, actuators, gateways, and software services across healthcare, manufacturing, smart-city, transportation, and agricultural systems. These environments support continuous monitoring and automated decision-making, but their heterogeneity, scale, and distributed control also create significant security exposure [1, 2]. Compromise of an IoT endpoint can affect confidentiality, integrity, availability, and, in cyber-physical settings, operational safety [3].

Conventional firewalls and signature-based intrusion-detection systems remain important, yet they often struggle with previously unseen attacks, dynamic device behavior, imbalanced traffic, and the limited computing capacity of edge nodes [4]. Trust management provides a complementary mechanism by evaluating whether a device or flow behaves consistently with expected interaction patterns [5, 6]. Neural and ensemble models can then use those behavioral signals to learn nonlinear decision boundaries that are difficult to express as fixed rules [7].

This study operationalizes that combination on the TON_IoT benchmark [8, 9]. A behavior-only trust score is calculated without using the ground-truth attack label and is appended to a common flow-feature set. Trust-guided and trust-agnostic Random Forest and LSTM-autoencoder models are trained and evaluated under identical preprocessing and data partitions. A grouped source-IP experiment is also used to test whether random splitting inflates performance through repeated device identities.

The study makes three contributions: (i) a reproducible trust-score formulation based on observable flow behavior; (ii) a controlled comparison of supervised and reconstruction-based detectors with and without trust features; and (iii) an explicit leakage check using disjoint source-IP groups and a reduced feature policy. The remainder of the paper reviews related work, defines the research problem, describes the materials and methods, reports the results, discusses their implications and limitations, and concludes with future directions.

1.1 Related Work

Trust-centered IoT security initially focused on routing, access control, and node reputation. Qin et al. used direct observations and neighbor recommendations to favor trustworthy routes [10], while Alrawais et al. examined security and privacy controls at the fog layer [11]. Mahalle et al. applied fuzzy trust to IoT access control [12]. These studies established trust as a dynamic security signal, although many formulations depend on peer recommendations or task-specific rules.

Neural anomaly detection has developed in parallel. Surveys of in-vehicle anomaly detection describe the diversity of machine-learning approaches and data representations [13]. Attention-based Bi-LSTM models capture temporal dependencies [14], hybrid CNN-GRU architectures combine spatial and temporal feature extraction [15], and TCN-BiLSTM models extend this idea to temporal faults [16]. Belief-based trust evaluation provides a complementary behavioral mechanism for wireless sensor networks [17], while deep-autoencoder research demonstrates the value and tuning challenges of reconstruction-based detection [7].

Hybrid and deployment-oriented methods increasingly combine deep learning, trust, and edge or fog computing. CNN-LSTM intrusion detection has been evaluated for industrial IoT traffic [18], while fog-computing research highlights scheduling and resource constraints that affect practical model placement [19]. Trust-weighted federated learning has been proposed for anomaly detection [20]. Healthcare IoT research emphasizes the need to combine communication quality, context, and security monitoring [21], and deep ensembles have been applied to unknown network attacks [22].

Recent work broadens the design space through graph neural networks [23], wearable-data anomaly frameworks [24], attention-based trust evaluation [25], Raspberry Pi-based edge detection [26], adversarial-robust hybrid networks [27], asynchronous federated learning [28], multi-agent trust-aware routing [29], swarm-optimized detection [30], cluster-aware federated graph trust [31], edge-oriented deep-learning integration [32], reinforcement-based trust adaptation [33], and blockchain-supported trust management [34]. These directions show growing interest in adaptive, distributed, and tamper-resistant trust mechanisms.

Two gaps motivate the present study. First, many trust models rely on labeled attacks, simulated recommendations, or infrastructure that may not be available at deployment. Second, intrusion-detection studies frequently emphasize a single random split even though repeated identity patterns can inflate performance [35, 36]. Trust-management reviews also identify interoperability, context dependence, and evaluation consistency as unresolved issues [37]. Accordingly, this work computes trust from observable flow behavior and reports both random and grouped source-IP evaluations.

1.2 Problem Statement and Research Question

IoT endpoints remain vulnerable to spoofing, command injection, unauthorized access, data theft, scanning, denial-of-service attacks, and malware. Static detectors may miss subtle or novel attacks, while overly sensitive anomaly models can overwhelm operators with false alerts. A useful detector must therefore improve discrimination without relying on unavailable labels during trust computation and without hiding dataset-specific dependencies.

The primary research question is: Does adding a behavior-derived trust score to otherwise identical anomaly-detection models improve predictive performance and reduce false positives? A secondary question examines whether any observed improvement persists when source IPs are separated between training and test sets and identity-correlated features are restricted.

2. MATERIALS AND METHODS

The proposed methodology follows a quantitative, experimental design and is realised as a single, reproducible Python pipeline. Raw TON_IoT network flow records are loaded and cleaned; a behaviour-only trust score is computed for every flow; the resulting trust-augmented data table is partitioned into stratified training, validation, and test sets; four detection models are trained and evaluated; and the same trust-guided Random Forest is finally re-evaluated under a strict grouped split to probe leakage. All experiments share the same random seed (42), the same preprocessing, and the same metric definitions to ensure a like-for-like comparison.

The pipeline implements (i) a trust-agnostic Random Forest binary classifier; (ii) a trust-guided Random Forest binary classifier obtained by appending the trust score and trust level to the feature matrix; (iii) an LSTM autoencoder trained on normal-traffic sequences only and used as an unsupervised anomaly detector; (iv) a trust-guided LSTM autoencoder with the trust features included; and (v) a multiclass Random Forest that assigns each flow to one of ten classes — normal, backdoor, DDoS, DoS, injection, MITM, password, ransomware, scanning, and XSS. For the LSTM models the reconstruction-error threshold is chosen by maximising F1 on the validation split, ensuring that both deep and ensemble models are tuned under identical model-selection criteria.

To address the well-known risk of source-IP leakage in network intrusion benchmarks, the pipeline also performs a strict validation experiment. Training and test sets are split by source IP using grouped sampling so that no source IP appears in both partitions, and the trust-guided Random Forest is re-trained both with the full feature set and with a reduced feature set in which port numbers and HTTP status codes are removed. This dual-mode strict evaluation distinguishes the genuine contribution of behavioural and trust features from the contribution of identity-correlated artefacts.

2.1 System Architecture

The system is organised in six modular layers, consistent with the architecture proposed in the original synopsis but instantiated concretely in software:

- **IoT Device Layer** — physical sensors, actuators, and gateways that generate the raw telemetry. In this study the layer is represented by the TON_IoT network capture, which records flow-level features such as connection state, byte and packet counts, duration, ports, and protocol identifiers.
- **Data Collection and Preprocessing Layer** — ingests the CSV records, drops non-discriminative dash columns (any object column with more than 95% "-" values), imputes missing numeric values with the median, imputes missing categorical values with the literal string "missing", standardises numeric features with z-score normalisation, and applies one-hot encoding to categorical features.
- **Trust Evaluation Module** — computes a behavioural trust score for every flow from connection-state quality, missed bytes, packet balance, byte balance, duration, source and destination reputations, and an interaction-confidence factor (Section 2.3).

- **Neural-Network Anomaly Detection Module** — hosts the LSTM autoencoder. The encoder compresses an eight-step sequence of flow vectors to a 48-dimensional latent representation; the decoder reconstructs the sequence; the per-sequence mean squared error is used as the anomaly score.
- **Decision Engine** — in the binary task, a Random Forest classifier produces calibrated probabilities and class labels using the trust-augmented feature vector; in the multiclass task, a Random Forest assigns one of ten attack labels.
- **Monitoring and Alert Dashboard** — a Streamlit dashboard renders dataset profile, trust distribution, model comparison, confusion matrices, and ROC curves, allowing an operator to inspect the model's behaviour interactively.

2.2 Dataset and Data Partitioning

All experiments use the publicly available TON_IoT network dataset developed at UNSW Canberra [8, 9], specifically the `train_test_network.csv` flow-level partition. After cleaning, the table contains 211,043 records and 44 columns: 50,000 normal flows and 161,043 attack flows. Ten classes are represented, as summarized in Table 1.

Table 1. Class distribution of the TON_IoT network capture used in this study.

Class	Number of flows
normal	50,000
backdoor	20,000
ddos	20,000
dos	20,000
injection	20,000
password	20,000
ransomware	20,000
scanning	20,000
xss	20,000
mitm	1,043
Total	211,043

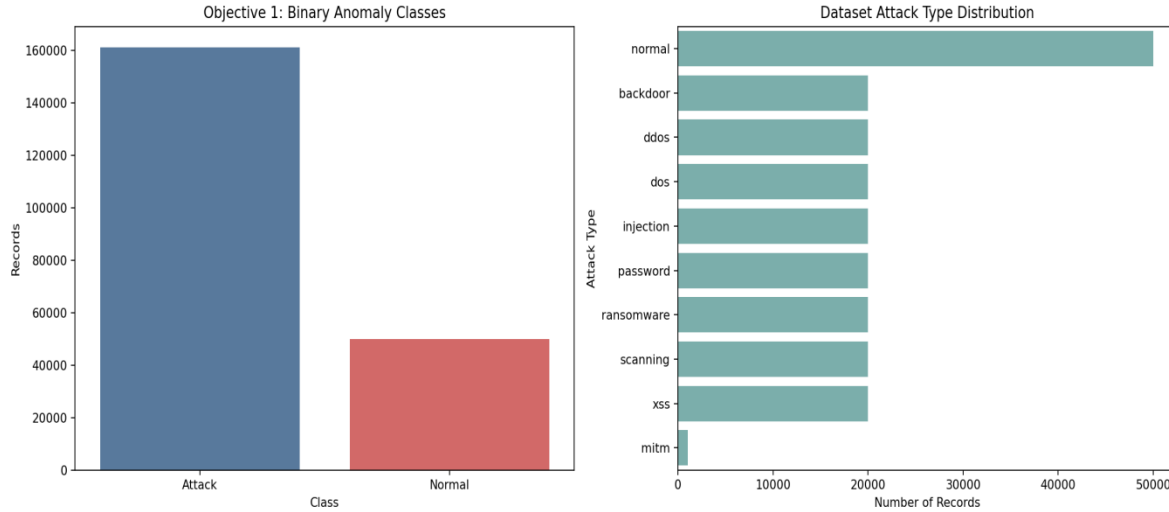


Figure 1. Dataset composition: binary (normal vs attack) distribution and attack-type composition.

The data is randomly split, with stratification on the binary label, into 70% training (147,729 rows), 15% validation (31,657 rows), and 15% test (31,657 rows). The class balance is preserved across splits, with an attack rate of approximately 76.3% in each partition. To probe spatial generalisation the data is also re-split with GroupShuffleSplit on the source-IP column, producing 206,189 training rows from 43 source IPs and 4,854 test rows from 8 disjoint source IPs. No flows from a given source IP appear in both partitions of the grouped split.

2.3 Trust Score Computation

The trust score is computed entirely from observable behavioural attributes of each flow, without any reference to the attack label. The score therefore behaves as a prior over benignness, derived from the same kind of signals that a network operator would intuitively use when judging whether a flow looks well-behaved.

Five base behavioural scores are calculated as follows.

Connection-state score (s_{conn}). The TON_IoT connection-state field is mapped to {SF, S1, S2, S3} $\rightarrow$ 1.0 (strong, fully established), {REJ, RSTO, RSTR, RSTOS0, RSTRH, SH, SHR} $\rightarrow$ 0.2 (weak, rejected or reset), and any other state $\rightarrow$ 0.55 (intermediate).

Missed-byte score (s_{miss}). A monotonically decreasing transform of missed bytes m , mapping zero loss to 1.0 and progressively penalising large losses:

$$s_{miss} = 1 / (1 + \ln(1 + m)) \quad (1)$$

Packet-balance score (s_{pkt}). A measure of bidirectional symmetry between the source and destination packet counts ps , pd :

$$s_{pkt} = 1 - |p_s - p_d| / (p_s + p_d + 1) \quad (2)$$

Byte-balance score (s_{byte}). The analogous symmetry measure for source and destination byte counts bs , bd :

$$s_{byte} = 1 - |b_s - b_d| / (b_s + b_d + 1) \quad (3)$$

Duration score (s_{dur}). A log-decay penalty on long-lived flows, since extreme durations frequently correspond to scanning or persistent connections:

$$s_{dur} = 1 / (1 + \ln(1 + d)) \quad (4)$$

These are linearly combined into a per-flow base behaviour score with empirically chosen weights that emphasise connection completeness:

$$s_{base} = 0.30 s_{conn} + 0.20 s_{miss} + 0.20 s_{pkt} + 0.15 s_{byte} + 0.15 s_{dur} \quad (5)$$

The base score is then contextualised using two reputation aggregates and one confidence factor:

- Source reputation r_{src} is the mean s_{base} across all flows that share the same src_ip . It rewards source IPs that consistently emit well-formed traffic.
- Destination reputation r_{dst} is the mean s_{base} across all flows that share the same dst_ip . It captures whether the destination is associated with predominantly clean interactions.
- Interaction confidence $c = \ln(1 + n_{src}) / \ln(1 + n_{max})$, where n_{src} is the number of flows observed from the source IP and n_{max} is the global maximum. It dampens the influence of singleton sources for which no reliable behavioural history exists.

The final trust score is:

$$trust = clip[0,1](0.60 s_{base} + 0.25 r_{src} + 0.10 r_{dst} + 0.05 c) \quad (6)$$

Each flow is additionally assigned a categorical trust level using fixed cut-points: low ($[0, 0.35]$), medium ($(0.35, 0.70]$), and high ($(0.70, 1.0]$). On TON_IoT the resulting trust score has mean 0.768 and standard deviation 0.091, with a minimum of 0.357 and maximum of 0.964. The

distribution and its dependence on flow class are illustrated in Figures 2 and 3.

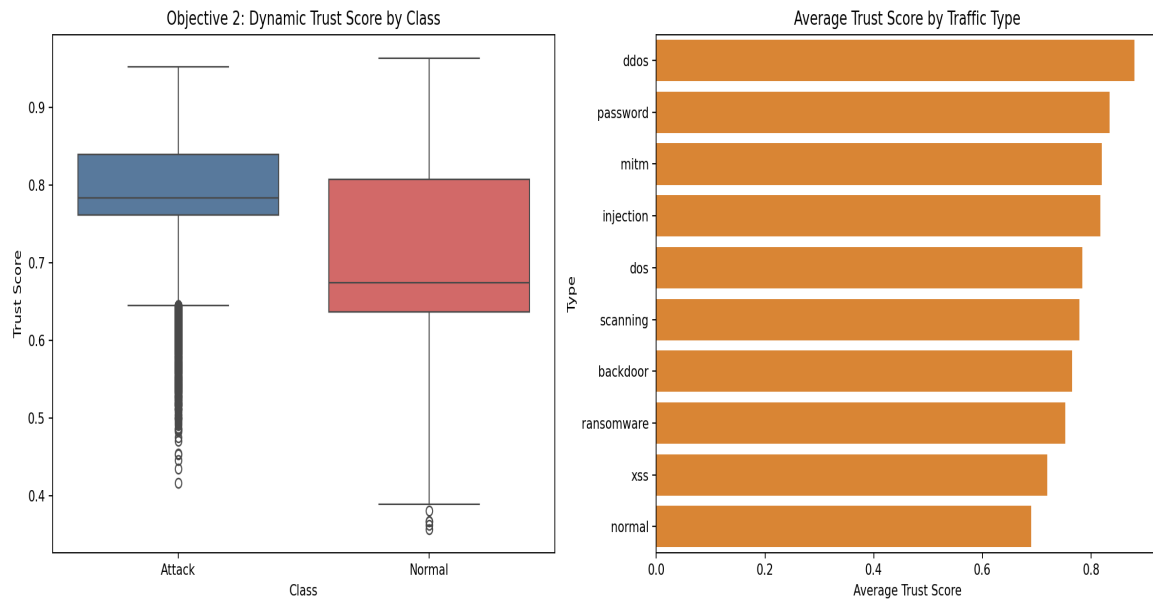


Figure 2. Trust-score analysis: distribution of trust scores conditioned on binary class and mean trust per attack type.

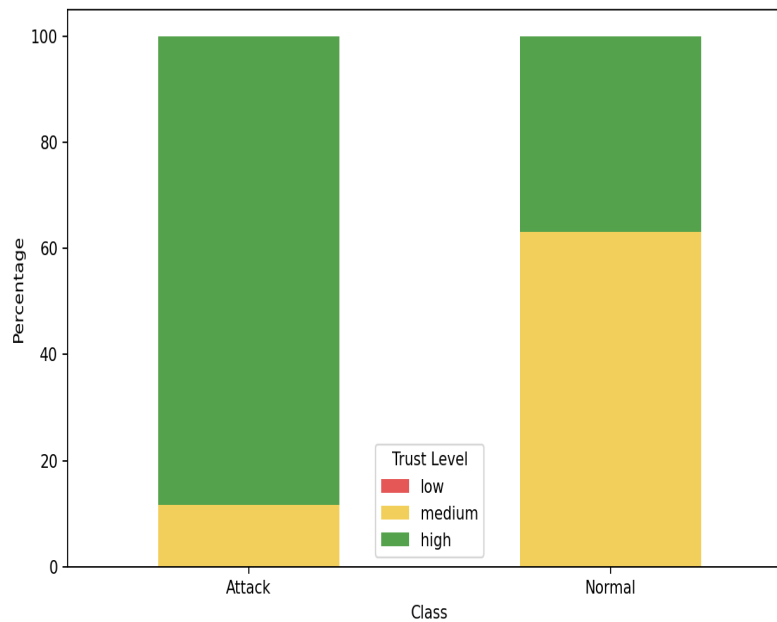


Figure 3. Composition of trust levels (low / medium / high) per binary class.

2.4 Model Implementation

2.4.1 Random Forest Classifiers

Both binary Random Forest models are scikit-learn pipelines comprising a ColumnTransformer (median imputation and z-score scaling for numeric features; constant imputation and one-hot encoding for categorical features) followed by a RandomForestClassifier with 120 estimators, maximum depth 18, balanced class weights, and

parallel training. The trust-agnostic baseline operates on 23 input features; the trust-guided variant operates on 25 features (the same 23 plus trust_score and trust_level). The multiclass Random Forest uses the same architecture but trains against the ten-class type column.

2.4.2 LSTM Autoencoder

The LSTM autoencoder is implemented in TensorFlow/Keras. It takes a sequence of eight consecutive flow vectors, encodes the sequence through a 48-unit LSTM

layer (tanh activation) followed by a 15% dropout, then applies a RepeatVector and a second 48-unit LSTM with `return_sequences=True`, and finally a TimeDistributed Dense layer that reconstructs the per-step feature vector. The model is trained with Adam and a mean-squared-error loss, batch size 256, for up to five epochs with early stopping on validation loss (patience 2). Training is performed on the normal-traffic subset only: the autoencoder learns the distribution of legitimate behaviour and is then used to score arbitrary sequences by their reconstruction error.

The decision threshold is chosen on the validation split by sweeping percentiles of the training reconstruction-error distribution and percentiles of the validation reconstruction-error distribution, selecting the candidate that maximises validation F1. The same threshold-selection procedure is

applied to both the trust-agnostic and the trust-guided LSTM variants.

2.4.3 Multiclass Detector

For the multiclass task a separate Random Forest with the same architecture is trained against the ten attack categories using the trust-guided feature set. Per-class precision, recall, and F1 are reported alongside macro and weighted averages.

2.5 Experimental Setup and Evaluation Metrics

All experiments are run from a single deterministic pipeline with `random_state=42`. The data partition is 70% training, 15% validation, 15% test, with stratification on the binary label, exactly as specified in the original research design. Hyperparameters are summarised in Table 2.

Table 2. Pipeline hyperparameters used in all experiments.

Hyperparameter	Value
<code>random_state</code>	42
<code>test_size</code>	0.15
<code>validation_size</code>	0.15
<code>rf_estimators</code>	120
<code>rf_max_depth</code>	18
<code>rf_class_weight</code>	balanced
<code>sparse_dash_drop_ratio</code>	0.95
<code>lstm_sequence_length</code>	8
<code>lstm_epochs</code>	5 (with early stopping, patience 2)
<code>lstm_batch_size</code>	256
<code>lstm_max_train_rows</code>	35,000 (normal only)
<code>lstm_max_eval_rows</code>	65,000
<code>anomaly_percentile</code>	97.5 (fallback initialisation)

Evaluation metrics for the binary task are accuracy, precision, recall, F1, false positive rate (FPR), and ROC-AUC. The confusion matrix is reported in absolute counts (TN, FP, FN, TP). For the multiclass task, per-class precision/recall/F1 with support, plus macro and weighted averages, are reported. Random and grouped splits are evaluated with the same metric definitions to enable direct comparison.

3. RESULTS

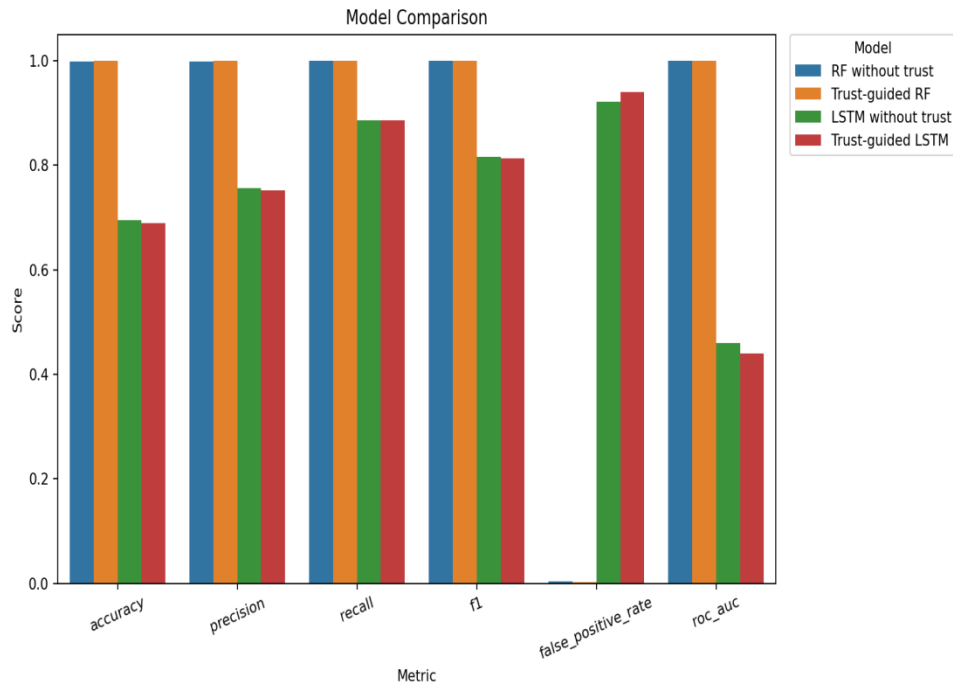
3.1 Overall Model Performance

Table 3 summarises binary performance on the random stratified test split for all four models. The trust-guided

Random Forest attains the highest measured F1 score (0.9995) and the lowest false positive rate (0.24%), narrowly outperforming the trust-agnostic Random Forest (F1 = 0.9992, FPR = 0.33%). Both LSTM-autoencoder variants are substantially weaker on this benchmark: while they recover an F1 score in the range 0.81–0.82, their FPR exceeds 0.92 because the model flags a large fraction of legitimate sequences as anomalous. The discrepancy is consistent with the well-known difficulty of unsupervised reconstruction-based detection on heterogeneous network captures, particularly when the legitimate-traffic class itself contains protocol variability that the autoencoder cannot fit in five epochs.

Table 3. Binary performance (normal vs attack) on the random stratified test split ($n = 31,657$).

Model	Acc.	Prec.	Recall	F1	FPR	ROC-AUC
Trust-guided Random Forest	0.9993	0.9993	0.9998	0.9995	0.0024	0.99999
Random Forest (no trust)	0.9987	0.9990	0.9994	0.9992	0.0033	0.99999
LSTM autoencoder (no trust)	0.6948	0.7561	0.8859	0.8158	0.9208	0.4602
Trust-guided LSTM autoencoder	0.6896	0.7520	0.8851	0.8132	0.9402	0.4392

*Figure 4. Comparison of accuracy, precision, recall, F1 and ROC-AUC across the four binary detectors.*

3.2 Confusion-Matrix Analysis

The confusion matrices for the two Random Forest models are reported in Figures 5 and 6. Out of 7,500 normal flows in the test set, the trust-agnostic model misclassifies 25 as attacks while the trust-guided model misclassifies only 18 — a relative reduction of 28%. False negatives drop even

more dramatically, from 15 to 4 out of 24,157 attack flows. The LSTM autoencoder confusion matrices (Figures 7 and 8) tell the opposite story: the reconstruction-error threshold optimised for F1 leans heavily towards recall, accepting a large number of false positives in order to retrieve most attacks.

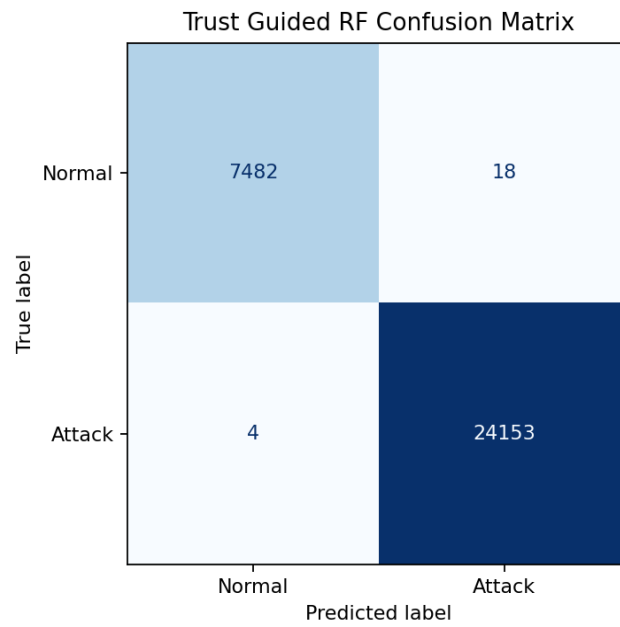


Figure 5. Confusion matrix — trust-guided Random Forest (random split).

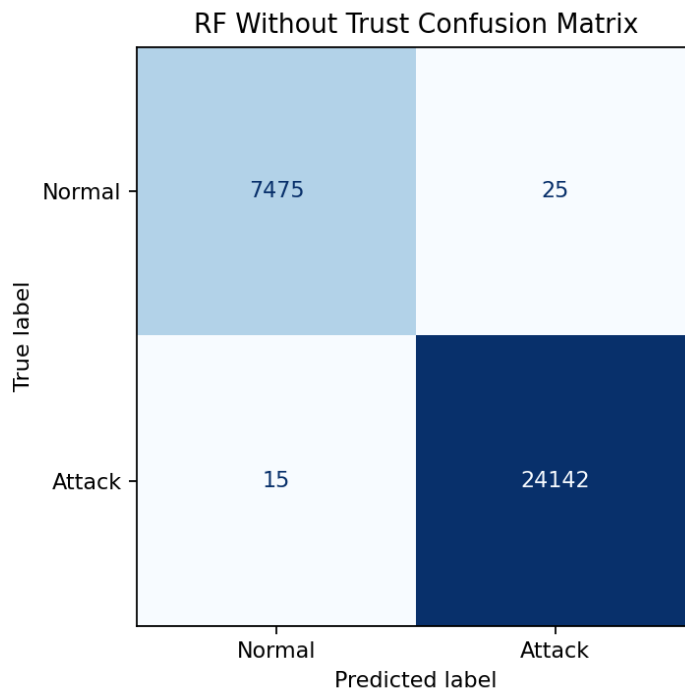


Figure 6. Confusion matrix — trust-agnostic Random Forest (random split).

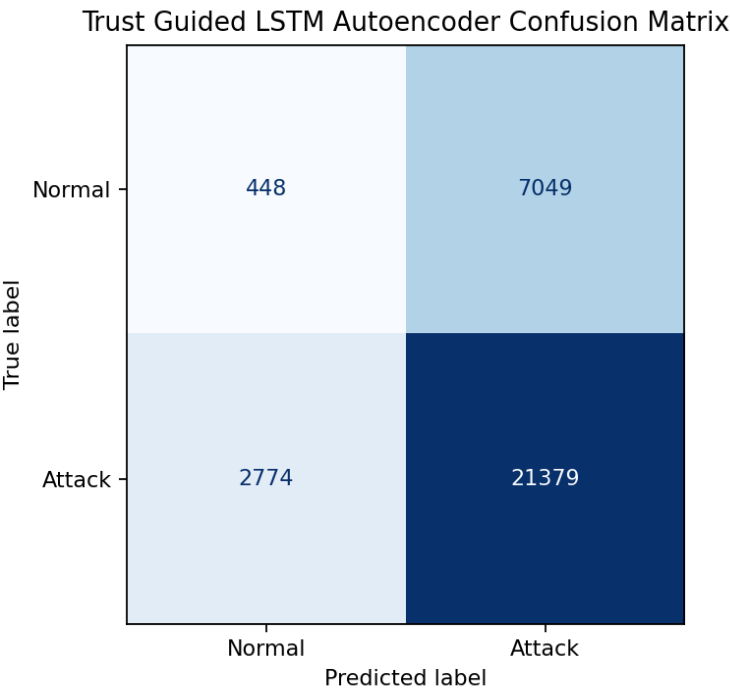


Figure 7. Confusion matrix — trust-guided LSTM autoencoder.

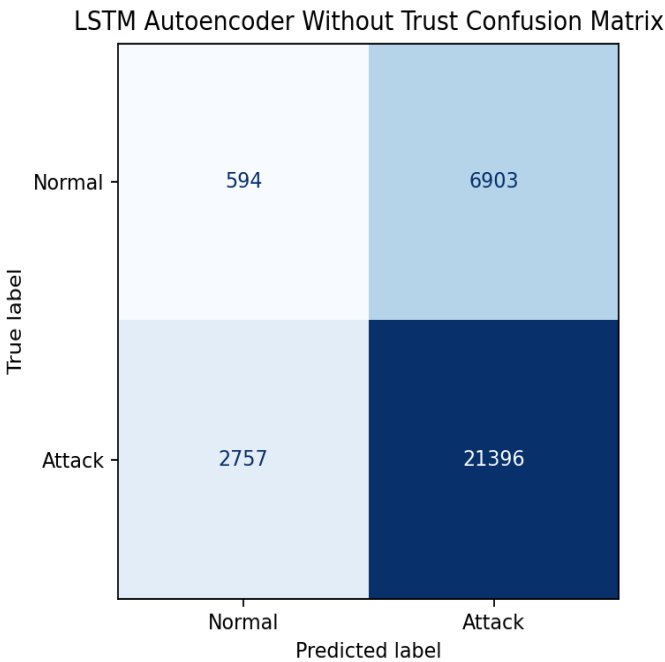


Figure 8. Confusion matrix — LSTM autoencoder without trust features.

3.3 Receiver Operating Characteristic Analysis

Figure 9 presents the receiver operating characteristic curves for the two Random Forest models. Both ROC-AUC values round to 0.99999. The trust-guided curve provides a

small advantage in the low-false-positive region, which is operationally important because even a modest false-positive rate can generate many alerts in high-volume networks.

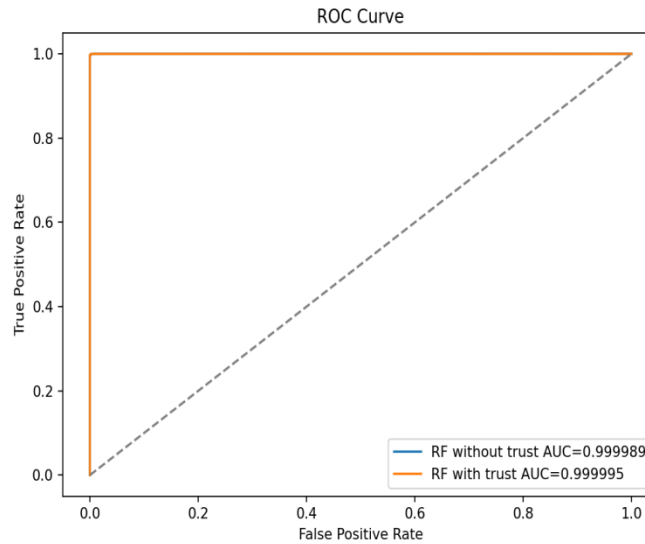


Figure 9. ROC curve for the Random Forest detectors with and without trust features.

3.4 Multiclass Attack Classification

Table 4 reports the ten-class performance of the trust-guided Random Forest. Overall accuracy is 0.9925, macro-averaged F1 is 0.9654, and weighted-averaged F1 is 0.9930.

MITM is the weakest class, with precision of 0.5623 and F1 of 0.7095. This result is consistent with its limited representation: MITM contributes 1,043 records overall and only 155 records in the test split, compared with approximately 20,000 records for most attack classes.

Table 4. Per-class performance of the trust-guided multiclass Random Forest on the random test split. Overall accuracy: 0.9925.

Class	Precision	Recall	F1	Support
backdoor	1.0000	1.0000	1.0000	3,037
ddos	0.9916	0.9877	0.9897	3,005
dos	0.9960	0.9801	0.9880	3,062
injection	0.9877	0.9847	0.9862	2,936
mitm	0.5623	0.9613	0.7095	155
normal	0.9989	0.9975	0.9982	7,500
password	0.9990	0.9890	0.9939	2,987
ransomware	0.9997	1.0000	0.9998	2,909
scanning	0.9924	0.9911	0.9918	3,033
xss	0.9960	0.9974	0.9967	3,033
Macro average	0.9524	0.9889	0.9654	31,657
Weighted average	0.9941	0.9925	0.9930	31,657

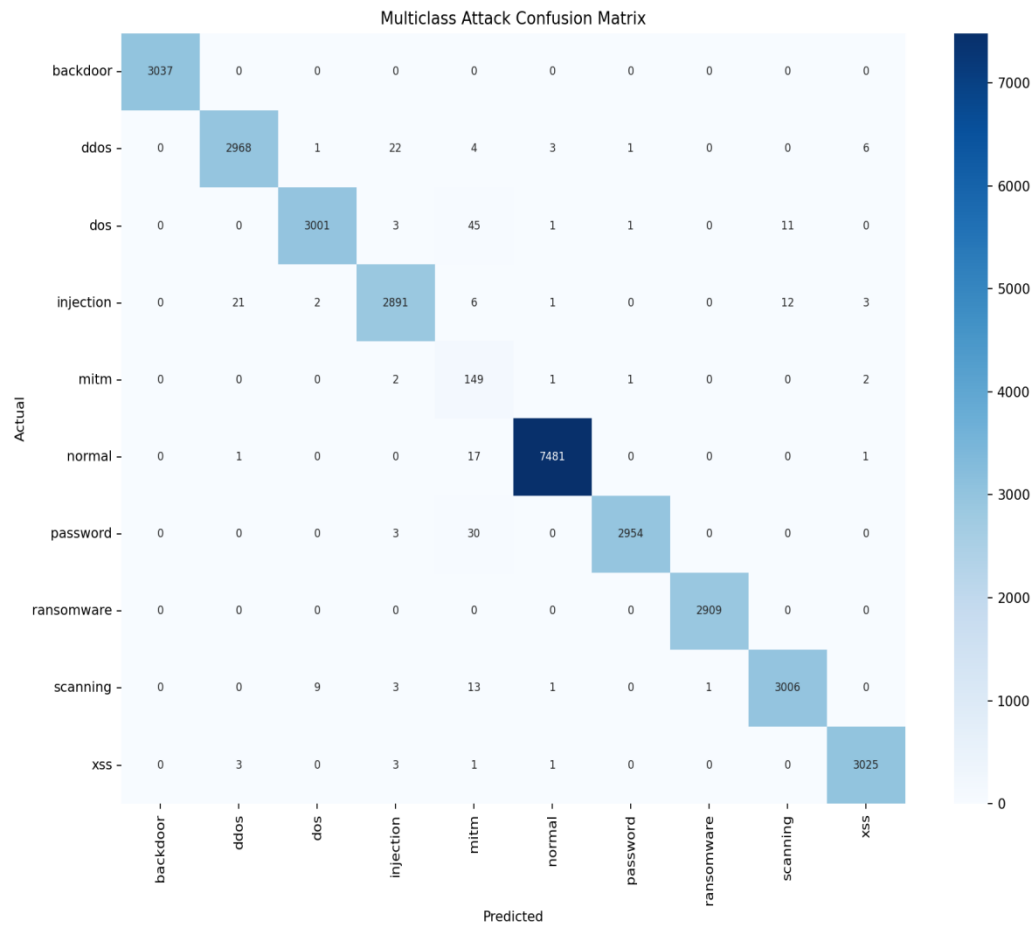


Figure 10. Confusion matrix — trust-guided multiclass Random Forest (ten attack categories).

3.5 Feature Importance

The Gini-based feature-importance ranking in Figure 11 identifies trust_score as the highest-ranked displayed feature, ahead of conventional indicators such as destination

port, path bytes, packet counts, and connection state. The result indicates that the engineered trust feature contributes substantial information to the Random Forest, although impurity-based importance should not be interpreted as causal evidence and may favor high-cardinality features.

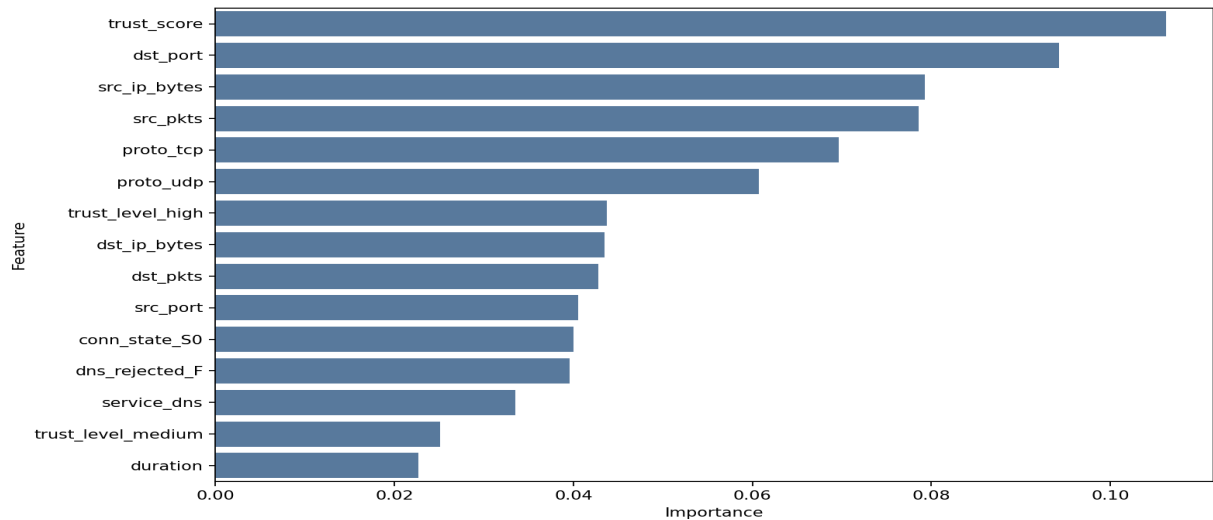


Figure 11. Top-15 features by importance — trust-guided Random Forest.

3.6 Trust-Guided versus Trust-Agnostic Models

The main comparison of the present research is conducted between a trust-agnostic Random Forest and a trust-guided Random Forest, where both types are trained,

tuned, and assessed in similar conditions with the only difference being the addition of two features in one of the types. The results of this comparison are presented in Table 5.

Table 5. Direct comparison of trust-agnostic versus trust-guided Random Forest on the random test split.

Metric	RF (no trust)	Trust-guided RF	Delta
Accuracy	0.9987	0.9993	+0.0006
Precision	0.9990	0.9993	+0.0003
Recall	0.9994	0.9998	+0.0004
F1	0.9992	0.9995	+0.0003
False positive rate	0.0033	0.0024	-0.0009 (-28%)
False positives (count)	25	18	-7
False negatives (count)	15	4	-11
ROC-AUC	0.99999	0.99999	≈ 0

The trust-guided variant uniformly improves over the baseline. The most consequential change is in the false-positive count: 18 versus 25, a 28% relative reduction. False negatives drop by 73% (from 15 to 4). These improvements arise from a feature-set difference of two columns out of twenty-five, which suggests that the trust score is encoding

information that the raw network features alone do not expose efficiently to the classifier — in particular, behavioural reputation aggregated across flows sharing the same source IP. Figures 12 and 13 visualise the false-positive reduction and the objective-level success summary, respectively.

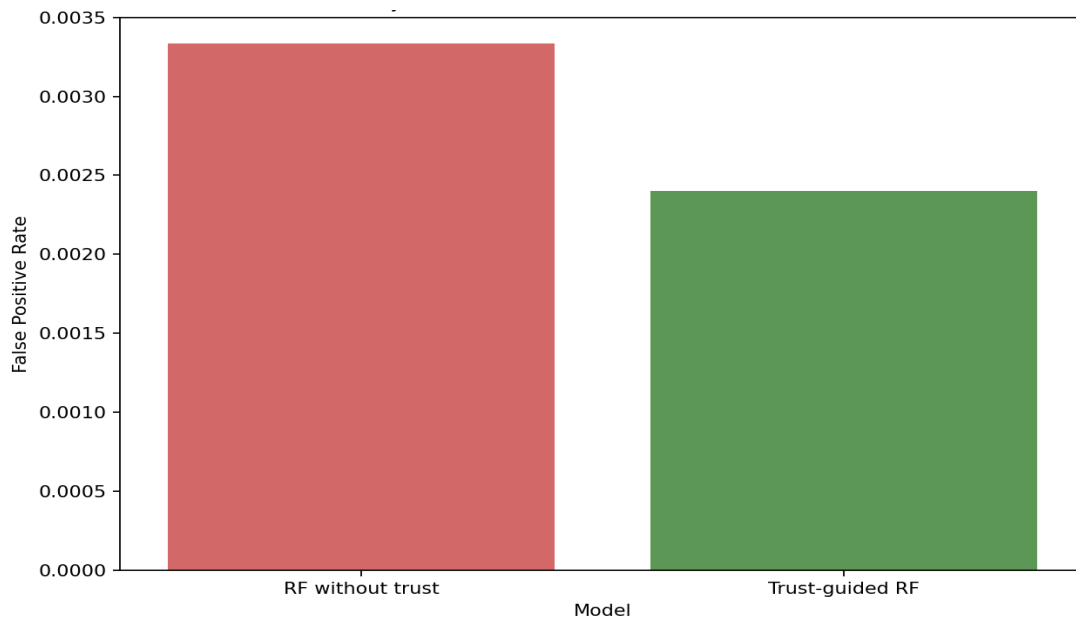


Figure 12. False positive rate reduction achieved by the trust-guided Random Forest.

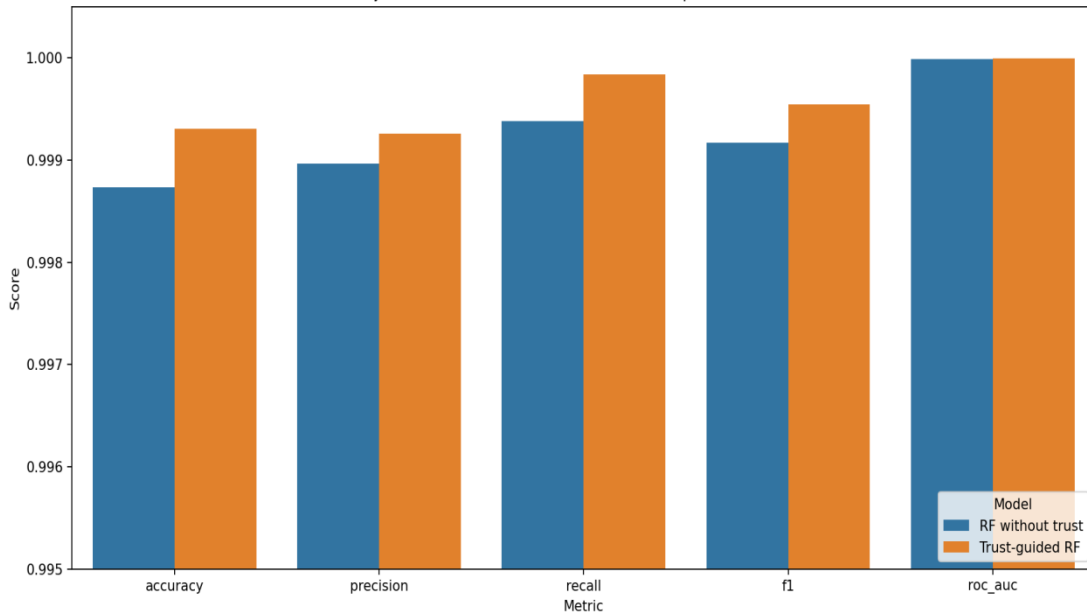


Figure 13. Baseline vs trust-guided success summary across binary metrics.

For completeness, Figure 14 compares the two LSTM-autoencoder variants. In this implementation the trust-guided LSTM autoencoder did not outperform the trust-agnostic LSTM autoencoder: both achieve similar F1 in the 0.81 range, and the trust-guided variant has a marginally higher FPR. We do not claim that the LSTM autoencoder outperformed the Random Forest. On the contrary, the trust-

guided Random Forest is the strongest measured detector in this study, and the LSTM autoencoder is reported as the realisation of the synopsis’s neural anomaly-detection objective whose measured performance was weaker than the ensemble baseline (Section 4.2 discusses the limitations).

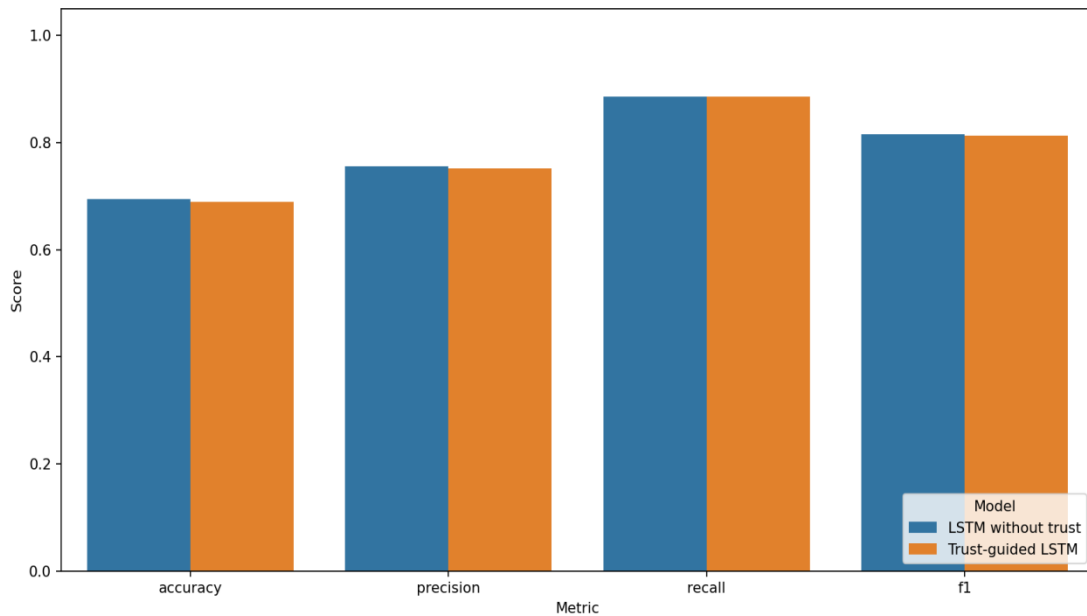


Figure 14. LSTM autoencoder — with vs without trust features.

3.7 Strict Validation and Leakage Check

Random splits on network intrusion datasets routinely yield optimistic estimates because the same source IPs (and therefore the same behavioural fingerprints) can appear in

both training and test partitions. Pattern leakage can therefore undermine reliability estimates [36]. To probe this, the data is re-split with GroupShuffleSplit on `src_ip`: training and test sets contain disjoint source IPs (43 vs 8), with 206,189 training rows and 4,854 test rows. The trust-

guided Random Forest is then trained under two feature policies: a full policy that keeps all 25 features, and a reduced policy that drops port numbers (src_port, dst_port), DNS request fields (dns_qclass, dns_qtype, dns_rcode), and

HTTP status code, on the grounds that these fields tend to correlate strongly with specific clients. Results are reported in Table 6 and visualised in Figures 15 and 16.

Table 6. Random versus strict grouped src_ip validation of the trust-guided Random Forest.

Experiment	Acc.	Prec.	Recall	F1	FPR	ROC-AUC
Random split, full features	0.9993	0.9993	0.9998	0.9995	0.0024	0.99999
Grouped src_ip, full features	0.9922	1.0000	0.9280	0.9627	0.0000	0.9965
Grouped src_ip, reduced features	0.7029	0.2587	0.9280	0.4046	0.3245	0.9661

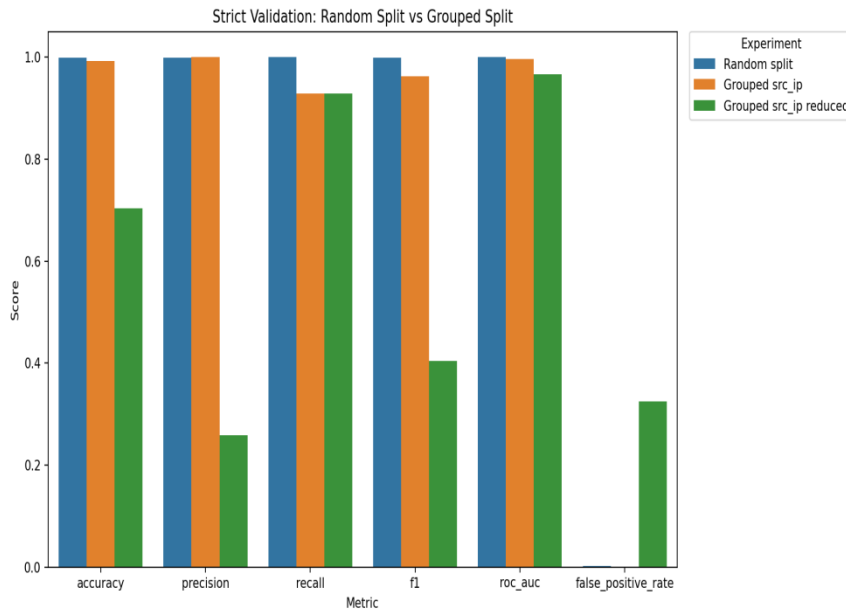


Figure 15. Trust-guided Random Forest — random split versus grouped src_ip split.

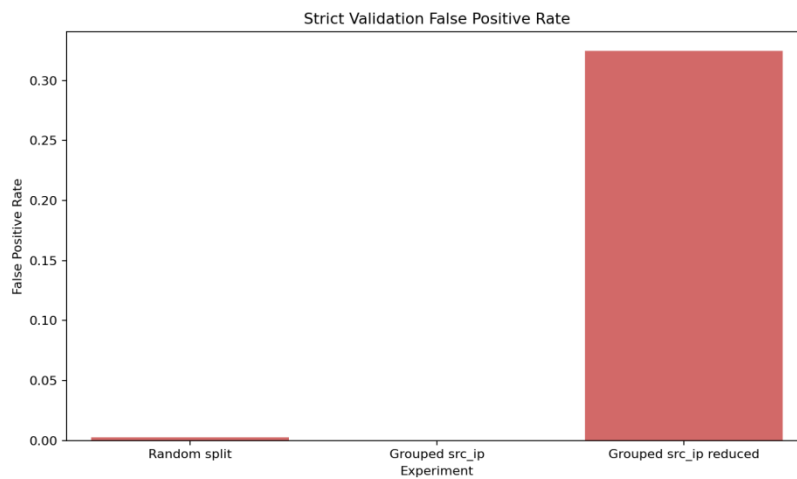


Figure 16. False positive rate under random split, grouped split (full features), and grouped split (reduced features).

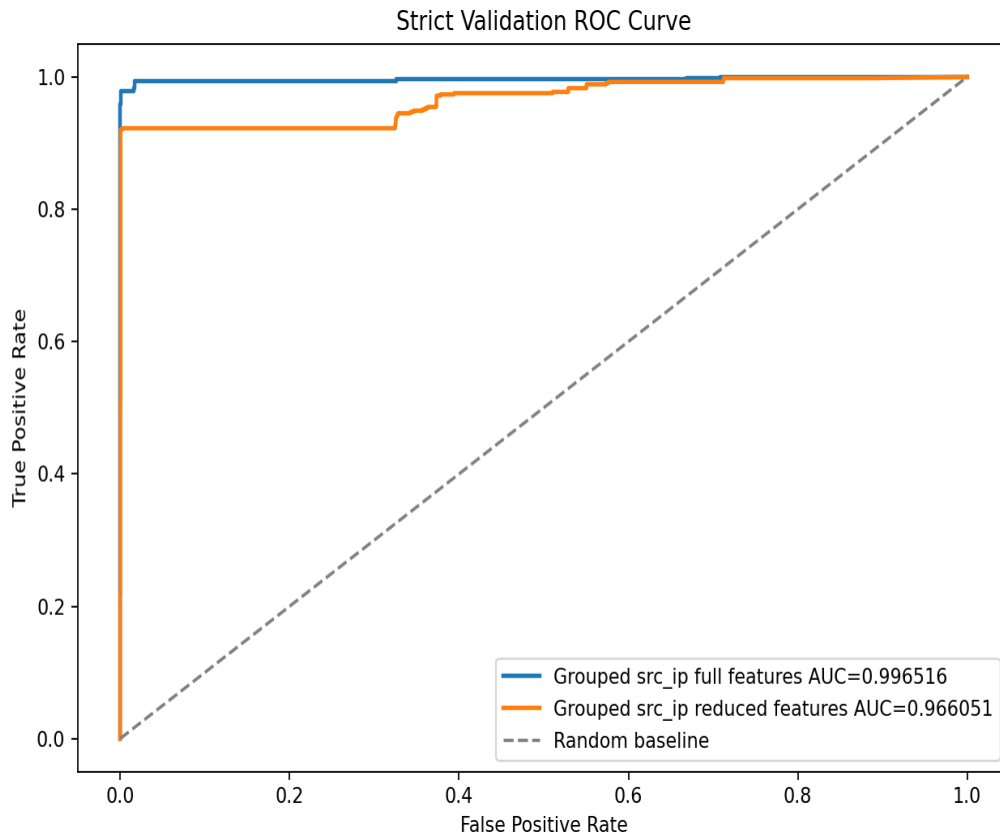


Figure 17. ROC curves under strict grouped *src_ip* validation.

The grouped evaluation produces two important findings. With the complete feature set, the trust-guided model reaches an F1 score of 0.9627 and an FPR of 0.0000 on previously unseen source-IP groups, although recall falls to 0.9280. When port, DNS, and HTTP status features are removed, F1 falls to 0.4046 and FPR rises to 0.3245. The retained ROC-AUC of 0.9661 indicates that ranking remains comparatively strong, but the operating threshold no longer transfers well. These results reveal substantial dataset-specific dependence and show why the random-split metrics should be treated as an upper-bound estimate rather than evidence of universal deployment performance.

4. DISCUSSION

4.1 Interpretation of Findings

The controlled Random Forest comparison supports the central proposition: adding *trust_score* and *trust_level* reduced both types of error on the random test split. False positives declined from 25 to 18 (28%), and false negatives declined from 15 to 4 (73%). Because preprocessing, training data, hyperparameters, and evaluation criteria were otherwise identical, the difference is attributable to the augmented feature representation within this experiment. The practical gain is primarily operational rather than numerical: fewer false alerts reduce analyst workload, while fewer missed attacks improve coverage.

The same benefit did not transfer to the reconstruction-based detector. The trust-guided LSTM autoencoder produced slightly lower F1 and higher FPR than its trust-agnostic counterpart. This contrast indicates that a trust feature can be useful to a supervised decision boundary without necessarily improving a model trained to reconstruct heterogeneous normal sequences. Training duration, the capped normal-traffic sample, sequence construction, and threshold selection are plausible contributors and require controlled ablation in future work.

The strict validation experiment materially qualifies the headline result. High performance with disjoint source IPs and the complete feature set suggests that the classifier can generalize beyond exact source identities, but the sharp deterioration under the reduced feature policy shows that ports, DNS fields, and HTTP status codes carry a large share of the discriminative signal. The framework should therefore be evaluated on external datasets, temporal splits, and live traffic before its trust score is considered deployment-ready.

4.2 Limitations

Several limitations of the current study should be acknowledged explicitly.

- Dataset dependence. All results are obtained on a single benchmark (TON_IoT). The capture is comprehensive but does not exhaust the diversity

of real-world IoT traffic, and absolute metrics may not transfer to other testbeds or to live deployment.

- Possible leakage and dataset dependence. The grouped experiment in Section 3.7 retains strong performance with full features but deteriorates sharply when identity-correlated fields are removed. Random-split values should therefore be interpreted as upper bounds.
- LSTM autoencoder performance. The reconstruction-based models attain F1 near 0.81 but very high false-positive rates. The five-epoch schedule, the 35,000-sequence training cap, heterogeneous normal traffic, and threshold sensitivity limit the conclusions that can be drawn about deep anomaly detection generally.
- Static trust formulation. The trust score is computed per flow with fixed weights. A live deployment would benefit from online updates, decay over time, and recalibration against ground truth where available.
- No cross-dataset evaluation. The framework was not evaluated on Bot-IoT or UNSW-NB15, despite both being mentioned in the original synopsis; this is left to future work.
- No real-device evaluation. The synopsis envisioned profiling inference latency on edge, gateway, and GPU hardware. The present paper is a software evaluation only.

5. CONCLUSIONS

This study presented a trust-guided anomaly-detection framework that converts observable flow behavior into a label-free trust score and supplies that score to ensemble and neural detectors. On the TON_IoT random test split, the trust-guided Random Forest achieved 99.93% accuracy, an F1 score of 0.9995, and an FPR of 0.24%, reducing false positives by 28% and false negatives by 73% relative to the trust-agnostic baseline. The LSTM autoencoder results were substantially weaker and did not improve after trust integration. Grouped source-IP validation further showed that the strongest metrics depend on the available feature policy. The evidence therefore supports a focused conclusion: behavioral trust features improved the supervised Random Forest in this experiment, but cross-dataset, temporal, and deployment validation are required before broader claims of generalization or operational readiness can be made.

5.1 Future Directions

Future work should first test the framework across Bot-IoT, UNSW-NB15, and newer zero-day captures using harmonized features and identical evaluation criteria. Temporal and leave-one-device-group-out validation should complement random and grouped splits. Second, the LSTM autoencoder should be retrained with a larger normal-only corpus, longer schedules, multiple sequence lengths, and calibrated thresholds; transformer, variational-

autoencoder, and graph-based alternatives should be assessed under the same compute budget. Third, the static trust score should be extended to an online, decay-sensitive formulation with cold-start handling, uncertainty estimates, and feedback-based recalibration. Fourth, the complete pipeline should be profiled on gateway and edge hardware for latency, memory, energy use, and alert throughput. Finally, federated and privacy-preserving variants should investigate trust-weighted aggregation, poisoning resistance, adversarial robustness, explainability, and operator-centered alert prioritization. These extensions would determine whether the observed reduction in false positives persists under distribution shift and real operational constraints.

Abbreviations

AI: artificial intelligence; CNN: convolutional neural network; DDoS: distributed denial of service; DNS: Domain Name System; DoS: denial of service; FPR: false-positive rate; GNN: graph neural network; HTTP: Hypertext Transfer Protocol; IoT: Internet of Things; LSTM: long short-term memory; MITM: man-in-the-middle; RF: Random Forest; ROC-AUC: area under the receiver operating characteristic curve.

Acknowledgments

The authors acknowledge UNSW Canberra for making the TON_IoT dataset available for academic cybersecurity research.

Funding Information

Author confirmation required before submission: the supplied manuscript did not include a funding declaration.

Author Contributions

Author confirmation required before submission: all authors should approve an ICMJE-compliant contribution statement, preferably using the CRediT taxonomy.

Conflict of Interest

Author confirmation required before submission: the supplied manuscript did not include a conflict-of-interest declaration.

Data Availability Statement

The TON_IoT dataset used in this study is publicly available from UNSW Research at <https://research.unsw.edu.au/projects/toniot-datasets> (accessed 26 August 2026).

REFERENCES

- [1] S. Nizetic, P. Solic, D. Lopez-de-Ipina Gonzalez-de-Artaza, and L. Patrono. "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future". *Journal of Cleaner Production*, vol. 274, pp. 122877, 2020. <https://doi.org/10.1016/j.jclepro.2020.122877>
- [2] A. Choudhary. "Internet of Things: A comprehensive overview, architectures, applications, simulation tools, challenges and future directions". *Discover Internet of Things*, vol. 4, no. 1, article 31, 2024. <https://doi.org/10.1007/s43926-024-00084-3>
- [3] Y. Li and Q. Liu. "A comprehensive review study of cyber-attacks and cyber security: Emerging trends and recent developments". *Energy Reports*, vol. 7, pp. 8176-8186, 2021. <https://doi.org/10.1016/j.egyr.2021.08.126>
- [4] H. H. A. Munshar, F. Jemili, O. Korbaa, and M. Alauthmaan. "Comprehensive analysis of intrusion detection systems for enhancing security in Internet of Things environments". *Discover Applied Sciences*, vol. 8, no. 4, article 473, 2026. <https://doi.org/10.1007/s42452-026-08458-8>
- [5] Y. Alghofaili and M. A. Rassam. "A Trust Management Model for IoT Devices and Services Based on the Multi-Criteria Decision-Making Approach and Deep Long Short-Term Memory Technique". *Sensors*, vol. 22, no. 2, article 634, 2022. <https://doi.org/10.3390/s22020634>
- [6] J. Caminha, A. Perkusich, and M. Perkusich. "A Smart Trust Management Method to Detect On-Off Attacks in the Internet of Things". *Security and Communication Networks*, vol. 2018, pp. 1-10, 2018. <https://doi.org/10.1155/2018/6063456>
- [7] I. D. Mienye and T. G. Swart. "Deep Autoencoder Neural Networks: A Comprehensive Review and New Perspectives". *Archives of Computational Methods in Engineering*, vol. 32, no. 7, pp. 3981-4000, 2025. <https://doi.org/10.1007/s11831-025-10260-5>
- [8] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar. "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems". *IEEE Access*, vol. 8, pp. 165130-165150, 2020. <https://doi.org/10.1109/ACCESS.2020.3022862>
- [9] N. Moustafa. "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets". *Sustainable Cities and Society*, vol. 72, pp. 102994, 2021. <https://doi.org/10.1016/j.scs.2021.102994>
- [10] D. Qin, S. Yang, S. Jia, Y. Zhang, J. Ma, and Q. Ding. "Research on Trust Sensing Based Secure Routing Mechanism for Wireless Sensor Network". *IEEE Access*, vol. 5, pp. 9599-9609, 2017. <https://doi.org/10.1109/ACCESS.2017.2706973>
- [11] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng. "Fog Computing for the Internet of Things: Security and Privacy Issues". *IEEE Internet Computing*, vol. 21, no. 2, pp. 34-42, 2017. <https://doi.org/10.1109/MIC.2017.37>
- [12] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad. "A fuzzy approach to trust based access control in Internet of Things". In *Proceedings of Wireless VITAE* 2013, pp. 1-5, 2013. <https://doi.org/10.1109/VITAE.2013.6617083>
- [13] O. Ozdemir, M. T. Isyapar, P. Karagoz, K. W. Schmidt, D. Demir, and N. A. Karagoz. "A Survey of Anomaly Detection in In-Vehicle Networks". *arXiv*, 2024. <https://doi.org/10.48550/arXiv.2409.07505>
- [14] S. Mishra, V. Kshirsagar, R. Dwivedula, and C. Hota. "Attention-Based Bi-LSTM for Anomaly Detection on Time-Series Data". In *Artificial Neural Networks and Machine Learning - ICANN 2021*, vol. 12891, pp. 129-140, Springer, 2021. https://doi.org/10.1007/978-3-030-86362-3_11
- [15] K. O. Adefemi, M. B. Mutanga, and O. A. Alimi. "A Hybrid CNN-GRU Deep Learning Model for IoT Network Intrusion Detection". *Journal of Sensor and Actuator Networks*, vol. 14, no. 5, article 96, 2025. <https://doi.org/10.3390/jsan14050096>
- [16] J. Bai, W. Zhu, S. Liu, C. Ye, P. Zheng, and X. Wang. "A Temporal Convolutional Network-Bidirectional Long Short-Term Memory Prediction Model for Temporal Faults in Industrial Equipment". *Applied Sciences*, vol. 15, no. 4, article 1702, 2025. <https://doi.org/10.3390/app15041702>
- [17] R. W. Anwar, A. Zainal, F. Outay, A. Yasar, and S. Iqbal. "BTEM: Belief based trust evaluation mechanism for Wireless Sensor Networks". *Future Generation Computer Systems*, vol. 96, pp. 605-616, 2019. <https://doi.org/10.1016/j.future.2019.02.004>
- [18] H. C. Altunay and Z. Albayrak. "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks". *Engineering Science and Technology, an International Journal*, vol. 38, pp. 101322, 2023. <https://doi.org/10.1016/j.jestech.2022.101322>
- [19] M. R. Alizadeh, V. Khajehvand, A. M. Rahmani, and E. Akbari. "Task scheduling approaches in fog computing: A systematic review". *International Journal of Communication Systems*, vol. 33, no. 16, pp. e4583, 2020. <https://doi.org/10.1002/dac.4583>
- [20] N. Chen, Y. Jin, Y. Li, and L. Cai. "Trust-based federated learning for network anomaly detection". *Web Intelligence*, vol. 19, no. 4, pp. 317-327, 2021. <https://doi.org/10.3233/WEB-210475>
- [21] S. M. Riazul Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak. "The Internet of Things for Health Care: A Comprehensive Survey". *IEEE Access*, vol. 3, pp. 678-708, 2015. <https://doi.org/10.1109/ACCESS.2015.2437951>

- [22] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh. "A Deep Learning Ensemble Approach to Detecting Unknown Network Attacks". *Journal of Information Security and Applications*, vol. 67, 103196, 2022. <https://doi.org/10.1016/j.jisa.2022.103196>
- [23] G. M., V. V. Snigdha, and Y. Jayadurga. "Anomaly Detection in IoT Networks using Graph Neural Networks". In *Proceedings of the 7th International Conference on Intelligent Sustainable Systems*, pp. 306-312, 2025. <https://doi.org/10.1109/ICISS63372.2025.11076391>
- [24] J. S. Sunny et al. "Anomaly Detection Framework for Wearables Data: A Perspective Review on Data Concepts, Data Analysis Algorithms and Prospects". *Sensors*, vol. 22, no. 3, article 756, 2022. <https://doi.org/10.3390/s22030756>
- [25] Y. Xu et al. "Attention-based neural networks for trust evaluation in online social networks". *Information Sciences*, vol. 630, pp. 507-522, 2023. <https://doi.org/10.1016/j.ins.2023.02.045>
- [26] T. Garalov and M. Elhajj. "Enhancing IoT Security: Design and Evaluation of a Raspberry Pi-Based Intrusion Detection System". In *Proceedings of the 2023 International Symposium on Networks, Computers and Communications*, pp. 1-7, 2023. <https://doi.org/10.1109/ISNCC58260.2023.10323656>
- [27] J. A. Gonzalez-Ramos, E. J. Abril, P. Fernandez, J. Prieto, and P. Chamoso. "Adversarial robustness evaluation of hybrid CNN-LSTM-transformer NIDS on evolving threats". *Journal of Information Security and Applications*, vol. 100, 104467, 2026. <https://doi.org/10.1016/j.jisa.2026.104467>
- [28] H. Xie, M. Xia, P. Wu, S. Wang, and K. Huang. "Decentralized Federated Learning With Asynchronous Parameter Sharing for Large-Scale IoT Networks". *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34123-34139, 2024. <https://doi.org/10.1109/JIOT.2024.3354869>
- [29] D. Dey and N. Ghosh. "iTRPL: An intelligent and trusted RPL protocol based on Multi-Agent Reinforcement Learning". *Ad Hoc Networks*, vol. 163, 103586, 2024. <https://doi.org/10.1016/j.adhoc.2024.103586>
- [30] M. Shoab and L. Alsbatin. "GRU Enabled Intrusion Detection System for IoT Environment with Swarm Optimization and Gaussian Random Forest Classification". *Computers, Materials & Continua*, vol. 81, no. 1, pp. 625-642, 2024. <https://doi.org/10.32604/cmc.2024.053721>
- [31] K. D. Seifi, P. Asghari, H. H. S. Javadi, and M. H. Alaeiyan. "FL-GCN-LG-Trust: A Federated Graph-Based Trust Framework for Secure Cluster Optimization and Intrusion Detection in Underwater Wireless Sensor Networks". 2025.
- [32] Y. Afaq and S. V. Akram. "Integration of deep learning with edge computing on progression of societal innovation in smart city infrastructure: A sustainability perspective". *Sustainable Futures*, vol. 9, 100761, 2025. <https://doi.org/10.1016/j.sfr.2025.100761>
- [33] G. Jean. *Reinforcement Learning for Trust Optimization: Applying reinforcement learning algorithms to dynamically adjust trust scores based on user feedback and article outcomes*, 2024.
- [34] E. Meybodian, S. Mostafavi, T. Dargahi, and V. Ranjbar Bafghi. "Secure and scalable trust management in IoT: A hierarchical blockchain-based approach". *Computers and Electrical Engineering*, vol. 127, 110631, 2025. <https://doi.org/10.1016/j.compeleceng.2025.110631>
- [35] S. Sagar, A. Mahmood, Q. Z. Sheng, W. E. Zhang, Y. Zhang, and J. K. Pabani. "Understanding the trustworthiness management in the social Internet of Things: A survey". *Computer Networks*, vol. 251, 110611, 2024. <https://doi.org/10.1016/j.comnet.2024.110611>
- [36] M. A. Bouke and A. Abdullah. "An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability". *Expert Systems with Applications*, vol. 230, 120715, 2023. <https://doi.org/10.1016/j.eswa.2023.120715>
- [37] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera. "Towards trustworthy Internet of Things: A survey on Trust Management applications and schemes". *Computer Communications*, vol. 160, pp. 475-493, 2020. <https://doi.org/10.1016/j.comcom.2020.06.030>