IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

Machine Learning and Deep Learning Based Intrusion Detection for MitM Attack Mitigation in IoT Networks

Aditi Joshi[1], Dr. J. K. Shimp[2], Prof. S. D. Satre[3], Dr. H. B. Jadhav[4]
[1] Student, Computer Engineering,
[2] Guide, Computer Engineering
[3] Assistant Professor, Computer Engineering,
[4] HOD, Computer Engineering
Adul Technical Campus, Ahilyanagar, Maharashtra, India

Abstract : *The Internet of Things (IoT) connectssensors, smart appliances, mobile devices and industrial equipment so that they can exchange informationandrespond to events. This connectivity is useful, but it also creates security risks when devices have limitedcomputing resources and communicate through different protocols. A Man-in-the-Middle (MitM) attackisonesuch risk: an attacker places themselves between communicating devices and may observe oraltertheexchanged data. This paper presents a machine-learning- based workflow for identifying traffic associatedwithMitM attacks in IoT networks. The proposed workflow includes traffic preparation, preprocessing,featureselection, model training, classification and evaluation. Support Vector Machine (SVM), RandomForestandLong Short-Term Memory (LSTM) are considered because they offer different ways to classifynetworktraffic. The supplied synopsis reports results from the referenced 2025 study in which Random Forestachieved94% accuracy, LSTM 92%, and SVM 85.7% under that study's experimental conditions. Thesefiguresare reported as prior-study results, not as results independently produced by this paper. Thediscussionalso identifies practical issues such as false alarms, limited device resources, dataset representativenessandthe need for testing in live IoT environments*

Key Words: Internet of Things, Man-in-the-Middleattack, intrusion detection, machine learning, deep learning, Random Forest, SVM, LSTM

INTRODUCTION

The Internet of Things (IoT) has become part of homes, workplaces and industrial environments. Devices such as sensors, smart cameras, connected appliancesand control units collect and exchange data to support monitoring and automation. However, these devicesdo not all have the same processing capacity, memory, energy supply or communication protocol. This variety makes it difficult to apply one security approach uniformly across an entire IoT network.

A Man-in-the-Middle (MitM) attack occurs when an unauthorized party interferes with communication between two legitimate endpoints. Depending on the situation, the attacker may listen to the traffic, redirect it or change messages before they reach their destination. Techniques associated with this threat include ARP cache poisoning, DNS spoofing, session hijacking, evil-twin access points and SSL/TLS-related interception. Such activity can affect the confidentiality and integrity of data and may disrupt the availability of a service.

Conventional protections such as encryption, authentication and network filtering remain important, but they do not remove the need to monitor suspicious behaviour. Machine learning (ML) can learn patterns from labelled traffic, while

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

deep learning (DL) can model more complex or sequential patterns. In this paper, SVM, Random Forest and LSTM are considered as complementary approaches for classifying traffic as normal or potentially malicious.

Detection is treated as a decision-support layer; it should complement, rather than replace, secure communication and access-control measures.

II. PROBLEM STATEMENT

IoT networks contain heterogeneous and often resource-constrained devices. Continuous monitoring can be difficult when devices generate large volumes of traffic or cannot run computationally expensive security

software locally. Rule-based mechanisms may also miss traffic patterns that have not been explicitly described in their rules. The research problem is therefore to examine whether

an automated traffic-classification pipeline can distinguish normal IoT communication from MitM-related activity with useful detection performance, while also considering false positives, processing cost and the requirements of real-time operation

III. OBJECTIVES AND SCOPE

The main objectives of this study are:

- To outline an automated intrusion-detection workflow suitable for analysing IoT network traffic.
- To classify traffic into normal and attack-related categories using prepared labelled data.
- To compare SVM, Random Forest and LSTM using consistent training, testing and evaluation procedures.
- To assess model behaviour with accuracy, precision, recall, F1-score and a confusion matrix.
- To identify practical improvements related to false alarms, generalisation and resource-aware deployment.

The scope is limited to the detection workflow and comparative evaluation described in the supplied synopsis and cited study. The paper does not claim that a fully deployed, real-time IoT defence system has been built or tested. Mitigation actions such as blocking a device or changing network routes would require a separate policy and validation stage.

IV. EXISTING SYSTEMAND RELATED WORK

Earlier work has explored multiple ways of detecting MitM activity in IoT environments. The supplied synopsis identifies lightweight gateway-based detection, classifiers designed around MQTT traffic, graph-based learning, regression modelling and combinations of feature extraction with supervised classification. These approaches show that traffic analysis can help identify suspicious behaviour, but they also highlight challenges that need to be considered before deployment.

A rule-based or signature-oriented detector can be efficient when an attack pattern is already known, but it may require frequent updates and can struggle with previously unseen variations. A gateway-based detector can reduce the burden on small IoT devices by moving analysis to a more capable network node; however, the gateway can become a performance bottleneck and may not see all relevant traffic. A model trained on

a specific protocol or dataset may perform less reliably when device types, traffic patterns or attack methods change.

The 2025 study by Ali and Al-Sharafi compares ML/DL approaches for MitM detection and is the primary reference for the model comparison discussed here [1]. Related work includes a survey of MitM attacks [2], graph-based learning for IoT MitM mitigation [3], and ML-based detection for MQTT devices [4]. These references provide context for the problem, but their methods and evaluation settings should not be assumed to be directly interchangeable.

The key limitation motivating the proposed workflow is not simply a lack of classifiers. It is the need for a fair comparison using the same prepared data and evaluation process, with special attention to false negatives, false positives and the cost of using a model in a constrained IoT setting. Accuracy alone may give an incomplete picture when attack and normal samples are unevenly represented

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

V. PROPOSED SYSTEM

The proposed system is a staged traffic-analysis pipeline. It starts with a prepared dataset containing normal network traffic and MitM-related attack samples. The records are cleaned and transformed, and relevant features are selected or extracted. The prepared data is then used to train three candidate classifiers: SVM, Random Forest and LSTM. Each model is evaluated on held-out test data, and the resulting metrics are compared before selecting a candidate for further validation.

SVM provides a margin-based classification approach, Random Forest combines the decisions of multiple decision trees, and LSTM is designed to learn patterns across sequences. Their different characteristics make the comparison useful: a model with high accuracy may still miss attacks or raise too many false alarms. For this reason, model selection should consider precision, recall and F1-score alongside accuracy, and should eventually include runtime and resource measurements.

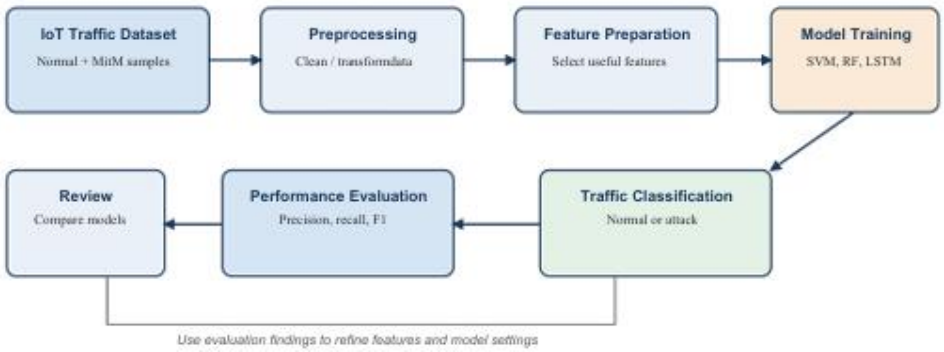
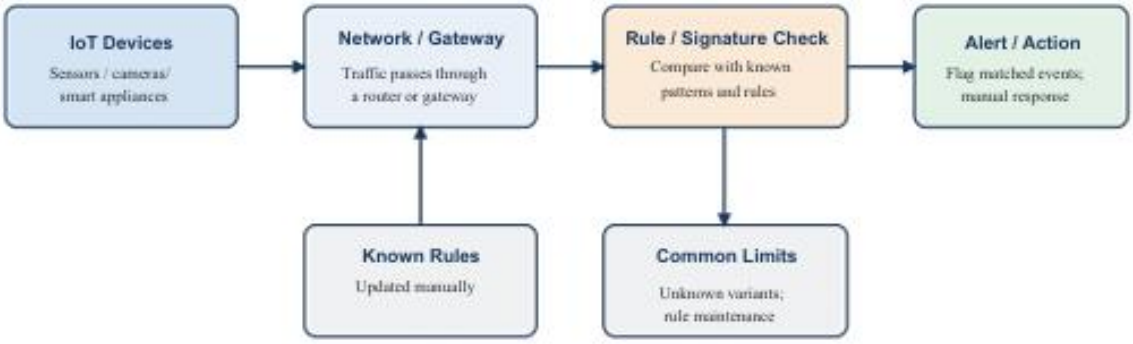


Fig. 1. Proposed ML/DL-based MitM traffic-detection workflow.

IV. EXISTING SYSTEM — DIAGRAM

In a conventional rule- or signature-based setup, IoT devices send traffic through a network or gateway. The detector checks the traffic against known patterns and rules. When a match is found, it can raise an alert for review or response. This approach can be efficient for recognised attacks, but it depends on the quality and freshness of the rules and may miss new or modified attack patterns.



Conceptual view of a conventional rule/signature-based detector

Note: The existing-system diagram complements the proposed ML/DL workflow already included in the paper. The proposed-system diagram shows dataset preparation, preprocessing, feature preparation, model training, classification and evaluation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

VI. METHODOLOGY

1) Dataset preparation: Use the prepared wireless-network dataset described in the supplied synopsis, containing normal traffic and MitM-related attack records. The synopsis indicates labels Normal = 0 and Attack1/Attack2 = 1. The original label definitions and dataset provenance should be checked before replication.

2) Preprocessing and feature preparation: Check for missing or invalid values, encode fields where necessary, and scale features if required by the selected algorithm. Feature selection should use training data only to avoid information leaking from the test set. Any class imbalance should be documented and addressed transparently.

3) Model development: Train SVM, Random Forest and LSTM using the same train/test split wherever appropriate. SVM classifies using a decision boundary; Random Forest combines multiple decision trees; LSTM processes sequential input and can learn temporal dependencies. Model settings and preprocessing steps should be recorded so that the experiment can be repeated.

4) Detection: Apply each trained model to unseen test traffic and assign a normal or attack-related label. In an operational network, an alert should be reviewed according to a defined response policy rather than automatically treated as proof of an attack.

5) Evaluation: Use a confusion matrix and calculate accuracy, precision, recall and F1-score. Accuracy describes the overall proportion of correct classifications. Precision measures how many predicted attacks were actual attacks; recall measures how many actual attacks were detected. F1-score balances precision and recall. These metrics should be read together because false alarms and missed attacks have different operational consequences

VII. COMPARATIVE RESULTS REPORTED IN THE SOURCE STUDY

The supplied synopsis reports the following results from Ali and Al-Sharafi [1]. They are included here to summarise the cited study and are not presented as newly generated experimental results for this draft.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	94.0%	0.698	0.864	0.772
LSTM	92.0%	0.495	0.823	0.573
SVM	85.7%	Not statedin synopsis	Not stated synopsisin	Not stated in synopsis

Table I. Values reported in the supplied synopsis for the cited study. The synopsis notes that SVM had 85.7% nominal accuracy but did not identify actual attacks in that test setting. This is an important reminder that a single accuracy value does not establish a detector's practical usefulness. The reported numbers depend on the source dataset, label distribution, model configuration and test procedure.

VIII. DISCUSSION AND RESEARCH GAP

The reported comparison suggests that Random Forest is a promising candidate for the specific experiment summarised in the source synopsis, while LSTM offers a way to learn sequential patterns in traffic. However, the precision reported for LSTM is lower than its recall, indicating that its attack predictions may include a notable number of false positives under that test setup. The source

synopsis also notes that the SVM result requires careful interpretation because its nominal accuracy did not translate into identifying attacks in the reported test setting.

A useful next research step is to test the models on more varied traffic, including different device types, traffic loads and attack variants. Evaluation should report the class distribution, data split, preprocessing details and per-class confusion matrix. A hybrid Random Forest–LSTM approach may be investigated, but its benefit must be demonstrated experimentally rather than assumed. For practical deployment, model size, inference time, memory use and gateway throughput should be measured alongside classification metrics. Another improvement is to separate detection from

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

mitigation. A detector can flag suspicious traffic, but actions such as isolating a device or blocking a connection can interrupt legitimate services if the alert is wrong. Any automated response should therefore have clear thresholds, logging, a recovery process and testing in a controlled environment before being used in a live network.

IX. LIMITATIONS

This paper draft is based primarily on the supplied synopsis and the references listed in it. It does not contain a new implementation, a new dataset analysis or independently reproduced results. The exact dataset composition, feature list, train/test split and full model configurations are not detailed in the synopsis, so they should be verified from the original article before attempting replication. The performance values in Table I must not be generalised to all IoT networks.

X. CONCLUSION

MitM attacks can threaten the confidentiality and integrity of communication in IoT networks, where device diversity and limited resources make security monitoring challenging. This paper outlines an intrusion-detection workflow that preprocesses network traffic, prepares features, trains SVM, Random Forest and LSTM classifiers, and evaluates their predictions using multiple metrics. The source study reports its strongest accuracy for Random Forest among the three models discussed, but precision, recall, false alarms and missed attacks must also inform model selection. Further work should test the approach on diverse and clearly documented datasets, measure its resource requirements, and validate any mitigation response in a controlled environment. These steps are necessary before claiming reliable real-time protection.

REFERENCES

1. M. A. Ali and S. A. H. Al-Sharafi, “Intrusion detection in IoT networks using machine learning and deep learning approaches forMitM attack mitigation,” Discover Internet of Things, vol. 5, article 48, 2025.
2. M. Conti, N. Dragoni, and V. Lesyk, “A survey of man in the middle attacks,” IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027–2051, 2016.
3. E. Nwafor, C. Schmidt, and H. Olufowobi, “IoT-MGSec: mitigating man-in-the-middle attacks in IoT networks using graph-basedlearning,” in Proceedings of the International Conference on Machine Learning and Applications (ICMLA), 2023, pp. 1261–1266.
4. A. B. M. Sultan, S. Mehmood, and H. Zahid, “Man in the middle attack detection for MQTT based IoT devices using differentmachine learning algorithms,” in Proceedings of ICAI, 2022, pp. 118–121.