IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15836

A Comparative Study of Lightweight Machine Learning Models for Detecting Suspicious Network Traffic

Dadavali S. P.

Department of Computer Science, Government First Grade College, Kengeri, Bengaluru, India

Abstract: The increasing use of computer networks and Internet-based services has increased the risk of suspicious and potentially malicious network activities. Traditional monitoring approaches may require continuous manual analysis and may become difficult to apply to large volumes of traffic. This paper presents a comparative evaluation of five lightweight machine learning models for binary classification of network traffic as Normal or Attack. The evaluated models are Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM). Experiments are conducted using the UNSW-NB15 dataset. A reproducible stratified sample of 20,000 training records and 10,000 testing records is used to keep the experiment computationally manageable. The models are evaluated using accuracy, precision, recall, F1-score, and confusion matrices. In the experiment, Random Forest achieves the highest accuracy (86.30%) and F1-score (88.81%), while SVM obtains the highest recall (99.47%). The findings show that conventional machine learning models can provide useful suspicious-traffic classification without requiring complex deep-learning architectures. Random Forest provides the best overall balance among the evaluated metrics under the experimental conditions of this study.

Keywords: Machine learning, network traffic, suspicious traffic detection, network security, classification, UNSW-NB15, intrusion detection.

I. INTRODUCTION

Modern computer networks generate a large and continuously changing volume of communication. At the same time, legitimate traffic is mixed with activities that may indicate scanning, unauthorized access, denial-of-service behavior, or other forms of misuse. Detecting these patterns promptly is therefore an important part of maintaining a secure network environment.

Many conventional security systems depend on predefined rules or signatures. Such mechanisms remain useful for recognized threats, but their effectiveness can be limited when traffic characteristics change or unfamiliar patterns appear. The increasing amount of network data also makes continuous manual inspection impractical. These factors provide motivation for using machine learning to assist automated traffic classification.

A trained classifier can learn relationships between network features and previously observed traffic classes and then apply those relationships to new observations. However, classifiers differ in computational cost and predictive behavior. Deep-learning solutions may require greater computational resources and more extensive training. For preliminary security analysis and environments with limited resources, conventional machine-learning methods can therefore provide a practical alternative.

This paper compares five conventional classifiers—Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM)—for identifying suspicious network traffic. The UNSW-NB15 dataset is used, and the task is simplified to two classes: Normal and Attack [1], [2].

The study follows a common pipeline consisting of data preparation, feature transformation, classifier training, testing, and metric-based evaluation. Accuracy, precision, recall, F1-score, and confusion matrices are reported. The main aim is to determine which of the selected lightweight models gives the most balanced result while keeping the experiment straightforward to reproduce with standard Python tools.

II. OBJECTIVES

1. To examine the suitability of conventional machine learning algorithms for suspicious network traffic detection.
2. To preprocess and prepare the UNSW-NB15 dataset for binary classification.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15836

3. To compare Logistic Regression, Decision Tree, KNN, Random Forest, and SVM using standard classification metrics.
4. To identify the comparatively better-performing lightweight model under the selected experimental conditions.

III. RELATED WORK

Machine-learning methods have been applied extensively to network-security classification and intrusion detection. Earlier experimental work commonly used benchmark datasets such as KDD'99. Later analysis identified several limitations in that dataset, contributing to interest in alternative and improved benchmarks, including NSL-KDD [3].

UNSW-NB15 was introduced as a more recent benchmark for network-traffic and intrusion-detection experiments. It combines normal activity with several contemporary attack behaviors and provides nine attack categories together with 49 features [1]. A later comparative analysis examined UNSW-NB15 against KDD'99 and highlighted differences in their statistical characteristics [2].

Conventional classifiers are useful baseline methods because they are available in standard machine-learning libraries and can be executed without specialized deep-learning hardware. Decision Trees use sequential feature-based decisions, whereas Random Forest aggregates predictions from multiple trees [4]. KNN uses distances between observations to determine a class [5]. SVM seeks a separating boundary with a maximum-margin formulation [6], while Logistic Regression supplies a comparatively simple linear baseline.

The Python Scikit-learn library provides implementations for preprocessing, classification, prediction, and evaluation [7]. Accordingly, this study concentrates on a direct and consistent comparison of five conventional classifiers instead of proposing a complex intrusion-detection architecture.

IV. DATASET AND PREPROCESSING

A. Dataset Description

The experiments use the UNSW-NB15 dataset. According to the dataset documentation, it combines normal network activities with generated contemporary attack behaviors and includes nine attack categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The dataset provides 49 features, including the class information, and the official train/test partition contains 175,341 training records and 82,332 testing records [1]. The supplied UNSW-NB15 training-set.csv and testing-set.csv files form the experimental input. Because the objective is a lightweight and reproducible comparison, 20,000 training records and 10,000 testing records are selected using stratified sampling with random state 42. This approach retains an approximately representative Normal/Attack distribution while reducing the computational load.

B. Binary Classification

The original label is used to define two classes. Label 0 represents Normal traffic and label 1 represents Attack traffic. The attack_cat field is not used as an input feature because it directly describes the attack category and would introduce target information into the model.

C. Data Preprocessing

Before classification, the data are separated into numerical and categorical attributes. Numerical missing values are replaced using the median, while categorical missing values use the most frequent category. Categorical attributes are transformed through one-hot encoding, with unseen test categories ignored. Numerical variables are standardized for the common preprocessing pipeline.

Numerical standardization is performed using $z = (x - \mu) / \sigma$, where x is the original value, μ is the training-set mean, and σ is the training-set standard deviation. The transformation is learned from the training data and then applied to the testing data to reduce information leakage.

V. MACHINE LEARNING MODELS

A. Logistic Regression

Logistic Regression is used as a simple baseline for binary classification. Its probability estimate can be written as $P(y=1|x) = 1 / (1 + e^{(-z)})$, where $z = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$. The model is computationally simple and provides a useful reference against nonlinear classifiers.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15836

B. Decision Tree
A Decision Tree classifies observations through a sequence of feature-based decisions. Internal nodes represent splitting conditions and terminal nodes represent predicted classes. The model can represent nonlinear relationships and does not require feature scaling for its decision rules.

C. K-Nearest Neighbors
KNN classifies an observation according to the classes of its nearest training observations. Euclidean distance can be expressed as $d(x,x_i) = \sqrt{\Sigma_j(x_j - x_{ij})^2}$. In this study, K=5 is used as the baseline configuration.

D. Random Forest
Random Forest combines predictions from an ensemble of decision trees rather than relying on a single tree. For classification, the final output can be represented by the majority class among the individual tree predictions. The experiment uses 100 trees and random state 42 [4].

E. Support Vector Machine
SVM seeks a decision boundary that separates the classes with a large margin. A linear decision function can be written as $f(x) = w^Tx + b$. An RBF kernel is used in the experiment with C=1.0 and gamma='scale' [6].

VI. EXPERIMENTAL METHODOLOGY

The complete workflow is: UNSW-NB15 dataset → stratified sampling → data inspection → preprocessing → categorical encoding → numerical scaling → model training → model testing → metric calculation → comparative analysis.

All experiments are implemented in Python using Pandas, NumPy, Scikit-learn, and Matplotlib. A pipeline-based approach is used so that preprocessing transformations are fitted on the training data and applied consistently to the test data.

The selected models are trained independently on the same prepared training sample and evaluated on the same test sample. No extensive hyperparameter search is performed because the purpose of the study is a simple comparison of conventional lightweight models rather than an optimization study.

A. Evaluation Metrics
Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 = 2 × (Precision × Recall) / (Precision + Recall)

Here, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Because attack detection is the main objective, recall and F1-score are considered alongside accuracy.

VII. RESULTS AND COMPARATIVE ANALYSIS

A. Overall Performance

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	80.77	75.11	97.33	84.79
Decision Tree	82.56	81.30	88.74	84.86
KNN	83.78	78.55	97.04	86.82
Random Forest	86.30	80.70	98.73	88.81
SVM	81.45	75.00	99.47	85.52

Table I shows that Random Forest achieved the highest accuracy (86.30%) and F1-score (88.81%) among the five evaluated models. SVM achieved the highest recall (99.47%), indicating that it missed very few attack instances, but its

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15836

precision was lower (75.00%). KNN achieved an accuracy of 83.78% and an F1-score of 86.82%. Decision Tree achieved 82.56% accuracy and 84.86% F1-score, while Logistic Regression achieved 80.77% accuracy and 84.79% F1-score.

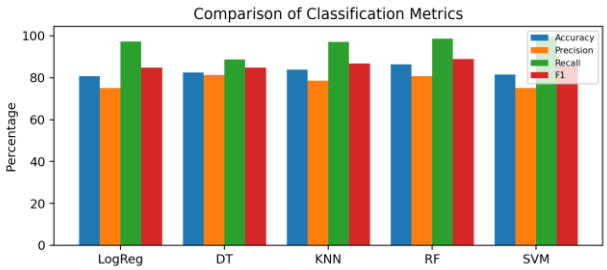

Fig. 1. Comparison of accuracy, precision, recall, and F1-score.

B. Confusion Matrix Analysis

Model	TN	FP	FN	TP
Logistic Regression	2718	1776	147	5359
Decision Tree	3370	1124	620	4886
KNN	3035	1459	163	5343
Random Forest	3194	1300	70	5436
SVM	2668	1826	29	5477

Random Forest produced 5,436 true-positive predictions and 70 false-negative predictions. This indicates strong attack-detection capability in the selected test sample. SVM produced only 29 false negatives, the lowest among the five models, but it also generated 1,826 false positives. This explains why its recall was highest while its precision was comparatively lower.

Decision Tree produced 3,370 true-negative predictions and 1,124 false positives. KNN produced 5,343 true positives and 163 false negatives. Logistic Regression produced 5,359 true positives and 147 false negatives. The confusion matrix results support the metric-based comparison and demonstrate that no single model is best on every metric.

VIII. DISCUSSION

The experiment shows that the classifiers make different trade-offs between detecting attacks and generating false alarms. Random Forest produced the strongest overall balance in the selected test sample, recording the highest accuracy and F1-score while retaining very high recall. The use of multiple decision trees may contribute to its comparatively stable classification behavior.

SVM may be useful when the primary concern is minimizing missed attacks because it produced the highest recall. At the same time, its larger number of false positives reduced precision. In operational settings, this trade-off matters because frequent false alarms can increase the effort required to investigate security alerts.

KNN provided strong recall and an intermediate F1-score, while Decision Tree provided a simpler rule-based model with moderate performance. Logistic Regression, although the simplest baseline, still detected a high proportion of attack instances. These results show that conventional models can provide useful baseline performance without deep-learning architectures.

The findings must be interpreted within the scope of the experiment. The study uses a 20,000-record training sample and 10,000-record testing sample from the official UNSW-NB15 train/test files rather than the complete dataset. The results therefore represent the selected experimental setup and should not be interpreted as universal performance claims for all network environments.

IX. CONCLUSION AND FUTURE WORK

A. Conclusion

This work evaluated five lightweight machine-learning classifiers—Logistic Regression, Decision Tree, KNN, Random Forest, and SVM—for binary suspicious-traffic classification using UNSW-NB15.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15836

Under the selected experimental setup, Random Forest recorded 86.30% accuracy and an 88.81% F1-score, both the highest among the compared models. SVM recorded the highest recall at 99.47%. These results indicate that conventional classifiers can provide useful traffic-classification performance without a complex deep-learning architecture. Within this experiment, Random Forest offered the most balanced combination of the reported measures.

B. Future Work

Future work can extend the experiments to the complete UNSW-NB15 dataset, evaluate additional classifiers and ensemble methods, and perform systematic hyperparameter optimization. Multiclass classification of individual attack categories can also be investigated. Feature-selection techniques may be used to reduce the number of input variables and examine whether comparable performance can be achieved with fewer features. Finally, the models can be evaluated on additional network-security datasets to study generalization across different traffic environments.

REFERENCES

[1] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. 2015 Military Communications and Information Systems Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
[2] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Information Security Journal: A Global Perspective, vol. 25, nos. 1–3, pp. 18–31, 2016, doi: 10.1080/19393555.2015.1125974.
[3] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. 2009 IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
[4] L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[5] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," IEEE Trans. Information Theory, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.
[6] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273–297, 1995, doi: 10.1007/BF00994018.
[7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[8] UNSW Research, "The UNSW-NB15 Dataset," University of New South Wales, Canberra, Australia. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset. Accessed: Aug. 18, 2026.

Data Availability Statement

The dataset used in this study is the publicly available UNSW-NB15 dataset. The experimental data files and processing scripts used to obtain the reported results should be made available by the authors upon reasonable request, subject to the dataset's original access conditions.

Conflict of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

AI Use Disclosure

Generative AI tools were used during preparation of portions of the manuscript. The author is responsible for verifying the technical content, experimental results, references, and conclusions and for ensuring that the final manuscript complies with the journal's AI policy.

