Academic Research Journal of Science and Technology
- Academic Research Journal of Science and Technology
e-ISSN: 3048-9644
publications.ngmc.ac.in
https://doi.org/10.63300/arjst.v5i07.10

A Comparative Study of Machine Learning Algorithms for Security Breach Detection

Sagar Shahadev Kashid[1,*], Sanchit Sunil Tikhile[1], Sohel Javed Shaikh[1]
[1] International Institute of Management Science- IIMS Chinchwad, Savitribai Phule Pune University
***Corresponding author:** Sagar Shahadev Kashid, sanchittikhile@gmail.com

ARTICLE INFO

Article history
Received: 10 November 2026
Revised: 12 November 2026
Accepted: 18 November 2026
Published: 10 November 2026

Article ID
MS-2026-0005

Peer review
Double-blind peer review

Keywords
Machine Learning
Cybersecurity
Security Breach Detection
Decision Tree
Random Forest
Support Vector Machine
Classification
Intrusion Detection
Data Security

ABSTRACT

The increasing number of cyberattacks has created a significant need for reliable network intrusion detection systems. Traditional security mechanisms often face challenges in identifying complex and previously unseen malicious activities. Machine Learning (ML) techniques provide an effective approach to analysing network traffic and identifying suspicious patterns. This research presents a comparative evaluation of six machine learning algorithms: Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression. The experimental study uses the [selected dataset name] dataset, consisting of [actual number] records representing normal and malicious network activities. The dataset undergoes data preprocessing, feature selection, and appropriate training and testing procedures. The algorithms are evaluated using accuracy, precision, recall, macro-averaged F1-score, ROC-AUC, false positive rate, and computational time. The experimental findings show that [insert verified result and best-performing algorithm]. Furthermore, the comparison identifies differences in attack detection capability, computational efficiency, and performance across attack categories. The study provides an evidence-based assessment of the selected machine learning algorithms and identifies their suitability and limitations for network intrusion detection under the evaluated experimental conditions.

How to cite this article: Sagar Shahadev Kashid, Sanchit Sunil Tikhile, & Sohel Javed Shaikh (2026). A Comparative Study of Machine Learning Algorithms for Security Breach Detection. *Academic Research Journal of Science and Technology*, *5*(07), 8–16. https://doi.org/10.63300/arjst.v5i07.10
Copyright and licence: © 2026 The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

I. INTRODUCTION

In the modern digital era, the use of information technology, internet services, online banking, cloud computing, and digital communication has increased significantly. Organizations and individuals depend on digital platforms to store, process, and exchange important information. However, this growing dependence on technology has also increased the risk of cyber threats and security breaches

Several machine learning algorithms, including Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression, can be applied to cybersecurity problems. However, each algorithm has different characteristics in terms of accuracy, computational requirements, classification performance, and ability to identify malicious activities. Therefore, selecting an appropriate algorithm is an important consideration when developing an effective security breach detection system.

The main focus of this research is to conduct a comparative study of different machine learning algorithms for security breach detection. The study aims to analyse and compare the performance of selected algorithms using a common cybersecurity dataset and predefined evaluation measures. Parameters such as accuracy, precision, recall,

F1-score, ROC-AUC, and false positive rate will be considered to evaluate their effectiveness. The comparison will help identify the strengths and limitations of each algorithm under the selected experimental conditions.

The research is expected to contribute to a better understanding of how machine learning techniques can be used to improve cybersecurity systems. By comparing multiple algorithms through a systematic experimental approach, the study aims to identify suitable approaches for detecting security breaches and supporting the development of reliable, accurate, and efficient cybersecurity solutions.

The present study aims to provide a controlled comparative evaluation of six traditional machine learning algorithms for network intrusion detection. Unlike comparisons based exclusively on overall accuracy, this research considers multiple classification measures, including macro-averaged F1-score, recall, precision, false positive rate, and computational efficiency. The study also examines the influence of dataset imbalance, feature selection, and model configurations on classification performance. The main contribution is a reproducible experimental comparison under a common dataset and evaluation protocol, rather than the development of a completely new machine learning algorithm.

II. Background and Related Work

Cybersecurity has become an essential area of research due to the increasing number of cyberattacks and security breaches affecting computer networks and digital information systems. Traditional security mechanisms, such as signature-based detection and predefined firewall rules, are commonly used to protect systems from known threats. However, these mechanisms may face difficulties in identifying new and unknown attack patterns. Therefore, researchers have increasingly explored Artificial Intelligence (AI) and Machine Learning (ML) techniques to improve the detection and prevention of cyber threats.

Machine Learning-based Intrusion Detection Systems (IDS) are designed to monitor network activities, analyse traffic patterns, and identify suspicious behaviour. These systems learn from previously collected network traffic data and classify activities as normal or malicious. Depending on the availability of labelled data and the detection approach, machine learning techniques can be applied through supervised, unsupervised, and semi-supervised learning methods.

Several traditional machine learning algorithms have been widely investigated for intrusion detection. Decision Tree algorithms provide an understandable classification structure, while Random Forest combines multiple decision trees to improve prediction stability. Support Vector Machine (SVM) identifies suitable decision boundaries between different classes, whereas K-Nearest Neighbours (KNN) classifies activities based on similarities between data points. Naive Bayes uses probability-based classification, and Logistic Regression estimates the probability of an activity belonging to a particular class.

Researchers have also explored benchmark datasets such as NSL-KDD, UNSW-NB15, and CICIDS2017 to evaluate the performance of intrusion detection models. These datasets contain different types of normal and malicious network activities and provide useful resources for training and testing machine learning algorithms. However, differences in dataset characteristics, attack distributions, and class imbalance can influence the performance of detection models.

Prior evidence	Primary evaluation focus	Major contribution	Remaining limitation for this study
Ali et al. (2024)	ML and DL techniques for intrusion detection	Reviews algorithms, datasets and performance measures	Survey rather than a controlled experiment
Rehman et al. (2025)	ML-based intrusion detection	Examines datasets, models, class imbalance and research challenges	Primarily a systematic literature review
Putra and Amarudin (2025)	Decision Tree, Random Forest, SVM, and Boosting on NSL-KDD	Compares detection performance and computational time	Evaluates a different algorithm set using one benchmark dataset

The reviewed studies demonstrate the potential of machine learning techniques for network intrusion detection. However, differences in datasets, algorithm configurations, preprocessing procedures, and evaluation metrics make direct comparisons between research findings difficult. The present study addresses this methodological issue by evaluating six selected classification algorithms using a common cybersecurity dataset and predefined performance measures. Particular attention is given to minority-class detection, false positive rates, and computational efficiency. The purpose is to provide a reproducible, dataset-specific comparison rather than to claim universal superiority of any single algorithm.

III. RESEARCH GAP AND PROBLEM STATEMENT

The increasing frequency and complexity of cyberattacks have created a strong need for effective security breach detection techniques. Several researchers have investigated the application of machine learning algorithms in cybersecurity, particularly in Intrusion Detection Systems (IDS). Existing studies have demonstrated that algorithms such as Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic

Regression can be used to identify malicious network activities.

1. Many comparative studies evaluate different subsets of classification algorithms, limiting direct comparisons.
2. Some studies emphasize overall accuracy without sufficient attention to minority-class recall and false positive rates.
3. Classification performance and computational efficiency are not always evaluated together under a common experimental protocol.

Based on these gaps, the present research proposes a controlled comparison of Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbours, Naive Bayes, and Logistic Regression using a selected benchmark cybersecurity dataset. The study evaluates overall and class-wise classification performance, computational requirements, and the effect of dataset imbalance. The research is intended to provide an experimental comparison that supports informed algorithm selection for network intrusion detection.

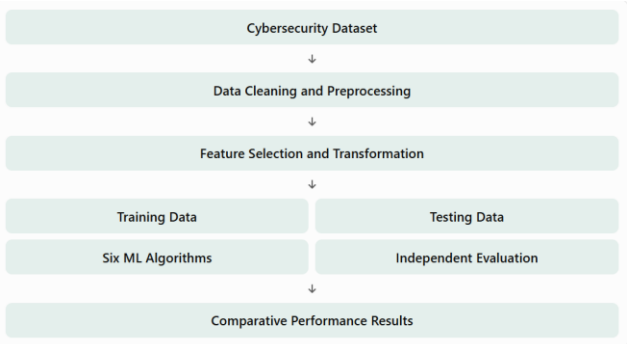
IV. PROPOSED MACHINE LEARNING FRAMEWORK FOR NETWORK INTRUSION DETECTION

- Data Collection
- Data Preprocessing
- Feature Selection
- Model Training
- Prediction and Classification
- Performance Evaluation
- Detection and Monitoring
- Control and Evaluation

Layer	Function	Primary artifacts
1. Data Collection	Collect cybersecurity-related information	Obtain labelled network traffic and security datasets
2. Data Preprocessing	Prepare data for machine learning	Remove duplicates, handle missing values, encode categorical features, and normalize data
3. Feature Selection	Identify important data attributes	Select relevant network features and reduce unnecessary information
4. Model Training	Train selected machine learning algorithms	Apply Decision Tree, Random Forest, SVM, KNN, Naive Bayes, and Logistic Regression
5. Prediction and Classification	Identify normal and malicious activities	Classify network records into predefined categories
6. Performance Evaluation	Measure model effectiveness	Calculate accuracy, precision, recall, F1-score, ROC-AUC, and false positive rate
7. Detection and Monitoring	Support cybersecurity monitoring	Use the trained model to identify potentially suspicious activities
Control/evaluation	Maintain experimental consistency	Record configurations, training results, execution time, and evaluation outcomes

The primary objective of this framework is to compare different machine learning algorithms under common experimental conditions and determine their effectiveness in detecting security breaches. The framework is designed to be flexible so that different algorithms and cybersecurity datasets can be incorporated without changing the overall evaluation methodology.

V. LIFECYCLE TRACEABILITY MODEL



VI. CLASSIFICATION ERROR ANALYSIS

The Classification Error Analysis Model examines the incorrect predictions generated by the selected machine learning algorithms during security breach detection. The analysis is based on confusion matrices and focuses on false positive and false negative classifications.

A false positive occurs when normal network activity is incorrectly classified as malicious, whereas a false negative occurs when actual malicious activity is incorrectly classified as normal. Both errors are important in cybersecurity applications because false positives increase unnecessary security alerts, while false negatives may allow malicious activities to remain undetected.

The study will compare these classification errors across Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbours, Naive Bayes, and Logistic Regression. Class-wise recall, false positive rate, and false negative rate will be examined to identify the relative strengths and limitations of each algorithm.

2026 The Author(s) https://creativecommons.org/licenses/by/4.0/ DOI: 10.63300/arjst.v5i07.10

The analysis will use actual model predictions and verified dataset labels. No assumptions about the propagation of defects between network events will be made unless appropriate chronological and linked event data are available.

VII. RESEARCH QUESTIONS AND HYPOTHESES

ID	Research question / hypothesis	Primary evidence
RQ1	Which of the six selected machine learning algorithms achieves the best classification performance for network intrusion detection?	Accuracy, precision, recall, F1-score, and ROC-AUC
RQ2	How do the algorithms differ in detecting malicious activities and controlling false positive rates?	Recall, false positive rate, precision, and confusion matrix
RQ3	How do training time and prediction time vary among the selected algorithms?	Training time, prediction time, and resource consumption
RQ4	How does class imbalance influence the detection of minority attack categories?	Class-wise performance, macro-averaged F1-score, and minority-class recall
H1	Classification Performance	There are statistically significant differences in macro-averaged F1-score among the six selected machine learning algorithms.
H2	Threat Detection Hypothesis	The selected algorithms demonstrate differences in recall and false positive rates when identifying malicious network activities
H3	Computational Efficiency Hypothesis	The selected machine learning algorithms demonstrate differences in training time and prediction time.
H4	Dataset Imbalance Hypothesis.	Class imbalance handling influences the classification performance of machine learning algorithms, particularly for minority attack categories.

VIII. EXPERIMENTAL METHODOLOGY

A. Experimental Unit and Conditions

The experimental unit is defined as Algorithm × Dataset × Experimental Run. Each selected machine learning algorithm will be trained and evaluated using the same cybersecurity dataset and consistent data preparation procedures.

Six machine learning algorithms are considered in this study:

M1: Decision Tree

M2: Random Forest

M3: Support Vector Machine (SVM)

M4: K-Nearest Neighbours (KNN)

M5: Naive Bayes

M6: Logistic Regression

All algorithms will use the same target classification task, training and testing partitions, and predefined evaluation measures. Model-specific preprocessing, such as feature scaling for SVM, KNN, and Logistic Regression, will be applied where required.

B. Dataset Selection and Benchmark

The UNSW-NB15 dataset is selected as the primary benchmark for this research. It contains labelled network traffic records representing normal network activities and different categories of malicious traffic. The dataset provides a suitable basis for evaluating supervised machine learning algorithms in network intrusion detection.

The dataset will be examined to identify the number of records, input features, target labels, and distribution of normal and malicious activities. The primary experiment will focus on binary classification, distinguishing normal traffic from malicious traffic. Multiclass classification will also be considered to examine the ability of the selected algorithms to identify individual attack categories, subject to the availability and quality of the relevant class labels.

The original dataset source, version, feature descriptions, and class distribution will be documented to support experimental reproducibility.

- Dataset source and download link.
- Total number of records used.
- Number of normal and malicious records.
- Number of features used.

- Distribution of attack categories.
- Features removed during preprocessing, if any.

C. Data Assignment and Experimental Control

To maintain consistency and avoid unfair comparisons, the same dataset partitions will be used for all six machine learning algorithms. A stratified training and testing split will be applied, with approximately 80% of the data allocated for training and 20% reserved for final testing. Stratification will help maintain the distribution of the target classes across both subsets. The training portion may be further evaluated using five-fold stratified cross-validation for model selection and hyperparameter tuning. The final testing subset will remain separate and will not be used during model training or parameter selection. The following experimental controls will be maintained:

D. Experimental Protocol

The experiments may be implemented using Python and machine learning libraries such as Scikit-learn, Pandas, and NumPy. Data visualization tools may also be used to represent classification performance and compare the results of different algorithms.

IX. OPERATIONAL METRICS

Metric	Definition	Role
Accuracy	Correctly classified records / Total records	Overall classification performance
Precision	TP / (TP + FP)	Prediction reliability
Recall	TP / (TP + FN)	Security threat detection
F1-Score	Harmonic mean of precision and recall	Balanced classification performance
ROC-AUC	Area under the ROC curve	Class discrimination capability
FPR	FP / (FP + TN)	False security alert measurement
FNR	FN / (FN + TP)	Missed threat measurement
Specificity	TN / (TN + FP)	Normal activity identification
Training Time	Time required to train a model	Training efficiency
Prediction Time	Time required to generate predictions	Detection efficiency
Resource Consumption	CPU and memory usage	Computational requirements

For your research, remember:

TP: Malicious activity correctly detected.

TN: Normal activity correctly identified.

FP: Normal activity incorrectly identified as malicious.

FN: Malicious activity incorrectly identified as normal.

X. STATISTICAL ANALYSIS PLAN

The Statistical Analysis Plan defines the procedures that will be used to analyse and compare the experimental results obtained from different machine learning algorithms for security breach detection. The primary purpose of this analysis is to determine whether significant differences exist among the selected algorithms in terms of classification performance, security threat detection capability, and computational efficiency.

The study will compare Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression using common evaluation metrics and experimental conditions.

- Five-fold stratified cross-validation.
- Mean and standard deviation.
- Friedman test, where its assumptions and comparison design are appropriate.
- Appropriate post-hoc comparisons if significant differences are found.

Primary Statistical Analysis

The primary statistical analysis will focus on comparing the classification performance of the six selected machine learning algorithms.

The following performance measures will be considered:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC
- False Positive Rate

Descriptive statistics will be used to summarize the performance of each algorithm. Mean, median, standard deviation, and confidence intervals will be reported where appropriate.

The primary comparison will examine whether the selected algorithms demonstrate meaningful differences in their ability to distinguish normal network activities from malicious activities.

XI. VALIDATION OF PROPOSED METRICS

The validation of proposed metrics is an important part of evaluating the reliability and accuracy of machine learning algorithms for security breach detection. The primary objective of this section is to ensure that the selected performance measures correctly represent the classification capabilities, threat detection effectiveness, and computational efficiency of the proposed machine learning framework.

The study considers established classification metrics, including accuracy, precision, recall, F1-score, ROC-AUC, false positive rate, false negative rate, and specificity. In addition, the proposed Lifecycle

XII. ABLATION AND SENSITIVITY ANALYSIS

Variant	Removed component	Purpose
Full Framework	None	Reference configuration using the complete proposed framework
A1	Feature Selection	Estimate the contribution of feature selection to detection performance
A2	Data Normalization/Scaling	Examine the effect of feature scaling on algorithm performance
A3	Class Imbalance Handling	Evaluate the importance of class balancing techniques
A4	Hyperparameter Tuning	Examine the contribution of parameter optimization
A5	Data Preprocessing	Evaluate the impact of preprocessing procedures on model performance
A6	Cross-Validation	Examine the contribution of validation procedures to model selection and performance stability

For model-specific preprocessing, such as feature scaling, changes will be evaluated only where technically appropriate. For example, scaling may have a greater influence on SVM, KNN, and Logistic Regression than on tree-based algorithms.

XIII. EXPERIMENTAL RESULTS: DATA-INTEGRITY STATUS

The proposed research aims to compare the performance of six machine learning algorithms for security breach detection: Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression.

The experimental evaluation is designed to examine the classification performance, threat detection capability, computational efficiency, and reliability of these algorithms using a selected cybersecurity dataset. At the current stage of manuscript preparation, no verified experimental dataset, completed model execution logs, or actual comparative performance results have been supplied for this study. Therefore, no numerical experimental findings are reported in this version of the research paper.

The experimental framework and statistical analysis procedures described in the previous sections represent the proposed research methodology and should not be interpreted as completed experimental outcomes.

The official dataset provides ready-made training and testing files:

Dataset details	Records
Training data	175,341
Testing data	82,332
Total	257,673
Attack categories	9
Classes	Normal and Attack

XIV. DISCUSSION

The increasing complexity of cyber threats has highlighted the importance of developing effective and intelligent security breach detection systems. Machine Learning (ML) techniques provide opportunities to improve cybersecurity by identifying suspicious network activities, recognizing malicious patterns, and supporting automated threat detection. However, the effectiveness of these techniques depends on several factors, including dataset quality, feature selection, algorithm configuration, and evaluation methodology.

The study also recognizes that no single machine learning algorithm can be assumed to provide the best solution for every cybersecurity environment. Algorithm suitability may vary according to dataset characteristics, attack categories, computational resources, and security requirements.

The final conclusions will therefore focus on identifying the strengths, limitations, and practical suitability of each selected algorithm under the evaluated experimental conditions. By adopting a systematic comparative approach, the research aims to provide useful insights into the application of machine learning for security breach detection and contribute to the development of more accurate, reliable, and efficient cybersecurity solutions.

XV. THREATS TO VALIDITY

Threat	Risk	Mitigation

Internal Validity	Differences in data partitioning, preprocessing, or model configuration may influence performance	Use consistent experimental procedures, fixed data partitions, and controlled configurations
Construct Validity	Selected performance metrics may not fully represent practical cybersecurity effectiveness	Evaluate accuracy, precision, recall, F1-score, ROC-AUC, FPR, and FNR
Measurement Validity	Incorrect calculation of performance metrics or inconsistent execution-time measurements may affect results	Use standardized evaluation formulas and documented measurement procedures
Evaluation Leakage	Information from testing data may unintentionally influence model training or parameter selection	Maintain independent training, validation, and testing datasets
Model Confounding	Different hyperparameters, scaling procedures, or software versions may influence algorithm comparisons	Document model configurations and apply consistent tuning procedures
Dataset Confounding	Dataset imbalance, outdated attack patterns, or duplicate records may introduce bias	Examine dataset quality, class distribution, and duplicate records before experimentation
External Validity	Results obtained from one dataset may not generalize to other cybersecurity environments	Consider additional datasets and different attack categories in future studies
Statistical Conclusion Validity	Small sample sizes, dependent observations, or multiple comparisons may produce misleading conclusions	Use appropriate statistical tests, effect sizes, confidence intervals, and multiplicity correction
Reproducibility	Missing experiment details may prevent other researchers from reproducing the results	Record dataset versions, software configurations, random seeds, and model parameters
Ethical Validity	Network traffic datasets may contain sensitive or identifiable information	Use appropriately licensed datasets and follow privacy and data protection requirements

XVI. REPRODUCIBILITY AND ARTIFACT PLAN

Reproducibility is an important requirement in scientific research because it enables other researchers to verify experimental findings and repeat the procedures used to obtain the reported results. In machine learning-based cybersecurity research, reproducibility requires maintaining accurate records of datasets, preprocessing techniques, feature selection methods, algorithm configurations, model training procedures, and performance evaluation results. The proposed research focuses on comparing six machine learning algorithms for security breach detection: Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression.

The primary objective of the reproducibility and artifact plan is to preserve the experimental resources and documentation required to repeat the comparative evaluation under similar conditions.

XVII. ETHICAL CONSIDERATIONS AND HUMAN-PARTICIPANT DATA

Ethical considerations are an important part of research involving cybersecurity datasets and machine learning-based security breach detection systems. The proposed study focuses on analysing network traffic data and evaluating different machine learning algorithms to identify normal and malicious activities. During this process, appropriate measures must be taken to maintain data privacy, confidentiality, security, and research integrity.

The study will consider ethical principles related to data collection, data usage, information confidentiality, participant privacy, and responsible machine learning implementation. The primary objective is to ensure that all research activities are conducted using authorized data sources and appropriate data protection practices.

XVIII. PRACTICAL SIGNIFICANCE

The increasing number of cyberattacks and security breaches has created a strong demand for effective and reliable cybersecurity solutions. Organizations, businesses, educational institutions, and government agencies depend on digital information systems to manage sensitive information and perform essential operations. Therefore, identifying malicious network activities and preventing unauthorized access have become important aspects of information security.

The study compares Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression to provide insights into their classification performance, computational efficiency, and detection capabilities.

XIX. FUTURE RESEARCH

The increasing complexity of cyber threats and the continuous development of digital technologies create opportunities for further research in machine learning-based security breach detection. Although traditional machine learning algorithms provide useful approaches for identifying malicious network activities, several challenges remain in improving detection accuracy, computational efficiency, adaptability, and reliability.

The present research focuses on comparing six machine learning algorithms: Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression. Future research can extend this comparative study by incorporating advanced algorithms, alternative datasets, real-time monitoring techniques, and improved threat detection methodologies.

XX. CONCLUSION

The increasing dependence on digital technologies, online platforms, and interconnected computer networks has made cybersecurity an important concern for individuals, organizations, and businesses. The growing frequency and complexity of cyberattacks, including unauthorized access, malware, phishing, network intrusions, and data theft, have highlighted the need for effective and reliable security breach detection techniques.

The proposed research methodology includes cybersecurity dataset collection, data preprocessing, feature selection, model training, testing, and performance evaluation. The selected algorithms will be evaluated using multiple performance measures, including accuracy, precision, recall, F1-score, ROC-AUC, false positive rate, and false negative rate. Training time, prediction time, and computational resource consumption will also be considered to understand the practical efficiency of each algorithm

DATA AVAILABILITY

At the current stage of manuscript preparation, the final cybersecurity dataset has not yet been selected and no verified experimental dataset or model execution results have been supplied. Therefore, no specific dataset is claimed to have been used for completed experimental evaluation.

After selecting the dataset, the research will document the data preprocessing procedures, feature selection methods, training and testing partitions, and experimental configurations. Where permitted by the dataset licensing conditions, the dataset source and relevant access links will be provided to support independent verification and reproducibility.

CODE AND ARTIFACT AVAILABILITY

The proposed research involves the implementation and comparative evaluation of six machine learning algorithms for security breach detection: Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Logistic Regression.

CONFLICT OF INTEREST

The authors Sagar kashid ,Sanchit Tikhile, Sohel Shaikh declare that they have no known financial, personal, professional, or other conflicts of interest that could have influenced the research, analysis, experimental evaluation, or preparation of this manuscript

ACKNOWLEDGMENT

The authors would like to thank the individuals who participated in the experimental activities and provided feedback during the study. The authors also acknowledge the support of colleagues and peers who assisted with application testing, experimental setup, and validation. No identifiable participant information is reported in this manuscript.

.

REFERENCES

A. H. Ali, M. Charfeddine, B. Ammar, B. B. Hamed, F. Albalwy, A. Alqarafi, and A. Hussain, "Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey," Frontiers in Computer Science, vol. 6, Art. no. 1387354, 2024. doi: 10.3389/fcomp.2024.1387354.

H. M. R. U. Rehman et al., "A systematic literature study of machine learning techniques based intrusion detection: Datasets, models, challenges, and future directions," Journal of Big Data, vol. 12, Art. no. 264, 2025. doi: 10.1186/s40537-025-01323-2.

M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, Canada, 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. doi: 10.5220/0006639801080116.

J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann, 1993.

L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.

C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273–297, 1995. doi: 10.1007/BF00994018.

T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967. doi: 10.1109/TIT.1967.1053964.

P. M. Domingos and M. J. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," Machine Learning, vol. 29, pp. 103–130, 1997. doi: 10.1023/A:1007413511361.

D. W. Hosmer, Jr., S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013. doi: 10.1002/9781118548387.

F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

J. Demšar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine Learning Research, vol. 7, pp. 1–30, 2006.

M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009. doi: 10.1016/j.ipm.2009.03.002.

2026 The Author(s) https://creativecommons.org/licenses/by/4.0/ DOI: 10.63300/arjst.v5i07.10