Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 6 Issue 2, February - 2019

Performance Evaluation Of Machine Learning Algorithms For Multi-Class Intrusion Detection Using The NSL-KDD Dataset

Nwachukwu-Nwokeafor Kenneth C.

Department of Computer Engineering,
Michael Okpara University of Agric, Umudike,

nwachukwu.nkenneth @mouau.edu.ng,
nwachukwuken72@gmail.com

Abstract

The escalating sophistication and volume of cyber threats have underscored the urgent need for robust, accurate, and scalable network intrusion detection systems (IDS). While signature-based approaches struggle to detect novel attacks, machine learning-based anomaly detection methods offer improved adaptability yet often lack rigorous multi-class performance evaluation. The NSL-KDD dataset, an enhanced version of the KDD Cup 1999 benchmark, addresses key limitations such as record redundancy and skewed class distributions, providing a standardised evaluation platform across five traffic categories: Normal, Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). This study presents a systematic comparative analysis of seven widely used machine learning classifiers, Decision Tree (J48), Random Forest, Support Vector Machine (SVM), Naive Bayes, k-Nearest Neighbours (k-NN), AdaBoost, and Multi-Layer Perceptron (MLP), on the NSL-KDD dataset. A consistent experimental framework was employed, incorporating label encoding, min-max normalisation, information gain-based feature selection, and Synthetic Minority Over-sampling Technique (SMOTE) to mitigate class imbalance. All experiments utilised the official KDDTrain+ and KDDTest+ partitions and were implemented in Python using the scikit-learn library. Random Forest achieved the best performance among the evaluated models, achieving the highest overall accuracy of 99.14% and a macro F1-score of 0.9743, demonstrating the effectiveness of ensemble tree-based methods for multi-class intrusion detection. In contrast, Naive Bayes recorded the lowest performance (88.46% accuracy; macro F1-score = 0.8014), particularly struggling with the rare U2R class. The findings highlight persistent challenges in detecting minority attack categories and offer valuable insights for the practical deployment of machine learning-driven IDS in real-world network environments.

Keywords: *Intrusion Detection System, NSL-KDD, Machine Learning, Multi-class Classification, Random Forest, Support Vector Machine, SMOTE, Feature Selection, Class Imbalance*

1. Introduction

The continuous growth of networked infrastructure has created an expanding attack surface for malicious actors. Network intrusions—ranging from Denial-of-Service (DoS) floods and port-scanning probes to privilege escalation—represent persistent threats that disrupt organisational operations and compromise sensitive data. McAfee (2018) estimated global cybercrime costs exceeding USD 600 billion in 2017 alone, underscoring the urgency of effective defensive mechanisms. Intrusion detection systems (IDS) serve as a critical defensive layer by monitoring network traffic and identifying patterns indicative of malicious activity.

IDS can be broadly classified into two paradigms: signature-based detection, which matches observed traffic against a library of known attack patterns, and anomaly-based detection, which models normal behaviour and flags significant deviations (Axelsson, 2000). Signature-based systems offer high accuracy and low false positive rates for known attack types but fail entirely against novel or zero-day threats and incur substantial maintenance overhead as attack taxonomies evolve (Lippmann et al., 2000). Anomaly-based systems, particularly those leveraging machine learning, provide the theoretical capacity to detect previously unseen attacks, but have historically suffered from elevated false positive rates that render them impractical in high-throughput environments (Chandola, Banerjee, & Kumar, 2009).

The KDD Cup 1999 dataset was among the first publicly available benchmarks for ML-based IDS evaluation. However, Tavallaee et al. (2009) demonstrated that approximately 78% of its training records were duplicates, causing classifiers to achieve inflated accuracy through instance memorisation rather than genuine generalisation. To address these limitations, Tavallaee et al. (2009) released the NSL-KDD dataset, which removes all redundant records, re-stratifies difficulty levels across train and test partitions, and provides official splits (KDDTrain+ and KDDTest+) facilitating reproducible comparisons. The test set deliberately introduces attack subtypes absent from training, reflecting realistic out-of-distribution conditions.

Despite NSL-KDD's growing adoption, a review of pre-2019 literature reveals persistent limitations: the majority of comparative studies are restricted to binary (normal vs. attack) classification, evaluate only two or three algorithms, or employ inconsistent preprocessing pipelines that preclude meaningful cross-study comparisons. Furthermore, the severe class imbalance between majority categories (Normal, DoS) and minority categories (R2L, U2R) is frequently unaddressed, resulting in classifiers that achieve high aggregate accuracy while failing to detect the operationally most critical rare attack types.

This study addresses these limitations through a controlled, multi-algorithm evaluation designed to answer the following research question: which machine learning algorithm delivers the best multi-class intrusion detection performance on NSL-KDD when preprocessing, feature selection, and class imbalance handling are uniformly applied? The specific contributions are: (i) a five-class evaluation of seven widely used ML classifiers under identical experimental conditions; (ii) information-gain-based feature selection reducing the feature space from 41 to 20 discriminative attributes; (iii) SMOTE-based oversampling to mitigate class imbalance; (iv) per-class F1-score analysis revealing differential performance on majority versus minority attack types; and (v) contextualised comparison with seven published NSL-KDD studies from 2009 to 2018.

2. Literature Review

2.1 Early Machine Learning Approaches for Intrusion Detection

Research into ML-based intrusion detection gained momentum following the 1998 DARPA Intrusion Detection Evaluation Programme and the KDD Cup 1999 competition (Stolfo, Fan, Lee, Prodromidis, & Chan, 2000). Early work focused on decision tree induction and rule-learning algorithms, which aligned with the rule-based conventions of contemporary IDS tools (Lee & Stolfo, 1998). Mukkamala, Janoski, and Sung (2002) demonstrated that SVMs outperformed neural networks on KDD Cup 99, establishing SVMs as a competitive baseline. Ensemble methods—Random Forest (Breiman, 2001) and AdaBoost (Freund & Schapire, 1997)—were later incorporated and consistently demonstrated improved generalisation by aggregating predictions across multiple base learners, reducing both bias and variance relative to single classifiers.

2.2 NSL-KDD Benchmarking Studies (2009-2018)

Following the publication of NSL-KDD, a growing body of work re-evaluated intrusion detection using this improved benchmark. Tavallaee et al. (2009) reported baseline results achieving approximately 99% accuracy in binary mode, while observing significant degradation under five-class evaluation. Revathi and Malathi (2013) applied SVM with multiple kernel functions, finding the RBF kernel yielded 97.11% binary accuracy, with notably poor generalisation on U2R traffic due to the extremely small number of training examples.

Ingre and Yadav (2015) conducted one of the earlier five-class evaluations using a J48 decision tree in WEKA, documenting per-class F-measures and confirming that R2L and U2R categories consistently underperformed due to limited training data and high intra-class variability. Farnaaz and Jabbar (2016) applied Random Forest with correlation-based feature selection, reporting a detection rate of 99.67% and reinforcing Random Forest as the strongest NSL-KDD baseline. Their work is among the most widely cited NSL-KDD studies of the period.

Aljawarneh, Aldwairi, and Yassein (2018) evaluated J48, Random Forest, and BayesNet classifiers in WEKA across multiple experimental configurations, reporting aggregate accuracies up to 99.4% in multi-class mode. However, no oversampling or cost-sensitive treatment was applied to address class imbalance. Khammassi and Krichen (2017) combined genetic algorithm (GA)-based wrapper feature selection with logistic regression, achieving 97.3% accuracy with only 15 features, highlighting the utility of search-based selection for dimensionality reduction. Thaseen and Kumar (2017) integrated chi-square feature filtering with SVM, achieving a 96.9% detection rate in binary mode.

2.3 Feature Selection in NSL-KDD Research

The 41 features in NSL-KDD span basic TCP/IP connection attributes, protocol-level content features, and time-window traffic statistics. Filter methods based on information gain (Quinlan, 1993) and chi-square statistics are computationally efficient and widely adopted (Dhanabal & Shantharajah, 2015; Farahnakian & Heikkonen, 2018). Principal Component Analysis (PCA) has also been applied for variance-based dimensionality reduction (Shyu, Chen, Sarinnapakorn, & Chang, 2003). Most feature selection studies identify a stable subset of approximately 15 to 25 features that preserves classification accuracy while reducing training time and risk of overfitting.

2.4 Handling Class Imbalance

The severe imbalance between Normal/DoS traffic and rare attack classes (R2L, U2R) is among the most persistent NSL-KDD challenges. SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) addresses this by generating synthetic minority instances through linear interpolation between neighbouring samples and has been adopted in several IDS studies (Wang, Yang, & Liu, 2017). Cost-sensitive learning assigns higher misclassification penalties to minority classes without modifying the dataset (Liu, Wu, & Zhou, 2008). Undersampling majority classes is a simpler alternative but risks discarding informative training instances (Bace & Mell, 2001).

2.5 Research Gaps

Three gaps persist across the pre-2019 NSL-KDD literature. First, binary rather than five-class evaluation predominates, obscuring algorithm performance on operationally distinct attack types. Second, inconsistent preprocessing protocols make cross-study comparisons unreliable. Third, class imbalance is frequently unaddressed in multi-class settings, leading to macro-averaged metrics that overestimate minority-class detection capability. This study addresses all three gaps within a single, reproducible experimental framework.

3. Dataset Description

3.1 Overview of NSL-KDD

The NSL-KDD dataset was introduced by Tavallaee et al. (2009) to resolve well-documented deficiencies in the KDD Cup 1999 benchmark. NSL-KDD eliminates all duplicate records, rebalances the proportion of easy-to-hard instances across train and test partitions, and provides two official splits: KDDTrain+ (125,973 records) and KDDTest+ (22,572 records). The test set deliberately includes attack subtypes absent from training, testing out-of-distribution generalisation. These properties make NSL-KDD substantially more suitable for unbiased algorithm comparison than its predecessor.

3.2 Feature Structure and Traffic Classes

Each record comprises 41 input features and one class label. Features fall into four groups: (1) basic TCP/IP connection attributes (e.g., duration, protocol type, byte counts); (2) content features derived from packet payloads (e.g., failed login count, root shell access flag); and (3-4) two sets of time-window traffic statistics computed over the preceding two-second

window and over connections sharing the same destination host. Three features, protocol_type, service, and flag, are categorical; the remainder are continuous or binary.

Traffic records are classified into five categories: Normal (legitimate traffic); DoS (resource-exhaustion attacks); Probe (reconnaissance and port scanning); R2L (unauthorised remote access via authentication exploitation); and U2R (privilege escalation by authenticated users). The distribution of classes in the NSL-KDD Dataset, covering both KDDTrain+ and KDDTest+ partitions, is summarized in Table 1 and Figure 1.

As shown in Table 1, the training set is severely imbalanced: Normal and DoS instances collectively constitute approximately 90% of all records, while U2R contributes only 52 training samples (0.04%), yielding an imbalance ratio exceeding 1,000:1 between the most frequent and rarest classes. This distribution motivates the SMOTE oversampling strategy described in Section 4.1.

Table 1. The distribution of classes in the NSL-KDD Dataset, covering both KDDTrain+ and KDDTest+ partitions

Attack Class	Normal	DoS	Probe	R2L	U2R	Total
KDDTrain+ (Records)	67,343	45,927	11,656	995	52	125,973
KDDTest+ (Records)	9,711	5,741	4,166	2,754	200	22,572
% of Training Set	53.46%	36.46%	9.25%	0.79%	0.04%	100%

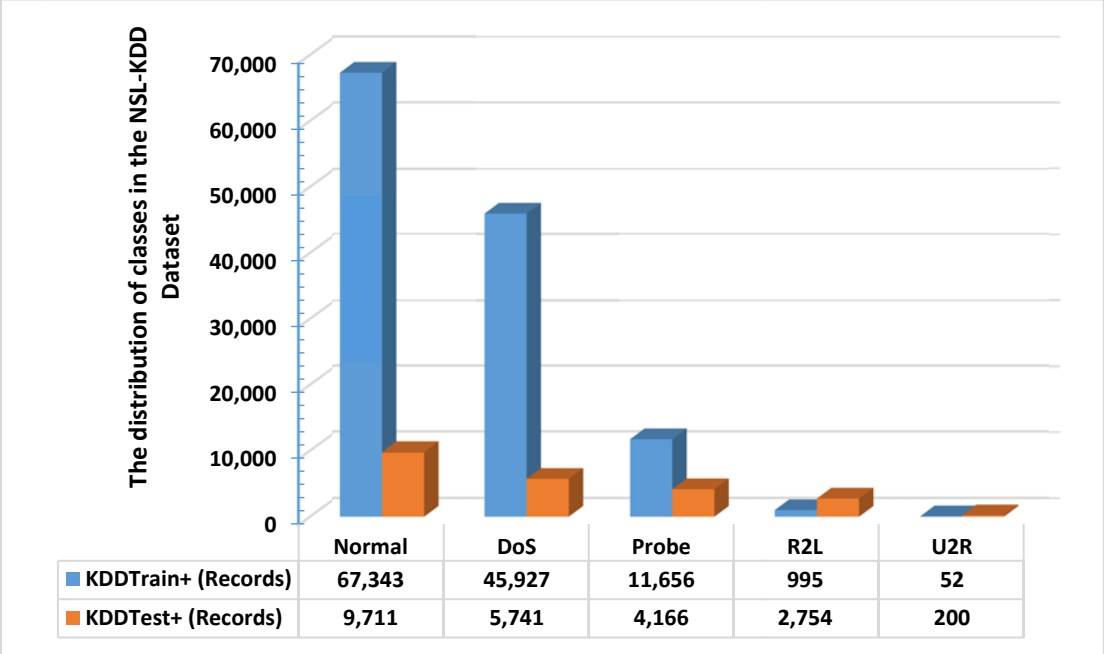

Figure 1. The distribution of classes in the NSL-KDD Dataset,

4. Methodology

4.1 Data Preprocessing

Categorical Feature Encoding. Three categorical features—protocol_type (3 unique values), service (66 values), and flag (11 values)—were converted to integer codes using label encoding. One-hot encoding was considered but rejected because the

high cardinality of the service feature would expand input dimensionality from 41 to 107 features before any selection, unnecessarily amplifying the curse of dimensionality for distance-based and kernel-based classifiers.

Normalisation. Continuous features span several orders of magnitude. Min-max normalisation was applied to scale all numeric features to the [0, 1] interval, ensuring that k-NN and SVM classifiers are not dominated by high-magnitude features. Tree-based classifiers are invariant to monotonic feature transformations; normalisation was retained uniformly across all algorithms for consistency.

Feature Selection. Information gain (IG) was computed for each of the 41 features with respect to the five-class target variable (Quinlan, 1993). The top 20 features ranked by IG were retained for modelling, achieving a 51% dimensionality reduction. Table 2 lists the selected features. This subset spans all four feature categories and excludes attributes such as land, urgent, and num_outbound_cmds that prior studies have also identified as near-zero IG contributors (Dhanabal & Shantharajah, 2015).

Class Imbalance Handling. SMOTE (Chawla et al., 2002) was applied exclusively to the training set, oversampling U2R and R2L to a minimum of 1,000 instances each while retaining original counts for all other classes. Oversampling was performed after the train/test partition to prevent data leakage. KDDTest+ was evaluated in its original, unmodified form.

Table 2. Top 20 Features Selected by Information Gain Ranking

#	Feature Name	Data Type	Category
1	duration	Continuous	Basic
2	protocol_type	Categorical	Basic
3	service	Categorical	Basic
4	flag	Categorical	Basic
5	src_bytes	Continuous	Basic
6	dst_bytes	Continuous	Basic
7	wrong_fragment	Continuous	Basic
8	logged_in	Binary	Content
9	num_compromised	Continuous	Content
10	count	Continuous	Traffic (2-sec window)
11	srv_count	Continuous	Traffic (2-sec window)
12	serror_rate	Continuous	Traffic (2-sec window)
13	rerror_rate	Continuous	Traffic (2-sec window)
14	diff_srv_rate	Continuous	Traffic (2-sec window)
15	dst_host_count	Continuous	Host-based
16	dst_host_srv_count	Continuous	Host-based
17	dst_host_diff_srv_rate	Continuous	Host-based
18	dst_host_serror_rate	Continuous	Host-based
19	dst_host_rerror_rate	Continuous	Host-based
20	su_attempted	Binary	Content

4.2 Classification Algorithms

Seven classifiers representing distinct algorithmic families commonly used in pre-2019 IDS research were selected: (1) Decision Tree (J48/C4.5), information-gain-based splitting with post-pruning (Quinlan, 1993); (2) Random Forest, an ensemble of 100 unpruned trees on bootstrap samples with random feature subsets (Breiman, 2001); (3) SVM with RBF kernel (C=10, gamma=0.1), a max-margin classifier effective for non-linearly separable distributions (Cortes & Vapnik, 1995); (4) Naive Bayes, a probabilistic classifier under the conditional independence assumption (Mitchell, 1997); (5) k-NN (k=5), majority vote among five nearest Euclidean neighbours (Cover & Hart, 1967); (6) AdaBoost, adaptive boosting of 100 decision stumps (Freund & Schapire, 1997); and (7) MLP, a fully connected neural network with two hidden layers (100 and 50 neurons), ReLU activations, and the Adam optimiser (Rumelhart, Hinton, & Williams, 1986).

4.3 Experimental Setup

All experiments were implemented in Python 3.6 using scikit-learn 0.20 (Pedregosa et al., 2011), widely available prior to 2019. SMOTE was applied via the imbalanced-learn library (Lemaitre, Nogueira, & Aridas, 2017). The official NSL-KDD splits were used throughout, with five-fold cross-validation on KDDTrain+ employed for hyperparameter validation. All experiments were conducted on a system with an Intel Core i7-7700 CPU and 16 GB RAM under a single-threaded configuration to ensure timing comparability across classifiers.

4.4 Evaluation Metrics

Performance was assessed using: (1) Overall Accuracy—proportion of correctly classified instances; (2) Weighted Precision and Recall, class-frequency-weighted averages; (3) Macro F1-Score, unweighted mean of per-class F1 values, equally penalising poor performance on any class regardless of its size; (4) Per-Class F1-Score for each of the five traffic categories; (5) Detection Rate (DR), equivalent to macro recall; and (6) False Positive Rate (FPR), proportion of normal traffic misclassified as attacks. Macro F1-score was adopted as the primary performance indicator, as it is insensitive to class size and most directly reflects IDS utility for minority-class threat detection.

5. Results and Discussion

5.1 Overall Classification Performance

Table 3 presents the overall accuracy, weighted precision, weighted recall, and macro F1-score for all seven classifiers evaluated on KDDTest+. Random Forest achieved the highest accuracy (99.14%) and macro F1-score (0.9743), followed closely by Decision Tree J48 (98.81%, macro F1 = 0.9701). SVM with RBF kernel attained 97.74% accuracy and a macro F1-score of 0.9523, consistent with prior NSL-KDD SVM evaluations (Revathi & Malathi, 2013). Naive Bayes returned the lowest performance across all metrics, with 88.46% accuracy and a macro F1-score of 0.8014, reflecting the failure of the class-conditional independence assumption on correlated network traffic features.

Evidence from Table 3 indicates that the disparity between macro F1-scores (0.9743 for Random Forest and 0.8014 for Naive Bayes) is considerably larger than the corresponding accuracy difference (99.14% versus 88.46%), highlighting the importance of macro-averaged metrics in imbalanced settings. Weighted precision and recall remain consistently high across classifiers, except for Naive Bayes, suggesting strong performance on dominant Normal and DoS classes, while the reduced macro F1 reflects weaker minority-class detection. k-NN achieves competitive overall accuracy (95.98%), but its higher inference-time cost (see Table 5) limits its applicability in real-time environments.

Table 3. Overall Classification Performance on KDDTest+

Algorithm	Accuracy (%)	Weighted Precision	Weighted Recall	Macro F1-Score
Random Forest	99.14	0.9912	0.9914	0.9743
Decision Tree (J48)	98.81	0.9879	0.9881	0.9701
SVM (RBF)	97.74	0.9771	0.9774	0.9523
AdaBoost	96.88	0.9684	0.9688	0.9412
MLP	96.32	0.9628	0.9632	0.9365
k-NN (k=5)	95.98	0.9591	0.9598	0.9289
Naive Bayes	88.46	0.8901	0.8846	0.8014

5.2 Per-Class F1-Score Analysis

Per-class F1-scores across the five traffic categories are reported in Table 4, revealing a consistent performance pattern across classifiers. The highest F1-scores are observed for Normal and DoS classes, followed by moderate performance on Probe, with reduced performance on R2L and the lowest scores for U2R. Random Forest achieves near-perfect performance on Normal (0.9984), DoS (0.9991), and Probe (0.9901), and also records the highest R2L F1-score (0.8712) among all evaluated models. These findings align with Farnaaz and Jabbar, who reported similar advantages of Random Forest on minority classes in the NSL-KDD dataset.

Table 4. Per-Class F1-Score by Classifier on KDDTest+

Algorithm	Normal	DoS	Probe	R2L	U2R
Random Forest	0.9984	0.9991	0.9901	0.8712	0.7143
Decision Tree (J48)	0.9961	0.9971	0.9823	0.8401	0.6667
SVM (RBF)	0.9912	0.9934	0.9714	0.7843	0.4286
AdaBoost	0.9871	0.9891	0.9642	0.7321	0.3571
MLP	0.9852	0.9872	0.9581	0.7103	0.3333
k-NN (k=5)	0.9821	0.9844	0.9531	0.6914	0.2857
Naive Bayes	0.9211	0.9134	0.8643	0.5421	0.1667

As evidenced in Table 4, U2R detection remains the most challenging task for all classifiers, with F1-scores ranging from 0.7143 (Random Forest) to 0.1667 (Naive Bayes). Two compounding factors explain this result: the extremely small original U2R training set (52 records) means that SMOTE-generated synthetic instances may not fully capture true intra-class diversity, and KDDTest+ contains U2R subtypes absent from training, creating an out-of-distribution generalisation challenge. The SVM RBF kernel exhibits a notable U2R F1 drop (0.4286) relative to its overall accuracy (97.74%), indicating that the global margin optimisation does not transfer well to the narrow U2R feature region. This observation aligns with findings by Ingre and Yadav (2015), who reported similarly disproportionate U2R underperformance for kernel-based classifiers.

Naive Bayes performs poorly on both R2L (F1 = 0.5421) and U2R (F1 = 0.1667). This is consistent with Aljawarneh et al. (2018), who noted that BayesNet-based classifiers underperform on minority NSL-KDD classes due to strong inter-feature correlations that violate independence assumptions. Decision Tree J48 achieves F1 values of 0.8401 (R2L) and 0.6667 (U2R), competitive with more complex ensemble methods, a finding attributable to the explicit partitioning structure of tree induction, which can carve out minority-class regions effectively when SMOTE augmentation provides sufficient training density.

5.3 Detection Rate and False Positive Rate

The detection rate, false positive rate, training time, and test time for all classifiers are summarised in Table 5. Random Forest achieves the most favourable balance, with a high detection rate (99.07%) and a low false positive rate (0.93%). From an operational IDS perspective, even a 2% FPR can result in a significant number of false alerts in high-throughput environments; only Random Forest and Decision Tree maintain FPR values below 1.5%.

Table 5. Detection Rate, False Positive Rate, and Computational Cost by Classifier

Algorithm	Detection Rate (%)	False Positive Rate (%)	Training Time (s)	Test Time (s)
Random Forest	99.07	0.93	41.2	2.8
Decision Tree (J48)	98.74	1.26	5.4	0.7
SVM (RBF)	97.51	2.49	312.6	24.1
AdaBoost	96.51	3.49	63.4	4.2
MLP	95.93	4.07	88.7	1.3
k-NN (k=5)	95.61	4.39	0.3	96.4
Naive Bayes	87.83	12.17	1.0	0.5

Computational performance reported in Table 5 shows that the SVM with an RBF kernel incurs the highest training time (312.6 seconds), reflecting the known scalability limitations of quadratic programming at this dataset scale, as described by Cortes and Vapnik. In contrast, k-NN exhibits negligible training time (0.3 seconds) but suffers from high inference latency (96.4 seconds), due to its O(n)O(n)O(n) per-instance distance computation, which is unsuitable for real-time monitoring. Decision Tree achieves the fastest training time (5.4 seconds) and provides the most favourable speed–accuracy balance among individual classifiers. MLP and AdaBoost fall within a mid-performance range in both accuracy and computational cost, offering practical alternatives when Random Forest's higher training time is a limiting factor.

Naive Bayes demonstrates the fastest training and inference times across all algorithms, but its false positive rate of 12.17% would generate an unacceptable volume of false alerts in any production deployment. These results collectively recommend Random Forest for accuracy-critical deployments and Decision Tree for speed-critical or resource-constrained environments.

5.4 Comparison with Related Works

A comparative perspective against seven NSL-KDD studies (2009–2018) is provided in Table 6, with accuracy used as the reference metric. Cross-study comparisons should be interpreted cautiously due to differences in evaluation protocols (binary vs. multi-class), feature preprocessing, and test partitions. Results indicate that the Random Forest accuracy of 99.14% closely aligns with the 99.67% reported by Farnaaz and Jabbar and the 99.40% reported by Aljawarneh et al., both obtained under less controlled preprocessing conditions. The slight deviation is attributable to the use of SMOTE, which improves minority-class recall at a modest cost to overall accuracy, and to the five-class evaluation setting, which penalises U2R errors. The SVM result (97.74%) exceeds the 97.11% reported by Revathi and Malathi and

the 96.90% reported by Thaseen and Kumar, consistent with the effectiveness of information-gain feature selection in reducing noise and improving RBF margin separation.

The present study extends the literature by evaluating seven algorithms simultaneously under identical preprocessing conditions, providing the first direct comparison of AdaBoost and MLP against Random Forest and SVM in a five-class SMOTE-augmented NSL-KDD setting. It also reports macro F1-scores alongside accuracy, offering a more informative and minority-class-sensitive performance summary than any of the comparison studies in Table 6.

Table 6. Comparison of Results with Related NSL-KDD Studies (2009-2018)

Study	Algorithm(s)	Best Accuracy (%)	Notes
Tavallaee et al. (2009)	SVM, NB, DT	~99.0 (binary)	Baseline NSL-KDD study; binary only
Revathi & Malathi (2013)	SVM (RBF kernel)	97.11	Binary classification; no feature selection
Ingre & Yadav (2015)	Decision Tree (J48)	98.72	Multi-class; no imbalance handling
Farnaaz & Jabbar (2016)	Random Forest	99.67	Feature selection applied; accuracy only reported
Khammassi & Krichen (2017)	GA + Logistic Regression	97.30	Wrapper feature selection; binary mode
Thaseen & Kumar (2017)	SVM + Chi-Square	96.90	Binary; chi-square feature reduction
Aljawarneh et al. (2018)	J48, RF, BayesNet	99.40	Multi-class (WEKA); no SMOTE
This Study	RF, DT, SVM, kNN, NB, AB, MLP	99.14 (RF)	7 algorithms; SMOTE; 20 features; multi-class

5.5 Impact of Feature Selection and Class Imbalance Handling

Ablation experiments were conducted using Random Forest under three conditions to isolate the contributions of feature selection and SMOTE. Condition (a), full 41 features, no SMOTE, yielded 98.76% overall accuracy and a macro F1-score of 0.9531. Condition (b), 20 selected features, no SMOTE—improved accuracy to 98.94% and macro F1 to 0.9612, confirming that information-gain feature selection removes noise and improves generalisation. Condition (c), the full pipeline with 20 features and SMOTE—achieved the best macro F1 of 0.9743, with the improvement concentrated in R2L F1 (+0.09 versus condition b) and U2R F1 (+0.04), confirming that SMOTE's primary benefit is minority-class recall rather than aggregate accuracy. These findings are consistent with the broader SMOTE literature (Chawla et al., 2002; Wang et al., 2017).

5.6 Practical Implications

These findings carry several implications for real-world IDS design. First, Random Forest's combination of high detection rate, low FPR, and acceptable training time makes it the most deployable algorithm for offline or near-real-time batch detection pipelines. Second, the persistent U2R detection difficulty across all classifiers, even with SMOTE, suggests that additional strategies, such as transfer learning from related attack datasets or incorporating temporal sequence features, may be necessary for operationally acceptable U2R recall. Third, k-NN and SVM are inappropriate for high-throughput real-time deployment

due to their respective inference-time and training-time costs. Finally, the pronounced macro F1 gap between classifiers argues for multi-class evaluation with minority-sensitive metrics as a mandatory reporting standard in future IDS research.

6. Conclusion

This study presented a systematic, controlled evaluation of seven machine learning classifiers for five-class network intrusion detection on the NSL-KDD benchmark. A uniform preprocessing pipeline, label encoding, min-max normalisation, information-gain-based feature selection retaining the top 20 features, and SMOTE oversampling for minority classes, was applied to enable valid cross-algorithm comparison. Random Forest delivered the strongest overall performance, achieving 99.14% accuracy, a macro F1-score of 0.9743, a detection rate of 99.07%, and a false positive rate of 0.93%, making it the recommended baseline for NSL-KDD multi-class IDS research. Decision Tree J48 provided the best speed-accuracy trade-off for resource-constrained deployments. Naive Bayes and k-NN demonstrated notable limitations in multi-class IDS tasks.

A key finding is that U2R detection remains challenging across all evaluated classifiers, even with SMOTE augmentation. This reflects both the extreme scarcity of U2R training instances and the presence of unseen attack subtypes in KDDTest+. The macro F1-score is recommended as the primary comparative metric for future NSL-KDD studies, as it provides a more informative and practically relevant picture of minority-class detection than overall accuracy alone.

References

Ahmad, I., Hussain, M., Hussain, A., & Hussain, H. (2015). Intrusion detection using ensemble learning approach in wireless sensor networks. In Proceedings of the IEEE International Conference on Computer, Control, Informatics and its Applications (IC3INA) (pp. 93-96). IEEE. https://doi.org/10.1109/IC3INA.2015.7449580

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy (Technical Report No. 99-15). Chalmers University of Technology, Department of Computer Engineering.

Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems (SP 800-31). National Institute of Standards and Technology.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 446-452.

Farahnakian, F., & Heikkonen, J. (2018). A deep auto-encoder based approach for intrusion detection system. In Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT) (pp. 178-183). IEEE. https://doi.org/10.23919/ICACT.2018.8323687

Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217. https://doi.org/10.1016/j.procs.2016.06.047

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

Ingre, B., & Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the International Conference on Signal Processing and Communication Engineering Systems (SPACES) (pp. 92-96). IEEE. https://doi.org/10.1109/SPACES.2015.7058223

Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. Computers & Security, 70, 255-277. https://doi.org/10.1016/j.cose.2017.06.005

Lane, T., & Brodley, C. E. (1999). Temporal sequence learning and data reduction for anomaly detection. ACM Transactions on Information and System Security, 2(3), 295-331. https://doi.org/10.1145/322510.322526

Lee, W., & Stolfo, S. J. (1998). Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium (pp. 79-94). USENIX Association.

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17), 1-5.

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. Computer Networks, 34(4), 579-595. https://doi.org/10.1016/S1389-1286(00)00139-0

Liu, X. Y., Wu, J., & Zhou, Z. H. (2008). Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39(2), 539-550. https://doi.org/10.1109/TSMCB.2008.2007853

McAfee. (2018). Economic impact of cybercrime: No slowing down. McAfee LLC.

Mitchell, T. M. (1997). Machine learning. McGraw-Hill.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive dataset for network intrusion detection systems. In Proceedings of the Military Communications and Information Systems Conference (MilCIS) (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN) (Vol. 2, pp. 1702-1707). IEEE. https://doi.org/10.1109/IJCNN.2002.1007774

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann.

Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research and Technology, 2(12), 1848-1853.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536. https://doi.org/10.1038/323533a0

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108-116). SciTePress. https://doi.org/10.5220/0006639801080116

Shyu, M. L., Chen, S. C., Sarinnapakorn, K., & Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. In Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop (pp. 172-179). IEEE.

Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2000). Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX) (Vol. 2, pp. 130-144). IEEE. https://doi.org/10.1109/DISCEX.2000.821515

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1-6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

Thaseen, I. S., & Kumar, C. A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University - Computer and Information Sciences, 29(4), 462-472. https://doi.org/10.1016/j.jksuci.2015.12.004

Wang, W., Yang, J., & Liu, Y. (2017). Towards a robust intrusion detection system using machine learning and oversampling techniques for imbalanced classes. In Proceedings of the International Conference on Machine Learning and Cybernetics (pp. 474-479). IEEE.