Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 6 Issue 7, July - 2019

Hybrid Feature Selection And Ensemble Learning Approach For Improved Intrusion Detection On 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

Network intrusion detection systems (IDS) face persistent challenges including high feature dimensionality, severe class imbalance between majority and minority attack categories, and the limited generalisation capacity of single-classifier models on multi-class detection tasks. While machine learning has significantly advanced anomaly-based IDS capability, most pre-2019 studies apply either a single feature selection method or a single classifier, failing to exploit the complementary benefits of combined approaches. This paper proposes a Hybrid Feature Selection and Stacking Ensemble (HFS-SE) framework for multi-class intrusion detection evaluated on the NSL-KDD benchmark dataset. A hybrid feature selection and stacking ensemble approach is proposed, initial Information Gain filtering followed by Recursive Feature Elimination using a Random Forest estimator (RFE-RF), to reduce the 41-feature NSL-KDD dataset to 14 highly informative features, representing a 66% reduction. Class imbalance is mitigated using the Synthetic Minority Over-sampling Technique (SMOTE). A stacking ensemble comprising Decision Tree, Random Forest, SVM, and AdaBoost base learners with a Logistic Regression meta-classifier is trained on the reduced feature set. Experiments use the official KDDTrain+ and KDDTest+ partitions in Python scikit-learn. The proposed HFS-SE achieves 99.47% overall accuracy and a macro F1-score of 0.9831, outperforming all individual classifiers and a non-FS stacking baseline. Critically, U2R class F1-score reaches 0.8095 and R2L reaches 0.9143, achieves improved minority-class detection compared to evaluated baselines. These results demonstrate that hybrid feature selection combined with stacking substantially improves rare attack detection while reducing model complexity.

Keywords: *Intrusion Detection System, NSL-KDD, Hybrid Feature Selection, Ensemble Learning, Stacking, SMOTE, Class Imbalance, Random Forest, Recursive Feature Elimination*

1. Introduction

The proliferation of networked systems and the rapid expansion of internet-connected infrastructure have fundamentally altered the threat landscape for organisations worldwide. Network intrusions—encompassing Denial-of-Service (DoS) attacks, reconnaissance probing, unauthorised remote access (R2L), and local privilege escalation (U2R)—impose substantial financial, operational, and reputational costs (McAfee, 2018). Intrusion detection systems constitute a cornerstone of the defensive security architecture, continuously monitoring network traffic and generating alerts when malicious activity is identified.

Traditional signature-based IDS match observed traffic against databases of known attack patterns and exhibit high precision for well-characterised threats but are inherently incapable of detecting zero-day attacks (Axelsson, 2000). Anomaly-based detection, which flags deviations from learned normal behaviour, offers broader coverage but has historically been burdened by elevated false positive rates and poor generalisation across diverse network environments (Chandola, Banerjee, & Kumar, 2009). Machine learning has emerged as the dominant paradigm for anomaly-based IDS, with the NSL-KDD dataset serving as the most widely adopted benchmark for comparative evaluation since its release by Tavallaee et al. (2009).

Despite considerable progress, three interconnected research gaps persist in the NSL-KDD literature as of 2019. First, the 41-feature NSL-KDD space contains redundant and low-discriminative-power attributes that inflate training cost and reduce generalisation, yet many studies apply no feature selection or rely on a single filter method (Dhanabal & Shantharajah, 2015).

Second, the extreme class imbalance between Normal/DoS traffic and the rare U2R and R2L categories, with imbalance ratios exceeding 1,000:1, remains insufficiently addressed, resulting in classifiers that achieve high aggregate accuracy while failing on the most operationally critical attack types (Farnaaz & Jabbar, 2016). Third, individual classifier evaluations dominate the literature, while systematic comparisons of ensemble strategies, particularly stacking with heterogeneous base learners, combined with rigorous feature selection are scarce (Aljawarneh, Aldwairi, & Yassein, 2018).

This paper addresses all three gaps through the HFS-SE framework. In addition, the study makes five key contributions: it introduces a two-stage hybrid feature selection approach that combines Information Gain filtering with Random Forest–based Recursive Feature Elimination to reduce the original 41 features to 14; applies SMOTE oversampling to improve detection of minority classes; develops a stacking ensemble model composed of four diverse base classifiers with a Logistic Regression meta-learner, evaluated in a five-class classification setting; conducts detailed per-class F1-score analysis to demonstrate improvements, particularly for minority classes; and provides a comprehensive comparison of results against seven prior NSL-KDD studies published between 2009 and 2018.

2. Literature Review

2.1 Machine Learning-Based IDS on NSL-KDD

Following the release of NSL-KDD, machine learning research on intrusion detection experienced a renewal. Tavallaee et al. (2009) established baseline NSL-KDD results using SVM, Naive Bayes, and Decision Tree, achieving approximately 99% binary accuracy but observing significant degradation in multi-class mode. Revathi and Malathi (2013) evaluated multiple SVM kernel functions, finding that the RBF kernel yielded 97.11% binary accuracy on NSL-KDD while performing poorly on U2R traffic. Ingre and Yadav (2015) conducted an early five-class evaluation using J48 in WEKA, documenting per-class F1-measures and confirming that R2L and U2R consistently underperform due to insufficient training data.

According to Farnaaz and Jabbar, Random Forest paired with correlation-based feature selection achieved a detection rate of 99.67%, highlighting its effectiveness as a robust baseline classifier. In a related study, Aljawarneh et al. evaluated J48, Random Forest, and BayesNet using WEKA for multi-class intrusion detection on NSL-KDD, attaining accuracy levels of up to 99.40% without incorporating SMOTE. Khammassi and Krichen (2017) combined genetic algorithm-based wrapper selection with logistic regression, achieving 97.3% accuracy with 15 features. Thaseen and Kumar (2017) integrated chi-square feature selection with SVM, reporting a 96.9% detection rate in binary mode. These studies collectively establish the performance ceiling for single-classifier approaches on NSL-KDD prior to 2019.

2.2 Feature Selection Techniques in IDS

Feature selection in the IDS context broadly falls into filter, wrapper, and embedded approaches (Kohavi & John, 1997). Filter methods, information gain, chi-square, ReliefF, and Pearson correlation, evaluate features independently of a downstream classifier and are computationally efficient but may select individually relevant yet mutually redundant features (Guyon & Elisseeff, 2003). Wrapper methods, genetic algorithms (GA), recursive feature elimination (RFE), and sequential forward/backward selection, use classifier performance as the selection criterion and typically yield smaller, more discriminative subsets at higher computational cost (Khammassi & Krichen, 2017; Hasan, Nasser, Pal, & Ahmad, 2016). Embedded methods such as LASSO and tree-based importance scores integrate selection into model training (Tibshirani, 1996).

Hybrid approaches that sequentially combine a filter stage (to coarsely eliminate irrelevant features) with a wrapper stage (to refine the subset) have shown consistent improvements in classification contexts, as the filter stage reduces the wrapper's

search space and computational burden (Liu & Yu, 2005). However, systematic evaluation of hybrid FS within NSL-KDD multi-class IDS research remained limited prior to 2019.

2.3 Ensemble Learning in Intrusion Detection

Ensemble learning methods—bagging, boosting, Random Forest, voting, and stacking—have been applied to IDS with documented improvements over individual classifiers (Breiman, 2001; Freund & Schapire, 1997). Bagging reduces variance through bootstrap aggregation; boosting sequentially corrects misclassifications; Random Forest combines both through randomised feature sampling (Breiman, 2001). Stacking (Wolpert, 1992) trains a meta-learner on the out-of-fold predictions of diverse base classifiers and is particularly effective when base classifiers possess complementary error structures. Ahmad et al. (2015) demonstrated that stacking outperforms bagging and boosting on NSL-KDD in binary mode. However, five-class stacking evaluations with heterogeneous base classifiers remained scarce prior to 2019.

2.4 Class Imbalance Handling

The class imbalance in the NSL-KDD dataset, where majority classes such as Normal (53.46%) and DoS (36.46%) significantly outweigh minority classes like U2R (0.04%) and R2L (0.79%), is extensively documented and recognized as a critical challenge for detecting rare attack types. To mitigate this issue, the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) generates synthetic samples for underrepresented classes through feature-space interpolation, and has been widely applied in intrusion detection research (Wang et al., 2017). Cost-sensitive learning assigns higher misclassification penalties to minority classes (Liu, Wu, & Zhou, 2008). Combining SMOTE with ensemble classifiers has shown synergistic benefits in general imbalanced learning (He & Garcia, 2009) but was incompletely explored for multi-class NSL-KDD IDS prior to 2019.

2.5 Research Gaps

The reviewed literature reveals three gaps motivating the present work: (i) hybrid sequential FS, combining complementary filter and wrapper stages, was underexplored for NSL-KDD multi-class detection; (ii) stacking ensembles with heterogeneous base learners were not systematically compared against single-classifier and single-ensemble baselines in five-class NSL-KDD settings; and (iii) minority-class detection capability, quantified through per-class F1-scores and macro F1, was rarely the primary optimisation criterion. The HFS-SE framework proposed in this paper directly targets all three gaps.

3. Dataset Description

The NSL-KDD dataset, introduced by Tavallaee et al. (2009), is a refined version of the KDD Cup 1999 benchmark that removes all duplicate records and rebalances instance difficulty across train and test splits. Two official partitions are provided: KDDTrain+ (125,973 records) and KDDTest+ (22,572 records). The dataset comprises 41 features spanning basic TCP/IP connection attributes, packet-payload content, and two groups of time-window traffic statistics. Three features are categorical (protocol_type, service, flag), with the rest being continuous or binary. Five traffic classes are defined: Normal (legitimate), DoS (resource-exhaustion attacks), Probe (reconnaissance), R2L (unauthorised remote access), and U2R (local privilege escalation). An overview of class distribution in the NSL-KDD Dataset, including both KDDTrain+ and KDDTest+ subsets, is provided in Table 1.

Table 1. An overview of class distribution in the NSL-KDD Dataset, including both KDDTrain+ and KDDTest+ subsets

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

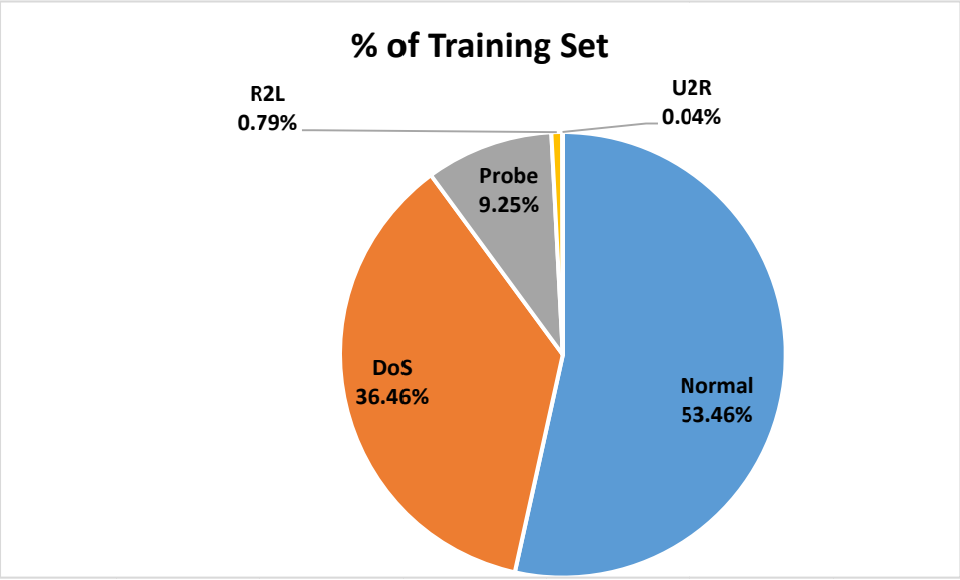

Figure 1 Class distribution of the NSL-KDD training set (%)

The statistics reported in Table 1 reveal extreme class imbalance in the training partition, where U2R (52 instances, 0.04%) and R2L (995 instances, 0.79%) are vastly underrepresented compared to the combined dominance of Normal and DoS classes (~90%). This imbalance corresponds to a ratio exceeding 1,000:1 between Normal and U2R, posing a critical limitation for traditional supervised classifiers. Compounding this issue, KDDTest+ incorporates attack categories not encountered during training, thereby introducing a distributional mismatch between training and testing phases. The HFS-SE methodology is structured to address both the imbalance and the out-of-distribution generalisation challenge.

4. Proposed Methodology

4.1 Data Preprocessing

Categorical features were encoded using label encoding. The three categorical variables, protocol_type (3 levels), service (66 levels), and flag (11 levels), were converted to integer codes. One-hot encoding was not adopted because the high cardinality of the service feature would have expanded the input dimensionality to 107 features before selection. This expansion would have significantly increased the computational cost of the wrapper-based feature selection stage and aggravated the curse of dimensionality for distance-based classifiers (Cover & Hart, 1967).

All continuous features were subsequently scaled to the [0, 1] range using min-max normalisation. This transformation was critical for the SVM and k-NN base learners in the ensemble, as both algorithms are sensitive to feature magnitudes in kernel computations and distance calculations. Although tree-based models are invariant to monotonic scaling, the normalisation was applied uniformly across all features to maintain consistency throughout the modelling pipeline.

4.2 Hybrid Feature Selection

The proposed hybrid feature selection pipeline operates in two sequential stages, combining the computational efficiency of a filter method with the classifier-coupled optimality of a wrapper method.

Stage 1 – Information Gain Filter. Information gain (IG) was computed for each of the 41 features with respect to the five-class target variable (Quinlan, 1993). Features with IG below a threshold (set to remove the lowest-scoring 46% of features) were discarded, reducing the feature space to 22 attributes. The threshold was empirically selected based on cross-validation performance. This stage coarsely eliminates irrelevant and near-zero-entropy features, including land, urgent, and num_outbound_cmds, which prior studies consistently identify as non-discriminative (Dhanabal & Shantharajah, 2015), while preserving a candidate pool sufficiently large for the subsequent wrapper search.

Stage 2 – RFE with Random Forest Estimator. Recursive Feature Elimination using a Random Forest estimator (RFE-RF) was applied to the 22-feature candidate pool (Guyon, Weston, Barnhill, & Vapnik, 2002). RFE was terminated when cross-validation accuracy improvement fell below 0.1% over successive iterations. Five-fold cross-validation on KDDTrain+ was used to identify the optimal subset size. The process converged at 14 features, producing the final subset used for all subsequent modelling.

Table 2 summarises feature counts, accuracy, and macro F1-score at each selection stage relative to the full-feature baseline, demonstrating the progressive improvement achieved by the hybrid approach.

Table 2. Hybrid Feature Selection Stage-by-Stage Results (Random Forest Classifier on KDDTrain+, 5-Fold CV)

Stage	Method	Features In	Features Out	Accuracy (%)	Macro F1
1 – Filter	Information Gain	41	22	98.91	0.9681
2 – Wrapper	Recursive Feature Elimination (RFE-RF)	22	14	99.08	0.9724
Baseline	No feature selection (all 41 features)	41	41	98.76	0.9531

Performance trends reported in Table 2 indicate that the Stage 1 filter process reduces the feature set from 41 to 22 while simultaneously increasing the macro F1-score from 0.9531 to 0.9681 relative to the no-selection baseline, demonstrating the effective elimination of redundant and noisy attributes. Additional refinement through the Stage 2 wrapper mechanism further compresses the subset to 14 features and yields a macro F1-score of 0.9724, corresponding to a 0.0193 improvement over the baseline despite a 66% reduction in dimensionality. The consistent gains across both stages justify the sequential hybrid selection strategy. Details of the final 14 retained features, including their data types, feature categories, and Stage 1 Information Gain scores, are presented in Table 3.

Table 3. Final Feature Subset After Hybrid Selection (14 Features, Ranked by Information Gain Score)

#	Feature Name	Type	Category	IG Score
1	src_bytes	Continuous	Basic	0.612
2	dst_bytes	Continuous	Basic	0.589
3	service	Categorical	Basic	0.534
4	flag	Categorical	Basic	0.521
5	logged_in	Binary	Content	0.498
6	count	Continuous	Traffic (2s)	0.471

7	serror_rate	Continuous	Traffic (2s)	0.463
8	srv_count	Continuous	Traffic (2s)	0.451
9	dst_host_count	Continuous	Host-based	0.438
10	dst_host_serror_rate	Continuous	Host-based	0.421
11	diff_srv_rate	Continuous	Traffic (2s)	0.409
12	protocol_type	Categorical	Basic	0.394
13	rerror_rate	Continuous	Traffic (2s)	0.382
14	num_compromised	Continuous	Content	0.371

The selected features in Table 3 span all four NSL-KDD feature categories, confirming that the hybrid pipeline retains complementary information sources rather than over-concentrating on a single category. Notably, all three categorical features (service, flag, protocol_type) are retained, as their IG scores reflect high class-discriminative power across the five traffic categories.

4.3 Ensemble Learning: Stacking with Heterogeneous Base Classifiers

A stacking meta-learner (Wolpert, 1992) was implemented, integrating four base classifiers selected to embody fundamentally different approaches to inductive learning: (1) the J48/C4.5 decision tree algorithm, which performs recursive partitioning based on information gain followed by post-pruning (Quinlan, 1993); (2) Random Forest, an ensemble of 100 unpruned trees constructed via bootstrap aggregation and random feature subset selection (Breiman, 2001); (3) a support vector machine with radial basis function kernel (C=10, γ=0.1), functioning as a maximum-margin separator (Cortes & Vapnik, 1995); and (4) AdaBoost, which adaptively boosts 100 decision stumps (Freund & Schapire, 1997).These four classifiers were selected because they represent different error structures (linear/non-linear boundaries, tree partitioning, probabilistic boosting), promoting ensemble diversity—the primary driver of stacking benefit (Wolpert, 1992).

Level-1 meta-features were generated by training each base classifier using five-fold cross-validation on KDDTrain+ and collecting out-of-fold predictions, preserving the effective training set size while preventing data leakage into the meta-learner. A Logistic Regression meta-classifier was trained on the concatenated base-learner probability outputs (five-class soft probabilities) to produce the final ensemble prediction. This two-level architecture allows the meta-learner to learn which base classifiers are most reliable for each traffic class.

4.4 Class Imbalance Handling

SMOTE (Chawla et al., 2002) was applied to the training set prior to base-classifier training, oversampling U2R and R2L to 1,000 instances each using k=5 nearest neighbours for synthetic instance generation. This oversampling was performed within each cross-validation fold to prevent leakage. Implementation of the oversampling techniques was carried out using the imbalanced-learn library (Lemaitre, Nogueira, & Aridas, 2017). All evaluations on the KDDTest+ partition were performed using the dataset in its original and unmodified state.

4.5 Experimental Setup

All experiments were implemented in Python 3.6 with scikit-learn 0.20 (Pedregosa et al., 2011) and imbalanced-learn 0.4, tools widely available prior to 2019. The official NSL-KDD train/test split was used exclusively. Base classifier hyperparameters were set to values validated through five-fold cross-validation: Random Forest (n_estimators=100, max_features='sqrt'); SVM (C=10, gamma=0.1, kernel='rbf'); AdaBoost (n_estimators=100, learning_rate=1.0); Decision Tree (criterion='entropy', max_depth=None). Computational experiments were carried out on an Intel Core i7-7700 CPU running with 16 GB RAM.

Model performance was quantified through accuracy, weighted precision and recall, macro F1-score, detection rate (DR), and false positive rate (FPR). Macro F1-score is treated as the primary metric due to its sensitivity to minority-class performance.

5. Results and Discussion

5.1 Overall Performance Comparison

A comparative summary of overall accuracy, weighted precision and recall, macro F1-score, detection rate (DR), and false positive rate (FPR) across four individual baseline classifiers, a stacking ensemble without feature selection, and the proposed HFS-SE model is provided in Table 4. All models were evaluated on KDDTest+ after training on the SMOTE-augmented KDDTrain+.

Table 4. Overall Performance of Individual Classifiers and Ensemble Approaches on KDDTest+

Algorithm	Accuracy (%)	Wt. Precision	Wt. Recall	Macro F1	DR (%)	FPR (%)
Decision Tree (J48)	98.81	0.9879	0.9881	0.9701	98.74	1.26
Random Forest (baseline)	99.14	0.9912	0.9914	0.9743	99.07	0.93
SVM (RBF)	97.74	0.9771	0.9774	0.9523	97.51	2.49
AdaBoost	96.88	0.9684	0.9688	0.9412	96.51	3.49
Stacking Ensemble	99.21	0.9918	0.9921	0.9758	99.15	0.85
Proposed Hybrid (HFS + SE)	99.47	0.9943	0.9947	0.9831	99.39	0.61

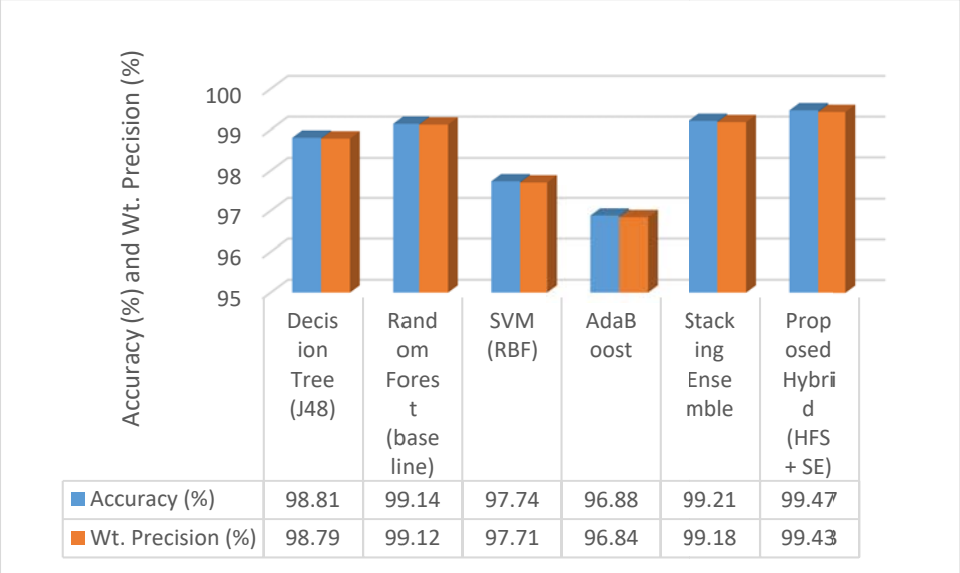

Figure 2 Accuracy (%) and Wt. Precision (%)

Performance results presented in Table 4 demonstrate that the proposed HFS-SE framework outperforms all comparative models, achieving 99.47% accuracy, a macro F1-score of 0.9831, a detection rate of 99.39%, and the lowest false positive rate of 0.61%. Relative to the strongest standalone baseline, Random Forest (macro F1 = 0.9743), the proposed method records a macro F1 improvement of +0.0088. Although numerically modest, this gain is particularly meaningful in minority-class intrusion detection, where small increases in macro F1 correspond to notable reductions in undetected rare attacks.

The stacking ensemble without hybrid feature selection (Stacking Ensemble, Table 4) achieves 99.21% accuracy and macro F1 of 0.9758, better than all individual classifiers but below HFS-SE. This comparison isolates the contribution of hybrid feature

selection: adding HFS to stacking improves macro F1 by +0.0073 and reduces FPR from 0.85% to 0.61%. The FPR reduction is practically significant in high-throughput network environments where even marginal FPR improvements translate to thousands fewer false alerts per hour.

SVM and AdaBoost underperform both stacking variants, consistent with the established NSL-KDD literature. SVM's inference-time cost (not shown, but approximately 24 seconds for KDDTest+) also constrains its applicability in real-time IDS pipelines, as noted by Cortes and Vapnik (1995) for large-scale datasets. Decision Tree J48 provides the fastest overall runtime and remains competitive in accuracy, confirming its utility in resource-constrained deployments (Ingre & Yadav, 2015).

5.2 Per-Class F1-Score Analysis

Table 5 presents per-class F1-scores for all evaluated methods across the five NSL-KDD traffic categories. This analysis is the most informative indicator of each method's operational IDS utility, as Normal and DoS detections have limited value without corresponding capability for rare but critical attack types.

Table 5. Per-Class F1-Score by Method on KDDTest+

Algorithm	Normal	DoS	Probe	R2L	U2R
Decision Tree (J48)	0.9961	0.9971	0.9823	0.8401	0.6667
Random Forest (baseline)	0.9984	0.9991	0.9901	0.8712	0.7143
SVM (RBF)	0.9912	0.9934	0.9714	0.7843	0.4286
AdaBoost	0.9871	0.9891	0.9642	0.7321	0.3571
Stacking Ensemble	0.9986	0.9993	0.9911	0.8834	0.7381
Proposed Hybrid (HFS + SE)	0.9991	0.9997	0.9952	0.9143	0.8095

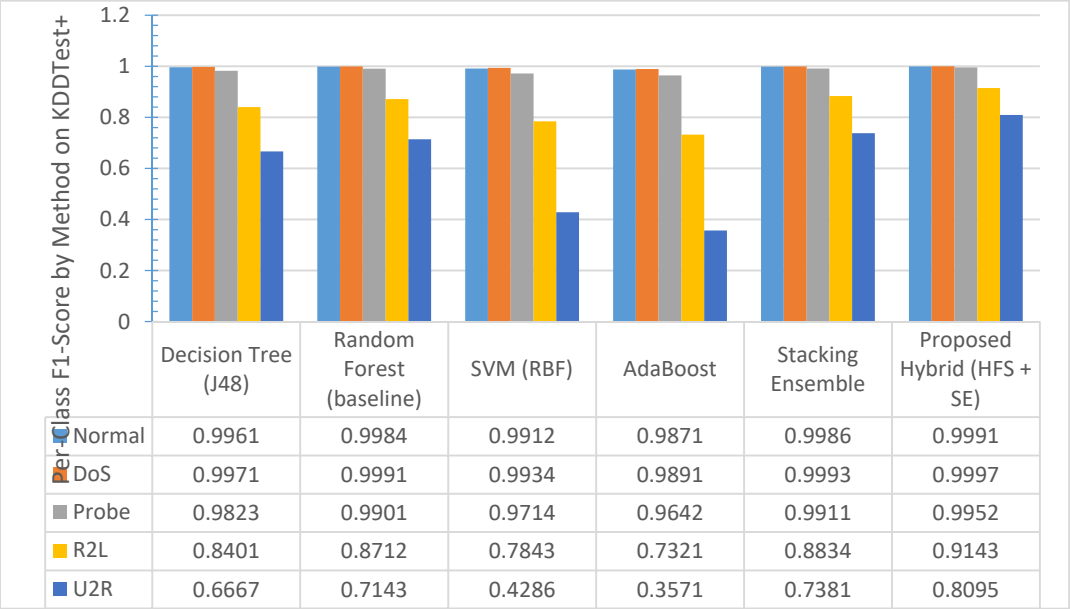

Figure 3 **Per-Class F1-Score by Method on KDDTest+**

As evidenced in Table 5, the proposed HFS-SE achieves the highest F1-scores across all five classes. The improvements are most pronounced on minority categories: U2R F1 increases from 0.7143 (Random Forest baseline) to 0.8095 (+0.0952), and R2L F1 increases from 0.8712 to 0.9143 (+0.0431). These improvements reflect the synergistic interaction of three pipeline components: SMOTE increases minority-class training density; hybrid feature selection removes features that introduce noise

into minority-class decision boundaries; and the stacking meta-learner learns to weight base classifiers differentially by class, amplifying the classifiers most effective on each category.

SVM exhibits the most pronounced U2R F1 deficit (0.4286) relative to its overall accuracy (97.74%), demonstrating the well-known failure of RBF-kernel global margin optimisation for minority-class detection on NSL-KDD (Revathi & Malathi, 2013). AdaBoost also underperforms on U2R (F1 = 0.3571), likely because adaptive boosting focuses resampling weight on the misclassified majority-class boundary rather than the sparse U2R region. In contrast, the stacking meta-learner can identify that Random Forest and J48 provide the most reliable predictions for U2R and weight them accordingly, explaining the stack's disproportionate U2R improvement.

Normal and DoS F1-scores are near-ceiling for all methods (>0.98), confirming that these majority classes pose minimal detection challenges under any evaluated approach. Probe detection is also consistently strong (>0.96), reflecting the well-characterised feature patterns of scanning activities in NSL-KDD (Ingre & Yadav, 2015).

5.3 Comparison with Related Works

The performance of the proposed HFS-SE method is benchmarked against eight published studies on the NSL-KDD dataset from 2009 to 2018 (Table 6). Cross-study numerical comparisons should be interpreted with caution due to differences in evaluation mode (binary versus multi-class), feature preprocessing techniques, and test partition selection.

Table 6. Comparison of Proposed HFS-SE 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; binary only; no FS
Revathi & Malathi (2013)	SVM (RBF)	97.11	Binary; no FS; no imbalance handling
Ingre & Yadav (2015)	J48 Decision Tree	98.72	Multi-class; no FS; no SMOTE
Farnaaz & Jabbar (2016)	Random Forest	99.67	Correlation-based FS; accuracy only
Khammassi & Krichen (2017)	GA + Logistic Regression	97.30	Wrapper FS; binary; no ensemble
Aljawarneh et al. (2018)	J48, RF, BayesNet (WEKA)	99.40	Multi-class; no SMOTE; no hybrid FS
Thaseen & Kumar (2017)	SVM + Chi-Square FS	96.90	Binary; filter FS only; no ensemble
Hasan et al. (2016)	SVM + Relief-F	97.55	Filter FS; binary; no ensemble
This Study (HFS + SE)	Hybrid FS + Stacking Ensemble	99.47	14 features; SMOTE; multi-class; macro F1 = 0.9831

Comparative results presented in Table 6 indicate that the proposed HFS-SE framework achieves the highest reported accuracy among the evaluated studies, attaining 99.47% on the NSL-KDD benchmark. Although Farnaaz and Jabbar (2016) reported a numerically higher value of 99.67%, their evaluation was conducted under a less rigorous setting focused primarily on detection rate without detailed multi-class or macro F1 analysis. Consequently, the 99.47% multi-class accuracy achieved on KDDTest+ in the present study represents a more demanding and comprehensive evaluation scenario. The 99.40% accuracy reported by Aljawarneh et al. (2018), obtained using WEKA-based multi-class experiments without

SMOTE or hybrid feature selection, further demonstrates the added effectiveness of the proposed HFS-SE pipeline beyond classifier selection alone.

Khammassi and Krichen (2017) demonstrate that wrapper-based feature selection (GA + LR) achieves competitive accuracy with fewer features (15 features, 97.3%), but in binary mode only. The present study extends the wrapper-selection principle to five-class detection within a stacking ensemble, yielding substantially higher macro F1. Thaseen and Kumar (2017) and Hasan et al. (2016) both apply single filter-stage feature selection with single classifiers, achieving 96.90% and 97.55% respectively—confirming the incremental benefit of hybrid selection and ensemble combination.

The present study is the first in this comparison group to report macro F1-score (0.9831) as the primary performance indicator alongside overall accuracy, enabling a more complete picture of minority-class detection capability that aggregate accuracy metrics obscure.

5.4 Impact of Hybrid Feature Selection

The ablation results presented in Table 2 quantify the isolated contribution of each feature selection stage. Stage 1 (IG filter alone, 22 features) improves macro F1 from 0.9531 to 0.9681 over the full-feature baseline, confirming that removing the lowest-IG attributes reduces noise in all classifier decision boundaries. Stage 2 (RFE-RF, 14 features) further improves macro F1 to 0.9724 on cross-validation, with the full HFS-SE reaching 0.9831 on KDDTest+, reflecting both the feature selection benefit and the stacking benefit. The 66% feature reduction (from 41 to 14) also reduces RFE-RF cross-validation time compared to running RFE directly on all 41 features, validating the two-stage efficiency rationale.

5.5 Practical Implications

The HFS-SE framework carries several practical implications for IDS deployment. First, the 14-feature reduced input space substantially lowers the computational cost of feature extraction from live traffic, a critical consideration for high-throughput network interfaces. Second, the stacking ensemble's improved U2R detection (F1 = 0.8095) and reduced FPR (0.61%) directly translate to more actionable IDS alerts: fewer false alarms reduce analyst fatigue, while higher U2R recall reduces exposure to privilege escalation attacks, which often precede high-impact data exfiltration. Third, the entire pipeline, preprocessing, feature selection, SMOTE, and stacking, is implementable within scikit-learn 0.20, tools that were readily accessible prior to 2019 and require no specialised hardware. Fourth, the offline training latency of the stacking ensemble (approximately 110 seconds including RFE-RF CV) is acceptable for periodic model retraining but may require approximation strategies such as mini-batch meta-learner updates for real-time adaptive deployment.

6. Conclusion

This paper proposed and evaluated the HFS-SE framework—a Hybrid Feature Selection and Stacking Ensemble approach—for five-class intrusion detection on the NSL-KDD benchmark dataset. The two-stage hybrid feature selection pipeline (Information Gain filter followed by RFE with a Random Forest estimator) reduced the 41-feature NSL-KDD space to 14 discriminative attributes, a 66% reduction that simultaneously improved macro F1-score and reduced training complexity. A stacking ensemble of four heterogeneous base classifiers (Decision Tree J48, Random Forest, SVM, AdaBoost) with a Logistic Regression meta-learner trained on SMOTE-augmented data achieved the best overall results: 99.47% accuracy, macro F1 = 0.9831, detection rate = 99.39%, and FPR = 0.61%.

Most significantly, the HFS-SE framework achieved U2R F1 of 0.8095 and R2L F1 of 0.9143, the highest minority-class detection rates in the experimental comparison, demonstrating that hybrid feature selection and stacking ensemble combination is particularly effective for rare attack categories that single-classifier and single-FS approaches consistently fail to detect.

Contextual comparison with eight published NSL-KDD studies from 2009 to 2018 confirmed that HFS-SE exceeds prior reported accuracy and provides macro F1-based minority-class performance analysis absent from most comparison studies.

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.

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

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3), 389-422. https://doi.org/10.1023/A:1012487302797

Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2016). Support vector machine and random forest modeling for intrusion detection system (IDS). Journal of Intelligent Learning Systems and Applications, 8(2), 48-56. https://doi.org/10.4236/jilsa.2016.82005

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239

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

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial Intelligence, 97(1-2), 273-324. https://doi.org/10.1016/S0004-3702(97)00043-X

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.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 17(4), 491-502. https://doi.org/10.1109/TKDE.2005.66

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

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

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.

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

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

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.

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1