

Hybrid Recursive Feature Elimination and Principal Component Analysis for Dimensionality Reduction in Network Intrusion Detection on the UNSW-NB15 Dataset

^{*1}Ayinde Ajibola Khadijat, ¹Issa Olawale Mustapha, ¹Saka Kamil Kayode, ^{1,2}Adefila Opeyemi Ademola and ¹Jimoh Lateefat Abiola

¹Department of Computer Science, Faculty of Computing, Engineering and Technology, Al-Hikmah University, Ilorin, Kwara State, Nigeria.

²Agricultural and Rural Management Training Institute (ARMTI), KM 18, Ilorin-Ajase-Ipo road, Kwara State, Nigeria.

*Corresponding authors' email: Ayindeajibola2@gmail.com

ABSTRACT

Machine-learning-based network intrusion detection is often limited by the high dimensionality of traffic records, whose redundant and correlated attributes increase computational cost. Recursive feature elimination (RFE) retains class-relevant attributes, whereas principal component analysis (PCA) decorrelates attributes without reference to class labels. This study evaluated a sequential pipeline in which RFE with a random-forest estimator retained 30 of the 42 attributes of the UNSW-NB15 dataset and PCA projected them onto 10 orthogonal components explaining 95.6% of their variance. The 175,341-record training partition of the dataset was split 70:30 by stratified sampling into 122,738 training and 52,603 testing records, and every transformer was fitted on the training records only. Four classifiers from different learning paradigms were evaluated with accuracy, precision, recall, F1-score, specificity, the area under the receiver operating characteristic curve (ROC-AUC) and the Matthews correlation coefficient (MCC), averaged over five runs, and compared with full-feature, RFE-only and PCA-only baselines. On the hybrid representation, the artificial neural network achieved the highest accuracy ($93.72 \pm 0.21\%$), ROC-AUC (0.982) and MCC (0.855), followed by k-nearest neighbour (93.41%; MCC 0.847), the support vector machine (91.56%; 0.806) and Gaussian naïve Bayes (86.82%; 0.698). All classifiers detected 90–99% of attacks but correctly identified only 76–87% of normal flows. Relative to the full attribute set, the hybrid representation reduced the input dimension by 76% with an MCC change between -0.041 and $+0.017$, and reduced k-nearest-neighbour testing time from 15.16 s to 0.73 s, but required 117.9 s of additional training for RFE and gave no test-time advantage over PCA alone. A web-based prototype supporting dataset upload, pre-processing, feature engineering, model training and prediction was also developed. The results indicate that the compact representation preserves detection performance while mainly reducing the cost of instance-based detection, and that the false-alarm rate and validation on the official testing partition must be addressed before deployment.

Received: 11 Oct. 2026

Accepted: 21 Oct. 2026

Published: 22 Oct. 2026

Keywords:

Feature Selection, Intrusion Detection, Machine Learning, Principal Component Analysis, Recursive Feature Elimination

INTRODUCTION

Cloud services, Internet of Things (IoT) networks and distributed enterprise networks have enlarged the attack surface available to adversaries, and intrusion detection systems (IDSs) remain a key component for identifying malicious network traffic. IDSs can be categorised as signature-based or anomaly-based: the former compares traffic with patterns derived from previously observed attacks, whereas the latter models normal traffic and flags deviations from it (Kasongo & Sun, 2020). Because anomaly-based detection can flag attacks that no stored signature describes, machine learning (ML) classifiers trained on labelled flow records have emerged as a popular approach to anomaly-oriented detection (Kasongo & Sun, 2020; Daoud et al., 2026).

Such classifiers are sensitive to the way traffic is represented. Each flow in benchmark datasets is described by dozens of attributes, and the performance of ML-based IDSs has been reported to decrease as the number of attributes increases, which suggests that attributes of low relevance should be pruned (Kasongo & Sun, 2020). UNSW-NB15 illustrates this

difficulty: the corpus is statistically more complex than the older KDD99 corpus (Moustafa & Slay, 2016), and published binary accuracies on it differ considerably depending on the classifier and feature subset used (Kasongo & Sun, 2020; Ghani et al., 2023). Dimensionality also has operational consequences, as the number of retained features has been shown to influence both training time and per-sample inference time (Daoud et al., 2026).

The literature distinguishes two families of dimensionality reduction. Feature selection retains a subset of the original attributes; recursive feature elimination (RFE), a wrapper method, repeatedly fits an estimator and removes the attribute least important to its decision function (Guyon et al., 2002). Feature extraction, in contrast, constructs new variables; principal component analysis (PCA) produces variables that are linear combinations of the original features and are mutually uncorrelated (Jolliffe & Cadima, 2016). Both methods have weaknesses. RFE uses the class labels but does not penalise redundancy among the attributes it retains, whereas PCA removes linear correlation without reference to the class labels. On UNSW-NB15, selection has been

reported to give higher accuracies and shorter runtimes when a moderate number of features is retained, whereas extraction was reported to be more robust when the dimension is reduced to a very small number and less sensitive to the number of retained dimensions (Ngo et al., 2024).

These complementary behaviours have inspired hybrid pipelines. Information gain has been combined with PCA before an ensemble of support vector, instance-based and multilayer perceptron learners (Salo et al., 2019), and information gain with random forest importance has been used to reduce the search space before RFE refines the feature subset for a multilayer perceptron (Yin et al., 2023). Closest to the present work, Ghani et al. (2023) applied RFE followed by PCA and supplied the resulting components to individual classifiers and a hard-voting ensemble. However, their evaluation centred on accuracy-type results, did not include a probabilistic learner whose assumptions interact directly with decorrelated inputs, and did not quantify the computational effect of applying RFE before PCA. Furthermore, Arp et al. (2022) found information leakage between training and test data in many of the security-related ML studies they analysed, which makes an explicit report of where each transformation is fitted an important aspect of such work.

The aim of this study is therefore to evaluate, under a controlled and leakage-free protocol, the sequential RFE–PCA representation of UNSW-NB15 previously applied by Ghani et al. (2023). The RFE–PCA sequence itself is not claimed as new. The study differs from Ghani et al. (2023) in four ways: it adds a probabilistic Gaussian naïve Bayes (GNB) model to the margin-based support vector machine (SVM), instance-based k-nearest neighbour (KNN) and artificial neural network (ANN); it reports specificity, ROC-AUC and the Matthews correlation coefficient (MCC) with confusion matrices, confidence intervals and significance tests; it isolates the contribution of each stage by full-feature, RFE only and PCA only baselines; and it measures the runtime of every stage, including the fitting of RFE. There are two research questions that direct the work. The first question is how do classifiers with different inductive biases perform on the representation obtained by RFE–PCA, compared to the full representation, RFE-only, and PCA-only representations? Second, how much does it cost to train a sequential pipeline versus these alternatives and how much does it cost to infer from a flow? The study also introduces a prototype which is available as a web-service that exposes the pipeline for use in the operation. The rest of the paper is a review of the related work, a description of methodology, presentation and discussion of results, and a suggestion for future research.

Related Works

This section reviews studies that reduced the dimensionality of intrusion detection data before classification, first those

using a single reduction technique and then those combining selection and extraction. The studies discussed are summarised in Table 1.

Feature Reduction for Network Intrusion Detection

Kasongo and Sun (2020) applied XGBoost importance scores as a filter to reduce the 42 UNSW-NB15 attributes to 19 and then tested SVM, KNN, logistic regression, ANN and decision tree classifiers in both binary and multiclass modes. On the official test partition, the reduced vector improved the binary accuracy of the decision tree to 90.85%, whereas the accuracy of logistic regression declined (88.13%) and that of the radial basis function SVM changed little (61–62%). They also observed that validation accuracies obtained from a split of the training partition exceeded the test-partition accuracies, which implies that reported results are influenced by the evaluation protocol applied to this dataset. Ngo et al. (2024) compared correlation-based selection with PCA-based extraction on UNSW-NB15 and found that the better approach depended on the target dimensionality, with a decision tree performing best under selection and a multilayer perceptron under extraction. Outside network flows, Daoud et al. (2026) evaluated compact feature sets for malware-laden URL detection and found that compact consensus subsets achieved similar accuracy while substantially reducing both training and per-sample inference time, highlighting the need to consider efficiency alongside accuracy when selecting compact subsets.

Hybrid Selection and Extraction Pipelines

Salo et al. (2019) combined information gain feature selection with PCA on the ISCX 2012, NSL-KDD and Kyoto 2006+ datasets and classified the reduced data with an ensemble of SVM, instance-based and multilayer perceptron classifiers, reporting that the resulting representation outperformed earlier methods. Yin et al. (2023) first used information gain and random forest importance to narrow the candidate attributes and then applied RFE to retain 23 of the 42 attributes, improving the accuracy of a multilayer perceptron from 82.25% to 84.24%; however, F1-scores for minority classes remained low, which the authors attributed to insufficient training samples. Ghani et al. (2023) selected 20 attributes using random-forest-based RFE and applied PCA with 10, 12 or 15 components, accounting for approximately 92%, 95% and 99% of the variance, respectively. With 15 components, their deep neural network achieved an accuracy of 95.08% and the voting ensemble 94.34%, slightly below the 95.85% obtained by the voting ensemble with all components. The reduced representation therefore largely retained, rather than exceeded, the accuracy obtained with the full representation.

Table 1: Related Feature Reduction Studies for Intrusion Detection

Study	Reduction strategy	Classifiers	Dataset	Evaluation protocol	Reported observation
Salo et al. (2019)	Information gain, then PCA	Ensemble of SVM, instance-based learner and multilayer perceptron	ISCX 2012; NSL-KDD; Kyoto 2006+	Not UNSW-NB15	Hybrid representation compared favourably with earlier methods
Kasongo and Sun (2020)	XGBoost importance filter (42 → 19)	SVM, KNN, logistic regression, ANN, decision tree	UNSW-NB15	Official training and testing partitions	Gains were classifier-dependent; decision tree improved, logistic regression declined
Yin et al. (2023)	Information gain and random forest filter, then RFE (42 → 23)	Multilayer perceptron	UNSW-NB15	Official training and testing partitions	Multiclass accuracy rose; minority attack classes remained weakly detected
Ghani et al. (2023)	RFE (42 → 20), then PCA (10–15 components)	SVC, KNN, deep neural network, voting ensemble	UNSW-NB15	Random 70:30 split	Reduced representation retained, rather than exceeded, full-feature accuracy

Study	Reduction strategy	Classifiers	Dataset	Evaluation protocol	Reported observation
Ngo et al. (2024)	Correlation-based selection versus PCA extraction	Several classifiers including decision tree and multilayer perceptron	UNSW-NB15	Official training and testing partitions	Preferred strategy depended on the number of retained dimensions
Daoud et al. (2026)	Harris hawks and bat algorithm subsets, union and intersection	Random forest, KNN, decision tree, extra trees	ISCX-URL2016	Not UNSW-NB15; stratified five-fold CV	Compact subsets reduced training and inference time with small accuracy loss

Note. Results obtained on the official UNSW-NB15 testing partition are not directly comparable with those obtained from random splits of the training partition, which tend to be higher (Kasongo & Sun, 2020).

Research Gap

Three observations emerge from this body of work. First, feature reduction may not benefit all classifiers equally (Kasongo & Sun, 2020; Ngo et al., 2024); a hybrid representation should therefore be examined across learners whose assumptions differ. Second, the previous RFE-PCA study on UNSW-NB15 did not examine a probabilistic classifier or imbalance-robust measures such as the MCC, which is high only when a classifier performs well on all four cells of the confusion matrix (Chicco & Jurman, 2020). Third, computational cost is seldom reported alongside detection quality in hybrid pipelines, although it is crucial for

deployment (Daoud et al., 2026). The present study addresses these three issues.

MATERIALS AND METHODS

The study followed the framework shown in Figure 1. The flow records were pre-processed and partitioned, RFE and PCA were then applied to create a compact representation, four classifiers were trained on this representation, and performance and runtime were measured on held-out records. The trained models were then embedded in a web-based prototype.

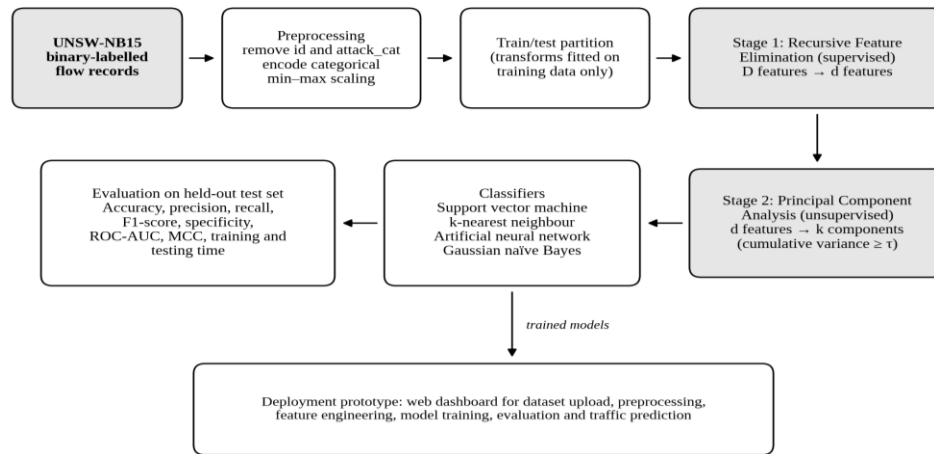


Figure 1: Proposed Sequential RFE-PCA Intrusion Detection Framework

Dataset and Preprocessing

The UNSW-NB15 dataset was used in this study. It was created in a cyber-range by combining real normal traffic with synthetic contemporary attacks (Moustafa & Slay, 2015) and is available as a training partition of 175,341 records and testing partition of 82,332 records, which contain normal traffic and nine attack types (Moustafa & Slay, 2016). The training partition was used and no other partitions. It contains 56,000 normal and 119,341 attack records and was split in a 70:30 ratio by stratified random sampling: 70% (122,738 records; 39,200 normal and 83,538 attack) formed the training set and 30% (52,603 records; 16,800 normal and 35,803 attack) formed the testing set. In both sets, class proportions were preserved using stratification. The official testing partition was not used for training, tuning or testing. Four classifiers from different learning paradigms were trained on the training set, and their performance was evaluated once on the testing set.

Each record was marked as "normal" (0) or "attack" (1). The record identifier was dropped, since it contains no traffic information; the multi-class attack-category was dropped, because it explicitly encodes the target; and $D = 42$ attributes (39 numeric and 3 categorical). The categorical attributes proto (133 values), service (13) and state (9) were label encoded. Label encoding is an arbitrary ordering of categories. It was retained because a random-forest estimator, used by RFE, splits based on thresholds and can isolate any

category with successive splits and because one-hot encoding would lead to an increase in the number of variables in the input from 42 to 192, its effect was tested by repeating the pipeline with one-hot encoding (Table 7). Min-max normalisation (Equation 1) was then applied to each attribute. The parameters fitted for the encoder, scaler and RFE were not changed when applied to the testing set; that is, no testing record affected any of the fitted parameters (Arp et al., 2022).

$$x'_{ij} = \frac{x_{ij} - \min_i(x_j)}{\max_i(x_j) - \min_i(x_j)} \quad (1)$$

Where x_{ij} is the value of attribute j for record i , x'_{ij} is its scaled value, and $\min_i(x_j)$ and $\max_i(x_j)$ are the minimum and maximum of attribute j over the training records. Testing values outside the training range were not clipped.

Stage 1: Supervised Filtering with Recursive Feature Elimination

Let $X \in \mathbb{R}^{N \times D}$ denote the preprocessed training matrix of N records and $D = 42$ attributes, and $y \in \{0, 1\}^N$ the corresponding labels. RFE initializes the active attribute set as $S_0 = \{1, 2, \dots, D\}$. At iteration t , a random forest of $B = 100$ trees is fitted to the training records restricted to S_t , and each active attribute i receives the ranking criterion c_i in Equation 2, its mean decrease in Gini impurity, where $\Delta G_{i,b}$ is the total impurity decrease produced by splits on attribute i in tree b . Guyon et al. (2002) used the squared weight w_i^2 because linear SVM weights can be negative; random-forest

importances are non-negative, so squaring would not change the ranking and is not needed. The least useful attribute is identified by Equation 3, and the active set is updated by Equation 4. The procedure repeats until $|S_t| = d = 30$, producing the filtered matrix $X_{RFE} \in \mathbb{R}^{N \times d}$. For seed 42, the twelve eliminated attributes were, in order of removal, is_ftp_login, ct_ftp_cmd, dwin, trans_depth, response_body_len, ct_flw_http_mthd, dloss, ct_src_dport_ltm, swin, service, ct_dst_sport_ltm and is_sm_ips_ports.

$$c_t^{(t)} = \frac{1}{B} \sum_{b=1}^B \Delta G_{t,b}^{(t)} \quad (2)$$

$$i^* = \arg \min_{i \in S_t} c_t \quad (3)$$

$$S_{t+1} = S_t \setminus \{i^*\} \quad (4)$$

Because elimination is guided by the labels, attributes that help separate attack from normal traffic are retained even when their variance is small. However, since each attribute is scored individually, several correlated attributes that are each informative can survive together, which motivates the second stage.

Stage 2: Decorrelation with Principal Component Analysis

The columns of X_{RFE} are mean-centred to form Z (Equation 5), where μ is the vector of training column means and $\mathbf{1}_N$ is a vector of N ones. The sample covariance matrix Σ (Equation 6) is eigendecomposed as in Equation 7, with eigenvalues ordered $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$. The number of retained components, k , is the smallest value for which the cumulative explained variance ratio reaches the threshold τ (Equation 8), and the hybrid representation is obtained by projecting onto the leading eigenvectors V_k (Equation 9). The components thus obtained are mutually uncorrelated (Jolliffe & Cadima, 2016). The mean vector and eigenvectors were estimated on the training data and then applied unchanged to the test data, as recommended to prevent data snooping (Arp et al., 2022). The two stages of the hybrid feature reduction are formulated as follows:

$$Z = X_{RFE} - \mathbf{1}_N \mu^T \quad (5)$$

$$\Sigma = \frac{1}{N-1} Z^T Z \quad (6)$$

$$\Sigma v_j = \lambda_j v_j, \quad j = 1, \dots, d \quad (7)$$

$$CEVR(k) = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^d \lambda_j} \geq \tau \quad (8)$$

$$X_{hybrid} = Z V_k, \quad V_k = [v_1, \dots, v_k] \quad (9)$$

Hyperparameter Selection and Runtime Protocol

The number of attributes retained by RFE, d , was chosen by five-fold stratified cross-validation on the training set of the seed-42 split only, and the chosen d and k were then kept fixed for all five runs. For each fold, the encoder, scaler and RFE were fitted on the training folds and a random-forest classifier

with the top d attributes was tested on the validation fold for various candidate values of $d \in \{5, 10, 15, 20, 23, 25, 30, 35, 42\}$. Mean validation MCC rose from 0.864 ± 0.004 at $d = 5$ to a maximum of 0.903 ± 0.004 at $d = 35$. The smallest d for which the mean MCC was within one standard deviation of the best was chosen, which resulted in $d = 30$ (0.900 ± 0.003). Then the number of principal components, k , was the minimum number of components required to explain $\tau = 0.95$ (95.58%) of the variance in the training data (Equation 8). For a fair comparison, PCA was also limited to 10 components. Classifier hyperparameters (Table 2) were not using the testing set to tune the parameters, and were not tuned beforehand.

Runtime was measured with Python's time.perf_counter on the machine described in Table 2, with no other experiments running. The time for each stage was measured individually: fitting the encoder and scaler, fitting RFE (random forests fit 12 random times, 42 to 30 attributes), fitting PCA, fitting each classifier, transforming the testing set and predicting. End-to-end training time: sum of the fitting stages; End-to-end testing time: sum of the test-time stages. Single-record latency was tested with 1,000 randomly sampled test records, each one of which was classified, with preprocessing. This experiment was repeated with five random seeds which resulted in a 70:30 split, different random forests in RFE, and different initializations for the SVM and ANN. For the seed-42 run, 95% confidence intervals were obtained from 1,000 bootstrap resamples of the testing set (Efron & Tibshirani, 1994), and classifiers were compared pairwise with McNemar's test (McNemar, 1947).

Classifiers and Experimental Configuration

Four classifiers were chosen to represent different inductive biases. The SVM builds a linear decision surface that is maximally wide. Performance of KNN relies directly on distances in the component space, using the most common label of the five nearest training neighbours. The ANN consists of 10 input units (one for each principal component), 64 and 32 units of hidden layers with rectified linear unit (ReLU) activation and logistic output unit. It was trained with the Adam optimiser (learning rate 0.001, batch size 256) to minimise binary cross-entropy for up to 50 epochs, with early stopping when the score on a 10% validation subset of the training set did not improve for five epochs. The GNB models each component with a class-conditional Gaussian likelihood under an assumption of conditional independence, which makes it a useful test of whether decorrelated inputs suit a probabilistic learner. All classifiers used the same representation, X_{hybrid} , and the same classifiers and settings were used for the full-feature, RFE-only and PCA-only baselines. Table 2 represents the setting for the configuration.

Table 2: Experimental Configuration

Component	Setting
Dataset and task	UNSW-NB15 official training partition (175,341 records; 56,000 normal, 119,341 attack); binary classification (normal versus attack)
Partitioning	Stratified 70:30 split: 122,738 training records (39,200 normal, 83,538 attack) and 52,603 testing records (16,800 normal, 35,803 attack); official testing partition not used
Original attributes	$D = 42$ (39 numeric, 3 categorical) after removing the record identifier and attack-category field
Categorical encoding	proto, service and state; label encoding (one-hot as a sensitivity check, Table 7)
Scaling	Min-max normalisation, fitted on training records only
RFE	Random forest, 100 trees, Gini importance; one attribute removed per iteration; $d = 30$, chosen by five-fold CV on the training set
PCA	Fitted on RFE output of training records; $k = 10$ components (cumulative explained variance $95.58\% \geq \tau = 0.95$); unwhitened
SVM	Linear kernel; $C = 0.11$
KNN	$k = 5$; Euclidean distance; uniform voting

Component	Setting
ANN	Hidden layers of 64 and 32 units, ReLU; logistic output; Adam, learning rate 0.001; batch size 256; up to 50 epochs with early stopping (patience 5)
GNB	Gaussian class-conditional likelihood; variance smoothing 1×10^{-9}
Repetitions	Five runs, seeds 42, 7, 21, 123 and 2026
Software	Python 3.13.16, scikit-learn 1.9.1, NumPy 2.5.3, pandas 3.0.5
Hardware	Intel Xeon processor at 2.1 GHz, 2 virtual cores, 7 GB RAM; Linux

Evaluation Metrics

Attack records were treated as the positive class, and TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively. Accuracy, precision, recall, specificity, F1-score and MCC (Equations 10–15) were used to quantify performance. Recall is the proportion of attacks correctly detected, specificity is the proportion of normal flows correctly identified as normal, and MCC summarises all four cells of the confusion matrix in a single coefficient that is less prone to inflation on imbalanced data than accuracy or F1-score (Chicco & Jurman, 2020). The area under the receiver operating characteristic curve (ROC-AUC) was computed from the classifier scores and provides a threshold-independent measure of each classifier's ranking ability. Runtime, confidence intervals and significance tests were obtained as described in the Hyperparameter Selection and Runtime Protocol subsection. The metrics are defined as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (12)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (13)$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (15)$$

RESULTS AND DISCUSSION

Feature Engineering Efficiency

The end-to-end times are reported as averages over 5 runs for ANN and KNN, and the SVM and GNB performed similar to ANN. The average and standard deviation of the time spent fitting RFE was 117.87 ± 1.95 s, as there are 12 iterations of fitting a 100-tree random forest to 122,738 records; while fitting PCA took 0.03 s. In contrast, the ANN was trained in 125.26 ± 2.65 s, while the PCA was trained in 8.24 ± 5.13 s and the full set of attributes was trained in 9.22 ± 2.23 s. This is a one-off cost, paid when the model is built or retrained. At test time, the benefit of dimensionality reduction depended on the classifier. For KNN, whose prediction cost grows with input dimension, classifying the 52,603 testing records took 15.16 ± 0.79 s with all 42 attributes but 0.73 ± 0.05 s with the 10 hybrid components, a 20.6-fold reduction. The testing time for every representation is less than 0.1 s ($\approx 1.7 \mu\text{s}$ per flow), for the ANN, SVM, and GNB. The testing time was not significantly different for the hybrid or PCA-only pipelines, since each generates 10 components. Single-records latency (3.7–5.0 ms) was mostly due to the preprocessing overhead and not to the classifier. These timings are measured offline only on one machine and have not been measured under live traffic.

Table 3: End-to-End Runtime of the Four Pipelines on UNSW-NB15 (Mean $\pm$ SD, Five Runs)

Classifier	Pipeline	Dims	Reduction fitting (s)	End-to-end training (s)	End-to-end testing (s)	Batch latency ($\mu\text{s}/\text{flow}$)	Single-record latency (ms)
ANN	Full features	42	–	9.22 ± 2.23	0.083 ± 0.012	1.6 ± 0.2	3.69 ± 0.47
ANN	RFE only	30	117.85 ± 1.95	125.23 ± 3.28	0.091 ± 0.012	1.7 ± 0.2	3.72 ± 0.10
ANN	PCA only	10	0.03 ± 0.00	8.24 ± 5.13	0.094 ± 0.023	1.8 ± 0.4	4.06 ± 0.73
ANN	RFE + PCA	10	117.87 ± 1.95	125.26 ± 2.65	0.088 ± 0.012	1.7 ± 0.2	3.90 ± 0.20
KNN	Full features	42	–	0.33 ± 0.04	15.156 ± 0.794	288.1 ± 15.1	7.21 ± 0.32
KNN	RFE only	30	117.85 ± 1.95	118.20 ± 1.98	13.187 ± 0.519	250.7 ± 9.9	8.16 ± 0.80
KNN	PCA only	10	0.03 ± 0.00	0.53 ± 0.04	0.727 ± 0.082	13.8 ± 1.6	4.54 ± 0.14
KNN	RFE + PCA	10	117.87 ± 1.95	118.37 ± 1.98	0.734 ± 0.047	13.9 ± 0.9	5.01 ± 0.39

Note. End-to-end training = encoder and scaler fitting + reduction fitting (every RFE iteration included) + classifier fitting. End-to-end testing = preprocessing + reduction transform + prediction for all 52,603 testing records. Single-record latency is the median over 1,000 records classified one at a time, including preprocessing.

Classification Performance

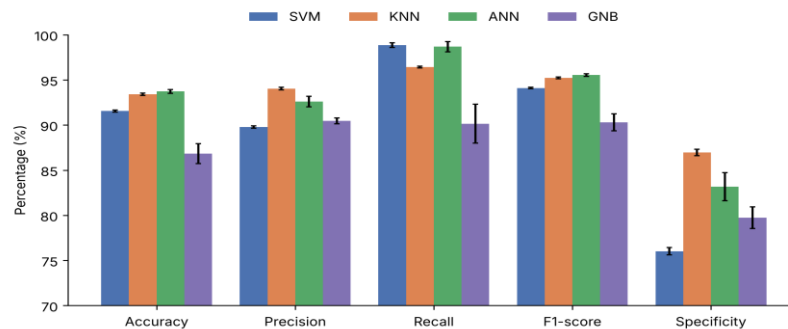
Table 4 and Figure 2 present the results of the four classifiers on the hybrid representation, averaged over five runs, and Table 5 gives the confusion matrices and 95% confidence intervals for the seed-42 run. The ANN achieved the highest accuracy ($93.72 \pm 0.21\%$), F1-score (95.54%), ROC-AUC (0.982) and MCC (0.855), followed by KNN (93.41%, MCC 0.847), the SVM (91.56%, 0.806) and GNB (86.82%, 0.698).

The results of the McNemar test indicated that each pairwise comparison was significant for the seed-42 run (all with $p < 0.001$), including the small difference between the ANN and KNN (1,239 vs. 1,033 discordant records, $p = 1.7 \times 10^{-5}$). The run-to-run standard deviations were small (≤ 0.21 percentage points of accuracy for the SVM, KNN and the ANN) while the GNB showed more variation (1.10). The ROC curves are presented in Figure 3.

Table 4: Performance of the Four Classifiers on the RFE-PCA Representation (Mean $\pm$ SD, Five Runs)

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)	ROC-AUC	MCC
SVM	91.56 $\pm$ 0.12		89.78 $\pm$ 0.13	94.10 $\pm$ 0.09	76.02 $\pm$ 0.39	0.894 $\pm$ 0.007	0.806 $\pm$ 0.003
KNN	93.41 $\pm$ 0.14	94.04 $\pm$ 0.15	96.43 $\pm$ 0.09	95.22 $\pm$ 0.10	86.96 $\pm$ 0.36	0.970 $\pm$ 0.001	0.847 $\pm$ 0.003
ANN	93.72 $\pm$ 0.21	92.60 $\pm$ 0.60	98.67 $\pm$ 0.58	95.54 $\pm$ 0.14	83.17 $\pm$ 1.56	0.982 $\pm$ 0.002	0.855 $\pm$ 0.005
GNB	86.82 $\pm$ 1.10	90.47 $\pm$ 0.33	90.15 $\pm$ 2.14	90.29 $\pm$ 0.95	79.74 $\pm$ 1.20	0.921 $\pm$ 0.004	0.698 $\pm$ 0.019

Note. Best value in each column in bold. Attack is the positive class. SVM = support vector machine; KNN = k-nearest neighbour; ANN = artificial neural network; GNB = Gaussian naïve Bayes; ROC-AUC = area under the receiver operating characteristic curve; MCC = Matthews correlation coefficient.

**Figure 2: Threshold-Based Performance of the Four Classifiers on the RFE-PCA Representation**

Note: Bars show means over five runs and error bars one standard deviation. The vertical axis begins at 70% to make between-classifier differences visible; differences should be judged from Table 4

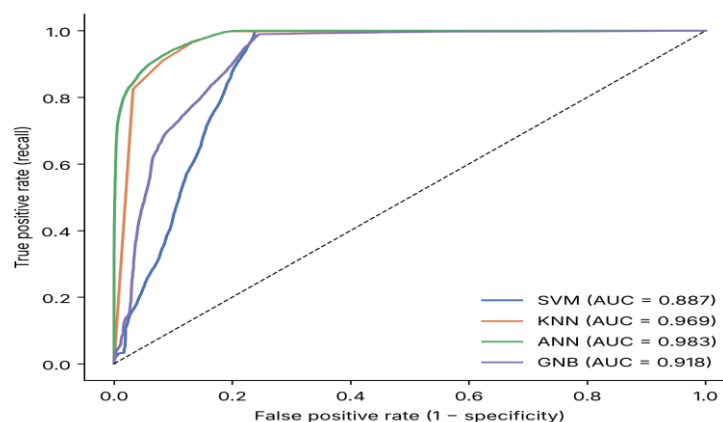
All four classifiers detected attacks far more reliably than they recognised normal traffic. Recall ranged from 90.15% (GNB) to 98.85% (SVM), whereas specificity ranged from 76.02% (SVM) to 86.96% (KNN). For the ANN in the seed-42 run, 3,030 of the 16,800 normal testing records (18.0%) were flagged as attacks, whereas only 267 of the 35,803 attacks (0.7%) were missed. The models are thus set up to raise false

alarms, then. It is typically more expensive to miss an attack, but this false alarm rate would take up a great deal of analyst time, and reducing it, for example by adjusting the decision threshold, is a priority before deployment. KNN offered the best balance between the two error types, with the highest specificity and the second-highest MCC.

Table 5: Confusion Matrices and 95% Confidence Intervals on the RFE-PCA Representation (Seed 42, n = 52,603)

Classifier	TN	FP	FN	TP	Accuracy 95% CI (%)	Specificity 95% CI (%)	MCC 95% CI
SVM	12,834	3,966	552	35,251	91.19–91.64	75.79–77.04	0.796–0.806
KNN	14,592	2,208	1,295	34,508	93.13–93.55	86.37–87.35	0.840–0.850
ANN	13,770	3,030	267	35,536	93.55–93.94	81.39–82.59	0.852–0.861
GNB	13,328	3,472	3,040	32,763	87.35–87.89	78.73–79.93	0.707–0.720

Note. Attack is the positive class. Normal support (TN + FP) = 16,800; attack support (FN + TP) = 35,803. Confidence intervals were obtained from 1,000 bootstrap resamples of the testing set.

**Figure 3: ROC Curves of the Four Classifiers on the RFE-PCA Representation (Seed 42)**

Note: The dashed diagonal indicates a random classifier; the area under each curve is given in the legend

The superiority of the ANN is consistent with Ghani et al. (2023), where deep neural network was the best individual learner on PCA components of RFE selected attributes, and the results of KNN outperforming SVM is consistent with their results. If the attacker can use interactions between the components that the linear SVM cannot detect, the learning system with low specificity (76.02%) can gain this advantage by the principal components being linear combinations of the attributes retained. KNN performed close to the ANN even though unwhitened principal components let the leading components dominate Euclidean distances. GNB was the weakest learner: PCA decorrelates the pooled training data, but decorrelation of pooled data does not imply independence within each class, and heavy-tailed flow attributes depart from the Gaussian likelihood that the model assumes. Its larger run-to-run variation (MCC SD 0.019) points the same way.

Contribution of Each Reduction Stage

Table 6 compares the four classifiers on the full attribute set, the 30 RFE-selected attributes, 10 principal components of all

attributes, and the 10-component hybrid representation, under identical splits and seeds. The input number of 42 attributes was reduced to 10 hybrid components, which resulted in a reduction of MCC by -0.041 for SVM, -0.010 for KNN and -0.014 for ANN, and an increase of MCC by 0.017 for GNB. The hybrid representation thus retained most of the information in the complete attribute set with only 24% of its dimensions, but failed to enhance detection of the three discriminative classifiers. For every classifier, the reduction in MCC by RFE was only slightly significant, while it was highly significant for GNB (MCC = 0.606 ± 0.060) due to the correlated attributes that RFE fails to remove in this case where the independence assumption is not met. PCA, on the other hand, increased MCC from 0.681 to 0.729 , matching the prediction of decorrelating inputs for a conditional independence model. Applying PCA after RFE rather than to all attributes changed MCC by at most 0.004 for the SVM, KNN and the ANN and lowered it by 0.031 for GNB, so on this dataset the supervised selection stage added little to PCA alone.

Table 6: Accuracy and MCC of Each Classifier on Four Representations (Mean $\pm$ SD, Five Runs)

Representation	Dims	Metric	SVM	KNN	ANN	GNB
Full features	42	Accuracy (%)	93.28 ± 0.05	93.83 ± 0.06	94.36 ± 0.12	86.13 ± 0.23
Full features	42	MCC	0.847 ± 0.001	0.857 ± 0.001	0.869 ± 0.003	0.681 ± 0.004
RFE only	30	Accuracy (%)	92.11 ± 0.09	93.71 ± 0.09	94.05 ± 0.19	79.90 ± 4.61
RFE only	30	MCC	0.818 ± 0.002	0.854 ± 0.002	0.862 ± 0.004	0.606 ± 0.060
PCA only	10	Accuracy (%)	91.81 ± 0.12	93.35 ± 0.06	93.83 ± 0.15	88.15 ± 0.23
PCA only	10	MCC	0.810 ± 0.003	0.846 ± 0.001	0.858 ± 0.003	0.729 ± 0.005
RFE + PCA	10	Accuracy (%)	91.56 ± 0.12	93.41 ± 0.14	93.72 ± 0.21	86.82 ± 1.10
RFE + PCA	10	MCC	0.806 ± 0.003	0.847 ± 0.003	0.855 ± 0.005	0.698 ± 0.019

To test whether the arbitrary order imposed by label encoding affected the results, the seed-42 run was repeated with one-hot encoding of proto, service and state, which expanded the input to 192 columns (Table 7). For the SVM, KNN and the ANN, the minimum change in MCC was 0.003 with the same run-to-run variation as seen in Table 4, using the same parameters as RFE, which retained its 30 columns and 10 PCA components. The one-hot columns were also sparse,

with 192 of them being binary, and their accuracy dropped to 48.02% for GNB (0.713 to 0.665), as many of them had a distribution which was not Gaussian. Label encoding was hence kept. One hot encoding also increased the time required for RFE fitting by approximately 119 times, from 118 s to 1,237 s: This is because 162 attributes had to be pruned compared to 12.

Table 7: MCC (Accuracy, %) on the RFE-PCA Representation with Label and One-Hot Encoding (Seed 42)

Encoding	Input columns	SVM	KNN	ANN	GNB
Label	42	0.801 (91.41)	0.845 (93.34)	0.856 (93.73)	0.713 (87.62)
One-hot	192	0.802 (91.49)	0.847 (93.41)	0.859 (93.84)	0.665 (84.95)

Comparison with Related Studies

Table 8 lists accuracies reported by studies that reduced the dimensionality of UNSW-NB15 before classification. The present accuracy of ANN achieved under the same protocol (random 70:30 split) was slightly less (93.72%) than Ghani et al. (2023), who reported an accuracy of 95.08% with their deep neural network (DNN) which employed standardisation, one-hot encoding, and 15 components. Kasongo and Sun

(2020) and Yin et al. (2023) evaluated on the official testing partition, which gives lower accuracy when compared to random partitions (Kasongo & Sun, 2020), and Yin et al. (2023) worked on a multiclass task which is not directly comparable. It is not that the pipeline improves on earlier work, but rather that it is comparable to the range reported in the comparison.

Table 8: Reported Accuracy of Feature Reduction Approaches on UNSW-NB15

Study	Reduction	Classifier	Task and partitioning	Accuracy (%)
Kasongo and Sun (2020)	XGBoost filter, 19 attributes	Decision tree	Binary; official test partition	90.85
Ghani et al. (2023)	RFE (20) then PCA (15)	Voting ensemble	Binary; 70:30 split	94.34
Ghani et al. (2023)	RFE (20) then PCA (15)	Deep neural network	Binary; 70:30 split	95.08
Yin et al. (2023)	Information gain and random forest filter, then RFE (23)	Multilayer perceptron	Multiclass; supplied partitions	84.24
This study	RFE (d = 30) then PCA (k = 10)	ANN	Binary; stratified 70:30 split of training partition	93.72

Practical Implications and Limitations

The trained pipeline was then incorporated into a web-based prototype using a Python web framework and an SQLite store that lets an analyst upload flow records, pre-process and feature engineer them, train and test models, and classify new flow traffic from a single dashboard. The major asset of this representation is when an instance-based classification technique is used and the testing time of KNN was reduced 20.6-fold (Table 3), whereas the others classifiers already use small prediction cost.

The findings are subject to several qualifications. The most important is that all results come from random splits of one partition of a single dataset with synthetic attack traffic; because random splits give higher accuracies than the official testing partition (Kasongo & Sun, 2020), performance on that partition and on other networks is likely to be lower. The false-alarm rate was high for every classifier (13–24% of normal flows flagged), and threshold adjustment was not explored. Only binary detection was evaluated and minority attack categories were not evaluated as they are hard to detect in UNSW-NB15 (Yin et al., 2023). The runtime has been measured offline on one machine, and the throughput under high rate traffic is yet to be determined. Hence, there is a need for a validation process on the official testing partition and a second dataset, false alarm reduction and streaming tests prior to deployment.

Conclusion

This study evaluated the sequential pipeline of Ghani et al. (2023), which selects class-relevant attributes of UNSW-NB15 using RFE and decorrelates them using PCA, with four learners of different inductive biases, under a leakage-free protocol in which every transformer was fitted on training records only and results were averaged over five runs. With respect to the first research question, the ANN achieved the highest accuracy (93.72%), ROC-AUC (0.982) and MCC (0.855), followed closely by KNN, then the SVM and GNB; all classifiers detected 90–99% of attacks but flagged 13–24% of normal flows as attacks. The ablation results demonstrated that neither reducing the 42 attributes to 10 hybrid components (PCA+RFE) caused a substantial change in MCC (from -0.041 to +0.017) nor did RFE have a significant contribution beyond PCA alone. As far as the second question is concerned, they added approximately 118 s of one-off training time, but with the same dimension gave no test-time improvement over PCA alone, and the primary improvement was a 20.6-fold decrease in KNN test time over the entire attribute set. These results imply that the representation will be useful primarily in instances where the instance-based detectors need to be deployed continuously, without establishing any benefit in deployment. Most important is that the evidence is the result of a random split of one synthetic dataset and is measured offline. Future work should include testing the pipeline on the official UNSW-NB15 test partition and at least one other dataset, calibrating the thresholds to reduce the false alarm rate, testing the pipeline in the live

setting to measure latency under traffic and the ability to extend the pipeline to multiclass detection, which could incorporate the handling of class imbalances, in order to map the component level decisions back to traffic attributes to enable interpretation by an analyst.

REFERENCES

- Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)* (pp. 3971–3988). USENIX Association.
<https://www.usenix.org/conference/usenixsecurity22/presentation/arp>
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), Article 6. <https://doi.org/10.1186/s12864-019-6413-7>
- Daoud, M. S., Abualhaj, M. M., Al-Khatib, S. S., Al-Zyoud, M., Abu-Shareha, A., Al-Aqrabi, H., Alsharaiah, M. A., & Anbar, M. (2026). A unified machine learning-based IDS/IPS framework with bio-inspired feature selection for real-time detection of malware-laden URLs. *Journal of Information and Communication Technology*, 25(3), 1–30. <https://doi.org/10.32890/jict2026.25.3.1>
- Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. Chapman and Hall/CRC.
<https://doi.org/10.1201/9780429246593>
- Ghani, H., Salekzamankhani, S., & Virdee, B. S. (2023). A hybrid dimensionality reduction for network intrusion detection. *Journal of Cybersecurity and Privacy*, 3(4), 830–843. <https://doi.org/10.3390/jcp3040037>
- 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>
- Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), Article 20150202. <https://doi.org/10.1098/rsta.2015.0202>
- Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, 7, Article 105. <https://doi.org/10.1186/s40537-020-00379-6>
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages.

- Psychometrika*, 12(2), 153–157. <https://doi.org/10.1007/BF02295996>
- Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/MilCIS.2015.7348942>
- Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1–3), 18–31. <https://doi.org/10.1080/19393555.2015.1125974>
- Ngo, V.-D., Vuong, T.-C., Van Luong, T., & Tran, H. (2024). Machine learning-based intrusion detection: Feature selection versus feature extraction. *Cluster Computing*, 27(3), 2365–2379. <https://doi.org/10.1007/s10586-023-04089-5>
- Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Computer Networks*, 148, 164–175. <https://doi.org/10.1016/j.comnet.2018.11.010>
- Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *Journal of Big Data*, 10, Article 15. <https://doi.org/10.1186/s40537-023-00694-8>

