Journal of Cybersecurity and Privacy

MDPI

Article

A Hybrid Dimensionality Reduction for Network Intrusion Detection

Humera Ghani *, Shahram Salekzamankhani and Bal Virdee

School of Computing and Digital Media, London Metropolitan University, London N7 8DB, UK
* Correspondence: hug0051@my.londonmet.ac.uk

Abstract: Due to the wide variety of network services, many different types of protocols exist, producing various packet features. Some features contain irrelevant and redundant information. The presence of such features increases computational complexity and decreases accuracy. Therefore, this research is designed to reduce the data dimensionality and improve the classification accuracy in the UNSW-NB15 dataset. It proposes a hybrid dimensionality reduction system that does feature selection (FS) and feature extraction (FE). FS was performed using the Recursive Feature Elimination (RFE) technique, while FE was accomplished by transforming the features into principal components. This combined scheme reduced a total of 41 input features into 15 components. The proposed systems' classification performance was determined using an ensemble of Support Vector Classifier (SVC), K-nearest Neighbor classifier (KNC), and Deep Neural Network classifier (DNN). The system was evaluated using accuracy, detection rate, false positive rate, f1-score, and area under the curve metrics. Comparing the voting ensemble results of the full feature set against the 15 principal components confirms that reduced and transformed features did not significantly decrease the classifier's performance. We achieved 94.34% accuracy, a 93.92% detection rate, a 5.23% false positive rate, a 94.32% f1-score, and a 94.34% area under the curve when 15 components were input to the voting ensemble classifier.

Keywords: network security; network traffic anomalies; intrusion detection; dimensionality reduction; principal component analysis; recursive feature elimination

check for updates

Citation: Ghani, H.; Salekzamankhani, S.; Virdee, B. A Hybrid Dimensionality Reduction for Network Intrusion Detection. *J. Cybersecur. Priv.* **2023**, *3*, 830–843. https://doi.org/10.3390/jcp3040037

Academic Editor: Panayiotis Kotzanikolaou

Received: 21 September 2023
Revised: 6 November 2023
Accepted: 10 November 2023
Published: 16 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Network traffic anomalies severely impact network services and users and can compromise network availability and operations [1]. Network intrusion detection systems are developed to counter this threat. The widespread application of machine learning techniques inspired computer network security researchers to apply machine learning techniques to solve network security issues [2,3]. In past years, various novel studies were produced to create network intrusion detection systems to identify traffic anomalies or intrusive packets using machine learning techniques [4].

Moustafa and Slay [5] pointed out that due to the wide range of network services, many types of protocols produce various packet features, some of which contain irrelevant and redundant information. Such features increase computational complexity and decrease accuracy [6]. Therefore, this research proposes a network intrusion detection system capable of reducing data dimensionality and accurately classifying normal and attack traffic. We will use the publicly available and widely used UNSW-NB15 dataset to conduct our experiment and employ feature selection and extraction techniques to reduce data dimensionality. We will perform traffic classification using multiple classifiers and feed their results to an ensemble classifier for receiving results based on a majority vote. Since both dimensionality reduction techniques are used, the proposed system is a hybrid dimensionality reduction system.

Two techniques for dimensionality reduction are feature selection and feature extraction [7]. Feature selection and feature extraction methods are widely employed in machine

and deep learning-based solutions to network intrusion detection problems [8–11]. Our research employed both techniques to reduce the data dimensionality.

In contemporary research, the following machine and deep learning techniques have been used extensively for detecting network traffic anomalies: Decision Tree [12], XGBoost [13], Support Vector Machine [14], Deep Neural Network [15], and Convolution Neural Network [16]. In addition, Ahmad et al. [17] mentioned that many researchers use ensemble techniques to achieve optimized performance in the network intrusion detection domain. We took advantage of the power of the Ensemble learner after exhaustively removing unuseful features. Our work classifies normal and attack traffic. To the best of the author's knowledge this combination of algorithms for feature selection, feature extraction, and classification has not been used before.

In this study, we propose a hybrid dimensionality reduction module that harnesses the power of feature selection and feature extraction approaches to reduce the input feature space since unuseful features increase computational complexity and decrease accuracy. Usha and Anuradha [18] compared three feature selection techniques: mutual information gain, the chai-square method, and recursive feature elimination. They reported that recursive feature selection selects optimal features compared to other techniques. Therefore, we used the Recursive Feature Elimination method in this research. This method selects the twenty most useful features from the input feature space. These selected features are given to a feature extractor that transforms them into components using the Principal Component Analysis technique. We chose Principal Component Analysis (PCA) as it is faster and since the first few principal components are computed and more interpretable than other techniques, such as Auto-encoder. We used an Ensemble learner for the classification task as it performs better than individual learners in most cases [19]. It does hard voting on individual predictions of SVC, KNC, and DNN classifiers and outputs the majority vote. We chose individual models from the machine and deep learning domains. We aimed to apply diverse models to our dataset instead of similar types like different tree-based algorithms. SVC is used because it can classify linearly and non-linearly separable data points. KNC is chosen because it clusters similar data points. Our data has different attack classes; therefore, it is anticipated we will achieve good evaluation results using it. DNN is also employed due to its ability to successfully classify non-linearly separable data points.

Classification is performed on the full dataset, twenty top-ranked features, and 10, 12, and 15 principal components. The evaluation results reveal the effectiveness of the proposed technique. Compared with a full feature set, the classification accuracy does not significantly impact when a reduced feature set is used. The main contributions of this paper are the following:

- A hybrid dimensionality reduction module is developed that performs its job in two phases. The first phase is feature selection, which selects the valuable features based on their rank using the Recursive Feature Elimination method. The second phase is feature extraction, where selected features are transformed into principal components by applying the Principal Component Analysis (PCA) technique.
- Classification is performed using SVC, KNC, and DNN to classify attack and normal traffic. The result of these classifiers is fed into the Ensemble learner, which produces final predictions based on the majority vote.
- The proposed research is compared with other contemporary research where the same dataset is used, and dimensionality reduction is performed before giving the input to the classifier. The experiment's results show that our scheme has better classification accuracy.

The remaining part of this paper is structured as follows. Section 2 discusses contemporary research in this area of research. Section 3 describes the material and methods used to accomplish this work. Section 4 presents and discusses the results and findings. Section 5 concludes this paper.

2. Related Works

Yin et al. [20] proposed an intrusion detection system for the UNSW-NB15 dataset. They performed feature selection in two stages. In the first stage, they computed the best features using information gain and Random Forest methods and selected the common features received from these algorithms. In the second stage, they gave these features to the RFE algorithm to eliminate the low-ranked features. For the classification, they used a multi-layer perceptron (MLP) classifier. They used only one classifier; however, experimentation with various classifiers could have potentially yielded improved classification metrics for this research.

Almaiah et al. [21] used UNSW-NB15 and KDD-Cup99 datasets for experiments. They reduced the input features using the PCA algorithm. For classification, they used SVC with four different kernels, namely, linear, polynomial, Gaussian radial bias, and sigmoid. They achieved the best classification metrics when the Gaussian radial bias kernel was used. It is important to highlight that SVC is perceived as a slow algorithm since it works to find the optimal separating hyperplane that maximizes the margin between classes while minimizing classification errors.

The work conducted by Lu and Tian [9] employed UNSW-NB15 and NSL-KDD datasets in their research. They used a stacked autoencoder for dimensionality reduction. Next, they developed a bi-directional long short-term memory model for classification. They improved this classifier by implementing an attention mechanism that modifies the weights to give more importance to key features. As a result, they achieved better results than traditional models.

Kasongo [22] proposed an intrusion detection system for industrial IoT networks. This researcher used the UNSW-NB15 dataset in his work. Feature selection is performed using a Genetic Algorithm where a Random Forest is used as a fitness function. Random Forest, Logistic Regression, Naive Bayes, Decision Tree, Extra-Trees (ET), and Extreme Gradient Boosting (XGB) classifiers are used for binary classification. This research achieved the best classification accuracy when using the RF classifier with 16 selected features. This research did employ different classifiers but did not leverage an ensemble model.

Zhou et al. [10] performed their experiments on the UNSW-NB15 dataset. Their dimensionality reduction method performs feature extraction without compromising the representation of rarely occurring labels. They accomplished this task by designing an encoder-decoder neural network associated with a specialized module that ensures extracted features have the power to represent rare labels. Next, they input these refined features to the estimator to perform classification. Their results show improvement in accuracy and reduction in false positive rate. An issue with the autoencoder is that it suffers from overfitting if the number of nodes increases [23].

The intrusion detection approach proposed by Kumar et al. [24] used the UNSW-NB15 dataset for training and real-time data for testing the proposed system. Feature selection is performed using the information gain technique, while classification is performed using C5, the Chi-square Automatic Interaction Detector (CHAID), Classification Regression Trees (CART), and the Quaternion Estimation (QUEST) algorithms. This research achieved better classification performance as compared to tree-based models. This system is designed to detect only five attack categories present in the dataset. However, a total of nine attack categories are present in the dataset.

Kasongo and Sun [25] used the UNSW-NB15 dataset in their research. They pruned the features using a filter-based feature selection technique combined with the XGBoost algorithm. They evaluated the predictive performance of reduced features over Support Vector Machine (SVM), K-Nearest Neighbor, Logistic Regression, Artificial Neural Network (ANN), and Decision Tree classifiers. They presented the model's classification performance over the full and reduced dataset. Their results show that, in general, the performance of classifiers did not drop when a reduced feature set was used. These researchers would have achieved better results if an ensemble had been used.

Gottwalt et al. [26] performed their experiments on the UNSW-NB15 dataset. They identified that multivariate correlation techniques are rarely used in dimensionality reduction. To counter this trend, they developed a feature selection technique named CorrCorr. This method, combined with an addition-based correlation detection, produced a feature set that was compared with the original feature set and PCA. The CorrCorr feature set showed better performance.

Salo et al. [9] performed their experiments on three datasets: ISCX 2012, NSL-KDD, and Kyoto 2006+. This research used a hybrid dimensionality reduction approach by combining information gain and Principal Component Analysis techniques. An ensemble classifier was created using the Support Vector Machine, an instance-based learning algorithm, and multi-layer perceptron for classification. The performance of the proposed system was evaluated on the accuracy, detection rate, and false positive rate. Results show that the proposed system outperformed individual approaches. Our research is related to this work, but instead of using information gain, we have used the recursive feature elimination technique since information gain cannot perform well when an attribute has a very high number of distinct values [27].

Moustafa and Slay [28] determined significant features of the UNSW-NB15 and KDDCup99 datasets. They replicated these datasets and used the association rule mining technique to generate the strongest features. They evaluated the predictive power of selected features using multiple classifiers and used accuracy and the false positive rate as evaluation metrics. Their results show that the accuracy of the KDDCup99 dataset is better than the UNSW-NB15 dataset; however, the false positive rate is lower. This research did not investigate if the identified features can be transformed to reduce input data size further without hitting the performance of classifiers.

In summary, contemporary research used a range of classification techniques on various network datasets. These techniques used PCA, autoencoders, genetic algorithms, and correlation-based methods for feature selection and reduction. These techniques utilized diverse classifiers, including SVC, MLP, DT, XGBoost, and Ensemble models for classification.

3. Material and Methods

3.1. Dataset

The UNSW-NB15 dataset was created using the IXIA PerfectStorm tool in the Australian Center for Cyber Security. It has records of contemporary normal and attack network traffic activities. Moustafa and Slay [28] mentioned that it has 45 features. A total of 42 features are input features; one is the record index vector, one is the label vector, while another vector has attack categories. Out of the 45 features, categorical features are 4, integer type are 30, and float type are 11. In this research, the UNSW-NB15 dataset is used for experiments because of the following reasons:

- This dataset is widely used in cybersecurity research due to the latest cyberattack traffic [29].
- This dataset has hybrid traffic activities. Normal traffic is real, while attack traffic is synthetic and contemporary [30].
- This dataset is considered complex as it has similar behavior of normal and attack traffic; therefore, it can be reliably used in network intrusion detection research [31].
- Another dataset that is widely used in this area of study is NSL-KDD (8, 9). This dataset has only four attack classes, while UNSW-NB15 has nine attack types. Due to its broader attack spectrum, UNSW-NB15 is used in our experiments compared to the NSL-KDD dataset.

3.2. Dimensionality Reduction

Dimensionality reduction techniques are employed to reduce the number of variables, reduce the computational complexity of high-dimensional data, improve the model's accuracy, improve the visualization, and understand the process that generated the data [6].

Padmaja and Vishnuvardhan [32] reported that dimensionality reduction transforms high dimensional data into low dimensions where newly transformed data is a meaningful representation of original data. Two main approaches to dimensionality reduction are feature selection and feature extraction.

Feature selection is the process of selecting the most valuable features from the feature space so that the feature space can be reduced; in comparison, feature extraction creates new features from the original data space using functional mapping [33]. The feature extraction process creates a low-dimensional representation of the feature space that preserves the most valuable information. Khalid et al. [6] identified two approaches to feature selection: the filter method and the wrapper method. These researchers identified three approaches for feature extraction: the performance measure, transformation, and the generation of new features.

3.2.1. Feature Selection

Feature selection methods select relevant features for model construction [34]. Widely used feature selection methods are filter methods, wrapper methods, and embedding methods. Filter methods evaluate the features based on their statistical properties, wrapper methods assess a subset of features using machine learning algorithms, and embedding methods learn feature importance as part of the model training process. In this research, to reduce the feature space, the most useful features are selected using the Recursive Feature Elimination technique, which ranks the features based on their predictive power. This method falls under the wrapper methods category.

Recursive Feature Elimination: Recursive Feature Elimination recursively reduces the feature space. RFE uses an RF classifier to assign feature weights in this research. Initially, it builds a model using all features, ranks the features based on their importance, and removes the smallest-ranked feature. Then, it builds the model again, using the remaining features, ranks them, and removes the least important feature [35].

Random Forest: A Random Forest is an ensemble of decision trees that can perform classification using a majority vote. Each decision tree uses a randomly selected sample of m features from the full set of p features. This technique ensures that trees are uncorrelated with each other hence their average result is less variable and more reliable. In addition, each tree uses a different sample of data, like the bagging approach. RF can successfully model high-dimensional data where features are non-linearly related and do not assume the data follow a particular distribution.

3.2.2. Feature Extraction

Feature extraction methods transform the existing features into lower dimensional space [36]. It is the process of deriving a reduced feature set from the original variables. A number of algorithms are available that can perform this transformation linearly and non-linearly. PCA is the one that can perform non-linear transformation.

Principal Component Analysis: Principal Component Analysis is a widely used feature extraction technique that performs orthogonal transformation of correlated variables into uncorrelated features. These new features are called principal components. This technique preserves a dataset's original high-dimensional variance into low-dimensional principal components. The PCA technique efficiently removes correlated features, and the reduced feature set improves the learning algorithm's performance. If X is a (N $\times$ D) dimensional-centered data matrix and its covariance matrix is as follows:

$$S = N^{-1}X^{T}X \tag{1}$$

Eigenvectors of the covariance matrix are computed as:

$$\frac{1}{N}X^{T}Xu_i = \lambda_i u_i \tag{2}$$

where u_i is the unit vector and indicates the direction and λ_i is the vector of eigen values. Multiplying both sides by X, we obtain the following:

$$\frac{1}{N} XX^T(Xu_i) = \lambda_i (Xu_i)$$
$$\text{putting } v_i = Xu_i$$
$$\frac{1}{N} XX^T v_i = \lambda_i v_i \quad (3)$$

Equation (3) is an eigenvector equation.

3.3. Ensemble Learning

An ensemble model is created by combining multiple machine learning models; the assumption is that the combined model will be better than individual models. Ensemble learners can be classified into the voting ensemble, bagging ensemble, boosting ensemble, and stacking ensemble. The voting ensemble can be further classified into two categories: hard voting and soft voting. Atallah and Al-Mousa [37] reported that the hard-voting approach combines the predictions made by individual models and produces a result based on a majority vote. Peppes et al. [19] explained that the soft voting approach adds the class probabilities predicted by individual models and produces a result based on the highest-class probability. In this research, we constructed a hard-voting ensemble for classifying the anomalies in the UNSW-NB15 dataset. We used predictions of the following individual models as input to our ensemble model: SVC, KNC, and DNN. The ensemble model performed majority voting on individual predictions and produced the final prediction as shown in Figure 1.

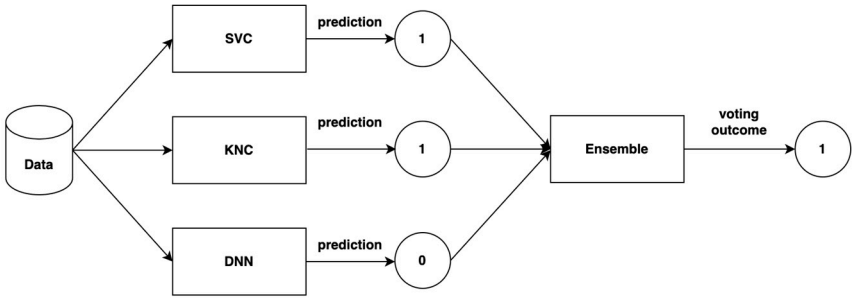

Figure 1. Ensemble Learning.

3.3.1. Support Vector Classifier

The Support Vector Classifier is a simple and powerful classifier. It can draw linear and non-linear class boundaries to classify the data points. To perform its job, it iteratively constructs a hyper-plane to differentiate classes. The main idea of this technique is to create a hyper-plane that creates a soft margin rather than a maximal margin. The maximal margin tries to minimize the error; since the margin is tiny, it is extremely sensitive to change in a single observation. Soft margin may misclassify some observations to achieve a better classification of most of the training observations and improved robustness of the individual observations. We try to maximize the soft margin represented with M in the equation below.

$$\underset{\beta_0,\ \beta_1,\ldots,\beta_p,\ \epsilon_1,\ \ldots,\epsilon_n M}{\text{maximize}} M \quad (4)$$

Constraints β_0, β_1, …, β_p ensure that observations are on the correct side of the hyper-plane. ε_1, …, ε_n M are the slack variables that allow individual observations to be on the wrong side of the margin. In this research, we used SVC with default parameter values.

3.3.2. K-Nearest Neighbor Classifier

K-Nearest Neighbor is a non-parametric proximity-based classifier. This algorithm works on the principle that close-by data points tend to belong to the same class. This algorithm is sensitive to redundant and irrelevant features. It relies on distances, so it is necessary to normalize the data before feeding it. To predict the class label of a point x_o, it finds the *K*-nearest neighbors to this point based on Euclidean distance. These closest points are represented by $\mathcal{N}_0$. Then, each of these neighbors votes for their class *j*, and the majority class wins.

$$\Pr(\mathrm{Y} = \mathrm{j} \,|\mathrm{X} = \mathrm{x}_0) = \frac{1}{\mathrm{K}} \sum_{\mathrm{i} \in \mathcal{N}_0} \mathrm{I}\,(\mathrm{y}_\mathrm{i} = \mathrm{j}) \tag{5}$$

To perform the experiments, we set the leaf size to 30.

3.3.3. Deep Neural Network Classifier

The Artificial Neural Network is a powerful machine learning model that is inspired by neurons in the human brain. The basic building blocks of ANN are the artificial neurons, which are simple processing units. Neurons connect with weights and biases; several layers of neurons can be stacked to produce a Deep Neural Network.

Activation Function: The activation function receives its input from neurons. Inputs of neurons are multiplied with respective weights; these values are added and given to the activation function as an input. The activation function applies a non-linear function on its input and produces non-linear mapping. Popular choices of activation functions are Logistic, ReLU, and Tanh. Using the GridSearchCV method of the Sickit-learn library, we performed an exhaustive search and found Tanh as the best activation function for our dataset.

Backpropagation: The backpropagation algorithm iteratively performs the following steps to train the network: takes input, randomly initializes the network weights, computes the output, evaluates the network performance, and backpropagates the network error. It computes the network error by taking the actual and predicted output difference. It backpropagates the network error by updating the network weights according to their share in the network error. The gradient calculation for updating the weights during the training process is shown below:

$$\frac{\delta \mathrm{C}}{\partial \mathrm{w}} = \mathrm{a}_{\mathrm{in}} \partial_{\mathrm{out}} \tag{6}$$

C represents the cost, w represents weight, a_{in} is the activation of neuron input to the w, and δ_{out} is the error of the neuron that outputs from w.

Optimization Algorithm: The optimizer minimizes the cost function by tweaking the model's parameters. To perform its job, the optimization algorithm shall know the learning rate (η) and gradients of the network parameters. The learning rate is a hyperparameter; its value, 10^{-5}, is set after performing the exhaustive search using the GridSearchCV method. Gradients of parameters are computed using the backpropagation algorithm. To find the desired values of the model parameters, many iterations of the optimization algorithm run on a complete training set. Popular choices of optimization algorithms are gradient descent, stochastic gradient descent, and Adam. In this research, we used Adam; this option is selected after performing a comprehensive search using the GridSearchCV method.

3.4. Proposed System Architecture

In this section, we describe the architecture of the proposed system designed to predict the class labels of the UNWS-NB15 dataset (see Figure 2).

1. Load the dataset in the development environment from a CSV file to the Pandas data frame.
2. Clean data to remove incorrect, incomplete, and redundant records and split the dataset into train and test sets for training and testing purposes.
3. Perform hybrid feature reduction:

- Identify and select valuable features by applying the RFE technique. This module will receive all input features and select the 20 top-ranked features for prediction.
- Perform feature extraction, using the PCA algorithm, on 20 top-ranked features to transform them into principal components. This step produces three sets of 10, 12, and 15 components.

4. Perform classification using SVC, KNC, and DNN on all three sets of extracted features.
5. Apply the majority voting ensemble technique to obtain the final classification of each extracted feature set.
6. Evaluate the model performance on the test dataset.

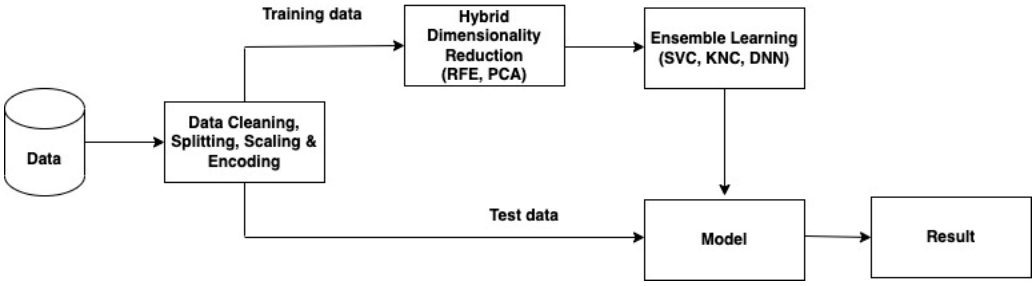

Figure 2. Proposed System Architecture.

3.5. Hardware and Software Platform

This experiment was performed using the Google Colaboratory development platform, a cloud-hosted Jupyter notebook service. It executes a user program in a private virtual machine associated with each user program. User programs run in GPU and TPU as per the availability of resources and the user's service model subscription. Furthermore, the program code is written in Python (version 3.10 2). Pandas library is used for data manipulation, and Sickit-learn is used for building machine learning models and evaluation.

3.6. Data Cleaning

Dirty data can lead to incorrect results. Data Cleaning is an essential step of the machine learning process. This step includes modifying the records with inaccurate, incomplete, duplicate, and improper formats. Inaccurate records can introduce noise and bias into the analysis, affecting the integrity of the results and predictions. Incomplete data, where certain records have missing values for specific features, can lead to biased or incomplete analysis. Most machine learning algorithms cannot handle missing values, and they might either ignore the incomplete records or introduce biases while filling in missing values. Duplicate records can skew the analysis and lead to overfitting in machine learning models. Removing duplicate records helps in creating a diverse and representative dataset, leading to more robust and accurate models. Improper formats can introduce inconsistencies in the dataset. Inconsistent formats make it difficult to analyze and process data uniformly. The following variables of the UNSW-NB15 dataset had some very low-frequency values: sttl, dttl, swin, dwin, trans_depth, ct_flw_http_mthd, is_ftp_login, ct_ftp_cmd. These values represent inaccuracy. Therefore, they were compared with the labels and replaced with a high-frequency alternative.

3.7. Data Pre-Processing

Data pre-processing prepares the data to feed in a machine learning model. This step includes feature scaling, missing values imputation, and encoding. At this stage, data were split into training (70%) and testing (30%) sets. The dataset is split into a training set and a test set to evaluate the model's performance on unseen data. This practice is essential to ensure that the machine learning model generalizes well to new, previously unseen data points. The UNSW-NB15 dataset does not have the missing values problem. Therefore,

this step was not required. However, feature scaling and encoding steps were performed, which are described below.

3.7.1. Feature Scaling

The feature scaling operation is applied to transform features on the same scale. Most machine learning models require data to be on the same scale except for a few, for example, Decision Tree and Random Forest. The UNSW-NB15 dataset features have high variance; therefore, they were transformed before being supplied to the machine learning model.

Standardization: Features were transformed by using the standardization technique. The standardization method transforms data around mean 0 and standard deviation 1. This method does not bind data within specified limits. Equation (7) shows a standardization formula where x represents an observation, μ represents the mean of the observations, and σ is the standard deviation.

$$x' = \frac{x - \mu}{\sigma} \tag{7}$$

3.7.2. Encoding

Categorical variables are those that do not have any natural order. The UNSW-NB15 dataset has three variables that have categorical values. These variables are proto, service, and state. The first variable carries the name of the transport layer protocol, the second variable keeps an application layer service name, and the third variable contains TCP connection state information. Since this data does not have any natural order, the One-Hot encoding is applied to transform them into a numeric form.

3.8. Evaluation Metrics

Model performance is evaluated using accuracy, detection rate (DR), false positive rate (FPR), f1-score, and area under the curve (AUC) metrics. Elements of these metrics can be retrieved from the confusion matrix where the confusion matrix is {TP, TN, FP, FN}. True positive (TP) means correctly classified attack packets. True negative (TN) means correctly classified normal packets. False positive (FP) means incorrectly classified attack packets, and false negative (FN) means incorrectly classified normal packets.

Accuracy represents the ratio of correctly identified packets versus total number of packets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

The detection rate represents the ratio of correctly identified attacks versus predicted attacks.

$$\text{Detection Rate} = \frac{TP}{TP + FN} \tag{9}$$

The false positive rate is the ratio of incorrectly identified attacks versus predicted normal.

$$FPR = \frac{FP}{FP + TN} \tag{10}$$

The F1-score is the harmonic mean of precision and recall (detection rate), where precision is defined as the accuracy of predicting normal packets as normal.

$$F1 - \text{score} = \frac{2(TP)}{2(TP) + FP + FN} \tag{11}$$

The area under the curve measures the overall performance of a binary classifier. It is expressed in terms of false positive rate and true positive rate, where the true positive rate (TPR) represents the proportion of normal packets that are correctly identified.

$$AUC = \sum_{i=1}^{n-1} \frac{1}{2} \left(FPR_{i+1} - FPR_i\right) \left(TPR_i - TPR_{i+1}\right) \tag{12}$$

4. Results

Research results are shown in Tables 1–5. Table 1 displays the evaluation metrics when all input features are used. Table 2 lists the names of the top twenty ranked features selected at the feature selection stage. Table 3 displays the evaluation metrics when these twenty selected features are used as the input. Table 4 illustrates the cumulative sum of PCA explained variance ratios computed over 10, 12, and 15 principal components, respectively. Table 5 presents the evaluation metrics when these three sets of principal components are used as input. Table 6 compares our results with contemporary research.

Table 1. Performance metrics using all features on the test set.

Performance Metrics	SVC	KNC	DNN	Voting Ensemble
Accuracy	92.7784	95.3662	95.7572	95.8536
Detection Rate	94.7302	93.2090	94.7330	94.9228
False Positive Rate	9.1760	2.4735	3.2170	3.2142
F1-score	92.9211	95.2671	95.7161	95.8174
AUC	92.7771	95.3677	95.7579	95.8542

Table 2. Feature Selection.

Names of 20 Top-Ranked Features.
dur, dpkts, sbytes, dbytes,
rate, sttl, dttl, sload, dload, sinpkt,
tcprtt, synack, smean, dmean, ct_srv_src,
ct_state_ttl, ct_dst_sport_ltm, ct_dst_src_ltm, ct_srv_dst, state_INT

Table 3. Performance metric using selected features on the test set.

Performance Metrics	SVC	KNC	DNN	Voting Ensemble
Accuracy	92.5200	93.9012	95.2433	94.1721
Detection Rate	94.7940	93.4715	95.4889	94.5418
False Positive Rate	9.7396	5.6717	5.0006	6.19518
F1-score	92.6650	93.8568	95.2407	94.17642
AUC	92.5271	93.8999	95.2441	94.1733

Table 4. Feature Extraction.

No. of Components	Cumulative Sum of PCA Explained Variance Ratio
10	0.919346104
12	0.953800048
15	0.989655398

The performance metrics obtained from the proposed research are presented in Tables 1, 3 and 5. A comparison of the results of the voting ensemble confirms that the reduced feature set does not decrease the classification performance substantially.

The results in Tables 1, 3 and 5 show that DNN outperformed other classifiers in general. The only exception is Table 1, where, in most cases, the voting ensemble gave better results. In two instances, Tables 1 and 3, KNC gave the best false positive rates at 2.4735% and 4.5394%, respectively. Similarly, at two places in Table 5, SVC gave the best detection rates at 94.8239% and 94.6183%.

Table 5. Performance metrics using extracted features on the test set.

Principal Components	Performance Metric	SVC	KNC	DNN	Voting Ensemble
10	Accuracy	92.2295	93.6275	94.3928	93.9026
	Detection Rate	94.8239	93.3035	94.5612	94.4074
	False Positive Rate	10.3614	6.0488	5.7752	6.6015
	F1-score	92.4216	93.6030	94.3989	93.9297
	AUC	92.2312	93.6273	94.3929	93.9029
12	Accuracy	92.2714	93.8300	94.4137	94.0465
	Detection Rate	94.6183	93.3272	94.4708	94.1620
	False Positive Rate	10.0930	5.6633	5.6437	6.0698
	F1-score	92.4748	93.8206	94.4367	94.0743
	AUC	92.2626	93.8319	94.4135	94.0460
15	Accuracy	92.4558	94.1261	95.0827	94.3453
	Detection Rate	94.4822	92.7937	94.9483	93.9240
	False Positive Rate	9.5737	4.5394	4.7826	5.2327
	F1-score	92.6109	94.0510	95.0797	94.3255
	AUC	92.4542	94.1271	95.0828	94.3456

Table 6. Performance comparison (UNSW-NB15).

Research	Dimensionality Reduction Technique	Features	Classifier	Accuracy
[25]	XGBoost	19	DT	90.85%
[38]	Rao Optimization Algorithm	19	SVM	92.5%
[22]	Genetic Algorithm	16	RF	87.61%
[21]	PCA	2	SVM-rbf	93.94%
[20]	IG, RF and RFE	23	MLP	84.24%
Proposed Approach	RFE and PCA	15	Voting Ensemble	94.3%

Comparing the performance of 10, 12, and 15 components (see Table 5), DNN remains the best classifier. The top metrics of the voting ensemble were received when 15 components were used. In this case, it gave 94.34% accuracy, a 5.23% false positive rate, a 94.32% f1-score, and 94.34% AUC. The only exception is the detection rate; the top metric of detection rate, 94.40%, is received when 10 components were used. The higher the number of components, the more data variance is captured and the better the representation is, which gave us better performance metrics.

Table 2 shows 20 top-ranked features, which are determined using the RFE technique at the feature selection stage. The cumulative sum of PCA is explained and the variance ratio is shown in Table 4. This table shows the outcome of the feature extraction process. Using 10, 12, and 15 components, we captured the following ratios of the cumulative sum of data variance: 0.9193, 0.9538, and 0.9896. The highest variance is captured when 15 components are used. When we increase the number of components, the cumulative sum of the data variance also increases. We captured 98.96% variance using only 15 components, far less than the input features space of size 42 we had.

We compared our work with contemporary research in the field of network anomaly detection (see Table 6). We achieved higher accuracy by utilizing a smaller number of

features in comparison to other methods, except for [21]. When comparing our ensemble model to the SVC radial bias function (SVC-rbf) employed by [21], our model proves to be simpler, more robust, and faster. The ensemble model simplifies the process by aggregating predictions from individual learners and then performs the final prediction, whereas SVC-rbf constructs intricate decision boundaries to capture complex non-linear data patterns. The Ensemble learner exhibits greater robustness, especially when diverse base models are used, whereas SVC-rbf is sensitive to the selection of hyperparameters. Additionally, the ensemble model is faster to train, whereas SVC-rbf requires significant time due to its non-linear transformation computations. Hence, it proves that our proposed hybrid dimensionality scheme is a good addition to anomaly detection research. The proposed research has several advantages. First, it reduces the number of features. Therefore, the model's complexity and processing time decrease. Second, considering that collective opinion is better than individual opinion, it employed a voting ensemble for classification, which uses predictions of individual classifiers and outputs a majority vote. Third, it presents performance metrics using three sets of principal components, proving that the best performance metrics are achieved when those components are used that capture the highest variance. Although a disadvantage of the proposed research is that using multiple classifiers takes more processing time, since we have reduced the number of features, this disadvantage is compensated.

5. Conclusions

The presence of redundant and irrelevant features negatively impacts model building and training. The proposed hybrid dimensionality reduction system, incorporating feature selection and extraction techniques, can reduce the input feature space and overcome this problem. The proposed system performs two processes: feature selection and feature extraction. The feature selection process selected the 20 best features from the input feature space. These features were reduced to 10, 12, and 15 principal components at the feature extraction stage. These principal components were given to the system one by one. In the first stage, classification was performed using SVC, KNC, and DNN. While at the second stage, these classification results were given to the voting ensemble classifier for final prediction. The performance of the proposed system was evaluated on the accuracy, detection rate, false positive rate, f1-score, and area under the curve metrics. The performance metrics confirm that reduced and transformed features did not decrease the classifiers' performance. In the future, we want to perform these experiments on different datasets to further test our proposed technique. In addition, we want to address the class imbalance issue in the dataset.

Author Contributions: Conceptualization, H.G., S.S. and B.V.; methodology, H.G.; software, H.G.; validation, H.G., S.S. and B.V.; formal analysis, H.G., S.S. and B.V.; investigation, H.G.; resources, H.G., S.S. and B.V.; data curation, H.G., S.S. and B.V.; writing—original draft preparation, H.G.; writing—review and editing, S.S., B.V. and H.G.; visualization, H.G.; supervision, S.S. and B.V.; project administration, S.S., B.V. and H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available in a publicly accessible repository. https://ieee-dataport.org/ (accessed on 2 January2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fernandes, G.; Rodrigues, J.J.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. *Telecommun. Syst.* **2019**, *70*, 447–489. [CrossRef]
2. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]

3. Mohamed, G.; Visumathi, J.; Mahdal, M.; Anand, J.; Elangovan, M. An Effective and Secure Mechanism for Phishing Attacks Using a Machine Learning Approach. *Processes* **2022**, *10*, 1356. [CrossRef]
4. Naseer, S.; Saleem, Y.; Khalid, S.; Bashir, M.K.; Han, J.; Iqbal, M.M.; Han, K. Enhanced network anomaly detection based on deep neural networks. *IEEE Access* **2018**, *6*, 48231–48246. [CrossRef]
5. Moustafa, N.; Slay, J. A hybrid feature selection for network intrusion detection systems: Central points. *arXiv* **2015**, arXiv:1707.05505.
6. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
7. Zebari, R.; Abdulazeez, A.; Zeebaree, D.; Zebari, D.; Saeed, J. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 56–70. [CrossRef]
8. Salo, F.; Nassif, A.B.; Essex, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Comput. Netw.* **2019**, *148*, 164–175. [CrossRef]
9. Lu, G.; Tian, X. An Efficient Communication Intrusion Detection Scheme in AMI Combining Feature Dimensionality Reduction and Improved LSTM. *Secur. Commun. Netw.* **2021**, *2021*, 6631075. [CrossRef]
10. Zhou, X.; Hu, Y.; Liang, W.; Ma, J.; Jin, Q. Variational LSTM enhanced anomaly detection for industrial big data. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3469–3477. [CrossRef]
11. Kasongo, S.M.; Sun, Y. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Comput. Secur.* **2020**, *92*, 101752. [CrossRef]
12. Bagui, S.; Walauskis, M.; DeRush, R.; Praviset, H.; Boucugnani, S. Spark configurations to optimize decision tree classification on UNSW-NB15. *Big Data Cogn. Comput.* **2022**, *6*, 38. [CrossRef]
13. Xu, W.; Fan, Y. Intrusion detection systems based on logarithmic autoencoder and XGBoost. *Secur. Commun. Netw.* **2022**, *2022*, 9068724. [CrossRef]
14. Jing, D.; Chen, H.B. SVM based network intrusion detection for the UNSW-NB15 dataset. In Proceedings of the 2019 IEEE 13th International Conference on ASIC (ASICON), Chongqing, China, 29 October–1 November 2019; pp. 1–4.
15. Dutta, V.; Choraś, M.; Kozik, R.; Pawlicki, M. Hybrid model for improving the classification effectiveness of network intrusion detection. In *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; Volume 12, pp. 405–414.
16. Man, J.; Sun, G. A residual learning-based network intrusion detection system. *Secur. Commun. Netw.* **2021**, *2021*, 5593435. [CrossRef]
17. Ahmad, I.; Haq, Q.E.U.; Imran, M.; Alassafi, M.O.; AlGhamdi, R.A. An efficient network intrusion detection and classification system. *Mathematics* **2022**, *10*, 530. [CrossRef]
18. Usha, P.; Anuradha, M.P. Feature Selection Techniques in Learning Algorithms to Predict Truthful Data. *Indian J. Sci. Technol.* **2023**, *16*, 744–755. [CrossRef]
19. Peppes, N.; Daskalakis, E.; Alexakis, T.; Adamopoulou, E.; Demestichas, K. Performance of machine learning-based multi-model voting ensemble methods for network threat detection in agriculture 4.0. *Sensors* **2021**, *21*, 7475. [CrossRef]
20. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *J. Big Data* **2023**, *10*, 1–26. [CrossRef]
21. Almaiah; Amin, M.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Al Hwaitat, A.K.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; et al. Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. *Electronics* **2022**, *11*, 3571. [CrossRef]
22. Kasongo, S.M. An advanced intrusion detection system for IIoT based on GA and tree based algorithms. *IEEE Access* **2021**, *9*, 113199–113212. [CrossRef]
23. Sankaran, A.; Vatsa, M.; Singh, R.; Majumdar, A. Group sparse autoencoder. *Image Vis. Comput.* **2017**, *60*, 64–74. [CrossRef]
24. Kumar, V.; Sinha, D.; Das, A.K.; Pandey, S.C.; Goswami, R.T. An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset. *Clust. Comput.* **2020**, *23*, 1397–1418. [CrossRef]
25. Kasongo, S.M.; Sun, Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *J. Big Data* **2020**, *7*, 1–20. [CrossRef]
26. Gottwalt, F.; Chang, E.; Dillon, T. CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques. *Comput. Secur.* **2019**, *83*, 234–245. [CrossRef]
27. Quinlan, J.R. Introduction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–107. [CrossRef]
28. Moustafa, N.; Slay, J. The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In Proceedings of the 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Kyoto, Japan, 5 November 2015; pp. 25–31.
29. Moualla, S.; Khorzom, K.; Jafar, A. Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset. *Comput. Intell. Neurosci.* **2021**, *2021*, 5557577. [CrossRef]
30. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.

31. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. A Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
32. Padmaja, D.L.; Vishnuvardhan, B. Comparative study of feature subset selection methods for dimensionality reduction on scientific data. In Proceedings of the 2016 IEEE 6th International Conference on Advanced Computing (IACC), London, UK, 27–29 August 2014; pp. 31–34.
33. Motoda, H.; Liu, H. Feature selection, extraction and construction. In *Communication of IICM (Institute of Information and Computing Machinery, Taiwan)*; Tamkang University: Taibei, Taiwan, 2002; Volume 5, p. 2.
34. Kocher, G.; Kumar, G. Analysis of machine learning algorithms with feature selection for intrusion detection using UNSW-NB15 dataset. *Int. J. Netw. Secur. Its Appl.* **2021**, *13*, 21–31.
35. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene Selection for Cancer Classification using Support Vector Machines. *Mach. Learn.* **2002**, *46*, 389–422. [CrossRef]
36. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Zarkasi, A. Automatic features extraction using autoencoder in intrusion detection system. In Proceedings of the 2018 International Conference on Electrical Engineering and Computer Science (ICECOS), Pangkal, Indonesia, 2–4 October 2018; pp. 219–224.
37. Atallah, R.; Al-Mousa, A. Heart disease detection using machine learning majority voting ensemble method. In Proceedings of the 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–6.
38. Abd, S.N.; Alsajri, M.; Ibraheem, H.R. Rao-SVM machine learning algorithm for intrusion detection system. *Iraqi J. Comput. Sci. Math.* **2020**, *1*, 23–27.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.