Efficient Network Intrusion Detection Using PCA-Based Dimensionality Reduction of Features

Razan Abdulhammed
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
rabdulha@my.bridgeport.edu

Miad Faezipour*
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
mfaezipo@bridgeport.edu

Hassan Musafer
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
hmusafer@my.bridgeport.edu

Abdelshakour Abuzneid
School of Engineering
University of Bridgeport
Bridgeport, CT, USA
abuzneid@bridgeport.edu

***Abstract*—Designing a machine learning based network intrusion detection system (IDS) with high-dimensional features can lead to prolonged classification processes. This is while low-dimensional features can reduce these processes. Moreover, classification of network traffic with imbalanced class distributions has posed a significant drawback on the performance attainable by most well-known classifiers. With the presence of imbalanced data, the known metrics may fail to provide adequate information about the performance of the classifier. This study first uses Principal Component Analysis (PCA) as a feature dimensionality reduction approach. The resulting low-dimensional features are then used to build various classifiers such as Random Forest (RF), Bayesian Network, Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) for designing an IDS. The experimental findings with low-dimensional features in binary and multi-class classification show better performance in terms of Detection Rate (DR), F-Measure, False Alarm Rate (FAR), and Accuracy. Furthermore, in this paper, we apply a Multi-Class Combined performance metric $Combined_{Mc}$ with respect to class distribution through incorporating FAR, DR, Accuracy, and class distribution parameters. In addition, we developed a uniform distribution based balancing approach to handle the imbalanced distribution of the minority class instances in the CICIDS2017 network intrusion dataset. We were able to reduce the CICIDS2017 dataset's feature dimensions from 81 to 10 using PCA, while maintaining a high accuracy of 99.6% in multi-class and binary classification.**

***Index Terms*—IDS, Imbalanced class distributions, Machine Learning, PCA.**

I. Introduction

A Network Intrusion Detection System (IDS) is a software-based application or a hardware device that is used to identify malicious behavior in the network [1]. In this research, we adopt machine learning based IDS with both multi- and binary-class classifications. For the multi-class classification, there are 15 classes, where each class represents either normal network flow traffic or one of 14 types of attacks.

This study accustoms Principal Component Analysis (PCA) for dimensionality reduction. As a proof-of-concept and to verify the feature dimensionality reduction ideas, the paper utilized the up-to-date CICIDS2017 intrusion detection and prevention dataset [2], [3] which consists of five separated data files. Each file represents the network traffic flow and specific the types of attacks for a certain period of time. The dataset was collected based on a total of 5 days, Monday through Friday. The traffic flow on Monday includes the benign network traffic, whereas the implemented attacks in the dataset were executed on Tuesday, Wednesday, Thursday and Friday. In this paper, we combined all CICIDS2017's files together and fed them through the PCA unit for a compressed and lower dimensional representation of all the data.

TABLE I
CICIDS2017 ATTACK DISTRIBUTION

Traffic Type	Size	Traffic Type	Size
DoS Hulk	231,073	DoS Slow HTTP Test	5,499
Port Scan	158,930	Botnet	1,966
DDoS	41,835	Web Attack: Brute Force	1,507
DoS GoldenEye	10,293	Web Attack: XSS	625
FTP Patator	7,938	Infiltration	36
SSH Patator	5,897	Web Attack: SQL Injection	21
DoS Slow Loris	5,796	HeartBleed	11

II. Related Work

This section provides a brief overview of the prior related work directly relevant to CICIDS2017, with a special emphasis on machine learning (ML) based approaches that utilize this dataset. Moreover, the section glances at work related to principal component analysis in ML-based IDS.

In [2], Sharafaldin et al. used a Random Forest Regressor to determine the best set of features to detect each attack family. The authors examined the performance of these features with different algorithms that included K-Nearest Neighbor (KNN), Adaboost, Multi-Layer Perceptron (MLP), Naïve Bayes, Random Forest (RF), Iterative Dichotomiser 3 (ID3) and Quadratic Discriminant Analysis (QDA). The highest precision value was 0.98 with RF and ID3 [2].

Aksu et al. [4] proposed a denial of service intrusion detection system that utilized the Fisher Score algorithm for features selection and Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Decision Tree (DT) as the classification algorithm. Their IDS achieved 99.7%, 57.76% and 99% success rates using SVM, KNN and DT, respectively.

Marir et al. [5] utilized a distributed Deep Belief Network (DBN) as the dimensionality reduction approach. The obtained features were then fed to a multi-layer ensemble SVM.

978-1-7281-1244-2/19/$31.00 ©2019 IEEE

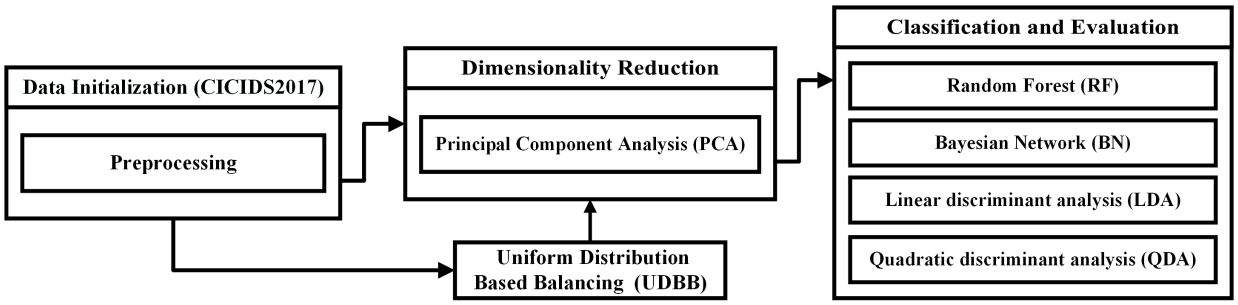

Fig. 1. Proposed Framework

The ensemble SVM was accomplished in an iterative reduce paradigm based on Spark [6]. Their proposed approach achieved an F-measure value equal to 0.921.

Bansal [7] proposed a Data Dimensionality Reduction (DDR) method for network intrusion detection. Their proposed scheme was evaluated by XGBoost (Extreme Gradient Boosting), SVM (Support Vector Machine), CTree (Conditional inference Trees) and Neural network (Nnet) classifiers. The number of selected features was 36 and the highest achieved accuracy was 98.93% with XGboost. Furthermore, the authors excluded Monday network traffic of the CICIDS2017 dataset, which is only benign traffic in their system.

Researchers have also employed Principal Component Analysis (PCA) for feature reduction in intrusion detection datasets [8], [9], [10], [11].

Xia et al. [12] implemented an IDS that used PCA as the feature reduction approach and Grey Neural Networks (GNN) as the classifier on the KDD-99 dataset. [12]. The researchers in [13] also defined the reduction rate and studied the efficiency of PCA for intrusion detection. The authors fulfilled their experiments utilizing Random Forest and C4.5 on KDD-CUP [14] and UNB-ISCX [15] datasets.

III. CICIDS2017 Dataset Structure

The CICIDS2017 dataset was collected based on real traces of benign and malicious activities of the network traffic. The total number of records in the dataset is 2,830,108. The benign traffic encompasses 2,358,036 records (83.3% of the data), while the malicious records are 471,454 (16.7% of the data). CICIDS2017 is one of the unique datasets that includes up-to-date attacks. Furthermore, the features are exclusive and matchless in comparison with other datasets such as AWID [16] [17], and CIDD-001 [18] [19]. For this reason, CICIDS2017 was selected as the most comprehensive IDS benchmark to test and validate the proposed ideas. Table I highlights the distribution of the attacks in the CICIDS2017 dataset. CICIDS2017 is a labeled dataset with a total number of 84 features including the last column corresponding to the traffic status (class label). The features were extracted by CICFlowMeter-V3 [20]. The output of CICFlowMeter-V3 is a CSV file that includes: Flow ID (1), Source IP (2) and Destination IP (4), Time stamp (7) and Label (84). The Flow ID (1) includes the four tuples: Source IP, Source Port, Destination IP, and Destination Port. To the best of our knowledge, most previous studies that utilized CICIDS2017 neglect Flow ID (1), Source IP (2), Destination IP (4), and Time stamp (7). In this paper, we utilized CICIDS2017 with respect to the listed features except the Flow ID (1) and Time Stamp (7). Thus, in our study, the total number of used features encompasses 82 features including the Label (84). The traffic features are explained in [21].

IV. Methodology and Experimental Procedures

This section gives an overview of our methodology and how we carried out our experiments. The procedure mainly includes prepossessing, dimensionality reduction using PCA, classification and evaluation (Figure 1).

A. Preprocessing

In this study, a preprocessing function is applied to the CICIDS2017 dataset by mapping the IP (Internet Protocol) address to an integer representation. The mapped IP includes the Source IP Address (Src IP) as well as the Destination IP Address (Dst IP). These two are converted to an integer number representation.

B. PCA-based Feature Reduction and Classification

PCA is a projection-based mechanism where the original dataset is projected into a subspace with lower dimensions, whilst retaining the essence of the original data. To reduce the features dimensionality from n-dimensions to k-dimensions using PCA, two phases are conducted. First, the data is preprocessed to normalize its mean and variance. Then, the covariance matrix, Eigen-vectors and Eigen-values are calculated to form the reduction phase.

Various classifiers such as RF, BN, LDA and QDA are used to build and test an ML-based IDS framework with reduced features and uniformly balanced class distributions. This study splits the data into training and testing sets with a ratio of 70:30.

C. Multi-Class Combined Performance Metric

In general, the overall accuracy is used to measure the effectiveness of a classifier. Unfortunately, in presence of imbalanced data, this metric may fail to provide adequate information about the performance of the classifier. Furthermore, the method is very sensitive to the class distribution and might be misleading in some way. Hamed et al. [22] proposed a combined performance metric to compare various binary classifier systems. However, their solution neglects

class distribution and can work only for binary classifications. In this paper, we apply the multi-class combined performance metric $Combined_{Mc}$ (or CM) with respect to class distribution to compare various binary class as well as multi-class classification systems through incorporating four metrics together (FAR, Accuracy, Detection Rate, and class distribution). The multi-class Combined performance metric can be estimated using the following equation.

$$Combined_{Mc} = \sum_{i=1}^{C} \lambda_i \left(\frac{Acc_i + DR_i}{2} - FAR_i \right) \quad (1)$$

where C is number of classes, and λ_i is the class distribution ($dist$), which can be estimated using the following formula.

$$dist \ = \lambda_i = \frac{Number\ of\ instances\ in\ class\ i}{Number\ of\ instances\ in\ the\ dataset} \quad (2)$$

The result of this metric will be a real value between -1 and 1, where -1 corresponds to the worst overall system performance and 1 corresponds to the best overall system performance. The confusion matrix of classification along with the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are used to derive this metric. Table II illustrates the pseudo-code for calculating this proposed combined metric.

D. Uniform Distribution Based Balancing (UDBB)

In many classification problems, almost all the instances are labeled as one class (called the majority, or negative class), while far fewer instances are labeled as the other class or classes (often called the minority class(es), or positive class(es)); usually the more important class(es) [19]. This subsection provides a glance at the Uniform Distribution Based Balancing (UDBB) technique. UDBB is based on learning and sampling probability distributions [23]. In this technique, the sampling of instances is performed following a distribution learned for each pair example of feature and class label. A uniform distribution is considered in order to re-sample new instances. According to [24], the Imbalance Ratio (IR) can be defined as the ratio of the number of instances in the majority class to the number of instances in the minority class, as presented in Equation 3.

$$Imbalance\ Ratio = \frac{Majority\ Class\ Instances}{Minority\ Class\ Instances} \quad (3)$$

For the CICIDS2017 dataset, IR is 5:1 and the total number of classes is 15 classes. To apply UDBB, a uniform number of instances $(I_{Resample})$ for each class is calculated from Equation 4.

$$I_{Resample} = \frac{Number\ of\ Instances\ in\ the\ dataset}{Number\ of\ Classes\ in\ the\ dataset} \quad (4)$$

V. Results and Discussion

All the simulations for obtaining the results were carried out using an Intel-Core i7 with 3.30 GHz and 32 GB RAM, running Windows 10. The results highlight the advantages of feature dimensionality reduction on CICIDS2017. From the research efforts in this work, we were able to reduce the dimensionality of the features in CICIDS2017 from 81 features to 10 features while maintaining a high accuracy in multi-class and binary class classification using the Random Forest classifier. The findings are discussed in following subsections.

A. Binary class (Bc) Classification

Table III displays the summary of the results obtained for binary classification (X representing the number of features). The Table highlights the results of the dimensionality reduction of the features in CICIDS2017 from 81 features to 10 features. The DR metric revealed that $(PCA - RF)_{Bc-10}$ is able to detect 98.8% of the attacks. In the same manner, $(PCA - RF)_{Bc-10}$ achieved an F-Measure of 0.997. The results from the classification using different classifiers assures that our reconstructing of new feature representation was good enough to achieve an overall accuracy of 99.6% with 10 features in binary classification using PCA.

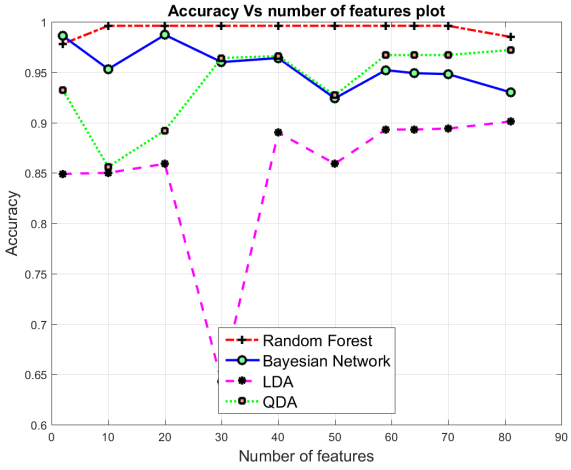

Fig. 2. Accuracy in terms of number of PCA components (Mc)

B. Multi-class (Mc) Classification

This study used the Acc, F-M, FP Rate (FPR), TP Rate (TPR), Precision, Recall, and the Combined multi-class metrics to evaluate the performance of multi-class classification. Table IV displays the summary of the results obtained for the dimensionality reduction of the features for CICIDS2017 from 81 to 10 features using PCA (X representing the number of features).

Figure 2 presents the resulting accuracies in terms of the number of principal components. What is striking about the resulting accuracies in Figure 2 is that the Random Forest classifier shows a constantly high accuracy for reduced features from 81 through 10. In contrast, the resulting accuracies of LDA and QDA cases were oscillatory. For QDA, the accuracy is wobbling between 66% with 10 features and 96.7% with 60 features. For LDA with 10 and 40 features, the accuracy is fluctuating between 85% and 96.6%, respectively. A detailed analysis summary of the proposed framework in terms of FPR, TPR, Precision and Recall are tabulated in Tables V and VI. Table V depicts the results with 10 features before applying UDBB (with the original class distributions), while Table VI shows the results using 10 features after

TABLE II
PSEUDO-CODE FOR THE PROPOSED $Combined_{Mc}$ METRIC CALCULATION

Calculate $Combined_{Mc}$ with respect to Class Distribution
Feed Confusion Matrix
For i =1 to C
Calculate the total number of **FP** for C_i as the sum of values in the i^{th} column excluding **TP**
Calculate the total number of **FN** for C_i as the sum of values in the i^{th} row excluding **TP**
Calculate the total number of **TN** for C_i as the sum of all columns and rows excluding the i^{th} row and column
Calculate the total number of **TP** for C_i as the diagonal of the i^{th} cell of CM
Calculate the total number of instances for C_i as the sum of the i^{th} row
Calculate the total number of instances in the dataset as the sum of all rows
Calculate Acc , DR , and FAR for each class C_i
Calculate the distribution of each C_i using Eq. 2
i ++
Calculate $Combined_{Mc}$ using Eq. 1

TABLE III
PERFORMANCE EVALUATION OF THE PROPOSED FRAMEWORK IN BINARY CLASSIFICATION USING PCA

	$(PCA-RF)_{Bc-X}$					$(PCA-BN)_{Bc-X}$					$(PCA-LDA)_{Bc-X}$					$(PCA-QDA)_{Bc-X}$				
	Acc	FAR	DR	F-M	CM_{Bc}	Acc	FAR	DR	F-M	CM_{Bc}	Acc	FAR	DR	F-M	CM_{Bc}	Acc	FAR	DR	F-M	CM_{Bc}
81	0.995	0.002	0.984	0.996	0.987	0.975	0.025	0.976	0.976	0.951	0.937	0.254	0.811	0.937	0.846	0.782	0.237	0.978	0.807	0.632
70	0.997	0.001	0.989	0.997	0.991	0.970	0.029	0.966	0.970	0.938	0.947	0.274	0.821	0.947	0.856	0.792	0.247	0.988	0.817	0.642
64	0.996	0.002	0.986	0.996	0.988	0.969	0.029	0.966	0.970	0.968	0.947	0.027	0.820	0.947	0.466	0.793	0.245	0.988	0.818	0.891
59	0.997	0.0017	0.989	0.997	0.991	0.968	0.030	0.963	0.969	0.934	0.947	0.027	0.823	0.947	0.858	0.794	0.244	0.988	0.819	0.646
50	0.996	0.0017	0.989	0.997	0.991	0.971	0.025	0.959	0.972	0.939	0.945	0.028	0.814	0.945	0.880	0.809	0.226	0.988	0.832	0.898
40	0.997	0.001	0.990	0.997	0.991	0.974	0.021	0.953	0.974	0.941	0.944	0.032	0.829	0.945	0.856	0.808	0.226	0.981	0.831	0.646
30	0.997	0.001	0.989	0.997	0.990	0.979	0.026	0.956	0.971	0.703	0.944	0.030	0.821	0.945	0.852	0.830	0.198	0.974	0.849	0.703
20	0.996	0.001	0.989	0.997	0.991	0.965	0.031	0.948	0.966	0.926	0.878	0.025	0.396	0.862	0.612	0.717	0.332	0.969	0.754	0.511
10	0.996	0.001	0.988	0.997	0.991	0.952	0.036	0.897	0.953	0.889	0.869	0.028	0.363	0.852	0.588	0.712	0.048	0.966	0.749	0.911

TABLE IV
PERFORMANCE EVALUATION OF THE PROPOSED FRAMEWORK IN MULTI-CLASS CLASSIFICATION USING PCA

	$PCA-RF)_{Mc-X}$			$(PCA-BN)_{Mc-X}$			$(PCA-LDA)_{Mc-X}$			$(PCA-QDA)_{Mc-X}$		
	Acc	F-M	CM_{Mc}	Acc	F-M	CM_{Mc}	Acc	F-M	CM_{Mc}	Acc	F-M	CM_{Mc}
81	0.985	0.995	0.986	0.930	0.953	0.925	0.901	0.914	0.801	0.972	0.974	0.961
70	0.996	0.988	0.988	0.948	0.964	0.942	0.894	0.906	0.735	0.967	0.975	0.967
64	0.996	0.997	0.986	0.949	0.966	0.955	0.893	0.906	0.745	0.967	0.975	0.985
59	0.996	0.995	0.987	0.952	0.967	0.917	0.893	0.906	0.677	0.967	0.975	0.880
50	0.996	0.996	0.987	0.924	0.941	0.916	0.859	0.880	0.679	0.927	0.946	0.885
40	0.996	0.997	0.9884	0.964	0.974	0.954	0.890	0.546	0.727	0.966	0.974	0.967
30	0.996	0.997	0.988	0.960	0.971	0.897	0.643	0.643	0.720	0.964	0.972	0.965
20	0.996	NAN	0.987	0.987	0.952	0.855	0.859	NAN	0.4805	0.892	0.886	0.886
10	0.996	NAN	0.986	0.953	0.964	0.946	0.850	NAN	0.363	0.856	0.886	0.886

applying UDBB. The weighted average result for all the attacks are presented in bold. The results confirmed that the proposed framework with the reduced feature dimensionality achieved a maximum precision value of 0.996 and an FPR of 0.010, confirming the efficiency and effectiveness of the intrusion detection process. However, $(PCA-RF)_{Mc-10}$ is unable to detect the HeartBleed attacks (noted as NAN in Table V). In this Table, the Recall and Precision values for HeartBleed and WebAttack:SQL are 0.00, 0.000 and 0.000, 0.000, respectively. A justification of such outcome could be due to the fact that the number of instances of these attacks from the original class distribution is equal to 11 and 21, respectively. Thus, these instances were miss-classified. To resolve this issue and to assure that the achieved accuracy is reflected due to the effective reduction approach, this paper applies UDBB to overcome the imbalanced class distributions of certain attacks in CICIDS2017.

Table VII shows the performance before and after applying the UDBB approach with 10 features and different classifiers (X). As observed, $(PCA-RF)_{Mc-10}$ achieved 99.6% and 98.8% before and after applying UDBB, respectively. In the same manner, $(PCA-QDA)_{Mc-10}$ achieved 85.6% and 98.9% before and after applying UDBB, respectively. The highest achieved F-M was obtained by $(PCA-QDA)_{Mc-10}$. However, the highest $CM_{(Mc)}$ achieved was 98.6% by $(PCA-RF)_{Mc-10}$.

The performance evaluation of $(PCA-X)_{Bc-10}$ and $(PCA-X)_{Mc-10}$ in terms of the time to build and test the model is presented in Table VIII (X represents the classifier). The lowest times to test the model were achieved by LDA with 2.96 seconds for multi-class and 5.56 seconds for binary class classification. Here, the Random Forest classifier that has the best detection performance, comes with the highest overhead in terms of the time to build and test the model. This is expected since Random Forest combines many decision trees into a single model and specifically in this work, the dataset has over 2.5 million instances in total.

Moreover, a visualization of the dataset with two PCA components before and after applying UDBB is displayed in Figures 3 and 4. This observation of the CICIDS2017 dataset visually represents how the instances are set apart. As displayed in Figure 4, the same type of instances were positioned (clustered) together in groups. This shows a significant improvement over the PCA visualization before

TABLE V
PERFORMANCE EVALUATION BEFORE APPLYING UDBB

	$(PCA-RF)_{Mc-10}$ Original Distribution			
	Recall	Precision	FP Rate	TP Rate
Benign	0.998	0.998	0.012	0.998
FTP-Patator	1.000	1.000	0.000	1.000
SSH-Patator	0.996	0.996	0.000	0.996
DDoS	0.877	0.900	0.001	0.877
HeartBleed	NAN	NAN	0.000	0.000
PortScan	1.000	0.998	0.000	1.000
DoSHulk	1.000	1.000	0.000	1.000
DoSGoldenEye	0.979	0.995	0.000	0.979
WebAttack: Brute Force	0.813	0.878	0.000	0.814
WebAttack:XSS	0.750	0.665	0.000	0.750
Infiltration	0.250	1.000	0.000	0.250
WebAttack:SQL	0.000	0.000	0.000	0.000
Botnet	0.960	0.991	0.000	0.960
Dos Slow HTTP Test	0.993	0.996	0.000	0.993
DoS Slow Loris	0.991	0.999	0.000	0.991
Weighted Average	**0.996**	**0.965**	**0.010**	**0.996**

TABLE VI
PERFORMANCE EVALUATION AFTER APPLYING UDBB

	$(PCA-RF)_{Mc-10}$ UDBB			
	Recall	Precision	FP Rate	TP Rate
Benign	1.000	1.000	0.000	1.000
FTP-Patator	1.000	1.000	0.000	1.000
SSH-Patator	1.000	1.000	0.000	1.000
DDoS	1.000	1.000	0.000	1.000
HeartBleed	1.000	1.000	0.000	1.000
PortScan	1.000	0.999	0.000	1.000
DoSHulk	0.999	1.000	0.000	0.999
DoSGoldenEye	1.000	1.000	0.000	1.000
WebAttack: Brute Force	0.945	0.891	0.008	0.945
WebAttack:XSS	0.884	0.943	0.004	0.884
Infiltration	1.000	1.000	0.000	1.000
WebAttack:SQL	1.000	0.998	0.000	1.000
Botnet	1.000	1.000	0.000	1.000
Dos Slow HTTP Test	0.999	0.999	0.000	0.999
DoS Slow Loris	0.999	0.999	0.000	0.999
Weighted Average	**0.988**	**0.989**	**0.001**	**0.988**

TABLE VII
PERFORMANCE EVALUATION OF $(PCA-X)_{Mc-10}$

Classifier	Acc	F-M	$CM_{(Mc)}$
Before applying UDBB			
PCA-Random Forest $(PCA-RF)_{Mc-10}$	0.996	NAN	0.9866
PCA-Bayesian Network $(PCA-BN)_{Mc-10}$	0.953	0.964	0.9464
PCA-LDA $(PCA-LDA)_{Mc-10}$	0.850	NAN	0.3626
PCA-QDA $(PCA-QDA)_{Mc-10}$	0.856	0.886	0.8862
After applying UDBB			
PCA-Random Forest $(PCA-RF)_{Mc-10}$	0.988	0.988	0.9882
PCA-Bayesian Network $(PCA-BN)_{Mc-10}$	0.976	0.977	0.9839
PCA-LDA $(PCA-LDA)_{Mc-10}$	0.957	0.957	0.6791
PCA-QDA $(PCA-QDA)_{Mc-10}$	0.989	0.990	0.8851

applying UDBB. After UDBB, the normal instances are very clearly clustered in their own group. Same is the case for other types of instances as well.

A comparison between the proposed framework and related work is highlighted in Table IX. The time to build the model in [2] was 74.39 seconds. This is while the time for our proposed system using Random Forest is 21.52 seconds with a comparable processor. Furthermore, our proposed intrusion detection system targets a combined detection process of all the attack families. In contrast to [4], our research proposes an IDS to detect all types of attacks embedded in CICIDS2017 and achieves 100% accuracy for DDoS attacks using $(PCA-RF)_{Mc-10}$ with uniform distribution based balancing (UDBB). Moreover, unlike [7], we kept all the files of the dataset that represent different classes of the network traffic. The authors in [7], [25], [26] reported the accuracy. Our proposed framework outperforms previous studies in terms of F-Measure and accuracy.

TABLE VIII
TIME TO BUILD AND TEST THE MODELS

Classifier	Time to Build the Model (Sec.)	Time to Test the Model (Sec.)
Binary-class Classification		
LDA	12.16	5.56
QDA	12.84	6.57
RF	752.67	21.52
BN	199.17	11.07
Multi-class Classification		
LDA	17.5	2.96
QDA	15.35	3.16
RF	502.81	41.66
BN	175.17	10.07

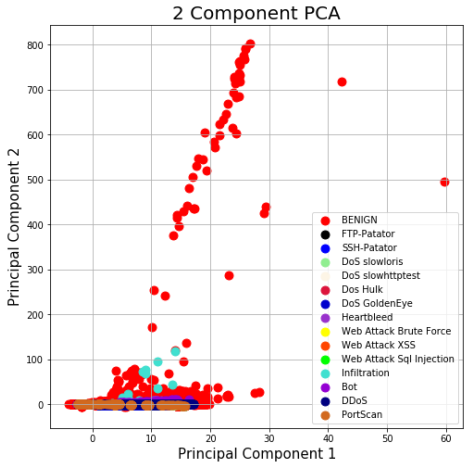

Fig. 3. 2D Visualization of PCA on CICIDS2017 without UDBB

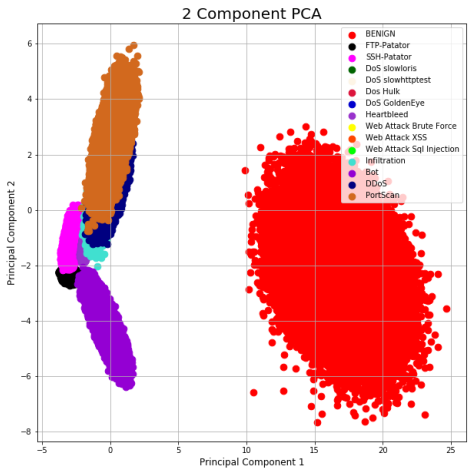

Fig. 4. 2D Visualization of PCA on CICIDS2017 with UDBB

VI. CONCLUSION AND FUTURE WORK

The aim of this research was to examine incorporating PCA for dimensionality reduction and the use of classifiers towards designing an efficient network intrusion detection system on the CICIDS2017 dataset. As exemplified from the obtained results, the PCA approach is able to preserve important information in CICIDS2017, while efficiently reducing the features dimensions in the used dataset, as well as presenting a reasonable visualization model of the data. The large number of decision trees that the Random Forest classifier produced by randomly selecting a subset of training samples and a subset of variables for splitting at each tree node, makes

TABLE IX
A COMPARISON OF THE PROPOSED FRAMEWORK AND PREVIOUS STUDIES

Reference	Classifier name	F-measure	Feature selection/extraction (Features Count)
[2]	KNN RF ID3 Adaboost MLP Naïve Bayes QDA	0.96 0.97 0.98 0.77 0.76 0.04 0.92	Random Forest Regressor (54)
[27]	MLP	0.948	Payload related features
[5]	SVM	0.921	DBN
[4]	KNN	0.997	Fisher Scoring (30)
[28]	XGBoost for DoS Attacks	0.995	(80)
[25]	Deep Learning for Port Scan Attacks	Accuracy 97.80	(80)
[25]	SVM for Port Scan Attacks	Accuracy 69.79	(80)
[7]	XGBoost	Accuracy 98.93	DDR Features Selections (36)
[26]	Deep Multi Layer Perceptron (DMLP) for DDoS Attacks	Accuracy 91.00	Recursive feature elimination with Random Forest
Proposed Framework	**Random Forest**	**0.996**	**PCA without UDBB (10)**
Proposed Framework	**Random Forest**	**0.988**	**PCA With UDBB (10)**

the Random Forest classifier less sensitive to the quality of training instances. Moreover, Random Forest is suitable, robust, and stable to classify high dimensional and correlated data as well as low dimensional data. Random Forest yielded better results in comparison with other classifiers. Features such as Subflow Fwd Bytes, Flow Duration, Flow Inter arrival time (IAT), PSH Flag Count, SYN Flag Count, Average Packet Size, Total Len Fwd Pck, Active Mean and Min, ACK Flag Count, and Init_Win_bytes_fwd are observed to be the discriminating features embedded in CICIDS2017 [2].

There are challenges and limitations including updating the training data for the IDS classifier models which calls for further investigations in the future. Newer IDS models should also be able to train themselves on-the-go as the packets traverse the network, for online network intrusion detection.

REFERENCES

[1] R. Abdulhammed, M. Faezipour, and K. Elleithy, *Intrusion Detection in Self organizing Network:A Survey*. NewYork: CRC Press Taylor Francis Group, 2017, ch. 13, pp. 393–449.

[2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of fourth international conference on information systems security and privacy, ICISSP*, 2018.

[3] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3: 322, pp. 1–27, 2019.

[4] D. Aksu, S. Üstebay, M. A. Aydin, and T. Atmaca, "Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm," in *International Symposium on Computer and Information Sciences*. Springer, 2018, pp. 141–149.

[5] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark," *IEEE Access*, 2018.

[6] A. Spark, "PySpark 2.4.0 documentation," 2018. [Online]. Available: https://spark.apache.org/docs/latest/api/python/index.html

[7] A. Bansal, "Ddr scheme and lstm rnn algorithm for building an efficient ids," Master's thesis, 2018.

[8] X. Xu and X. Wang, "An adaptive network intrusion detection method based on pca and support vector machines," in *International Conference on Advanced Data Mining and Applications*. Springer, 2005, pp. 696–703.

[9] A. George and A. Vidyapeetham, "Anomaly detection based on machine learning: dimensionality reduction using pca and classification using svm," *International Journal of Computer Applications*, vol. 47, no. 21, pp. 5–8, 2012.

[10] K. Ibrahimi and M. Ouaddane, "Management of intrusion detection systems based-kdd99: Analysis with lda and pca," in *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*. IEEE, 2017, pp. 1–6.

[11] A. Boukhamla and J. C. Gaviro, "Cicids2017 dataset: performance improvements and validation as a robust intrusion detection system testbed," *International Journal of Information and Computer Security*, 2018, in press.

[12] D. Xia, S. Yang, and C. Li, "Intrusion detection system based on principal component analysis and grey neural networks," in *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, April 2010, pp. 142–145.

[13] K. K. Vasan and B. Surendiran, "Dimensionality reduction using principal component analysis for network intrusion detection," *Perspectives in Science*, vol. 8, pp. 510–512, 2016.

[14] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, "The uci kdd archive of large data sets for data mining research and experimentation," *ACM SIGKDD explorations newsletter*, vol. 2, no. 2, pp. 81–85, 2000.

[15] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[16] Kolias, Constantinos and Kambourakis, Georgios and Stavrou, Angelos and Gritzalis, Stefanos, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[17] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Alessa, "Effective features selection and machine learning classifiers for improved wireless intrusion detection," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2018, pp. 1–6.

[18] M. Ring, S. Wunderlich, D. Grdl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," in *Proceedings of the 16th European Conference on Cyber Warfare and Security*, Conference Proceedings, pp. 361–369.

[19] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," *IEEE Sensors Letters*, vol. 3, no. 1, pp. 1–4, Jan. 2019.

[20] University of New Brunswick, "Canidian Institute of Cybersecurity, cicflowmeter," 2017, [Online; accessed 16-October-2017].

[21] CIC, "Canidian Institute of Cybersecurity, list of extracted traffic features by cicflowmeter-v3," https://www.unb.ca/cic/datasets/ids-2017.html, 2017, [Online; accessed 16-October-2017].

[22] T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," *Computers & Security*, vol. 73, pp. 137–155, 2018.

[23] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Improving the performance of naive bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2072–2080, 2011.

[24] V. García, J. S. Sánchez, and R. A. Mollineda, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 13–21, 2012.

[25] D. Aksu and M. A. Aydin, "Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*. IEEE, 2018, pp. 77–80.

[26] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*. IEEE, 2018, pp. 71–76.

[27] G. Watson, "A comparison of header and deep packet features when detecting network intrusions," Tech. Rep., 2018.

[28] A. Bansal and S. Kaur, "Extreme gradient boosting based tuning for classification in intrusion detection systems," in *International Conference on Advances in Computing and Data Sciences*. Springer, 2018, pp. 372–380.