International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Innovative Multi-Dataset Performance Analysis of Machine Learning Classifiers based on Features Reduction for Intrusion Detection

Salim Q. Mohammed[1], Mohammed A. El Sheikh Hussein[2]

Submitted: 04/10/2023 **Revised**: 25/11/2023 **Accepted**: 08/12/2023

Abstract: Computer users receive millions of internet packets every day, some are regular normal usage packets, others are packets sent by intruders for illegal purposes. With the increased numbers of users, regular countermeasure methods are no longer effective and Machine Learning is a key tool to deal with this increase of user numbers and attack types. Three well-known datasets, KDD99, UNSW NB15, and CICIDS2017 are used as a framework for a comprehensive comparison study were the proposed models deployed for performances measurements using a number of features reduction methods. Nine machine learning models that start with K-Nearest Neighbor, Logistic Regression, Support Vector Machine-Linear, Stochastic Gradient Descent, Nave Bayes, Decision Trees, Random Forest, Gradient Boosting to Adaboost are applied to the reduced features datasets of KDD99, UNSW NB15, and CICIDS2017. For a variety of features reductions, the accuracy and F1 score metrics have been used to evaluate and analyze each model's performance. For KDD99, the achieved accuracy and F1 scores are 99.9663% and 99.979%, respectively, and with the UNSW NB15, the values are 95.1968% and 96.2473%, respectively. Finally, the CICIDS2017 dataset values of 99.7004% for accuracy and 99.7515% for F1 were obtained. Random forest classifier showed the highest performances values using all the three datasets, and the innovative features reduction by 80% gave better outcomes of accuracy and F1, surpassing other state-of-the-art surveyed researches.

Keywords: *Binary Classifiers, Logistic Regression, Supervised Machine Learning Algorithms, Support Vector Machine (SVM).*

1. Introduction

Sniffers and attackers are increasing by numbers and used techniques. This means that regular techniques cannot cope with this rise in numbers and attack types. IDSs (Intrusion Detection Systems) were used a long time ago to reduce risks of attacks, which are considered as potential efforts to eliminate or reduce impact of these imminent threats and attacks [1, 2]. Various techniques, like cryptography, firewall, and access control were used to improve the security of data networks, but attacks are always evolving both in quantity and versatility [3, 4]. Nowadays, machine learning based IDS gained a lot of attention in the academic research community as well as by well-known companies like Google, Facebook and Intel, etc. which build a robust IDS capable of predicting, detecting and analyzing attacks. The goal is to increase the detection accuracy, minimize false alarms for both false positive and false negative detections [5, 6, 7].

1Ph.D. Student Technical College of Engineering, Sulaimani Polytechnic University
Sulaimanyah, Iraq
salim.muhammed@spu.edu.iq
2Ph.D.in Computer Science – Engineering College of Engineering, University of Sulaimani
Sulaimanyah, Iraq
mohammedabdullah.hussein@ univsul.edu.iq

In 2014, Mahmood et al. focused on a machine learning-based intrusion detection system employing unsupervised machine learning for binary detection. It has utilized the unsupervised K-means clustering technique with k=2 to label the data inputs into two classes: normal and abnormal. The NSL- KDD dataset was used with 41 selected attributes, which were then reduced down to the 23 most important features using IG (Information Gain). There are 60% training sets and 40% test sets in the proposed approach dataset. The research findings demonstrated that the technique had a high TPR (True Positive Rate) of 97.2% and a good accuracy of 97.22%, with a low FPR (False Positive Rate) of 2.9%. When compared to the scenario of employing all of the dataset's input features, the method to improve with fewer features required less processing time [8].

Almseidin et al. used different machine learning algorithms, such as J48, RF (Random Forest), RT (Random Tree), Decision Table, MLP (Multilayer Perceptron), NB (Naive Bayes), and Bayes Network, in 2017 for the categorization of intrusion detection systems. The models were carried out using the KDD dataset, with an emphasis on metrics for FP (False Positive) and FN (False Negative) rates. The decision table has a low FNR (False Negative Rate) of 0.2% and a high FPR (False

Positive Rate) of 7.3%, meaning that 7.3% of the data packets were inaccurately classified as intrusions [9].

In 2018, Belouch et al. worked on improving efficiency of detections. The IDS performance was evaluated for four supervised classifiers, SVM (Support Vector Machine), NB, DT (Decision Trees) and RF using Apache Spark. The UNSW-NB15 dataset was used with all its features and the obtained accuracy was 97.49 % using RF classifier [10].

The year of 2018 had seen the proposal of Classical machine learning methods and DNNs (Deep Neural Networks) for network IDS in cyber security by Vigneswaran et al. Both training and testing made use of the KDD99 dataset. DNN algorithms were compared to more established machine learning techniques, such as binary classifiers Boost, DT, KNN (K-Nearest Neighbor), Linear Regression, NB, RF, SVM-Linear, and SVM-rbf (Radial Basis Function). Using 0.1 learning rate and 1,000 epochs, the DNN was applied to a variety of layers, ranging from one to five layers. Best performance was achieved by DNN with three hidden layers compared with other used models [11].

In 2019, Devi et al. presented an IDS classifier using several supervised machine learnings; Logistic Regression, Decision Trees, KNN, SVM, RF, Adaboost, MLP and NB. The KDD99 and NSL-KDD datasets have been used. RF showed better performances for both datasets among others. 99.0 % and 99.7 % of accuracy have been achieved for mentioned datasets, respectively [12].

Also, an improved IDS classifier employing agent clustering and KNN models on preliminary edge detection was proposed by Sandosh et al. in 2019. In order to remove undesirable outlier data instances, the KDD99 dataset was initially preprocessed. The K-means clustering approach using an agent-based clustering subgroup clusters the unlabeled data. KNN has launched identification attacks to categorize the data received into recognized normal data and suspicious attack data. The enhanced intrusion detection system combining agent-clustering and KNN models performed better than conventional classifier models, according to empirical findings. According to a separate metric, the suggested model outperformed previous models in terms of accuracy, achieving 92.23% and a FNR of 0.7% [13].

Meryem et al. in developed a method for a hybrid intrusion detection system using machine learning techniques. Misuse detection and typical pattern signatures were coupled with NSL-KDD to enhance the model's detection capability for both anomaly and signature detections. K-means algorithm was used to cluster unlabeled data with KNN in accordance with the design. The KNN model has superior precision for all five classes, according to experimental results, with 98.80% accuracy, 99.80% precision, 98.80% recall, and a FPR of 0.9% [14].

Mohan et. al. focused on data mining classifications for IDS in 2020. PCA (Principal Component Analysis) was used to reduce the number of dimensional features and choose the NSL-KDD dataset. They employed RF, NB, Random Tree, and J48 as binary classifier models. The empirical findings showed that the RF outperformed all other classifiers in terms of performance, with an accuracy of 99.78% and a FPR of 0.1% [15]. In the same year, Abrar et al. sorted the data into five multi-classes using a variety of machine learning classifiers, including KNN, SVM, LR (Logistic Regression), MLP, NB, RF, DT, and ETC (Extra-Tree Classifier): four for intrusive data and one for regular data. The goal of the implementation was to increase detection prediction rates by reducing the number of very complicated features. The NSL-KDD dataset is initially preprocessed utilizing four distinct sub-groups of reduced dataset features. According to experimental findings, RF, DT, and ETC all performed with above 99% accuracy for all invasive classes in all sub-groups [16].

In 2020, Fitni et al. employed ensemble learning and feature selection techniques to improve an Anomaly-Based IDS. They used different classifier models such as Regression, DT and Gradient Boosting to detect intrusions with selecting 23 features of CIC-IDS2017 dataset. The accuracy of 98.8 % has been achieved by RF classifier [17].

Additionally, Iman et al. proposed an enhancement to the IDS in 2020 using the best Random Forest parameters to resolve the Boruta algorithm's difficulty with infinite loops. Entropy and Gini index were used as preprocessing with the NSL-KDD dataset's estimated selected features. The number of trees and various depth factors were employed with the Random Forest classifier. The experiment findings showed that the proposed design, which had a depth parameter of 7 to alleviate the infinite loop in the Boruta algorithm, and improved the running time and the number of iterations [18]. Waskle et al. at 2020 also presented a method for IDS based on unsupervised machine learning algorithm. They employed PCA to reduce dataset dimensionality. The RF classifier achieved accuracy of 96.78 % [19].

A hybrid IDS combining K-means, Random Forest, and DL (Deep Learning) algorithms were proposed by LIU et al. in 2021. They used a multi-stage architecture utilizing the Spark platform's K-means clustered with Random Forest binary classifier unsupervised machine learning technique. The model was trained and tested using the

NSL-KDD and CIC-IDS2017 datasets. To further classify data that had been altered by the first and second phases as normal or under attack, a deep learning stage was introduced. The response was rapid, with a considerable increase in accuracy. The study result demonstrated that the proposed technique, with quick response and minimal training time, obtained a high TPR for all different types of attacks. With the NSL-KDD dataset, the proposed model that is presented by Authors has achieved an accuracy of 85.24% and 99.91% using NSL-KDD and CIS-IDS2017 datasets, respectively [20]. SETH et al. developed an intelligent IDS by the same year (2021), employing numerous algorithms to identify various types of intrusions. The training and testing parts of the model's implementation employed the CIC-IDS2018 dataset. Performance of various machine learning methods, including RF, KNN, EGB (Extreme Gradient Boosting), Histogram Based GB (Gradient Boosting), Light GBM, DT and ETC, was assessed in terms of a number of parameters. The results revealed that the model had high invasive detection rates and a 97.4% accuracy rate [21].

Mohammed et al. in 2022 evaluated the performances of many traditional machine learning classifier models, KNN, SVM, NB, DT, RF, SGD (Stochastic Gradient Descent), GB (Gradient Boosting) and AdaBoosting models applied KDD99 dataset. The obtained accuracy and F1 score were 99.96% and 99.97%, respectively by Random Forest classifier [22].

This study aimed at developing reliable IDSs that are efficient in predicting, identifying, and evaluating attacks. Therefore, the objective reduces false alarms for both false positive and false negative suspected cases while also increasing the detection accuracy. Building an effective intrusion detection model that requires minimal training time and memory storage is challenging, particularly for online networks where thousands of terabytes are transported via the networks. The features of the input data necessary to train the model can be reduced in an effective way to achieve the specified objective. The input data is normalized using a standard equation to provide the best performance, and the reduction of the features has been utilized to decrease data attributes to 8 features, or a reduction of 80% of features per dataset. It uses the KDD99, UNSW-NB15, and CIC-IDS2017 datasets to assess the performance of several classifiers. The use of feature selection and reduction highlights how the importance of the features will vary depending on the dataset. Online IDS, which are directly linked to the internet, as a result, demand little processing and offer speedy detection. There are two speed-up advantages because there are only 8 features used and the information is extracted from the packet's header rather than the payload.

2. Methodology

The structural block diagram of the used scheme in this work is shown in **Fig.1**, it comprises multiple blocks, starting with the dataset, preprocessing, model training, test set, IDS classifier, and performance assessment blocks [23, 24]. The system works as follows: For each dataset, the data is prepared using the data wrangling preprocessing stage, which is essentially divided into two components, the first is feature selection. The eight highest ranks are utilized for each dataset after the most informative features are ranked using the information gain technique. Then, to improve the performance of the classifier model, all numerical values of the datasets for all samples are scaled to be between 0 and 1. Then, at a ratio of 70% to 30%, the data was split into train and test sets. The 30% test unseen data is then classified using the suggested model. Each model's performance is determined using a variety of indicators, including accuracy and F1 score.

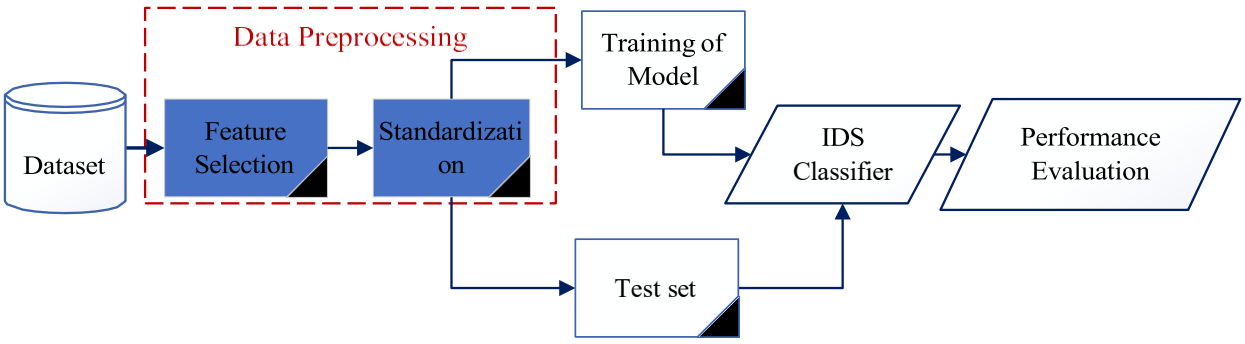


Figure 1. The used scheme

2.1 Datasets

Three well-known popular semi-structural datasets have been downloaded (on March 5th,2022) for preparation to train and test the performance of the used models, as follows:

2.1.1 KDD99 Dataset

Due to the requirement for a sizable reliable dataset for intrusion detection systems, the KDD99 (Knowledge Discovery and Data Mining) dataset is utilized. You can obtain the well-known standard benchmark dataset below to assess the effectiveness of machine learning-based IDS:

http://
kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

It has 5 million instances in the training dataset, 3 million instances in the testing dataset, and 10% of those instances are freely employed by the users. This translates to 494021 data points in the training set and 311029 data points in the testing set. There are five types that make up the KDD99 result or targeted labels. The first one is for regular data, while the others are for attack data. The types of attacks include Denial of Service, probing or port-scanning, Root to Local, and User to Root. The test set comprises 39 attack kinds, 17 of which are unknown attacks, while the training set has 22 attack types [25, 26, 27]. There are 41 feature attributes in the KDD99 dataset, nine of which contain discrete type features, and the remaining 36 are continuous types [22].

2.1.2 UNSW_NB15 Dataset

The Australian Center for Cyber Security generated the UNSW_NB15 dataset in 2015. There are almost two million records total with 49 features, ten classes, one normal class, and the remaining attack classes [28, 29, 30].

2.1.3 CICIDS2017 Dataset

The Canadian Institute of Cyber Security (CIC) developed the CICIDS2017 dataset in 2017. It consists of 15 classes. One is normal and 14 are abnormal data classes [31].

2.2 Data Preprocessing

The preprocessing of the datasets has been performed at first using information gain technique to select the most informative features in each dataset and rank them from higher to lower values. Highest most informative eighth features were selected per each dataset. The selection is highly important to reduce the dimensionality of the dataset for several reasons, such as decreasing overfitting, creating less complex classifier models that generalize data successfully, lowering the complexity of calculations, lowering the amount of memory that must be stored, shortening training times, and lowering false alarm rates. The confusion matrix, accuracy, error rate, precision, recall, false alarm, detection rate and f-score are just a few examples of the various performance evaluation metrics that have been employed. The implementation and simulations of the models have been achieved using Scikit-Learn library of Python 3 program. Pre-processing steps for a high dimensional features dataset with target labels are:

1) Utilizing one-hot encoding to change category features into numerical values.

2) Using IG ratio to select the most informative features and remove irrelevant or redundant features. For two random variables, the value of MI (Mutual Information), which estimates dependencies in input features, varies from 0 to 1. When the two random variables are independent, it equals zero, and when there are dependencies, it approaches values near one. Actually, it gauges how much knowledge can be obtained from one random variable given another [32, 33].

3) Standardizing input features values using the below equation which is called the standardization equation:

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x} \quad (1)$$

Here, μ_x is a mean value and σ_x is a standard deviation value [32]. This Eq. (1) reduces the model's sensitivity to the min contrast in min-max scaling, which limits the range of possible values for the data and preserves the relevant information about outliers.

2.3 Models' Description

Employing supervised machine-learning algorithms is to classification. The input data are split into training and testing sets, with 70% serving as training data and 30% serving as test data, and the model is trained to enable intrusion detection and classifications. The input data is divided into the normal and abnormal classes [34, 35]. The nine supervised machine learning algorithms used in this work are written below:

1 -KNN (K-Nearest Neighbor)
2 -LR (Logistic Regression)
3 -SVM-Linear
4 -SGD (Stochastic Gradient Descent)
5 -NB (Naïve Bayes)
6 -DT (Decision Trees)
7 -RF (Random Forest)
8 -GB (Gradient Boosting)
9-Adaboost

The performance of all nine models has been evaluated for three semi-structural datasets with a dedication of 70% for training and 30% for testing. This ratio remained

constant with respect to number of normal and attack records in each model as shown in **Table 1**.

Table 1. Number of normal and attack records in training and testing sets.

No.	Dataset	Type	Data Set		Total Classes
			Training 70%	Testing 30%	
1-	KDD99 Total: 494,021	Normal	68,094	29,184	97,278
		Attack	277,720	119,023	396,743
2-	UNSW-NB15 Total: 257,673	Normal	65,100	27,900	93,000
		Attack	115,271	49,402	164,673
3-	CIC-IDS2017 Total: 1,042,557	Normal	289,438	124,045	413,483
		Attack	440,351	188,723	629,074

2.4 Evaluation Metrics for Binary Classification

The four metrics that are utilized with these classifiers are arranged in a confusion matrix as follows

- TP refers to an attack that the model actually detected.
- TN is regular data that the model accurately detects.
- FP data is just regular data, but the model interprets it as an intrusion.
- FN attacks are actually detected as false positives by classifiers, which can lead to catastrophic security vulnerabilities. **Table 2** provides information about the confusion matrix [32, 33, 36].

Table 2. Confusion matrix.

Data Types	Predicted Classes	
	Predicted Normal	Predicted Attack
Normal Data	TN	FP
Attack Data	FN	TP

Evaluation performance is fullfilled by Accuracy, Error Rate, FPR, Precision (Specificity), Recall (Detection Rate), and F1-Score metrics using equations shown in **Table 3** [24].

Table 3. The formulas of the metrics.

No.	Measures	Equations
1-	Accuracy	(TP+TN) / (TP+TN+FP+FN)
2-	Error Rate	1- Accuracy
3-	FPR	FP / (TN+FP)
4-	Precision	TP / (TP+FP)
5-	Recall	TP / (TP+FN)
6-	F1-Score	(2* Precision* Recall) / (Precision+ Recall)

3. Analysis And Results

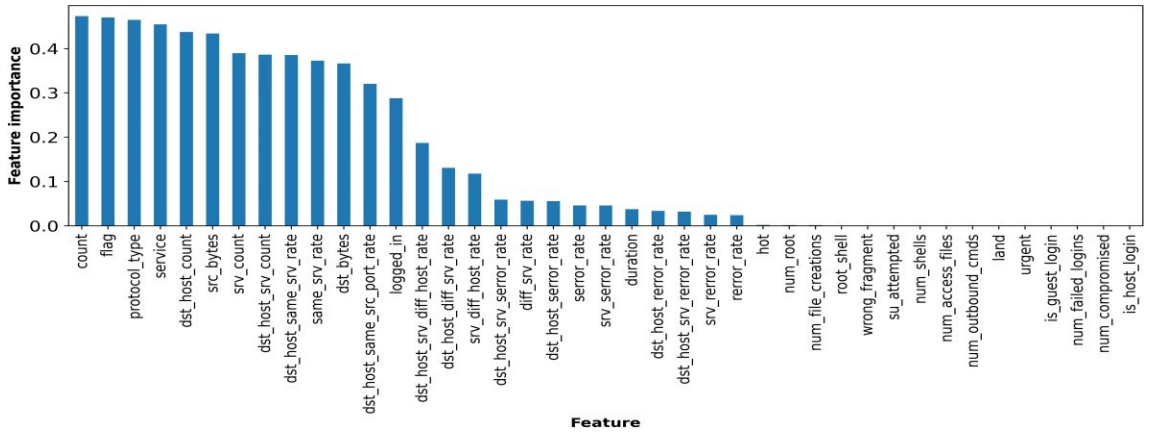
For all nine models and using all the datasets (KDD99, UNSW-NB15, and CIC-IDS2017), performances were evaluated using all features, 8 features, 6 features, 4 features, 2 features, and 1 feature. IG was used to select highest ranked informative features. Programming language Python 3.9.7 with libraries; NumPy 1.22.3, SciPy 1.7.1, Scikit-learn 1.0.2, Matplotlib 3.4.2, and Pandas 1.3.4 was used to do the required processing. The used operating system is Window 10 and the hardware platform is an MSI laptop with core i5 processor (i5-6267) and 6G bytes of RAM. The results are as below:

1) The eight most important feature scores for KNN, Logistic Regression, Linear SVM, SGD, and Naïve Bayes classifiers, ranked by IG using KDD99, UNSW-NB15, and CIC-IDS2017 datasets are shown in **Table 4**, arranged from highest to lowest.

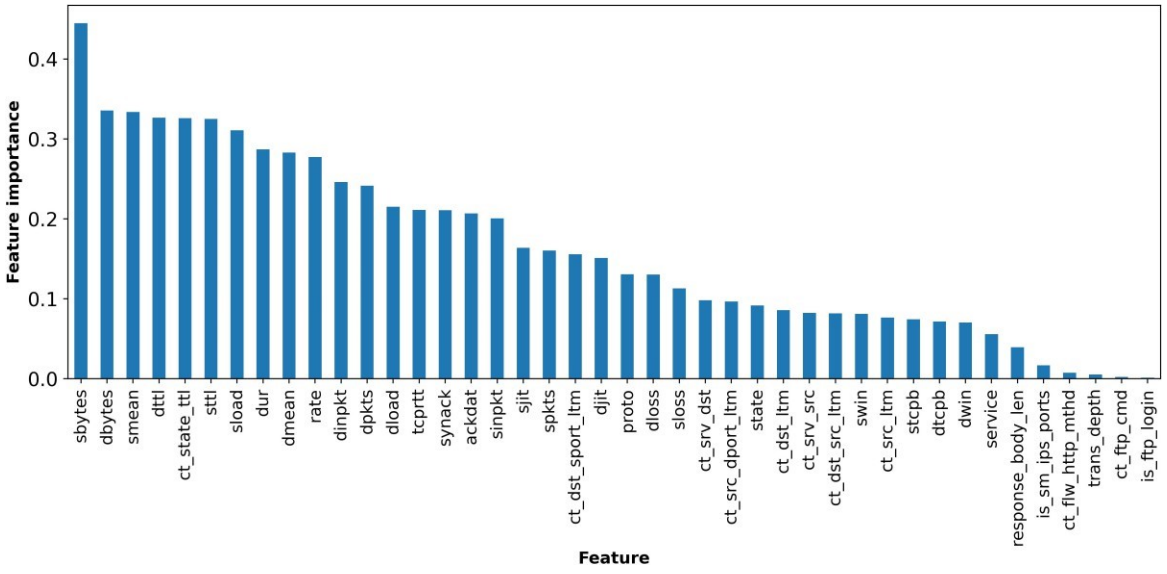
Table 4. The most 8 important features per dataset ranked by IG.

Dataset	No.	Feature	Name	IG Value
KDD99	1	f_{23}	count	0.473365
	2	f_4	flag	0.470284
	3	f_2	protocol_type	0.464793
	4	f_3	service	0.454624
	5	f_{32}	dst_host_count	0.437785
	6	f_5	src_bytes	0.434419
	7	f_{24}	srv_count	0.389739
	8	f_{33}	dst_host_srv_count	0.386193
UNSW-NB15	1	f_6	sbytes	0.444441
	2	f_7	dbytes	0.336140
	3	f_{26}	smean	0.333561
	4	f_{10}	dttl	0.326025
	5	f_{31}	ct_state_ttl	0.324081
	6	f_9	sttl	0.321886
	7	f_{11}	sload	0.310437
	8	f_0	dur	0.286249
CIC-IDS2017	1	f_{52}	Average Packet Size	0.524382
	2	f_0	Destination Port	0.494630
	3	f_{42}	Packet Length Variance	0.485854
	4	f_{41}	Packet Length Std	0.485663
	5	f_{40}	Packet Length Mean	0.475881
	6	f_5	Total Length of Bwd Packets	0.470539
	7	f_{65}	Subflow Bwd Bytes	0.469983
	8	f_{12}	Bwd Packet Length Mean	0.448970

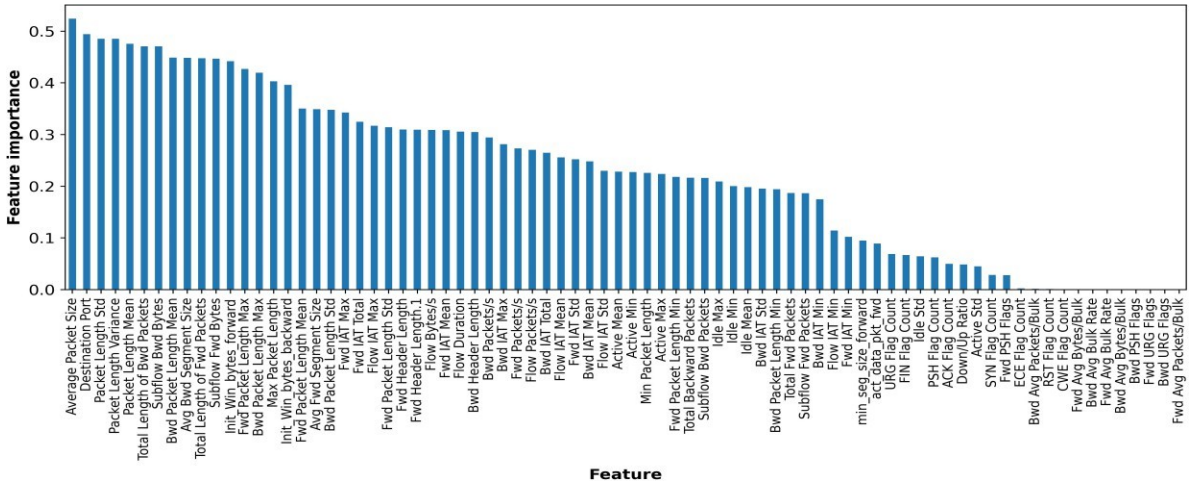
Also, the feature importance scores by dataset for the mentioned models using IG are ranked from the most important to the least one shown graphically in **Fig. 2.**



(a) KDD99 dataset



(b) UNSW-NB15 dataset



(c) CIC-IDS2017 dataset

Figure 2. Features importance per dataset ranked by IG.

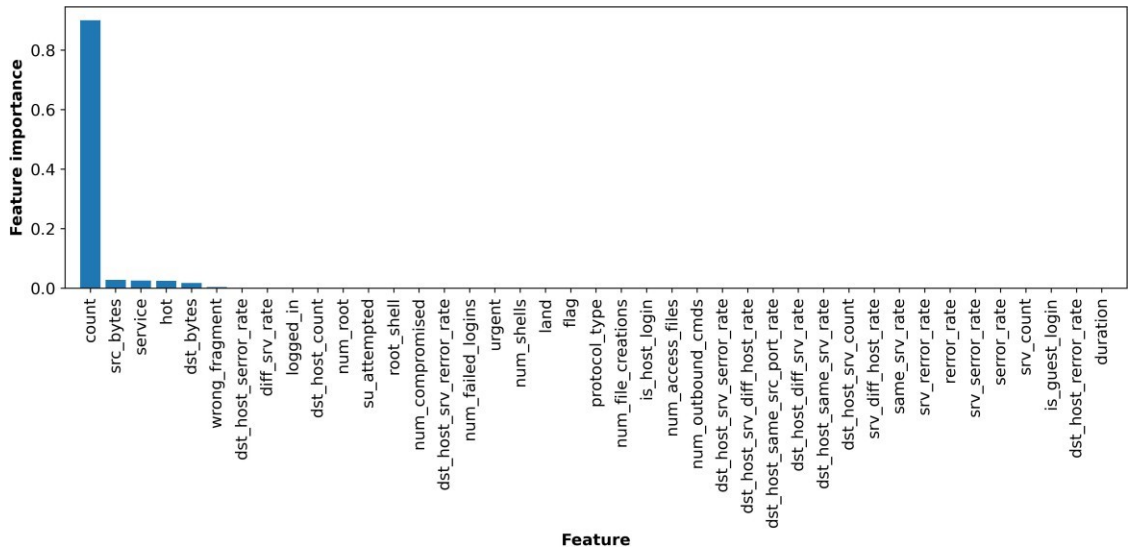
2) Performance evaluation metrics of KNN, RF, LR, Linear SVM, SGD, Naïve Bayes, DT, GB, and Adaboost classifiers using KDD99, UNSW-NB15 and CIC-IDS2017 datasets using different features selection processes are shown in **Table 5**.

Table 5. Performance evaluation metrics per model and dataset.

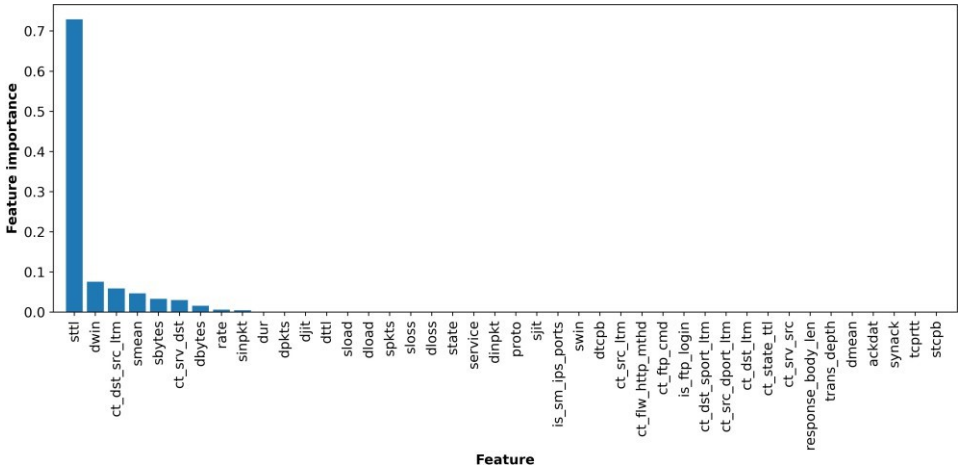
Model	Dataset	Metrics %	All F	8F	6F	4F	2F	1F
KNN	KDD99	Accuracy	99.96	99.93	99.92	99.94	99.49	89.78
		F1-score	99.97	99.96	99.95	99.96	99.68	93.93
	UNSW-NB15	Accuracy	93.96	94.05	92.81	92.66	91.99	73.42
		F1-score	95.28	95.37	94.35	94.24	93.69	75.76
	CIC-IDS2017	Accuracy	99.95	98.59	98.58	98.41	97.63	92.40
		F1-score	99.96	98.83	98.81	98.67	98.06	93.78
LR	KDD99	Accuracy	99.76	98.35	98.06	97.95	97.99	98.01
		F1-score	99.85	99.97	98.78	98.72	98.74	98.75
	UNSW-NB15	Accuracy	91.06	83.22	82.67	82.66	81.66	64.27
		F1-score	93.16	87.31	86.92	87.10	86.50	77.01
	CIC-IDS2017	Accuracy	98.63	78.99	79.07	78.28	78.15	76.97
		F1-score	98.86	84.22	84.28	83.56	83.37	82.31
Linear SVM	KDD99	Accuracy	99.94	99.57	98.95	98.57	98.39	97.99
		F1-score	99.97	99.73	99.35	99.10	98.99	98.74
	UNSW-NB15	Accuracy	94.75	91.41	91.05	89.05	86.99	86.99
		F1-score	95.95	93.53	93.28	91.75	90.62	90.62
	CIC-IDS2017	Accuracy	99.78	90.95	90.25	89.84	88.07	79.91
		F1-score	99.82	92.05	91.44	91.07	89.39	84.97
SGD	KDD99	Accuracy	99.74	98.29	98.05	98.04	97.96	98.01
		F1-score	99.84	98.93	98.77	98.76	98.72	98.75
	UNSW-NB15	Accuracy	90.85	87.13	85.56	85.56	85.05	85.50
		F1-score	93.01	90.83	89.80	89.80	89.39	89.75
	CIC-IDS2017	Accuracy	98.44	80.59	81.50	80.23	79.53	77.49
		F1-score	98.71	85.50	86.27	85.19	84.58	82.78
Naïve Bayes	KDD99	Accuracy	98.46	97.82	97.44	97.10	97.98	95.10
		F1-score	99.03	98.63	98.39	98.20	98.73	96.86
	UNSW-NB15	Accuracy	81.74	80.82	81.30	80.95	86.18	86.02
		F1-score	86.38	85.31	85.69	85.64	90.11	90.01
	CIC-IDS2017	Accuracy	87.36	76.71	76.85	76.87	75.86	77.38
		F1-score	88.47	82.08	82.20	82.22	81.30	82.68
DT	KDD99	Accuracy	99.61	99.61	99.61	99.54	99.04	98.01
		F1-score	99.75	99.75	99.75	99.71	99.39	98.75
	UNSW-NB15	Accuracy	92.03	92.03	91.98	91.71	87.44	87.10
		F1-score	93.89	93.89	93.86	93.67	90.57	90.79
	CIC-IDS2017	Accuracy	97.46	97.46	97.36	96.62	94.63	83.62
		F1-score	97.93	97.93	97.85	97.27	95.51	88.02

RF	KDD99	Accuracy	99.98	99.97	99.69	99.47	99.01	98.00
		F1-score	99.99	99.98	99.81	99.67	99.38	98.75
	UNSW-NB15	Accuracy	96.52	95.20	94.51	90.51	88.79	87.12
		F1-score	97.29	96.25	95.71	92.72	91.90	90.77
	CIC-IDS2017	Accuracy	99.95	98.65	98.22	98.21	97.69	94.66
		F1-score	99.96	98.87	98.54	98.54	98.11	95.54
GB	KDD99	Accuracy	99.70	97.73	97.73	97.73	96.89	95.15
		F1-score	99.82	98.58	98.58	98.58	98.07	96.91
	UNSW-NB15	Accuracy	92.60	92.27	92.11	91.48	87.22	87.06
		F1-score	94.47	94.22	94.09	93.62	90.90	90.79
	CIC-IDS2017	Accuracy	98.43	98.35	98.07	97.50	94.75	83.63
		F1-score	98.70	98.63	98.39	97.92	95.62	88.03
Ada boost	KDD99	Accuracy	99.94	99.84	99.79	99.81	99.58	98.43
		F1-score	99.96	99.90	99.87	99.88	99.74	99.03
	UNSW-NB15	Accuracy	94.18	93.47	93.61	87.38	84.18	70.27
		F1-score	95.48	94.93	95.05	89.95	87.60	79.59
	CIC-IDS2017	Accuracy	99.83	99.70	99.58	99.22	97.03	90.68
		F1-score	99.86	99.75	99.65	99.35	97.52	91.90

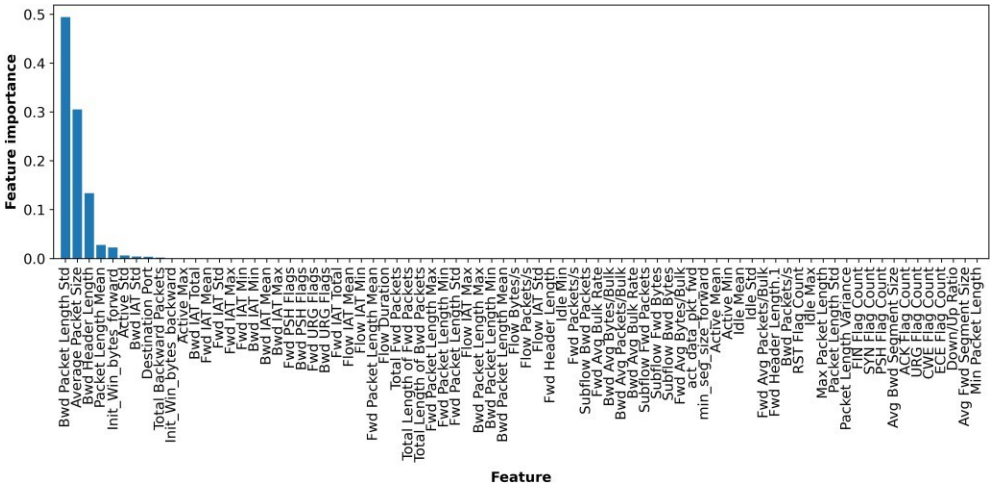
3) Most informative features for KDD99, UNSW-NB15, and CIC-IDS2017 datasets are used by Decision Trees classifier are shown graphically in **Fig. 3**



(a) KDD99 dataset



(b) UNSW-NB15 dataset



(c) CIC-IDS2017 dataset

Figure 3. Decision trees features importance per dataset ranked by IG.

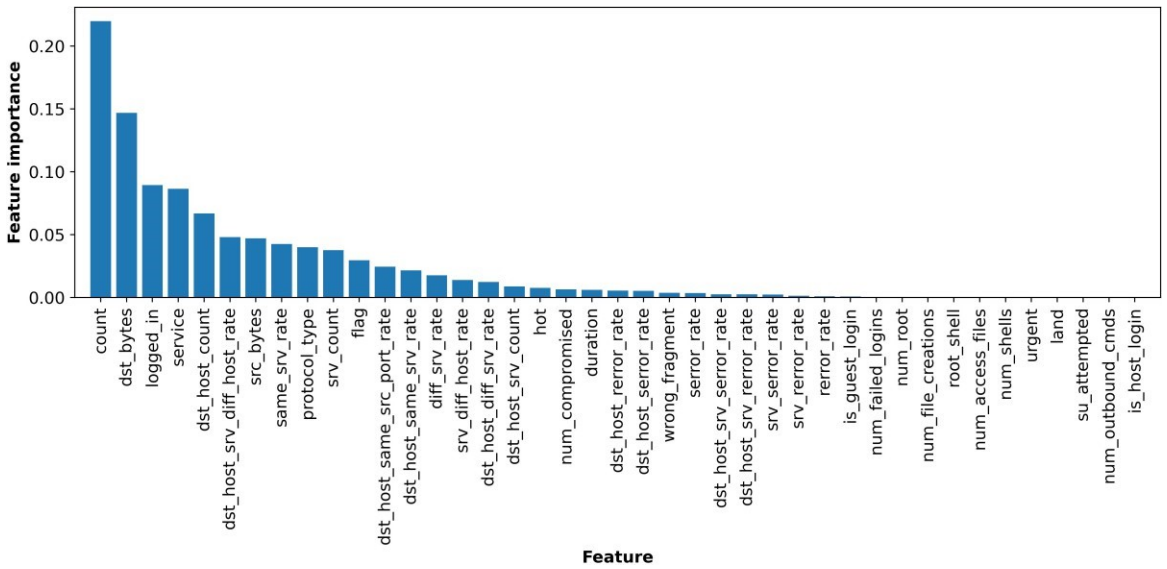
Analyzing feature importance per each dataset using Decision Trees classifier shows that most useful information is contained in few features. For example, the count feature (f_{23}) contains about 90 % of information for KDD99 dataset. The sttl feature (f_9) comprises more than 70% of information for UNSW-NB15 dataset, and the Bwd packet length Std feature (f_{13}) has nearly 50 % of information needed by the classifier using the CIC-IDS2017 dataset. **Table 6** shows feature importance scores per each dataset.

Table 6. Decision trees 8 important features per dataset ranked by IG.

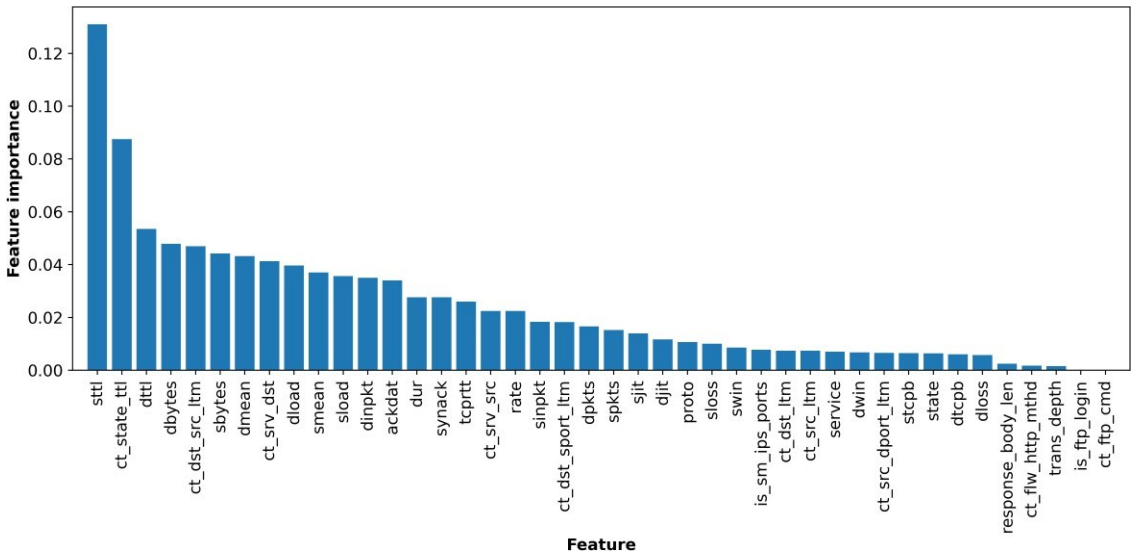
Dataset	No.	Feature	Name	IG Value
KDD99	1	f_{23}	count	0.900023
	2	f_5	src_bytes	0.028105
	3	f_3	service	0.025532
	4	f_{10}	hot	0.024754
	5	f_6	dst_bytes	0.017468
	6	f_8	wrong_fragment	0.003316
	7	f_{38}	dst_host_serror_rate	0.000670
	8	f_{30}	diff_srv_rate	0.000092

UNSW-NB15	1	f_9	sttl	0.729187
	2	f_{22}	dwin	0.075873
	3	f_{35}	ct_dst_src_ltm	0.058835
	4	f_{26}	smean	0.046731
	5	f_6	sbytes	0.032990
	6	f_{40}	ct_srv_dst	0.030048
	7	f_7	dbytes	0.015772
	8	f_8	rate	0.005970
CIC-IDS2017	1	f_{13}	Bwd Packet Length Std	0.494237
	2	f_{52}	Average Packet Size	0.305257
	3	f_{35}	Bwd Header Length	0.133945
	4	f_{40}	Packet Length Mean	0.027938
	5	f_{66}	Init_Win_bytes_forward	0.022834
	6	f_{71}	Active Std	0.006080
	7	f_{27}	Bwd IAT Std	0.004128
	8	f_0	Destination Port	0.003560

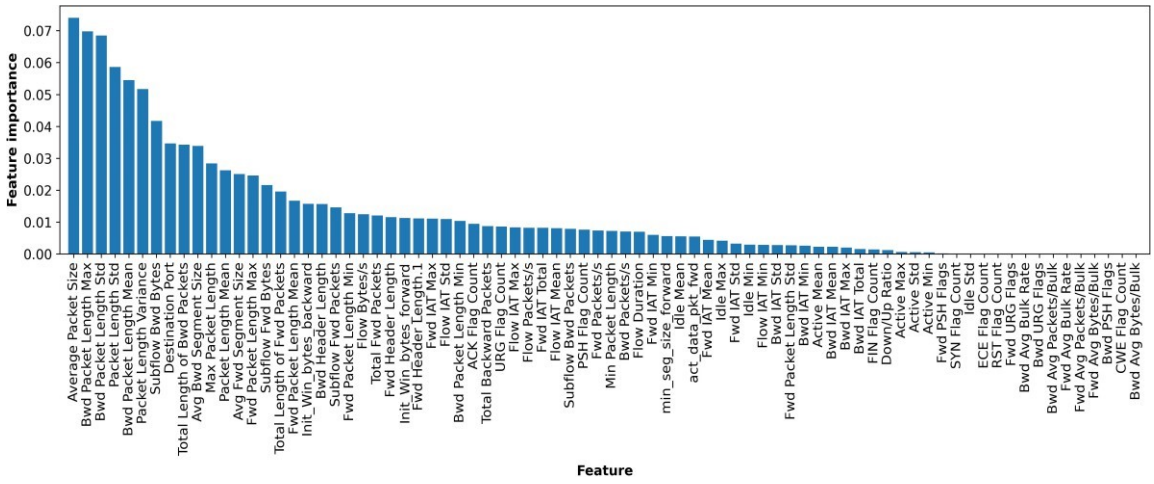
4) Most important features of datasets; KDD99, UNSW-NB15, and CIC-IDS2017 used by Random Forest classifier, ranked by IG are shown graphically in **Fig. 4**.



(a) KDD99 dataset



(b) UNSW-NB15 dataset



(c) CIC-IDS2017 dataset

Figure 4. Random forest features importance per dataset ranked by IG.

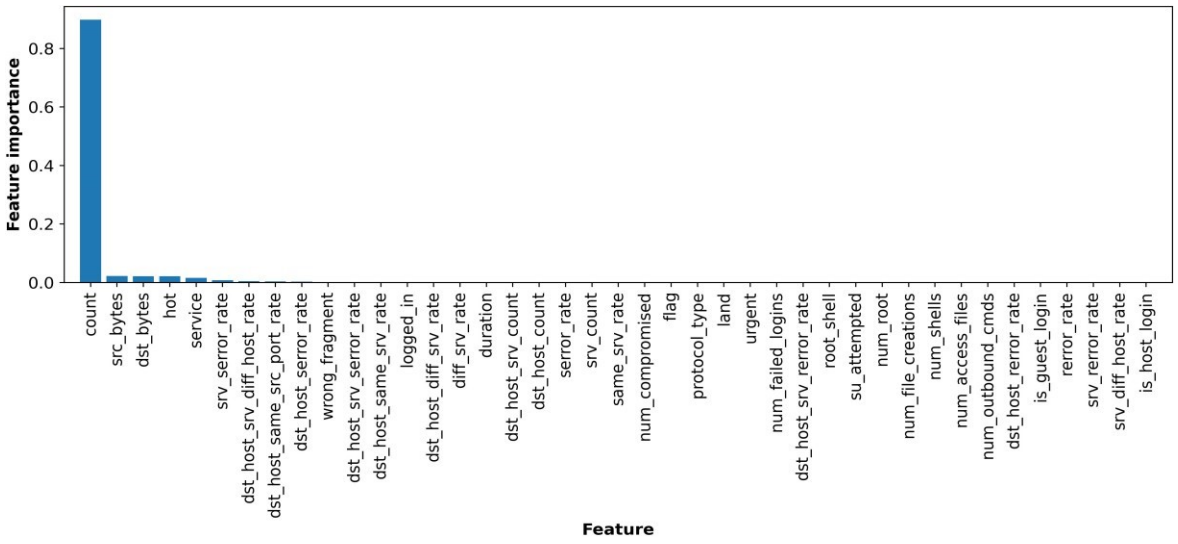
Eight most important features using KDD99, UNSW-NB15 and CIC-IDS2017 datasets by Random Forest classifier are shown in **Table 7**.

Table 7. Random forest 8 important features per dataset ranked by IG.

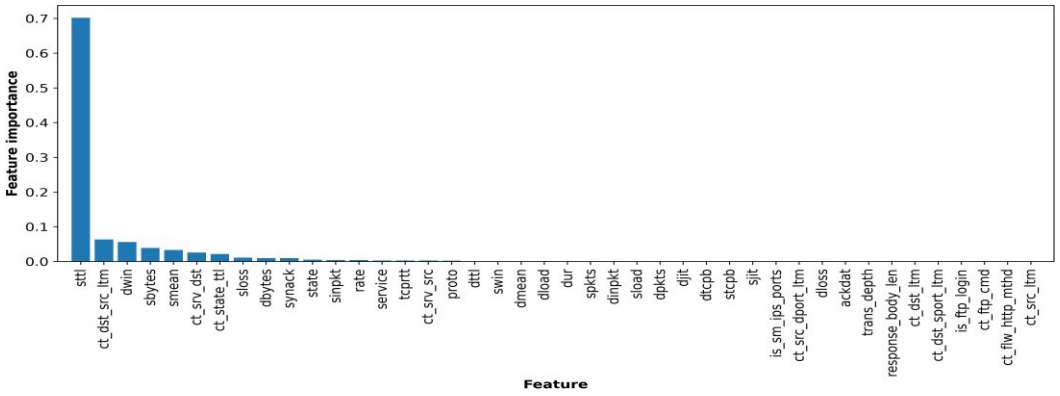
Dataset	No.	Feature	Name	IG Value
KDD99	1	f_{23}	count	0.219827
	2	f_6	dst_bytes	0.146822
	3	f_{12}	logged_in	0.089388
	4	f_3	service	0.086515
	5	f_{32}	dst_host_count	0.066904
	6	f_{37}	dst_host_srv_diff_host_rate	0.047942
	7	f_5	src_bytes	0.046921
	8	f_{29}	same_srv_rate	0.042519
UNSW-NB15	1	f_9	sttl	0.131009

	2	f_{31}	ct_state_ttl	0.087504
	3	f_{10}	dttl	0.053480
	4	f_{7}	dbytes	0.047857
	5	f_{35}	ct_dst_src_ltm	0.046928
	6	f_{6}	sbytes	0.044261
	7	f_{27}	dmean	0.043237
	8	f_{40}	ct_srv_dst	0.041297
CIC-IDS2017	1	f_{52}	Average Packet Size	0.074091
	2	f_{10}	Bwd Packet Length Max	0.069817
	3	f_{13}	Bwd Packet Length Std	0.068512
	4	f_{41}	Packet Length Std	0.058631
	5	f_{12}	Bwd Packet Length Mean	0.054541
	6	f_{42}	Packet Length Variance	0.051758
	7	f_{65}	Subflow Bwd Bytes	0.041719
	8	f_{0}	Destination Port	0.034718

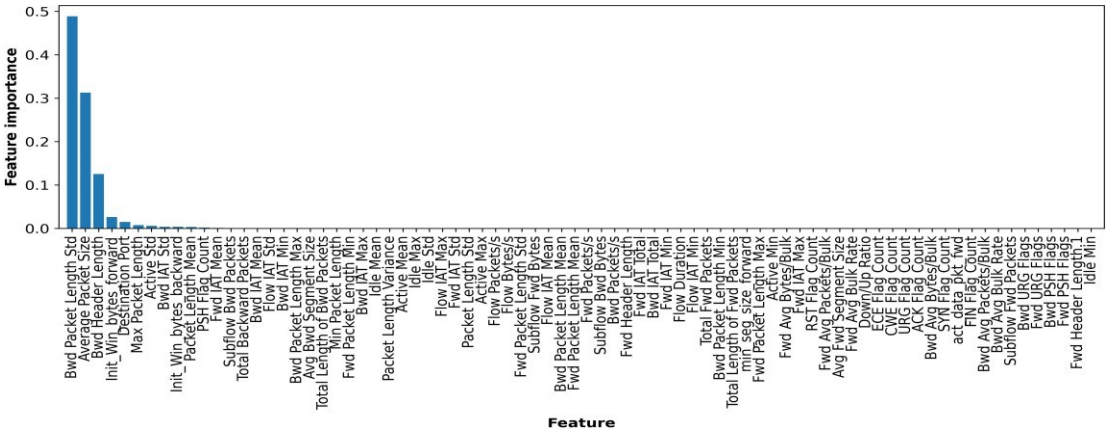
5) Most important features using KDD99, UNSW-NB15 and CIC-IDS2017 datasets by Gradient Boosting classifier, ranked by IG appear graphically in **Fig. 5**. It clearly shows that all needed information is contained only in one or two feature(s).



(a) KDD99 dataset



(b) UNSW-NB15 dataset



(c) CIC-IDS2017 dataset

Figure 5. Gradient boosting classifier features importance per dataset ranked by IG.

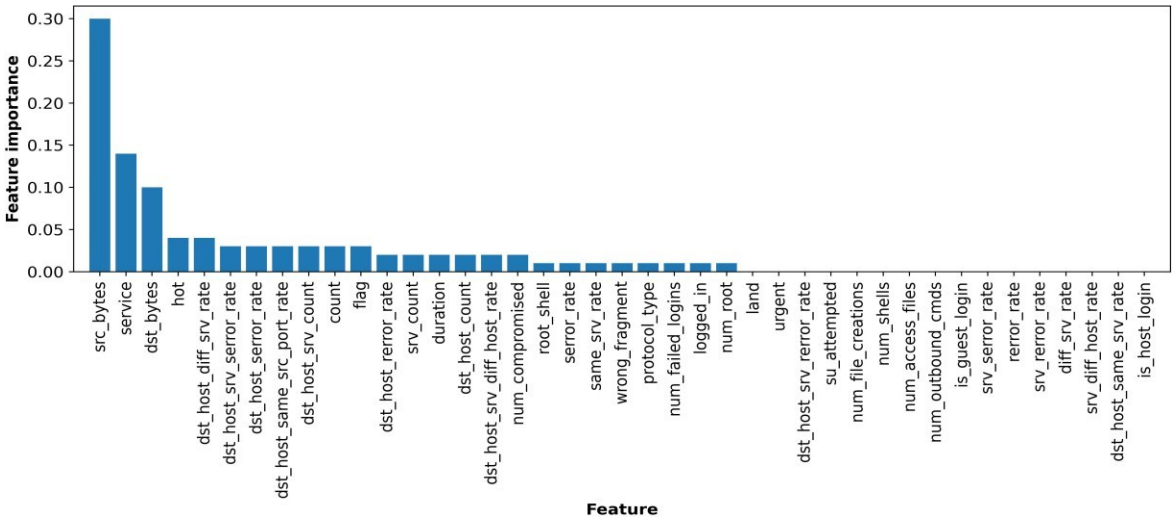
Table 8 includes the eight most important features for KDD99, UNSW-NB15, and CIC-IDS2017 datasets using Gradient Boosting model, ranked by IG.

Table 8. Gradient boosting 8 important features per dataset ranked by IG.

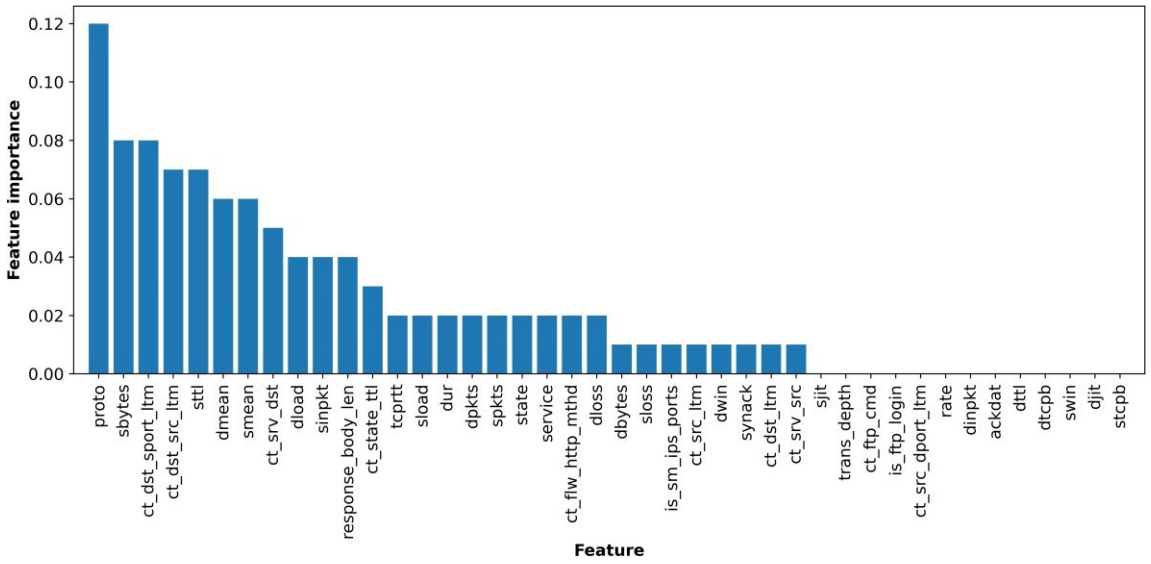
Dataset	No.	Feature	Name	IG Value
KDD99	1	f23	count	0.898363
	2	f5	src_bytes	0.022166
	3	f6	dst_bytes	0.021552
	4	f10	hot	0.021497
	5	f3	service	0.015555
	6	f26	srv_serror_rate	0.007729
	7	f37	dst_host_srv_diff_host_rate	0.004546
	8	f36	dst_host_same_src_port_rate	0.003254
UNSW- NB15	1	f9	sttl	0.702101
	2	f35	ct_dst_src_ltm	0.063536
	3	f22	dwin	0.056489
	4	f6	sbytes	0.039116
	5	f26	smean	0.033319
	6	f40	ct_srv_dst	0.026155

	7	f31	ct_state_ttl	0.021445
	8	f11	sloss	0.01109
CIC- IDS2017	1	f13	Bwd Packet Length Std	0.488723
	2	f52	Average Packet Size	0.312806
	3	f35	Bwd Header Length	0.125434
	4	f66	Init_Win_bytes_forward	0.026147
	5	f0	Destination Port	0.01512
	6	f39	Max Packet Length	0.007966
	7	f71	Active Std	0.005975
	8	f27	Bwd IAT Std	0.003924

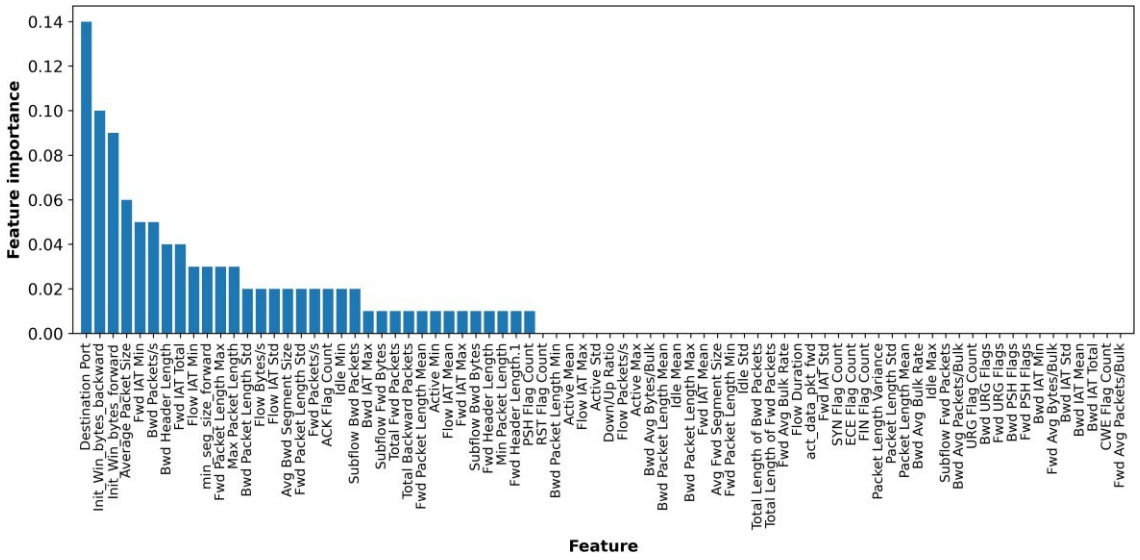
6) Fig. 6 is the most important features of KDD99, UNSW-NB15 and CIC-IDS2017 datasets using Adaboost classifier, ranked by IG graphically. The figure shows that useful information is distributed on many features.



(a) KDD99 dataset



(b) UNSW-NB15 dataset



(c) CIC-IDS2017 dataset

Figure 6. Adaboost features importance per dataset ranked by IG.

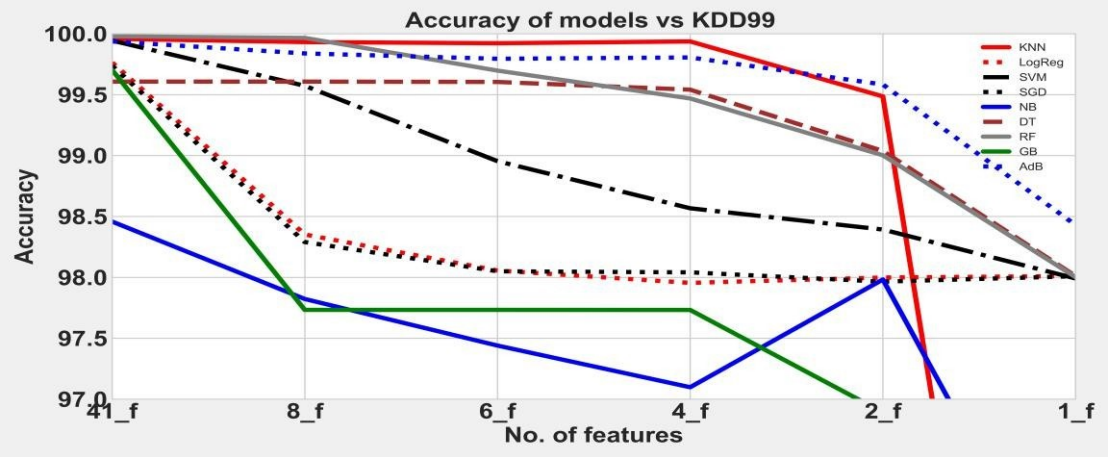
In **Table 9**, eight most important features for KDD99, UNSW-NB15 and, CIC-IDS2017 datasets using Adaboost classifier, ranked by IG are explained.

Table 9. Adaboost 8 important features per dataset ranked by IG.

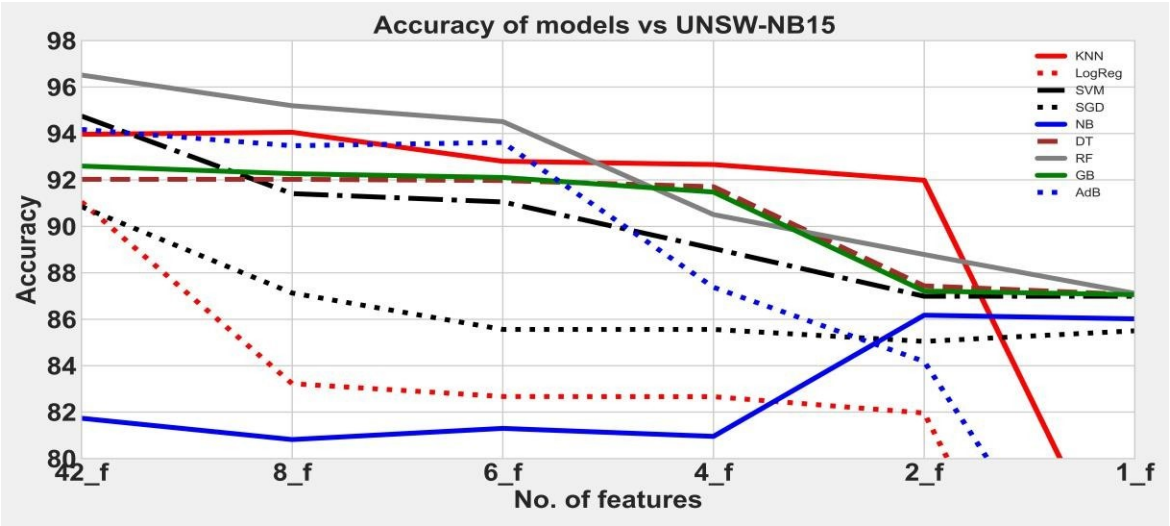
Dataset	No.	Fea-ture	Name	IG Value
KDD99	1	f_5	src_bytes	0.300000
	2	f_3	service	0.140000
	3	f_6	dst_bytes	0.100000
	4	f_{10}	hot	0.040000

	5	f35	dst_host_diff_srv_rate	0.040000
	6	f39	dst_host_srv_serror_rate	0.030000
	7	f38	dst_host_serror_rate	0.030000
	8	f36	dst_host_same_src_port_rate	0.030000
UNSW-NB15	1	f1	proto	0.120000
	2	f6	sbytes	0.080000
	3	f34	ct_dst_sport_ltm	0.080000
	4	f35	ct_dst_src_ltm	0.070000
	5	f9	sttl	0.070000
	6	f27	dmean	0.060000
	7	f26	smean	0.060000
	8	f40	ct_srv_dst	0.050000
CIC-IDS2017	1	f0	Destination Port	0.140000
	2	f67	Init_Win_bytes_backward	0.100000
	3	f66	Init_Win_bytes_forward	0.090000
	4	f52	Average Packet Size	0.060000
	5	f24	Fwd IAT Min	0.050000
	6	f37	Bwd Packets/s	0.050000
	7	f35	Bwd Header Length	0.040000
	8	f20	Fwd IAT Total	0.040000

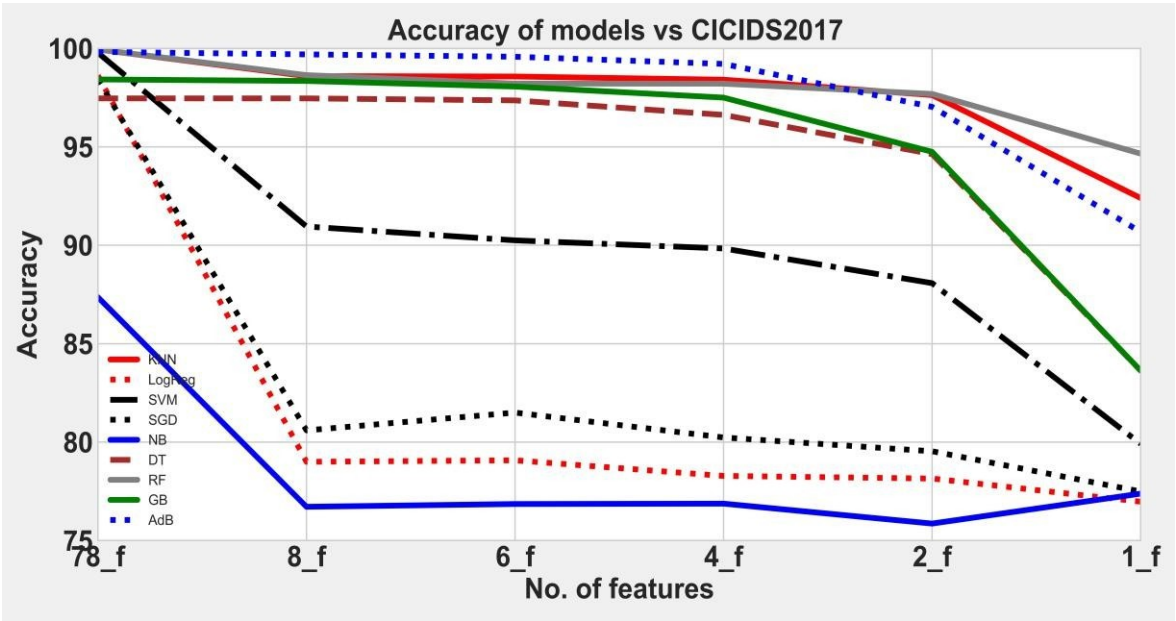
7) Using the KDD99, UNSW-NB15, and CIC-IDS2017 datasets, the accuracy of all nine models for all features, eight features, six features, four features, two features, and one feature are presented in Fig. 7. According to **Tables 12, 13, and 14**, the RF model has the greatest accuracy for all features and 8F. KNN model consistently outperforms other models for all features, including eight, six, and four. When the property number reduced, the majority of models perform worse.



(a) KDD99 dataset



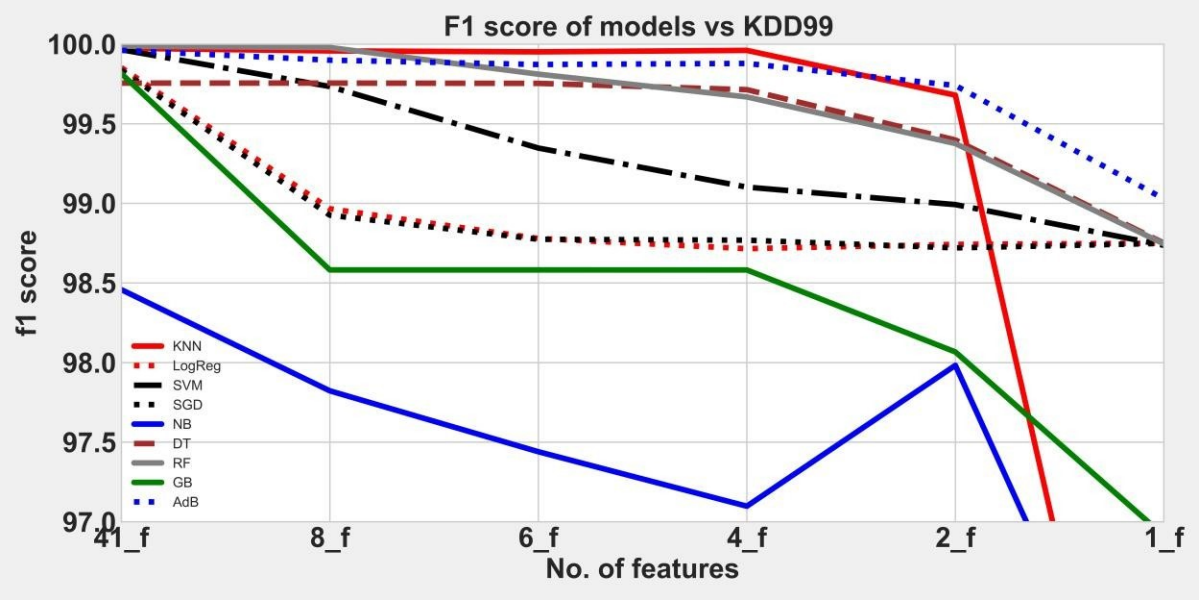
(b) UNSW-NB15 dataset



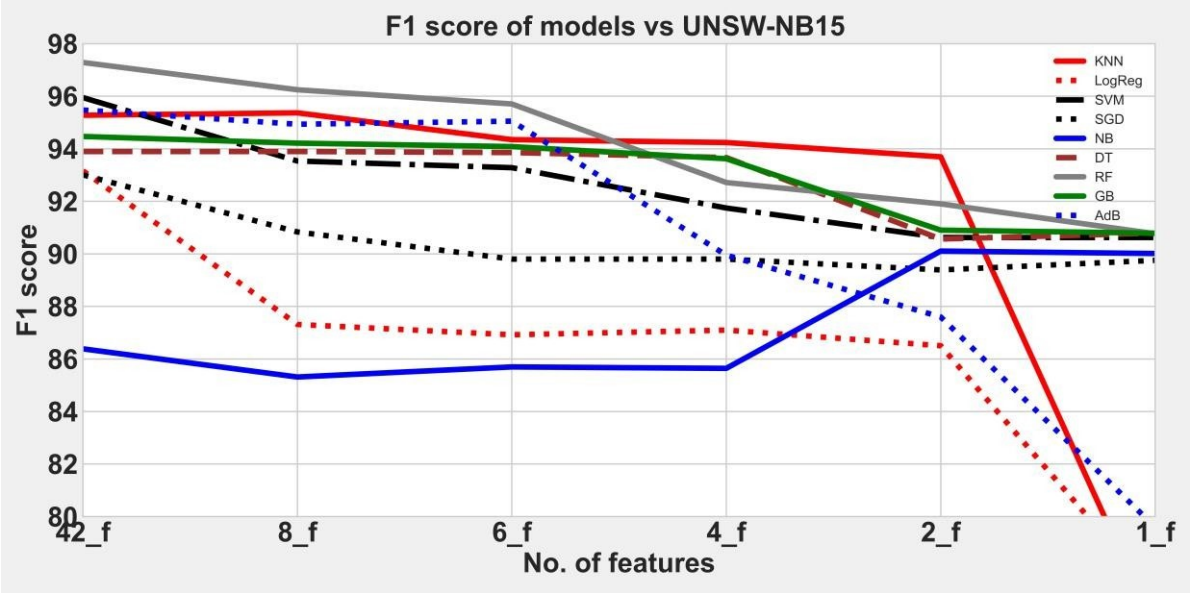
(c) CIC-IDS2017 dataset

Figure 7. Accuracy per model and dataset.

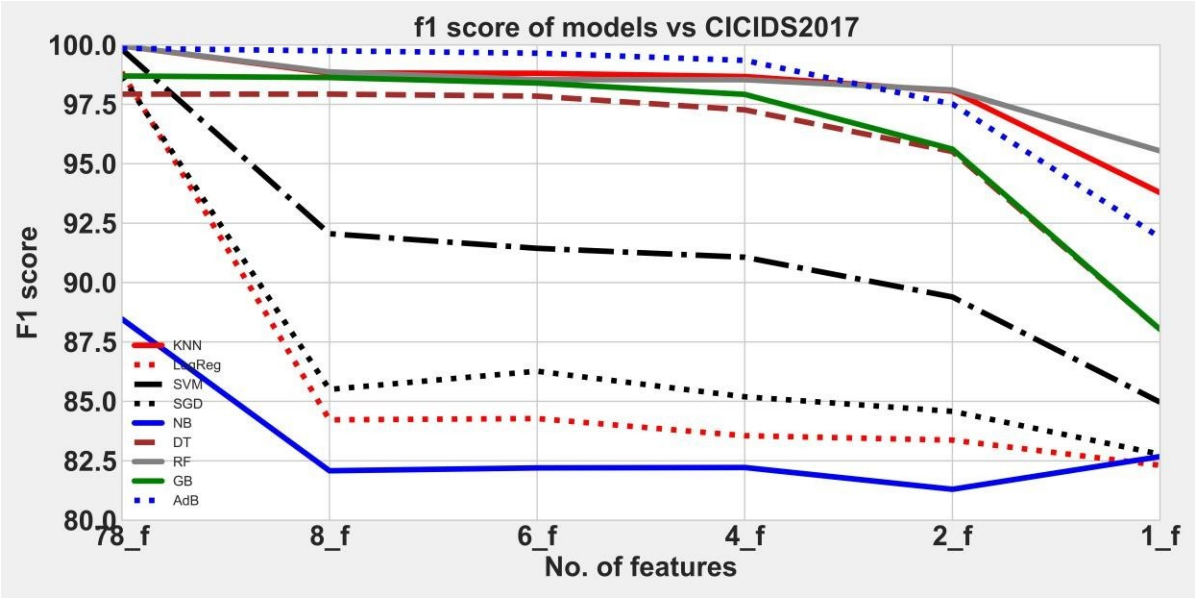
8) The F1-score metric for all models by 8, 6, 4, 2, and 1 feature(s) using KDD99, UNSW-NB15 and CIC-IDS2017 datasets are reported in **Fig. 8**.



(a) KDD99 dataset



(b) UNSW-NB15 dataset



(c) CIC-IDS2017 dataset

Figure 8. F1 score per model and dataset.

4. Comparison And Discussion

The outcomes of this work are compared with what was obtained by [8, 11, 37]. Results are tabulated and shown in **Table 10**. The table clearly shows the better achieved values and the contribution of this work (second column).

Table 10. Comparing this work and others.

Type	This work	[8]	[11]	[37]
Parameter	Value %	Value %	Value %	Value %
Accuracy	99.9615	97.22	92.7	99.6
Error Rate	0.0385	2.78	-	-
True Positive Rate	99.9621	97.2	-	-
False Positive Rate	0.0411	2.9	-	-
Precision	99.9899	97.2	99.9	99
Recall	99.9622	97.2	91	98
F1_score	99.9761	97.2	95.3	99.8

Key points that could be deduced from this work:

1- Accuracy values using Decision Trees classifier by all features and 8 features were equal for all the three datasets, as shown in **Table 11**. The reason for this is that this classifier obtains the most important information from many features and not few ones, as in **Fig. 7**.

Table 11. Decision trees classifier accuracy.

No.	Dataset	All features accuracy %	8 features accuracy%
1-	KDD99	99.6073	99.6073
2-	UNSW-NB15	92.0274	92.0274
3-	CIC-IDS2017	97.4649	97.4639

2- Accuracy and F1 scores for all classifier models have the highest values with KDD99 dataset, then the CIC-IDS2017 dataset and the last is the UNSW-NB15 dataset. This is due to imbalances between attacks and normal record numbers in the datasets. The KDD99 dataset is with 80.3% attack records, which means it is biased toward attacks, while the percentages in the UNSW-NB15 and CIC-IDS2017 datasets are 63.9% and 60.33%, respectively. Also, the number of records (datapoints) in the CIC-IDS2017 dataset is four times the number of records in the UNSW-NB15 dataset, as it was shown in **Table 1**. With the increasing datapoints, accuracy and F1 score values will enhance.

3- The highest accuracy and F1 score values for KDD99 dataset are in **Table 12**.

Table 12. KDD99 dataset highest performance values.

No.	Used features	Classifier	Accuracy%	F1 score%
1-	All	Random Forest	99.9798	99.9874
2-	8	Random Forest	99.9663	99.979
3-	6	KNN	99.9224	99.9517
4-	4	KNN	99.9372	99.9609
5-	2	Adaboost	99.5837	99.7405
6-	1	Adaboost	98.4299	99.0284

4- The highest Accuracy and F1 score values for UNSW-NB15 dataset appear in **Table 13**.

Table 13. UNSW-NB15 dataset highest performance values.

No.	Used features	Classifier	Accuracy%	F1 score%
1-	All	Random Forest	96.5176	97.2862
2-	8	Random Forest	95.1968	96.2473
3-	6	Random Forest	94.5124	95.7074
4-	4	KNN	92.6638	94.239
5-	2	KNN	91.9886	93.6952
6-	1	Random Forest	87.1194	90.7671

5- The highest Accuracy and F1 score values for CIC-IDS2017 dataset are illustrated in **Table 14**.

Table 14. CIC-IDS2017 dataset highest performance values.

No.	Used features	Classifier	Accuracy%	F1 score%
1-	All	Random Forest	99.9485	99.9573
2-	8	Adaboost	99.7004	99.7515
3-	6	Adaboost	99.5808	99.6524
4-	4	Adaboost	99.2157	99.3494
5-	2	Random Forest	97.6906	98.1055
6-	1	Random Forest	94.6558	95.5412

6- Although Random Forest classifier has the highest Accuracy and F1 score values using all features, KNN classifier shows better Detection Rates (Recall), which means less False Negative rates, a point needs to be considered in some application fields.

5. Conclusion

In the scope of this work, many binary classifiers that work based on rules, distances and probability approaches were realized using three widely used semi-structural datasets. The most commonly used evaluation metrics are applied to evaluate the classifiers, supported by tables and figures throughout the paper and the outcomes are shown below:

- Reducing the required numbers of features by 80% achieved a dual enhancement effect of

increasing the speed of detection using less memory.

- Performances values have been improved. Best performance value was with the KDD99 dataset, with an accuracy of 99.96% and an error rate of 0.038%.

Minors to the above points are:

- Information gain technique is used for feature selection; therefore, eight features are used instead of all features to improve model's performances.
- The models were implemented with all features, 8 features, 6 features, 4 features, 2 features and 1 feature. The Random Forest classifier showed a unique outcome and the best performance for all datasets after feature reduction by 80%.
- The packet header, not the data, is used to extract the features. Consequently, online IDSs, which are directly connected to the internet, require minimum processing and provide quick detection. Due to working with 8 features only and extracting data from the packet's header rather than the payload, a dual speed-up enhancement is achieved.
- Random Forest classifier is our nominee as it achieved the best performances and metrics values in all of the three datasets.

References

[1] Mebawondua, J., 2020. Network Intrusion Detection System using Supervised Learning Paradigm. Elsevier, 24 July. Doi:10.1016/j.sciaf.2020.e00497

[2] Al-Garadi, M., 2020. A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Communications Surveys & Tutorials, 22(3), pp. pp.1646- 1685.Doi:10.1109/comst.2020.2988293

[3] Azhagiri, M., Rajesh, D. A. and Karthik, D. S., 2015. Intrusion Detection and Prevention System: Technologies and Challenges. International Journal of Applied Engineering Research, 10(87). https://www.researchgate.net/publication/287208734_intrusion_detection_and_prevention_system_tchnologies_and_challenges

[4] Anwar, S., 2017. From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions. Algorithms. MDPI algorithms, 10(2), p. 39.Doi:10.3390/a10020039

[5] Gupta, A. R. b. and Agrawal, J., 2020. A Comprehensive Survey on Various Machine Learning Methods used for Intrusion Detection System. 9th IEEE International Conference on Communication Systems and Network Technologies, 16 June. pp. 282-289.Doi:10.1109/csnt48778.2020.9115764

[6] Ahmad, Z., 2021. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1).Doi:10.1002/ett.4150

[7] Daniya, T., Kumar, K. S., Kumar, B. S. and Kolli, C. S., 2021. A Survey on Anomaly based Intrusion Detection System. ELSEVIER, 12 March. Doi:10.1016/j.matpr.2021.03.353

[8] Mahmood, D. Y. and Hussein, M. A., 2014. Feature based Unsupervised Intrusion Detection. International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8(9), pp. 1515-1519.https://www.researchgate.net/publication/317730391_feature_based_unsupervised_intrusion_detection

[9] Almseidin, M., Alzubi, M., Kovacs, S. and Alkasassbeh, M., 2017. Evaluation of Machine Learning Algorithms for Intrusion Detection System. In 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, September. https://www.researchgate.net/publication/322328775_evaluation_of_machine_learning_algorithms_for_intrusion_detection_system

[10] Belouch, M., 2018. Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, Volume 127, pp. pp.1-6.Doi:10.1016/j.procs.2018.01.091

[11] Rahul, V., KP, S. and Poornachandran, P., 2018. Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. In 2018 9th International conference on computing, communication and networking technologies (ICCCNT). IEEE, July. pp. 1-6.Doi:10.1109/icccnt.2018.8494096

[12] Devi, R. R. and Abualkibash, M., 2019. INTRUSION DETECTION SYSTEM CLASSIFICATION USING DIFFERENT MACHINE LEARNING ALGORITHMS ON KDD-99 AND NSL-KDD DATASETS. International Journal of Computer Science & Information Technology (IJCSIT), 11(3).Doi:10.5121/ijcsi.2019.11306

[13] Sandosh, S., Govindasamy, V. and Akila, G., 2020. Enhanced Intrusion Detection System via Agent Dlustering and Classification based on Outlier Detection. Peer-to-Peer Networking and Applications, 13(3), pp. 1038-1045.Doi:10.1007/s12083-019-00822-3

[14] Meryem, A. and Ouahidi, B. E., 2020. Hybrid Intrusion Detection System using Machine Learning.

Network Security, May, 2020(5), pp. 8-19.Doi:10.1016/s1353-4858(20)30056-8

[15] Mohan, L., Jain, S., Suyal, P. and Kumar, A., 2020. Data mining Classification Techniques for Intrusion Detection System. IEEE, 12th International Conference on Computational Intelligence and Communication Networks, 20 December. Doi:10.1109/cicn49253.2020.9242642

[16] Abrar, I., Ayub, Z., Masoodi, F. and Bamhdi, A. M., 2020. A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. In 2020 International Conference on Smart Electronics and Communication (ICOSEC). IEEE, September. pp. 919-924.Doi:10.1109/icosec49089.2020.9215232

[17] Fitni, Q. a. R. K., 2020. Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pp. (pp. 118-124). IEEE. Doi:10.1109/iaict50021.2020.9172014

[18] Iman, A. N. and Ahmad, T., 2020. Improving Intrusion Detection System by Estimating Parameters of Random Forest in Boruta. In 2020 International Conference on Smart Technology and Applications (ICoSTA). IEEE, February. 1-6. Doi:10.1109/icosta48221.2020.1570609975

[19] Waskle, S. P. L. a. S. U., 2020, July. Intrusion detection system using PCA with random forest approach. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. (pp. 803-808). IEEE. Doi:10.1109/icesc48915.2020.9155656

[20] Liu, C., Gu, Z. and Wang, J., 2021. A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning. IEEE Access, May, Volume 9, pp. 75729-75740.Doi:10.1109/access.2021.3082147

[21] Seth, S., Chahal, K. K. and Singh, G., 2021. A Novel Ensemble Framework for an Intelligent Intrusion Detection System. IEEE Access, 29 September, Volume 9, pp. 138451-138466.Doi:10.1109/access.2021.3116219

[22] Mohammed, S. Q. and Hussein, M. A., 2022. Performance Analysis of different Machine Learning Models for Intrusion Detection Systems. Journal of Engineering, 28(5).Doi:10.31026/j.eng.2022.05.05

[23] Agrawal, D. and Agrawal, C., 2020. A Review on Various Methods of Intrusion Detection System. Computer Engineering and Intelligent Systems, 31 January .11(1).Doi:10.7176/ceis/11-1-02

[24] Bertoli, G. D. C., 2021. An End-To-End Framework for Machine Learning-Based Network Intrusion Detection System. IEEE Access, 27 July, Volume 9, pp. 106790-106803.Doi:10.1109/access.2021.3101188

[25] Zhang, B., 2018. Network Intrusion Detection Method Based on PCA and Bayes Algorithm. Security and Communication Networks, Research Article, 17 October. Doi:10.1155/2018/1914980

[26] Zhu, H., Liu, W., Sun, M. and Xin, Y., 2017. A Universal High-Performance Correlation Analysis Detection Model and Algorithm for Network Intrusion Detection System. Mathematical Problems in Engineering. Doi:10.1155/2017/8439706

[27] Xin, Y., 2018. Machine learning and deep learning methods for cybersecurity. IEEE Access, pp. 35365-35381.Doi:10.1109/access.2018.2836950

[28] Hooshmand, M. a. G. I., 2020. Feature selection approach using ensemble learning for network anomaly detection. CAAI Transactions on Intelligence Technology, 5(4), pp. pp.283-293. Doi:10.1049/trit.2020.0073

[29] Al-Daweri, M. S., Ariffin, K. A. Z., Abdullah, S. and Senan, M., 2020. An Analysis of the KDD99 and UNSW-NB15 Datasets for the Intrusion Detection System. Symmetry, 12(10), p. 1666.Doi:10.3390/sym12101666

[30] Li, G., Yan, Z., Fu, Y. and Chen, H., 2018. Data Fusion for Network Intrusion Detection: A Review. Security and Communication Networks. Doi:10.1155/2018/8210614

[31] Abdulhammed, R. M. H. A. A. F. M. a. A. A., 2019. Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics, 8(3), p. p.322.Doi:10.3390/electronics8030322

[32] Raschka, S., Liu, Y.and Mirjalili, V., 2022. Machine Learning with PyTorch and Scikit-Learn. Birmingham B3 2PB, UK.: Packt Publishing.

[33] Klosterman, S., 2021. Data Science Projects with Python. UK: Birmingham B3 2PB.

[34] Mukhopadhyay, S., 2018. Advanced Data Analytics using Python: with Machine Learning, Deep Learning and nlp Examples. Kolkata, West Bengal, India: Apress.

[35] Müller, A. C. and Guido, S., 2016. Introduction to Machine Learning with Python: A Guide for Data Scientists. First ed. s.l.:O'Reilly.

[36] Salih, A. and Abdulazeez, A., 2021. Evaluation of Classification Algorithms for Intrusion Detection System. A Review. Journal of Soft Computing and Data Mining

(JSCDM), 15 April, 2(1), pp. 31-40.Doi:10.30880/jscdm.2021.02.01.004

[37] Hidayat, I., Muhammad, Z. A. and Arshad, A., 2023. Machine Learning -Based Intrusion Detection System:An Experimental Comparison. Journal of Computational and Cognitive Engineering, Volume 2(2), pp. 88-97.Doi:10.47852/bonviewjcce2202270