ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse93942020.pdf
https://doi.org/10.30534/ijatcse/2020/93942020

A Comparative Evaluation of supervised and unsupervised algorithms for Intrusion Detection

Venkata Ramani Varanasi[1],Shaik Razia[2]
[1]Department of Computer Science and Engineering, Research Scholar, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
[2]Department of Computer Science and Engineering, Associate Professor, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India

ABSTRACT

Intrusion Detection is an essential feature of Network Security that monitors and detects any intrusions in the network. The current frameworks are inadequate to handle network security attacks, which are increasing rapidly with Internet usage. Various machine learning techniques have been proposed to detect network intrusions. One of the problems with the datasets is imbalanced classes. This paper emphasizes the implementation of di□erent machine learning algorithms for network-based intrusions, analyses data imbalance and its impact on classification and anomaly detection. A pair of balanced and imbalanced datasets, NSL-KDD and CICIDS are considered as benchmark datasets for evaluation. Random Forest classifier is used to determine the best set of features for feature selection. The set of supervised and unsupervised algorithms selected for the implementation include - K-Nearest Neighbors, Naive Bayes,Random Forest, Decision Trees, K-Means,Logistic Regression, Isolation Forest, and Local Outlier Factor. Implementation results indicate that in case of supervised learning, Random Forest outperforms the other methods, whereas K-Means performs better than other unsupervised learning methods.

Key words: Data balancing, Intrusion Detection, Machine Learning, Supervised Learning, Unsupervised learning.

1. INTRODUCTION

Intrusion Detection System (IDS) is a security monitoring system that examines the network behavior to check that the activity in the network is normal. If not, it takes necessary actions depending on the severity.
IDS can be categorized as three levels;NIDS (Network Intrusion Detection System), HIDS (Host Intrusion Detection System), WIDS (Web Intrusion Detection System).NIDS detects intrusions across the network. HIDS detects host specific intrusions. WIDS detects the Web server attacks. When the user enters a search phrase, the web bots are included in the path of request and response to or from the server so that intermediate agents are bypassed to connect to the server. They make the original contents as visitedand will affect the target system by malfunctioning [1]. Based on the detection mechanism, IDS is categorized as Misuse and Anomaly. Misuse based IDS learns patterns with the training data. Anomaly-based IDS can detect the behavior that is different from normal behavior in the network. Signature-based or Misuse based IDS detects only known attacks but Anomaly-based IDS can detect novel attacks.
This paper focuses on NIDS using Machine Learning. Machine Learning methods included in this paper are K-Nearest Neighbors, Naive Bayes, Random Forest, Decision Trees, K-Means,Logistic Regression, Isolation Forest, and Local Outlier Factor.Procedures for all the algorithms listed above are discussed in detail in the following sections.

2. COMPARATIVE STUDY

This paper compares the following algorithms.

2.1. Logistic Regression

It is a classification model that uses a logistic function to predict the probabilities of events with the data fit to it. It uses a sigmoid function to map predicted values to the probabilities. The logistic function is used by this model is represented by Eq. (1).

$$\log\left[\frac{p(x)}{1-p(x)}\right] = \beta_0 + x\beta \qquad (1)$$

To predict a class that data belongs to, this method uses a threshold value. Based on the predicted value greater than the threshold, it can be classified accordingly.

2.2. Random Forest

This paper uses the Random Forest (RF) algorithm for classification. It builds a set of N decision trees, each associated with k random number of data samples. For each of the new samples, the algorithm constructs N trees that predicts the category and assign a new sample to the category that gains the majority vote. It is an ensemble method of learning that uses bagging in which a strong learning group is created from a set of weak learners[2].

2.3. Decision Trees

This paper uses Decision trees for classification. Decision trees split the data using if-then-else conditions of the features. The decision tree's core components are a branch, a leaf node, and a decision node. Classification begins at the decision node, tests the features guided by that node, going down the tree at that point, then comparing the estimation of the features in the given sample. For attribute selection at each decision node, it uses one of the techniques called information gain using entropy, gini index.

2.4. Naïve Bayes

Baye's theorem is used in Naive Bayes method. It uses the assumption that every pair of features are conditionally independent when the value of the class variable is given. We use the Naïve Baye's classification rule as Eq (2).

$$\hat{y} = argmax_y \, p(y) \prod_{i=1}^{n} p(x_i | \, y) \qquad (2)$$

The different naive Bayes classifiers differ by the distribution of probabilities as given in Eq.(3)

$$P(x_i|y) = \frac{P(y|x_i)P(x_i)}{P(y)} \qquad (3)$$

In this paper Gaussian Naïve Bayes is used for classification. The likelihood of the features using Gaussian Naïve Bayes is given by Eq. (4).

$$p(x_i|y) = \frac{1}{\sqrt{(2\pi\sigma_y^2)}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \qquad (4)$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

2.5. K - Nearest Neighbors

In this, each time a new sample is to be classified, it computes k-instances that are nearest to the required one. The k-closest neighbors can be computed using one of the Hamming distance, Minkowski, Euclidean distance, Manhattan distance.

2.6. K-means

Kmeans is an unsupervised learning method that involves iterative calculations that tend to divide the dataset into K distinct clusters where each data point belongs to only one group. It first chooses k number of clusters and calculates k centroids and then assigns each data point to the closest centroid. Again compute the new centroid of each cluster and then reassign each data point to the nearest cluster centroid and repeat this process till convergence.

2.7. Isolation Forest

Isolation Forest, also called iForest, is an unsupervised learning algorithm that works to isolate anomalies that are 'few and di□erent' in the feature space compared to normal data points. iForest separates the samples by arbitrarily choosing an attribute and choosing a split value between the maximum and minimum estimations of that chosen attribute. This split relies upon to what extent it takes to isolate the points. Random partitioning of random trees in a forest produces shorter paths, they are considered as anomalies.

2.8. Local outlier Factor

It is an anomaly detection method based on unsupervised learning that computes local density based on nearest neighbors. It compares local densities of the data points to the densities of its neighbors and identifies the outliers.

3. RELATED WORK

The section presents various works carried out by some of the authors on NSL-KDD and CICIDS in the form of Table 1.

4. METHODOLOGY

4.1 Experiment steps for Supervised Learning

The experiment is carried out using the steps given below: "Data set selection, Data preprocessing, Feature Selection using Random Forest, Build the models using selected features, Train the models, Test the models, Compare the performance of the models".

Data sets selection

NSL-KDD and CICIDS-2017 datasets are selected as benchmark datasets as the IDS research community already adopts these datasets. NSL-KDD is selected because it is the traditional one, and CICIDS-2017 isselected because it is the dataset with all types of up-to-date attacks. NSL-KDD is the improved version of KDD-CUP-99, an acronym for Knowledge Discovery in Databases. CIC-IDS-2017 dataset is developed by Canadian Institute for Cybersecurity.
NSLKDD [13] and CICIDS [14] are used for binary classification. The data proportions for binary classes (normal and attack data) identifies that NSLKDD is almost balanced and CICIDS is imbalanced.

Data Preprocessing

Preprocessing is a crucial phase in which raw data can be transformed into a standardized format. It includes data cleaning (handling null or missing values, deleting unneeded variables, handling categorical values), data normalization or scaling, data balancing, separating target variables, and splitting data into train and test.

Feature Selection

In data preprocessing, the number of features may increase if we apply one-hot encoding for categorical columns. Even otherwise, selecting a subset of features from the existing features plays a vital role because it a□ects the performance of the model.
Random Forest with feature importance is used for feature selection. Random Forest uses ensemble learning by combining a set of Decision Trees with controlled variance. Majority voting can be used for deciding the predictions. As the number of trees increases, the model variance decreases. Random Forests are resistant to overfitting. Because of all these reasons, Random Forests are chosen for feature selection. A random forest classifier with a threshold of 0.01 is chosen for selecting features.

Build the models using selected features

With the subset of features selected in the previous step, the following models arebuilt. K-Nearest Neighbors, Gaussian Naive Bayes, Random Forest, Decision Trees, Logistic Regression.

Train the models

Having the features selected for our dataset, the models can be trained using the train data.

Test the models

Here we use the test data to predict the labels in it and evaluate the performance metrics.

Compare the performance metrics of the models

The performance metrics used to evaluate the models for prediction are the Confusion matrix, F1-Score, Precision, Recall, Area under ROC curve, and Accuracy.

Table 1: previous works related to CICIDS and NSL-KDD datasets:

Author	Year	Dataset	Feature Selection method used	Classification model used	Performance of the model
LukmanHakim et.al. [3]	2019	NSL-KDD	Information Gain, Gain Ratio, ReliefF selection, Chisquare,	J48, Random Forest, Naïve Bayes, KNN	Performance is significant though there is a slight drop in accuracy
Ripon Patgiri et.al.[4]	2018	NSL-KDD	Recursive Feature Elimination (RFE).	Random Forest Support Vector Machine	SVM outperforms RF.
Manjula C. Belavagi et.al. [5]	2016	NSL-KDD	-	Gaussian Naive Bayes, Support Vector Machine, Random Forest, LogisticRegression	RF outperforms other methods
ApichitPattawaro et.al. [6]	2018	NSL-KDD	Attribute ratio	K-Means, XGBoost	Accuracy-84.41% Detection rate - 86.36% false alarm rate - 18.20%
Yi Yi Aunget.al. [7]	2018	KDD 99	-	k-means	-
Muhammad Shakil Pervez et.al. [8]	2014	NSL-KDD	Merge of feature selection and classification	SVM	91% to 99% accuracy
Safura A. Mashayak, et.al. [9]	2019	NSL-KDD	Recursive Feature Elimination	Decision Tree, Random Forest	Accuracy 99%
Razan Abdulhammed et.al. [10]	2019	CICIDS 2017	Dimensionality Reduction using Auto Encoder, PCA	Random Forest, Bayesian network, LDA, QDA	-
Mr. Ketan Sanjay Desale et.al. [11]	2015	NSL-KDD	Genetic Algorithm	Naive Bayes and J48	-
Jorge Meira et.al. [12]	2018	NSL-KDD, ISCX	-	Nearest Neighbors, K-means, Auto Encoder, Isolation Forest	Accuracy 60%
Sharafaldin et al. [13]	20 18	CICIDS 2017	Random Forest Regressor	Multi-Layer Perceptron, Adaboost, KNN, Random Forest, QDA, ID3, Naïve Bayes	Precision rate 0.98 with Random Forest and ID3
Aksu, D et.al. [14]	2018	CICIDS 2017	Fisher Score algorithm	Support Vector Machine, K-Nearest Neighbors, Decision Tree	Accuracies - 99.7%, 57.76% and 99%

4.1.1. Supervised learning using NSL-KDD dataset

This dataset has 41 feature columns and one label column. The 41 features are grouped into three categories: basic features related to TCP/IP connections, tra□c features associated with the service or host, and content features extracted from packet contents. There are five di□erent types of labels that categorizing the data as normal or attack. The attacks are classified into four types: DOS, Probing, U2R, R2L.

DOS: To make the network resources unavailable to the user.

Probing: To explore the fragility in the network that can lead to attacks
U2R: Invader that has user privileges but trying to get admin privileges.
R2L: Invader that has illegitimate access to the remote system.

Data Preprocessing

Preprocessing includes the following steps.
2. All the values of the column, num_outbound_cmds contain zero for all the rows. So it is deleted because it does not a□ect the performance.
.3. There are three categorical values protocol type, service, flag. One hot encoding is applied for categorical features of both train and test datasets. For protocol type, there are three unique values in train and test data sets. There are 70 unique values in the train data set and 64 unique values in the test data set for service. For the flag, there are 11 unique values for train and test datasets. All the protocol type and flag categorical values are one-hot encoded. All the 70 categories

In this paper, binary classification of the data as normal or attack is used. The authors have used KDDTrain+ and KDDTest+ datasets for implementation. KDDTrain+ has 125973 samples and KDDTest+ has 22544 samples.
1. In NSL-KDD dataset, there are no null values or missing values.

in the train data set and 64 categories in the test dataset are one-hot encoded for service. The remaining six categories that are missing in the test dataset are filled with zeros.
4. The target label ‘class’ is encoded as 0 for normal data and 1 for attack data using Label Encoder.
5. All the one-hot encoded data is scaled to put them in the range between 0 and 1. Standard Scaler is used for this purpose.
6. For binary classification, data is almost balanced, so no resampling techniques are used. Data balancing is identified as shown in Figure 1.

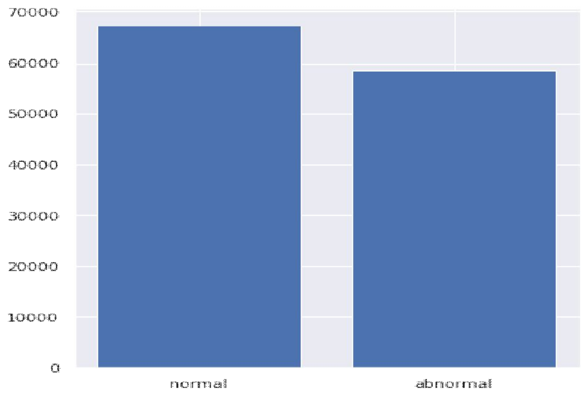

Figure 1: Data balancing for NSL-KDD

class 0: normal: 6734333
class 1: anomaly: 5863034
Proportion: 1.15:1

After completing the data preprocessing step, the shapes of train and test data are:
Train shape: (125973, 121)
Test shape: (22544, 12)

Feature Selection

Random Forest is selected for feature selection. Out of 121 features, 26 features are selected based on the threshold value of feature importance 0.01. Due to this, the data set size is reduced to
Train shape: (125973, 26)
Test shape: (22544, 26)
The selected features include:
[protocol_type_tcp, protocol_type_icmp, service_ecr_i, service_http, flag_SF,flag_S0, service_private, srv_count, dst_host_srv_count,di□_srv_rate, same_srv_rate srv_serror_rate, dst_host_count, logged_in, dst_host_srv_serror_rate, dst_host_rerror_rate,dst_host_srv_rerror_rate, dst_host_srv_di□_host_rate, dst_host_same_srv_rate, dst_host_serror_rate, src_bytes, dst_bytes,dst_host_di□_srv_rate, dst_host_same_src_port_rate, serror_rate, count]

Build the models using selected features

All the models ‘K-Nearest Neighbors, Gaussian Naive Bayes, Random Forest, Decision Trees, Logistic Regression’ are implemented using the subset of 26 features selected out of 121 features.

Train the models

All the models are trained using the train data as for cls in classifiers:
trained_model=cls.fit(X_train, Y_train)

Test the models

The models are tested with test data as
Y_pred = trained_model.predict(X_test)

Compare the performance metrics of the models

The results are given in Table 2.

Table 2: Results of Supervised learning with Random Forest feature selection using NSL-KDD

Model	Accuracy	F1 Score	Precision	Recall	AUC	Confusion matrix
Logistic Regression	0.722453	0.740513	0.619913	0.9 19369	0.853823	[[7359 5474] [783 8928]]
Decision Tree	0.754524	0.772488	0.642920	0.967459	0.780515	[[7615 5218] [316 9395]]
Random Forest	0.765037	0.780925	0.652543	0.972196	0.948926	[[7806 5027] [270 9441]]
Gaussian NB	0.743390	0.744738	0.651559	0.869014	0.819417	[[8320 4513] [1272 8439]]
K-Nearest Neighbors	0.764105	0.778545	0.653569	0.962619	0.809692	[[7878 4955] [363 9348]]

ROC curve for supervised learning using NSL-KDD

ROC curve for supervised learning is obtained as shown in Figure 2.
The curve indicates that Random forest occupies more area.

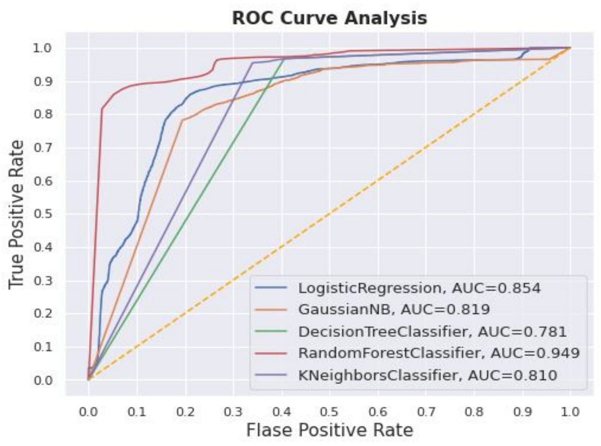

Figure 2. ROC Curve for supervised learning with NSLKDD dataset

4.1.2. Supervised learning using CICIDS-2017 dataset

The dataset is available in two formats: PCAP files and CSV files. The authors have used CSV files for implementing their models. All these files are combined to form 78 feature columns and one label column. There are 15 di erent types of attacks. They are 'BENIGN, DoS GoldenEye, DoSSlowhttptest,DoS Hulk, DoS slowloris, Heartbleed, PortScan,DDoS, Infiltration,DoS Slow HTTP Test, SSH-Patator,FTP-PatatorBot, Web Attack-Sql Injection, Web Attack- XSS,Web Attack-Brute Force'. Authors have used binary classification to identify the tra c as normal or attack.

Data Preprocessing

Preprocessing includes the following steps.

1. CICIDS dataset contains infinity values and null values. Infinity values are replaced with NaN values. All null values are replaced with the mean of the column containing the null value.
2. Eight columns are containing 0 for all the rows. The columns are:
   [Bwd Avg Bulk Rate, Bwd Avg Packets/Bulk, Bwd Avg Bytes/Bulk, Fwd Avg Bulk Rate, Fwd Avg Packets/Bulk, Fwd Avg Bytes/Bulk, Bwd URG Flags, Bwd PSH Flags]
   The above features are deleted as they do not a ect the performance.

3. There are no categorical values in the dataset.
4. The target label 'Label' is encoded as zero for normal data and one for attack data using Label Encoder. Target labels are separated from the remaining features.
5. The data is scaled to put it in the range between 0 and 1. Standard Scaler is used for this purpose.
6. Data is identified as imbalanced for binary classification as shown in Figure 3.

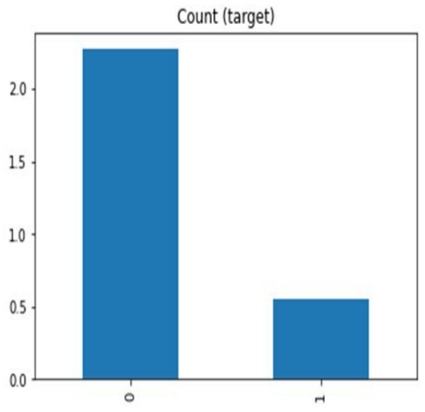

Figure 3. Data balancing for CICIDS dataset

Date shape: (2830743, 70)
class 0: Benign: 2273097
class 1: Anomaly: 557646
Proportion: 4.08 : 1

7. The data is split into train data and test data. The test data size is 25% of the total data. After the data split, the size of the train and test data is:
Train_X shape: (2123057, 70)
Test_X shape: (707686, 70)
Train_y shape: (2123057,)
Test_y shape: (707686,)
8. A 'Near Miss Under sampling' technique is used for resampling the train data. Using this technique train data is resampled to the average of the total samples, the reason behind that is, if we use near-miss under sampling to resample
to the number of samples in the minority class, the datamay cause underfitting.
Before Under Sampling, counts of label '1': 418679
Before UnderSampling, counts of label '0': 1704378
After UnderSampling, counts of label '1': 418679
After UnderSampling, counts of label '0': 675288
After UnderSampling, the shape of train_X: (1093967, 70)
After UnderSampling, the shape of train_y: (1093967,)

Feature selection

Random Forest classifier is used for feature selection. Out of 70 features, 27 features are selected based on the threshold value of feature importance 0.01. Because of this, the data set size is reduced to
Train_X shape: (1093967, 27)
Test_X shape: (707686, 27).
The selected features include:
[Total Length of Fwd Packets, Total Fwd Packets, Total Backward Packets, Destination Port, Fwd Packet LengthMax, Fwd Packet Length Mean, Bwd Packet Length Min, Bwd Packet Length Max, Bwd Packet Length Std, Bwd Packet Length Mean, Flow Packets/s, Flow IAT Max, Fwd Packets/s, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance, Average Packet Size, Avg Bwd Segment Size, Avg Fwd Segment Size, Subflow Bwd Packets, Subflow Fwd Bytes, Subflow Fwd Packets, Init Win bytesbackward, Init Win bytesforward, act data pkt fwd, Idle Max].

Build the models using selected features

All the models "Logistic Regression, Random Forest, Decision Tree, Gaussian Naive Bayes, K- Nearest Neighbors" are implemented using the subset of 27 features selected out of 70 features.

Train the models

All the models are trained using the train data.
for cls in classifiers:
trained_model = cls.fit(train_X, train_y)

Test the models

The models are tested with test data as
Y_pred = trained_model.predict(test_X)

Compare the performance metrics of the models

The performance metrics obtained from the experiment are given in Table 3.

Table 3: Results of Supervised learning with Random Forest feature selection using CICIDS

Model	Accuracy	F1 Score	Precision	Recall	AUC	Confusion matrix
Logistic Regression	0.823021	0.592122	0.540815	0.654184	0.897242	[[491531 77188] [48057 90910]]
Decision Tree	0.891597	0.774368	0.654829	0.947296	0.910645	[[499328 69391] [7324 131643]]
Random Forest	0.937743	0.841484	0.841460	0.841509	0.986115	[[546686 22033] [22025 116942]]
Gaussian NB	0.696664	0.3792802	0.317034	0.471939	0.766184	[[427436 141283] [73383 65584]]
K-Nearest Neighbors	0.906897	0.805871	0.682306	0.984089	0.950408	[[505043 63676] [2211 136756]]

Hyper parameters used with the models in supervised learning

Hyper parameters used in the supervised learning algorithms are given in Table 4.

Table 4: Hyper parameters used in supervised learning

Model	Hyper parameters used
Logistic Regression	C = 1.0, Penalty = ‘L2’ Solver = ‘lbfgs’
Decision Tree	Criterion = ‘gini’
Random Forest	n_estimators = 100
K-Nearest Neighbors	n_jobs = -1, algorithm = ‘auto’ metric = ‘minkowski’

ROC curve for supervised learning using CICIDS dataset

ROC curve is obtained as shown in Figure 4. The curve indicates that Random forest occupies more area under curve.

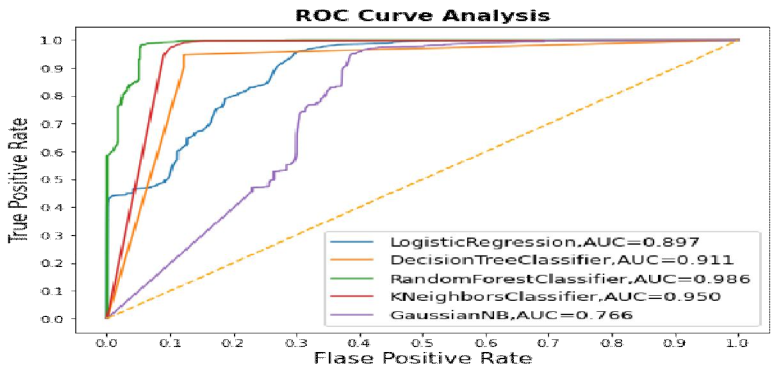

Figure 4: ROC Curve for supervised learning with CICIDS

4.2. Experiment steps for unsupervised learning

The steps used for the experiment are given in below.
Data set selection, Data preprocessing, Select the model for anomaly detection, Classification results.

4.2.1. Unsupervised learning using NSL-KDD dataset

After data preprocessing (as with supervised learning), unsupervised learning models: K-means, Isolation Forest, Local outlier factor are selected for the identification of clusters and anomaly detection. After processing is done results are obtained as given in Table 5 and Table 6.

Table 5: Results of unsupervised learning using NSL-KDD

Model	**Clusters**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Contingency matrix**
K-Means	[0,1] 0 normal 1 anomaly	0.88	[0.99,0.82]	[0.76,0.99]	[0.86,0.89]	[[54185 17278] [757 76297]]
Isolation Forest	[-1,1] 1 normal -1 anomaly	0.56	[0.73,0.55]	[0.15,0.95]	[0.25,0.69]	[[10777 60686] [4075 72979]]
Local outlier factor	[-1,1] 1 normal -1 anomaly	0.49	[0.34,0.50]	[0.07,0.87]	[0.12,0.64]	[[5041 66422] [9811 67243]]

Table 6: Results of unsupervised learning using NSL-KDD

Model	**Adjusted random score**	**Adjusted mutual info score**	**Homogeneity score**	**Complete-ness score**	**V_measure**	**Fowlkes mallows score**
K-Means	0.5732	0.5389	0.52588	0.55262	0.53892	0.79415
Isolation Forest	0.0154	0.0268	0.0197	0.04202	0.0268	0.64678
Local outlier factor	-0.00020	0.00895	0.00658	0.01402	0.0089	0.64068

4.2.2. Unsupervised learning using CICIDS dataset

As part of data preprocessing, infinity columns are replaced with NaN. All null values are replaced with the mean of their corresponding columns. The columns with all zero values are deleted. Data normalization is done to set the data values between 0 and 1. All target labels are encoded as 0 for normal and 1 for attack data. All target labels are separated from the remaining independent variables. We need to feed these independent features to the models to learn the patterns and to prepare clusters. The number of clusters is taken as two. Predicted labels are compared with actual labels, and results obtained are given in Table 7 and Table 8.

Table 7: Results of unsupervised learning using CICIDS

Model	**Clusters**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Contingency matrix**
K-Means	[0,1] 0-normal 1-anomaly	0.79	[0.84,0.46]	[0.91,0.31]	[0.88,0.37]	[[2078680 194417] [389423 168223]]
Isolation Forest	[-1,1] 1-normal -1-anomaly	0.79	[0.45,0.83]	[0.23,0.93]	[0.30,0.88]	[[126033 431613] [157042 2116055]]
Local Outlier factor	[-1,1] 1-normal -1-anomaly	0.56	[0.55,0.73]	[0.07,0.95]	[0.24,0.68]	[[10477 60486] [4099 72999]]

Table 8: Results of unsupervised learning using CICIDS

Model	**Adjusted random score**	**Adjusted mutual info score**	**Homogeneity score**	**Complete-ness score**	**Vmeasure**	**Fowlkes mallows score**
K-Means	0.1781	0.0628	0.0556	0.07216	0.06285	0.77735
Isolation Forest	0.1387	0.0439	0.03634	0.0554	0.04391	0.78415
Local Outlier factor	0.0147	0.02468	0.0187	0.04102	0.02652	0.6366

Hyper parameters used with the models in unsupervised learning

Hyper parameters used in the unsupervised learning algorithms are given in Table 9.

Table 9: Hyper parameters used with the models in unsupervised learning

Model	Hyper parameters used
K-Means	init = ‘k-means++’ n_clusters = 2
Local Outlier Factor	contamination='auto', n_jobs= -1
Isolation Forest	contamination=0.1, n_estimators=100

4. RESULTS AND DISCUSSIONS

In supervised learning, with the NSL-KDD [15] dataset, among all the models that are used, Random forest and K-NN are showing better performance than other models with an accuracy of 76%. For all the models, recall values are higher than precision values, which means that false negatives are lesser than false positives. From a network security perspective, it is required to have a less false-negative rate. With the CICIDS [16] dataset, the Random forest outperforms other models with an accuracy of 93%. Precision and recall values are almost the same for the random forest. Also, it occupies more area in the ROC curve plot. After Random forest, KNN and Decision Tree algorithms show better performance. The metrics classification report, confusion matrix,f1 score, recall,accuracy, precision, are evaluated and presented in the tables. In unsupervised learning, with NSL-KDD and CICIDS datasets, K-means is showing better accuracy. However, the problem observed is that it depends on the random seed. The best accuracy observed is 88% with NSL-KDD and 79% with CICIDS. A new column is added with the actual labels [0, 1] changed to [1, -1] in both the datasets, comparing the outlier labels with the actual labels and then evaluating all the metrics for Isolation forest and Local outlier factor algorithms. The outliers are represented with a negative one value. Vmeasure is the harmonic mean of homogeneity and completeness score. Fowlkes mallows score is the geometric mean of pairwise precision and recall values. The Adjusted random score,Completeness score,Homogeneity score, Adjusted mutual info score, Vmeasure, and Fowlkes mallows score are used for internal evaluation based on the data[17]. Other metrics accuracy, precision, recall, and f1 score are used for external evaluation to quantify the quality of predictions.

5. CONCLUSION

This paper presents a comparative study of supervised and unsupervised algorithms using NSL-KDD and CICIDS datasets. For supervised learning, a random forest is used for feature selection. The threshold value of 0.01 for feature importance is used for feature selection in training and testing. Using these features, the models are evaluated for both the datasets. With CICIDS, since the data is imbalanced, Near Miss under-sampling is used for balancing the data. The result of this under-sampling data with the selected features using random forest, the models are evaluated and quantified the predictions. Unsupervised learning models are used for clustering and anomaly detection. With supervised learning, Random forest and KNN are performs better than other algorithms. With unsupervised learning, K-Means performs better.

REFERENCES

S. Ponmaniraj, Dr. Tapas Kumar , Dr. Amit Kumar Goel,**Web Intrusion Detection System through Crawler’s Event Analysis.** *IJATCSE, ISSN 2278-3091, Volume 9, No.3, May - June 2020, 2503 – 2507*,https://doi.org/10.30534/ijatcse/2020/03932020

1. Tadepalli Anish Deepak, Burra Vijaya Babu , Burra Meghana, **XGBoost Classification based Network Intrusion Detection System for Big Data using PySparkling Water.** *IJATCSE, ISSN 2278-3091, Volume 9, No.1, Jan–Feb 2020, 377 – 382*,https://doi.org/10.30534/ijatcse/2020/55912020
2. Lukman Hakim, RahillaFatma, Novriandi. (2019). **Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset.***2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE),* Jember, Indonesiahttps://doi.org/10.1109/icomitee.2019.8920961
3. Ripon Patgiri, UditVarshney, Tanya Akutota, and RakeshKunde. (2018) **An Investigation on Intrusion Detection System Using Machine Learning**. *2018 IEEE Symposium Series on Computational Intelligence (SSCI),* Bangalore, India,https://doi.org/10.1109/ssci.2018.8628676
4. Manjula C. Belavagi and BalachandraMuniyal. (2016) **Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection.***Procedia Computer Science, Volume 89, 2016, Pages 117-123*.
5. Apichit Pattawaro, Chantri Polprasert. (2018). **Anomaly-Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique**. *2018, 16th International*

Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand.https://doi.org/10.1109/ictke.2018.8612331

6. Yi Yi Aung, Myat Myat Min. (2018**). An Analysis of K-means Algorithm Based Network Intrusion Detection System.** *Advances in Science, Technology and Engineering Systems Journal* Vol. 3, No. 1, 496-501 (2018). https://doi.org/10.25046/aj030160
7. Muhammad Shakil Pervez, and Dewan Md. Farid. (2014). **Feature Selection and Intrusion classification in NSL-KDD Cup 99 Dataset Employing SVMs.***The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*Dhaka, Bangladeshhttps://doi.org/10.1109/skima.2014.7083539
8. Safura A. Mashayak, Balaji R. Bombade. (2019). **Network Intrusion Detection Exploitation Machine Learning Strategies with the Utilization of Feature Elimination Mechanism.***International Journal of Computer Sciences and Engineering* Vol. 7(5), May 2019, E-ISSN: 2347-2693. https://doi.org/10.26438/ijcse/v7i5.12921300
9. Razan Abdulhammed, Hassan Musafer, Ali Alessa, Miad Faezipour and Abdelshakour Abuzneid. (2019). **Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection.***Electronics* 2019, 8(3), 322. https://doi.org/10.3390/electronics8030322
10. Mr. Ketan Sanjay Desale, Ms. Roshani Ade (2015). **Genetic Algorithm based Feature Selection Approach for Effective Intrusion Detection System**. 2015 *International Conference on Computer Communication and Informatics (ICCCI -2015)*, Jan. 08 – 10, 2015, Coimbatore, INDIA. https://doi.org/10.1109/iccci.2015.7218109
11. Jorge Meira. (2018). **Comparative Results with unsupervised Techniques in Cyber Attack Novelty Detection.** *Proceedings of XoveTIC Congress*, A Coruña, Spain 2018, 2(18),1191. https://doi.org/10.3390/proceedings2181191
12. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. (2018). **Toward generating a new intrusion detection dataset and intrusion traffic characterization.** *Proceedings of the Fourth International Conference on Information Systems Security and Privacy, ICISSP*, Funchal, Madeira, Portugal, 22–24 January 2018. https://doi.org/10.5220/0006639801080116
13. Aksu, D.; Üstebay, S.; Aydin, M.A.; Atmaca, T. (2018). **Intrusion Detection with Comparative Analysis of Supervised Learning Techniques and Fisher Score Feature Selection Algorithm**. *International Symposium on Computer and Information Sciences; Springer: Berlin*, Germany, 2018; pp. 141–149. https://doi.org/10.1007/978-3-030-00840-6_16.
14. **NSL-KDD Data Set** [Online], accessed in June 2020. Available at: https://www.unb.ca/cic/datasets/nsl.html/
15. **CICIDS 2017 Data Set**[Online]. accessed in June 2020. Available:https://www.unb.ca/cic/datasets/ids2017.html
16. Clustering metrics accessed from https://scikit-learn.org/stable/modules/clustering.html
17. Shaik Razia,M. Bhanusri, **Maize Leaf Disease Detection Using Convolution Neural Network**, , *Jour of Adv Research in Dynamical & Control Systems* 12 (02), 1154
18. Shaik Razia, P Gopi Krishna,**Implementation of Environment Gases Monitoring System using LoRa Gateway in Smart Cities with IoT Technology**, Jour *of Adv Research in Dynamical & Control Systems* 12 (02), 1109.
19. Shaik Razia, P.Vyshanavi,**Secure Sharing Of Files Using Key Image Encryption And Storing In Aws**, Jour of Adv Research in Dynamical & Control Systems 12 (02), 1147.
20. A Roy, S Razia, N Parveen, AS Rao, **Fuzzy rule based intelligent system for user authentication based on user behaviour**, *Journal of Discrete Mathematical Sciences and Cryptography*, vol-23,issue-2, pages 409-417, Taylor & Francis.
21. Shaik Razia, **Prediction Of Rainfall Using Hybrid Neural Network**, *Jour of Adv Research in Dynamical & Control Systems* 12 (02), 1119
22. *S Razia, PS Kumar, AS Rao,* **Machine learning techniques for Thyroid disease diagnosis: A systematic review**, ,*Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough …* Springer, Cham, 2020, pages 203-212.
23. Shaik Razia, M.R.Narasingarao, **Development and analysis of support vector machine techniques for early prediction of breast cancer and thyroid,***Journal of Advanced Research in Dynamical and Control Systems* 9 (Sp– 6)
24. MS Shaik Razia, P Siva Kumar , Dr.N.Anbazhaghan, **Prediction of cardiovascular Disease Using Classification Techniques With High Accuracy**, *Jour of Adv Research in Dynamical & Control Systems* 12 (02), 1134