Network Anomaly Uncovering on CICIDS-2017 Dataset: A Supervised Artificial Intelligence Approach

2022 IEEE International Conference on Electro Information Technology (eIT) | 978-1-6654-8009-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/eIT53891.2022.9814045

Pankaj Jairu
Department of Information Systems
St. Cloud State University
St. Cloud, MN, USA
pankaj.jairu@go.stcloudstate.edu

Akalanka B. Mailewa
Department of Computer Science and Information Technology
St. Cloud State University
St. Cloud, MN, USA
amailewa@stcloudstate.edu

***Abstract*— In today's world, businesses and services are shifted to a digital transformation. As a result, network traffic has tremendously increased over the years. With that, network threats and attacks are growing and with that, the importance of intrusion detection systems has increased. The traditional signature-based approach to intrusion detection is not sufficient to detect intrusions, so anomaly-based intrusion detection came into play. There are many methods to Anomaly-based intrusion detection methods that can classify unknown network attacks. To detect network anomalies, Machine Learning and Deep Learning techniques are applied, and a considerable number of studies are done in this field. This research presents classification models built using supervised Machine Learning algorithms. The algorithms Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, Decision Tree and Random Forest on multiple datasets of realistic evaluation dataset CICIDS-2017. The results show that Random Forest outperforms other supervised algorithms with as high as 99.93% accuracy using 14 features selected using Pearson's correlation coefficient method.**

Keywords—IDS/IPS, Attacks, Security, Deep Learning, Machine Learning, Threats, Vulnerabilities

I. INTRODUCTION

Over the past decade, more and more businesses and organizations are digitizing their confidential data. This has increased the volume of network traffic, with data being created at a very large scale. Computer networks have expanded tremendously over the last decade especially with the emergence of new devices and services like cloud computing and Internet of Things (IoT). The security of this data is a big challenge. Also, attacks on networks have increased significantly and Network Intrusion is acknowledged to be the most danger to security [1][2]. Attacks like Denial of Service (DoS), Zero-day attacks and Advanced Persistent threats (APT) have been significant problems in today's information technology global community. This is where the idea of Intrusion Detection System (IDS) comes handy.

Intrusion Detection System (IDS) are hardware and software systems that can identify such harmful behaviors. The main objective of the Intrusion Detection System (IDS) is to observe the behavior of the system, identify attacks and generate alarms so that appropriate actions can be taken to prevent any harmful consequences [2]. Intrusion can be detected using two classification techniques i.e., signature-based and anomaly based. Signature-based, also known as pattern-based anomalies are looked against a list of patterns the database already has. Signature-based intrusion detection comes with a drawback – it is unable to learn by itself, any anomalous patterns and intrusions within raw data. Anomaly-based intrusion detection can point out a normal or benign activity and look for anything that is anomalous. It can learn any abnormal pattern based on Machine Learning and Deep Learning concepts. The inputs of an IDS could be traffic logs, application logs, file system changes, packets, etc. that are monitored, and output is the label for each input [2]. Numerous research studies have been conducted in the field of Machine Learning (ML) and Deep Learning (DL) because they can learn trends of malicious behaviors while reducing false alarms [9]. Several authors have attempted to do a comprehensive survey on Machine Learning and Deep Learning techniques for anomaly detection [2][4][5]. It turns out that much of the research in this area is based on shallow Learning techniques which requires a lot of time, effort and resources and their effectiveness depends on the expertise and extent of knowledge of the researchers in the field [10].

Network Intrusion Detection using Machine Learning (ML) and Deep Learning (DL) is one of the most significant developments in the field of information security. There is a competition among researchers, leading companies, and economies to advance Deep Learning and Artificial Intelligence. In some cases, Artificial Intelligence has exceeded human Intelligence, like the modern mobile applications, decision to predict stocks, decision to predict movie ratings, etc. Although DL and ML in detecting network attacks have accomplished a lot, there are still areas where effectiveness is lacking. There could be more precision, accuracy and performance of the algorithms that help classify these attacks to prevent them.

With the increase in the volume of network traffic, with data being created at a very large scale. Computer networks have expanded hugely over the last decade and especially with the emergence of new devices and services like cloud computing and Internet of Things (IoT), attacks on networks globally have increased significantly [34]. Malware, spear-phishing, ransomware top the list of cybersecurity threats. Besides those many other network intrusion attacks like denial of service, Zero-day attacks and advanced persistent threats (APT) have been reported as significant problems in today's information technology global community. APTs can be dangerous and

978-1-6654-8009-3/22/$31.00 ©2022 IEEE

costly as these are powerful attacks launched by malicious actors against government and private organizations with the intent of causing great damage. Thus, the main objective of this study is to evaluate the effectiveness of machine learning models by using various performance metrics. The performance metrics we used for this study is Accuracy, Precision, recall and F-1 score. The goal of this study is to test the performance of various machine-learning algorithms on the various categories of subset data of realistic evaluation dataset CICIDS-2017. It was expected that our machine-learning model comprising feature selection using Pearson's correlation coefficient coupled with these algorithms would increase the accuracy on the CICIDS2017 dataset. This would be the contribution of this study in the field of application of machine learning on anomaly detection. To achieve these objectives, we have defined three research questions (RQ) as follows:

RQ1: How to use "Pearson Correlation Coefficient" as a feature selection tool contribute to the performance of the proposed models?

RQ2: Do the models reduce the computational costs for the Intrusion Detection System?

RQ3: What supervised machine-learning model performs the best in detecting anomalies on this dataset?

In addition, in order to understand our research methods very clearly, we have defended most important terms and phrases as follows:

A. Self-taught Learning (STL)

Self-taught Learning (STL) is a deep learning approach consisting of two stages for the classification. The first stage is learning a good feature representation from a large collection of unlabeled data called Unsupervised Feature learning. The second stage consists of applying this learnt representation to labeled data and is used for the classification task [17]. There are different approaches used for unsupervised feature learning, such as Sparse Autoencoder, K-Means Clustering, Restricted Boltzmann Machine (RBM) and Gaussian Mixtures [7].

B. Machine Learning and Deep Learning Algorithms

In recent years, Machine Learning and Deep Learning algorithms in anomaly detection have garnered huge interest [4][23]. Anomaly-based intrusion detection is essentially a classification problem and Machine Learning and Deep Learning algorithms have proven to be useful in Network Intrusion Detection [5][6]. Machine Learning is a branch of Artificial Intelligence, and it gives computers the ability to learn without being explicitly programmed [23]. Deep Learning is an advanced field in Machine-Learning research, and it simulates the human brain style to analyze and interpret data. Deep Learning is essentially an advancement of the Machine Learning process, and it is derived and formulated from the Artificial Neural Network. It is believed that Deep Learning algorithms are the most significant breakthrough of the century, which significantly drives applications towards Artificial Intelligence [11]. Traditional Machine Learning methods used for intrusion detection such as Support Vector Machine (SVM), Decision Tree, Linear Regression, Hidden Markov Model etc. have shallow architecture and are not capable of handling intrusion detection in modern data environments [24]. The idea of Deep Learning was proposed by Hinton [25] and it is a Machine-Learning method based on characterization of data learning. Some examples of Deep Learning algorithms include Convolutional Neural Network (CNN), LSTM (Long Short-Term Memory), Deep Boltzmann Machine (DBM), etc.

C. Logistic Regression

Logistic regression is a predictive analysis algorithm, and it is based on the concept of probability. It is used for classification problems. It is used for binary classification that uses a logistic function called a sigmoid function for prediction. Although its name makes it sound like a regression algorithm, logistic regression is a classification algorithm.

D. Kernelized Support Vector Machine (SVM)

Support Vector Machine comprises a set of supervised learning methods. It is one of the most simple and common ML algorithms used to categorize different types of data in SVM. It is a non-probabilistic method. It creates a hyper-plane or a multiple hyper-plane in a boundless dimensional input vector to classify the instances. It is a powerful model and performs well on a variety of datasets. It has been used to identify network intrusion quickly and accurately [20][41]. However, it requires very meticulous and careful data pre-processing of the data and tuning of parameters.

E. K-Nearest Neighbor

The KNN is a classification algorithm inspired from Standard Euclidean Distance (SED) that exists between two points in the same space [8]. It is a very simple and easy to implement algorithm and there is no need to build a model and optimize parameters. However, the algorithm performs very slowly with the increase in number of examples or variables. The two important parameters in KNN algorithms are: number of neighbors and the way distance between data points are measured. The default distance used is the Euclidean distance, which works well.

F. Naive Bayes

Naive Bayes algorithm is a supervised learning algorithm based on Bayes' theorem, which assumes conditional independence between every pair of features given the value of the class variable. It is easy to implement an algorithm, but it requires the predictors to be independent. Since most realistic cases have predictors that are dependent, the performance of the classifier is affected negatively. Naïve Bayes Classifiers are efficient and the reason being that they learn parameters by looking at each feature individually and they collect statistics from each feature. There are three classes of Naïve Bayes Classifiers implemented in ScikitLearn: BernoulliNB, MultinomialNB and GaussianNB. For this study, GaussianNB was used because it can be applied to any continuous data [31]. The dataset used in this study is comparatively high-dimensional and GaussianNB is mostly used on very high-dimensional data. The GaussianNB model requires very less training time and makes predictions

G. Decision tree

Decision tree is a supervised ML algorithm used to classify data. The architecture of a decision tree comprises the category nodes, the internal nodes and a root node. Decision trees are the

building blocks of Random Forest. Decision trees are simple and easy to implement and can handle high dimensional data. One advantage of running Random Forest (RF) is that we have to specify fewer parameters compared to other machine learning methods like support vector machines (SVM), Artificial Neural Network (ANN) [28]. Decision trees make their decision (classification) by learning a hierarchy of if/else questions. In the language of Machine Learning, these if/else questions are known as 'tests. To build a tree, the algorithm looks for all tests and discovers the one that represents the target variable the most [31]. The main downside to Decision trees is that they suffer from overfitting problems, and they are poor at generalizations.

H. Random Forest (RF)

According to the random forest classifier proposed by Breimanis [29], it is essentially a decision tree concept that is constructed by using many decision trees. It takes thousands of input variables without deleting variables and classifies them based on their importance [28]. It is an ensemble of classification trees. In random forest, a collection of individual tree structured classifiers can be mathematically expressed as below:

$\{h(x, \theta_k), k = 1, 2, \ldots i \ldots\}$ [30]

Where h represents RF classifier, $\{\theta_k\}$ stands for random vectors distributed independently identical and each tree has an input for the most famous class at input variable x.

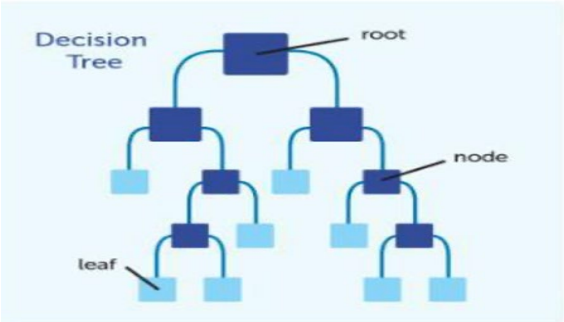

Fig.1. Structure of a decision tree.

As discussed above, a random forest is basically a collection of decision trees, and the trees are slightly different from one another. The issue of decision trees suffering from overfitting of training data is solved by random forests. Random forest is a strong classifier.

II. BACKGROUND AND RELATED WORK

M. Masdari and H. Khezri [7] present an in-depth survey and taxonomy of the fuzzy misuse detection-based Intrusion Detection System approaches that are meant to improve the security of computer systems. Detecting misuse is one of the main branches of intrusion detection which intends to prevent known security attacks regarding a set of signatures that are previously known. Fuzzy misuse detection is based on fuzzy logic which means vague or imprecise information. Fuzzy logic makes use of linguistic variables which can be defined by using fuzzy sets. Fuzzy clustering technique is widely used by misuse detection methods in order to improve the IDS performance [7][8]. Wang et al. proposed a model called 'Integrated Deep Learning Model' which contains three modules, namely: data preprocessing module, intrusion detection module and detection and classification module [12]. Their data pre-processing module contains steps like feature extraction, data conversion, data normalization, etc. The intrusion detection module follows preprocessing and determines the entire network structure and training parameters, which uses the training sets to train the model and saves the model for testing after training. The final module, which is the detection and classification module, is where it displays the classification results to the user. The algorithms used include the Stacked Denoising Autoencoder and single-hidden layer feed forward network known as Extreme Learning Machine. Compared to traditional ML models their module achieved better detection resulting in intrusion detection [12].

M. Choraś and M. Pawlicki [13], make use of a tool/method called hyper-parameter optimization to improve the algorithms in their model. The model was based on artificial neural networks. How a neural network performs is affected by activation function. The commonly used activation functions are sigmoid, hard-sigmoid, Rectified Linear Unit (ReLU) and Hyperbolic Tangent (tanh). Grid search method helps optimize the neural network. Since IDSs are complex, they need a hyperparameter tuning or optimization process to improve system performance [14]. Another important tool to get effective results during Deep Learning studies is dimensionality reduction. It searches for a feature vector, which communicates the nature of the dataset but avoids the task of representing every single feature of the dataset. There are many ways to dimensionality reduction; some common ones include: Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA) [13] during data pre-processing stage. An essential step in data preprocessing is Feature Selection. A detailed review of Feature Selection techniques is available on the paper published by Mauro et. Al [3] as rank-based Feature Selection, meta-heuristic feature selection, nature-inspired Feature Selection, evolutionary Feature Selection. The authors point out that most studies in the field of intrusion detection ignore feature selection because they use the benchmark KDD99 dataset, which makes feature selection irrelevant. Feature Selection processes have been found to accelerate the overall training process significantly although the accuracy drops i.e., from 0.9993 to 0.9971 in their experiment. Some studies have shown that the semantics of network traffic are different. It attempts to re-express the semantic space of intrusion detection traffic so that it helps better distinguish abnormal traffic. It was thought that there is a significant difference in the narrative semantics. According to Z. Wu et al. [15], both raw and converted network traffic are semantically re-encoded. It was concluded from their experiment that semantic re-encoding technology has limited performance improvement.

M. Ring et al [18], provide a literature survey of existing network-based intrusion detection datasets. The paper presents a comprehensive taxonomy containing the nature of the data, data volume, recording environment and basis of evaluation. The authors broadly classify network traffic capture into packet-based or flow-based format. Packet-based data is usually captured in pcap format and has payload as opposed to flow-based data which contains many Meta information about

network connections. Metadata like source IP address, source port, destination IP address, destination port and transport protocol are used to evaluate the datasets used for intrusion detection. However, among the metadata information, the 'Labeled' and 'format' are the most decisive properties while looking for network-based data sets.

In another research I. F. Kilincer et al. [16], the authors have made a comprehensive study of the datasets used in intrusion detection studies. It is found that studies are generally done on a small number of datasets which doesn't provide a clear idea about the performance of the datasets and only a limited number of classifiers were used. A notable dataset is NSL-KDD dataset which is the benchmark dataset in intrusion detection studies which is an improved version of the KDD Cup '99 dataset that was developed in the year 1999. It does not represent today's modern network attacks [6]. It contains old network traffic and does not have real-time properties. The authors I. F. Kilincer [16] further points out that there is only a limited number of anomaly detection studies done on some realistic evaluation datasets like CSE-CIC-IDS-2017, CSE-CIC-IDS-2018. Aldweesh [19] points out that obtaining traffic from simulated environments overcomes this issue by using recent datasets. A dataset of interest that could be used in anomaly detection studies could be the Canadian Institute of Cybersecurity i.e., the CIC-generated CIC-IDS-2017 dataset. The dataset CSE-CIC-IDS-2017 was created by Sharafaldin et al. [17] to fulfill the gap of dataset that represents today's realistic network attacks. This dataset contains benign and common network attacks that are like true real-world data. The dataset includes the results of the network traffic analyses using CICFlowMeter with various labels based on the timestamp, source, and destination IPs, source and destination ports, etc. [6][21].

Elmasri et al. [21] did a similar study using the CICIDS2017 dataset. The authors used a model comprising K Nearest Neighbors (KNN), enhanced KNN and Local Outlier Factor (LOF) techniques. They employed a full sample size. The study uses a semi-supervised anomaly detection approach. It is reported that they utilized the Principal Component Analysis (PCA) and normalization using the Sklearn python library. It was found from their studies that both LOF and KNN performed better than the simple KNN algorithm.

The paper published by Maseer et al.[22], the authors propose a benchmarking approach of CICIDS2017 and evaluate the performance of 10 Machine Learning algorithms, which contains, 7 supervised algorithms and 3 unsupervised algorithms. The supervised algorithms include k-Nearest Neighbor, Support Vector Machine, Decision tree, Random Forest, Artificial Neural network, Naïve Bayes and Convolutional Neural Networks. The three unsupervised algorithms include K-means clustering, Expectation - Maximization (EM) clustering and Self-Organizing Maps (SOM) algorithms. Their studies found that K-Nearest Neighbor, Decision Tree and Naïve Bayes algorithms achieved excellent results while Self-organizing maps and Expectation-Maximization clustering achieved poor results.

A study conducted by Lopez et al. [26] on the CICIDS2017 dataset took a subset of the section of the original dataset that contained the DoS/DDoS attacks captured. Algorithms used to conduct their studies include Logistic Regression, K-Nearest neighbors, Random Forest, Naïve Bayes, Multi-Layer Perceptron and Dense Neural Networks. There are still more varieties of data traffic present in the CICIDS2017 dataset which need to be studied like Brute Force, FTP-Patator, SSH-Patator, Web Attack, Botnet, and Port Scan using broader inclusion of algorithms.

III. METHODOLOGY AND EXPERIMENTS

The data for this study is secondary data i.e. collected by other researchers from the Canadian Institute of Cybersecurity [1] and the dataset is very realistic.

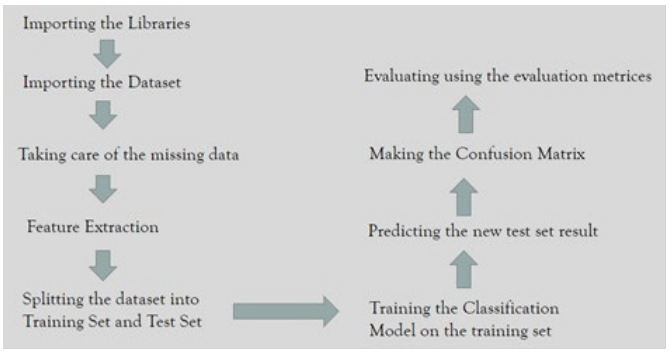

Fig.2. Flow Chart of the method used in the calculation.

In this study, a small subset of data from the CICIDS2017 is taken to optimize the Machine Learning model that can help the attacks mentioned in the table above. The dataset comprises attacks captured using CICFlowMeter [16] with timestamp, source and destination IPs, source and destination ports, protocols and type of attack.

A. *Hardware and Software Environment*

Operating System: Windows 10 Home

Processor: AMD Ryzen 5 3600 6-Core Processor, 3.6 GHz

Installed RAM: 16.0 GB

Startup Disk: McIntosh HD

Software Environment: Python 3.9.4 64-bit

B. *Design of the Study*

The study is mathematical computation in nature. Our model uses Pearson Correlation Coefficient as the feature elimination technique and various supervised Machine Learning classifiers for performing classification. The python libraries that are useful in the study are Scikit-learn, Numpy, Pandas, Keras, matplotlib, TensorFlow, and Pytorch. The calculations are performed on a jupyter Notebook using python. In order to perform the calculation, first the required python libraries were imported. Then, the dataset is imported and it is analyzed. As with every dataset, we need to take care of missing data and select appropriate features. The 'scikit learn' library comes very handy when using necessary resources in python.

Feature selection is a very important task as it helps reduce the computational complexity and eliminate unnecessary and irrelevant features while enhancing the performance of IDS [34][35][38]. Correlation-based feature selection have been

found to improve classification accuracy and reduce the dimensionality of dataset [36][37]. The correlation function called from scikit learn library is used to obtain a confusion matrix. A correlation coefficient is a measure of the degree to which variation in one variable is related to variation in one or more variables [32][34.

The value of correlation coefficient can range from -1 to 1. If the value of correlation is close to +1, there is a very strong positive relationship between the variables and a value close to -1 indicates that there is a very strong negative relationship between the variables. Basically, if the sign of the correlation is opposite, it shows the direction of the relationship between variables [33]. So, the value of correlation tells us the relationship between variables. Feature selection In the case of continuous variables, if the two values are highly correlated, they contribute as the same factor to the target result, and so appropriate selection of features can be done.

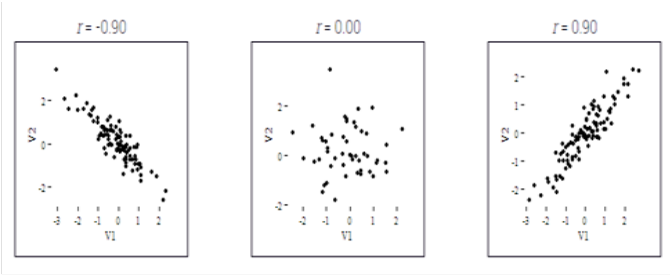

Fig.3. Correlation coefficient defines the relationship between attributes.

```
def correlation(dataset, threshold):
    col_corr = set()
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j]) > threshold:
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr
```

```
corr_features = correlation(X_train, 0.8)
len(set(corr_features))
```

Fig.4. Set up for Pearson's correlation coefficient.

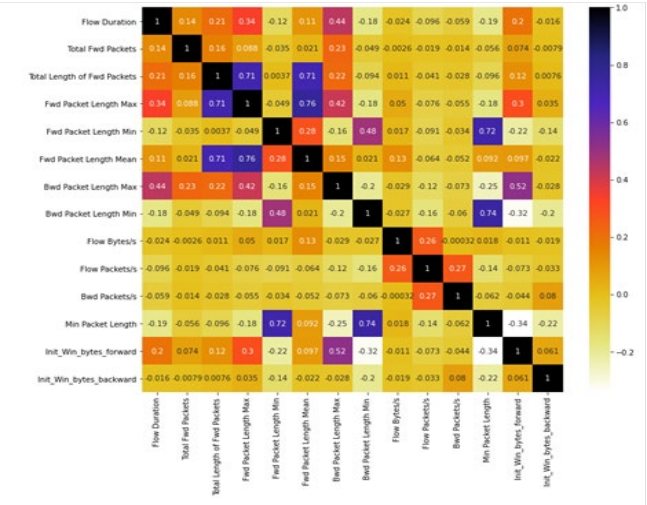

Fig.5. Pearson's correlation plot showing features considered for this study.

After the analysis of the Pearson's correlation plot, the final 14 features that were selected shown as follows:

- Use Total Flow Duration
- Total Forward Packets
- Total Length of Forward Packets
- Forward Packet's maximum length
- Forward Packet's minimum length
- Forward Packet's mean length
- Backward Packet maximum length
- Backward Packet minimum length
- Flow Bytes per second
- Flow Packets per second
- Backward Packets per second
- Minimum Packet Length
- Initial Window Bytes (Forward)
- Initial Window Bytes (Backward)

C. *Description of each features selected*

- Total Flow Duration: The total duration of flow in microseconds.
- Total Forward Packets: Total packets in forward direction.
- Total Length of Forward Packets: Total size of packet in forward direction.
- Forward Packet's maximum length: The maximum size of packet in forward direction.
- Forward Packet's minimum length: The minimum size of packet in forward direction.
- Forward Packet's mean length: The mean size of packet in forward direction.
- Backward Packet maximum length: The maximum size of packet in backward direction.
- Backward Packet minimum length: The minimum size of packet in backward direction.
- Flow Bytes per second: The number of flow bytes per second.
- Flow Packets per second: The number of flow packets per second.
- Backward Packets per second: The number of backward packets per second.
- Minimum Packet Length: The minimum length of a packet.
- Initial Window Bytes (Forward): The total count of bytes sent in the initial window in the forward direction.

- Initial Window Bytes (Backward): The total count of bytes sent in the initial window in the forward direction.

Therefore, this helped in selecting appropriate features for this study. Usually, cluster analysis is done to serve this purpose in the case of unsupervised studies [43], [45] but we conducted the study to see the performance by supervised algorithms.

After the feature selection process, the datasets were imported and using scikit-learn's train test split function, the data was split into 80 % training set and 20 % test set. After this, the classifier i.e., machine learning model's parameters were defined, and the model was trained on a training set. The model was tested on the test set. The prediction is observed through a confusion matrix. Thereafter, a classification report is generated for each dataset and algorithm, which shows the traffic classified into 'BENIGN' and attack type or types. It also shows various other metrics like precision, recall, f-1 score and support for further analysis and conclusion.

D. Performance Metrics

As discussed above, in order to measure the performance of machine learning algorithms, we use some metrics like accuracy, precision, recall, and F-1-score. The performance indicators used for classification problems are based on the below mentioned four possibilities:

- True Positive (TP): correct classification attack packets as attacks.
- True Negative (TN): correct classification normal packets as normal.
- False Positive (FP): normal activity that is incorrectly labeled as intrusive by IDS.
- False Negative (FN): intrusive activity that is classified as normal.

E. The accuracy, the precision, recall and F1-score

1. Accuracy

The accuracy rate is the main prediction indicator for the several machine and deep learning classifiers. It is simply the measure of how correctly the model classifies. Where, TP = True Positive, TN = True Negative, FP = False positive, FN = False Negative.

Accuracy *= (TP + TN) / (TP + FP + TN + FN)*

2. Precision

It is the ratio of correctly identified positive observations to all the predicted positive observations. In other words, Precision measures the number of correct instances retrieved divided by all retrieved instances [39]. The precision is intuitively the ability of the classifier not to label as positive for a sample that is negative [40].

Precision *= True Positive / (True Positive + False Positive)*

3. Recall

Recall is the ratio of correctly identified positive cases to all the observed cases. In other words, recall measures the number of correct instances retrieved divided by all correct instances [39]. The recall is the ratio TP / (TP + FN) where TP is the number of true positives and FN the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples [40].

4. F-1 score

It is the harmonic mean of precision and recall. It is needed when we want to find a balance between Precision and Recall.

F-1 score *= 2 * (Precision x Recall) / (Precision + Recall)*

5. The CICIDS-2017 Dataset

CIC-IDS2017 has benign and common attacks, which is very similar to true real-world data [1]. It also has the result of network traffic analysis using CICFlowMeter with labeled flows based on the timestamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files) [1].

TABLE I. TABLE OF INTRUSION IN THE CICIDS-2017 DATASET

No	Group of intrusion	Type of Intrusion
1	Normal	Benign
2	Denial of Service (DoS)	Botnet, DDoS, DoSGoldenEye, DoS Hulk, DoSSlowhttp, DoSSlowloris
3	Password attack	FTP-Patator, SSH-Patator, Web-Attack-Brute-Force
4	Probing	Port Scan
5	Vulnerability	Heartbleed Attack, Infiltration, Web-Attack-SQL-Injection, Web-Attack-XSS

They captured the data between July 3, 2017 and July 7, 2017 for a total of 5 days. The implemented attacks contain Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS [1]. CICIDS2017 is a very huge dataset, which has approximately 3 million network flows in different files [1] [27]. In CICIDS2017, there is no specified training or test sets to be used in the experiments. So, for this study, only 10% of this dataset was selected for training and testing so that we can reduce training and testing time or the training and testing time would be very lengthy. In addition, the computer used for this study suffered memory error while trying to take a bigger size of datasets for calculations. The selection of those 10% of the dataset was done randomly by using the sampling without replacement technique to ensure the diversity of traffic records and avoiding overfitting. It has several datasets listed under different categories where there are eight different categories of datasets within the main folder containing the datasets. The objective was to perform study on each dataset separately. Therefore, instead of combining these different files into one, machine learning study was performed in each category of dataset separately. However, some datasets were avoided from the study like the 'Monday-WorkingHours.pcap' dataset as it contained only normal benign traffic and 'Friday-WorkingHours-Morning.pcap_ISCX' was avoided because of only one class problem. The detailed study and results obtained from the classification is listed in the next section in this article.

IV. RESULT

In this section, we do quantitative evaluation of the performance of algorithms on different datasets. As discussed above, the CICIDS-2017 dataset is a very huge dataset. Below is a list of tables showing the accuracy and other metrics of the performance of algorithms on different datasets within the CICIDS-2017 dataset. A similar study by [42] on the benchmark NSL-KDD dataset shows random forest as a strong classifier outperforming all other machine-learning classifiers. It is very important what choice of dataset we take for machine learning studies. Machine learning methods cannot work without representative data. In order to be able to perform decent anomaly detection, using network-level and kernel-level data contribute a lot [45]. In the classification reports below, there is a column for a metric called 'Support'. Support is the total occurrence of that class in the considered dataset. Support does not change among models and it helps examine the evaluation process.

A. Tuesday-WorkingHours.pcap_ISCX

The data captured under this category includes Brute Force Attacks conducted on Tuesday, July 4, 2017. Below is the list of tables containing classification done using different classifiers.

TABLE II. LOGISTIC REGRESSION ON TUESDAY-WORKINGHOURS.PCAP_ISCX(ACCURACY=0.9693477992572563)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.97	1	0.98	86399
FTP-Patator	0	0	0	1548
SSH-Patator	0	0	0	1182

TABLE III. KERNELIZED SVM ON TUESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.8969812591179441)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.90	1.00	0.95	7993
FTP-Patator	0.00	0.00	0.00	918

TABLE IV. NAÏVE BAYES- GAUSSIAN ON TUESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.8846369655481988)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.90	0.98	0.94	7993
FTP-Patator	0.08	0.01	0.02	918

TABLE V. DECISION TREE CLASSIFIER ON TUESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY =0.8311076197957581)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.90	0.91	0.91	7993
FTP-Patator	0.16	0.14	0.15	918

TABLE VI. RANDOM FOREST ON TUESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.8837391987431265)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.90	0.98	0.94	7993
FTP-Patator	0.21	0.05	0.08	918

For this subset dataset, logistic regression outperformed the other class of classifiers. The logistic regression algorithm did the classification with accuracy of 96.9 %.

B. Wednesday-workingHours.pcap_ISCX

The data captured under this category includes Denial of Service attacks. There are various kinds of DoS attacks recorded that day i.e. Wednesday, July 5, 2017. DoS Slowloris, DoS Slowhttptest, DoS Hulk, and DoS GoldenEye were the different kinds of attacks recorded under this subset category. Below is the list of tables containing classification done using different classifiers.

TABLE VII. LOGISTIC REGRESSION ON WEDNESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.9166305447189712)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.92	1.00	0.96	12699
FTP-Patator	0.33	0.01	0.02	1143

TABLE VIII. TABLE 8: KERNEL SVM ON WEDNESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.9187978615806964)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.93	0.99	0.96	12699
DoS slowloris	0.53	0.15	0.23	1143

TABLE IX. NAÏVE BAYES- GAUSSIAN ON WEDNESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.23797139141742524)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.96	0.18	0.30	12689
DoS slowloris	0.09	0.93	0.17	1153

TABLE X. DECISIONTREE CLASSIFIER ON WEDNESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.9073833261089438)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.95	0.95	0.95	12699
DoS slowloris	0.43	0.41	0.42	1143

TABLE XI. RANDOM FOREST ON WEDNESDAY-WORKINGHOURS.PCAP_ISCX (ACCURACY = 0.9299956653662765)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.95	0.98	0.96	12699
DoS slowloris	0.61	0.41	0.49	1143

From the above observation, we can see that the Random Forest Classifier outperformed all other classifiers in terms of accuracy with a 92.99 % detection rate.

C. Thursday-WorkingHours-Morning WebAttacks.pcap_ISCX

The data captured under this category includes Web Attacks. The three types of web attacks recorded that day i.e. Thursday, July 6, 2017 were Brute Force attack, cross-site Scripting and SQL Injection. Below is the list of tables containing classification done using different classifiers.

TABLE XII. LOGISTIC REGRESSION ON THURSDAY-WORKINGHOURS-MORNING WEBATTACKS.PCAP_ISCX (ACCURACY = 0.9985302763080541)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	3398
Web Attack / Brute Force	0.00	0.00	0.00	4

TABLE XIII. KERNEL SVM ON THURSDAY-WORKINGHOURS-MORNING WEBATTACKS.PCAP_ISCX (ACCURACY =0.9988242210464433)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	3398
Web Attack / Brute Force	0.00	0.00	0.00	4

TABLE XIV. NAÏVE BAYES- GAUSSIAN ON THURSDAY-WORKINGHOURS-MORNING WEBATTACKS.PCAP_ISCX (ACCURACY = 0.6102292768959435)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	0.61	0.76	3398
Web Attack / Brute Force	0.00	0.75	0.00	4

TABLE XV. DECISION TREE CLASSIFIER ON THURSDAY-WORKINGHOURS-MORNING WEBATTACKS.PCAP_ISCX (ACCURACY = 0.9985302763080541)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	3398
Web Attack / Brute Force	0.00	0.00	0.00	4

TABLE XVI. RANDOM FOREST CLASSIFIER ON THURSDAY-WORKINGHOURS-MORNING WEBATTACKS.PCAP_ISCX (ACCURACY = 0.9993016759776536)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	5726
PortScan	0.00	0.00	0.00	2

Looking at the accuracy rate above, it looks like Random Forest Classifier out-performed by a small margin. Naïve Bayes Gaussian did classification with low accuracy.

D. Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX

The data captured under this category includes Port Scanning Attacks conducted on Friday, July 7, 2017. Below is the list of tables containing classification done using different classifiers.

TABLE XVII. LOGISTIC REGRESSION ON FRIDAY-WORKINGHOURS-AFTERNOON-PORTSCAN.PCAP_ISCX (ACCURACY = 0.9996508379888268)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	5726
PortScan	0.00	0.00	0.00	2

TABLE XVIII. KERNEL SVM ON FRIDAY-WORKINGHOURS-AFTERNOON-PORTSCAN.PCAP_ISCX (ACCURACY = 0.9996508379888268)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	5726
PortScan	0.00	0.00	0.00	2

TABLE XIX. NAÏVE BAYES ON FRIDAY-WORKINGHOURS-AFTERNOON-PORTSCAN.PCAP_ISCX (ACCURACY = 0.770949720670391)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	0.77	0.87	5726
PortScan	0.00	1.00	0.00	2

TABLE XX. DECISION TREE CLASSIFIER ON FRIDAY-WORKINGHOURS-AFTERNOON-PORTSCAN.PCAP_ISCX (ACCURACY =0.9991270949720671)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	5726
PortScan	0.00	0.00	0.00	2

TABLE XXI. RANDOM FOREST CLASSIFIER ON FRIDAY-WORKINGHOURS-AFTERNOON PORTSCAN.PCAP_ISCX (ACCURACY = 0.9993016759776536)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	5726
PortScan	0.00	0.00	0.00	2

It looks like multiple classifiers were able to predict well. A thing to note is that the Naïve Bayes Classifier had a low accuracy rate of prediction.

E. Friday-WorkingHours-Afternoon-DDos.pcap_ISCX

The data captured under this category includes Port Scanning Attacks conducted on Friday, July 7, 2017. Below is the list of tables containing classification done using different classifiers.

TABLE XXII. LOGISTIC REGRESSION ON FRIDAY-WORKINGHOURS-AFTERNOON-DDOS.PCAP_ISCX (ACCURACY = 0.9665411034788389)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.97	1.00	0.98	4025
DDoS	0.98	0.70	0.82	488

TABLE XXIII. KERNEL SVM ON FRIDAY-WORKINGHOURS-AFTERNOON-DDOS.PCAP_ISCX (ACCURACY =0.9684453111150505)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	0.98	0.99	0.99	4017
DDoS	0.95	0.81	0.88	496

TABLE XXIV. NAVIE BAYES-GAUSSIAN ON FRIDAY-WORKINGHOURS-AFTERNOON-DDOS.PCAP_ISCX(ACCURACY =0.8526479060491913)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	0.83	0.91	4017
DDoS	0.43	1.00	0.60	496

TABLE XXV. DECISION TREE CLASSIFIER ON FRIDAY-WORKINGHOURS-AFTERNOON DDOS.PCAP_ISCX (ACCURACY =0.9984489253268336)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	4017
DDoS	0.99	0.99	0.99	496

TABLE XXVI. RANDOM FOREST CLASSIFIER ON FRIDAY-WORKINGHOURS-AFTERNOON DDOS.PCAP_ISCX (ACCURACY = 0.9988920895191669)

Predicted Anomaly	Precision	Recall	F-1 score	Support
BENIGN	1.00	1.00	1.00	4017
DDoS	1.00	0.99	0.99	496

From the above observation of the Friday-WorkingHours-Afternoon-DDos.pcap_ISCX, it can be observed that Random Forest Classifier was able to classify the data with highest accuracy of 99.889 %. Therefore, in summary, from the observation of the above five dataset results, it can be concluded that random forest classifiers outperformed other algorithms in most cases. Decision trees came close to the performance of Random Forest, which is not surprising as random forest is composed of decision trees. In addition, Logistic Regression performed well on some of these cases. Following, figure 6 presents a 3-D graph to illustrate the comparison of accuracy in classification of different supervised learning algorithms.

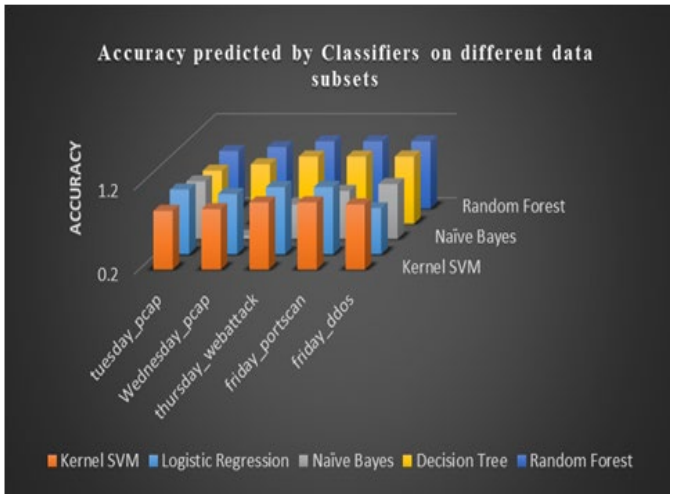

Fig.6. Accuracy in classification of different supervised learning algorithms.

V. CONCLUSION

The main objective of this article was to present a study to design and test the performance of some supervised machine learning algorithms: logistic regression, kernel Support Vector Machine, Gaussian Naïve Bayes, Decision Tree and Random Forest on the CICIDS2017 dataset for network anomaly detection. The performance of classifiers i.e., algorithms was compared based on accuracy, precision, recall and F-1 score. It was found that the Random Forest algorithm essentially yielded the highest accuracy among the classifiers compared. So, to answer the initial research question, whether the Pearson's correlation coefficient used as a feature selection technique enhances the model's performance; it can be credited to contribute to this high efficiency. However, the downside of this feature selection approach is that it is time consuming because it is an iterative process.

REFERENCES

[1] M. R. Ayyagari, N. Kesswani, M. Kumar, and K. Kumar, "Intrusion detection techniques in network environment: a systematic review," Wirel. netw., vol. 27, no. 2, pp. 1269– 1285, 2021.

[2] S. Gamage and J. Samarabandu, "Deep Learning methods in network intrusion detection: A survey and an objective comparison," J. Netw. Comput. Appl., vol. 169, no. 102767, p. 102767, 2020.

[3] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," Eng. Appl. Artif. Intell., vol. 101, no. 104216, p. 104216, 2021.

[4] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of Deep Learning-based network anomaly detection," Cluster Comput., vol. 22, no. S1, pp. 949–961, 2019.

[5] Roshan Ramprasad Shetty, Akalanka Mailewa Dissanayaka, Susan Mengel, Lisa Gittner, Ravi Vadapalli, and Hafiz Khan. 2017. Secure NoSQL Based Medical Data Processing and Retrieval: The Exposome Project. In Companion Proceedings of the10th International Conference on Utility and Cloud Computing (UCC '17 Companion). ACM, New York, NY, USA, 99-105.

[6] Akalanka Mailewa Dissanayaka, Roshan Ramprasad Shetty, Samip Kothari, Susan Mengel, Lisa Gittner, and Ravi Vadapalli. 2017. A Review of MongoDB and Singularity Container Security in regards to HIPAA Regulations. In Companion Proceedings of the10th International Conference on Utility and Cloud Computing (UCC '17 Companion). ACM, New York, NY, USA, 91-97.

[7] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems," Appl. Soft Comput., vol. 92, no. 106301, p. 106301, 2020.

[8] S. Garg and S. Batra, "Fuzzified cuckoo based clustering technique for network anomaly detection," Comput. Electr. Eng., vol. 71, pp. 798–817, 2018.

[9] Adeyanju, I. A., O. O. Bello, and M. A. Adegboye. "Machine learning methods for sign language recognition: A critical review and analysis." Intelligent Systems with Applications 12 (2021): 200056. DOI: https://doi.org/10.1016/j.iswa.2021.200056

[10] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning approach to network intrusion detection," IEEE trans. emerg. top. comput. intell., vol. 2, no. 1, pp. 41–50, 2018.

[11] Akalanka Mailewa Dissanayaka, Susan Mengel, Lisa Gittner, and Hafiz Khan. Vulnerability Prioritization, Root Cause Analysis, and Mitigation of Secure Data Analytic Framework Implemented with MongoDB on Singularity Linux Containers. In The 4th International Conference on Compute and Data Analysis -2020 (ICCDA-2020). San Jose, CA.

[12] Z. Wang, Y. Liu, D. He, and S. Chan, "Intrusion detection methods based on integrated Deep Learning model," Comput. Secur., vol. 103, no. 102177, p. 102177, 2021.

[13] M. Choraś and M. Pawlicki, "Intrusion detection approach based on optimised Artificial neural network," Neurocomputing, 2020.

[14] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using Deep Learning and hyperparameter optimization," J. Inf. Secur. Appl., vol. 58, no. 102804, p. 102804, 2021.

[15] Z. Wu, J. Wang, L. Hu, Z. Zhang, and H. Wu, "A network intrusion detection method based on semantic Re-encoding and Deep Learning," J. Netw. Comput. Appl., vol. 164, no. 102688, p. 102688, 2020.

[16] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine Learning methods for cyber security intrusion detection: Datasets and comparative study," Comput. netw., vol. 188, no. 107840, p. 107840, 2021.

[17] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018.

[18] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Comput. Secur., vol. 86, pp. 147–167, 2019.

[19] A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep Learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,” Knowl. Based Syst., vol. 189, no. 105124, p. 105124, 2020.

[20] Akalanka Mailewa Dissanayaka, Susan Mengel, Lisa Gittner, and Hafiz Khan. Dynamic & portable vulnerability assessment testbed with Linux containers to ensure the security of MongoDB in Singularity LXCs. In Companion Conference of the Supercomputing-2018 (SC18).

[21] T. Elmasri, N. Samir, M. Mashaly, and Y. Atef, “Evaluation of CICIDS2017 with qualitative comparison of Machine Learning algorithm,” in 2020 IEEE Cloud Summit, 2020, pp. 46–51.

[22] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, “Benchmarking of Machine Learning for anomaly based intrusion detection systems in the CICIDS2017 dataset,” IEEE Access, vol. 9, pp. 22351–22370, 2021.

[23] Imteaj, Ahmed, and M. Hadi Amini. "Leveraging Asynchronous Federated Learning to Predict Customers Financial Distress." Intelligent Systems with Applications (2022): 200064. DOI: https://doi.org/10.1016/j.iswa.2022.200064

[24] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, “A hybrid Deep Learning model for efficient intrusion detection in big data environment,” Inf. Sci. (Ny), vol. 513, pp. 386–396, 2020.

[25] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[26] A. D. Lopez, A. P. Mohan, and S. Nair, “Network traffic behavioral analytics for detection of DDoS attacks,” SMU Data Science Review, vol. 2, no. 1, p. 14, 2019.

[27] Simkhada, Emerald, Elisha Shrestha, Sujan Pandit, Upasana Sherchand, and Akalanka Mailewa Dissanayaka. "SECURITY THREATS/ATTACKS VIA BOTNETS AND BOTNET DETECTION & PREVENTION TECHNIQUES IN COMPUTER NETWORKS: A REVIEW, In The Midwest Instruction and Computing Symposium. (MICS), North Dakota State University, Fargo, ND, April 5-6 2019.

[28] Wang, Wentao, Kavya Reddy Mahakala, Arushi Gupta, Nesrin Hussein, and Yinglin Wang. "A linear classifier based approach for identifying security requirements in open source software development." Journal of Industrial Information Integration 14 (2019): 34-40. DOI: https://doi.org/10.1016/j.jii.2018.11.001

[29] L. Breiman, “Random forests,” Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[30] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” IEEE Commun. Surv. Tutor., vol. 21, no. 1, pp. 686–728, 2019.

[31] Wanigasekara, Chathura, Ebrahim Oromiehie, Akshya Swain, B. Gangadhara Prusty, and Sing Kiong Nguang. "Machine learning-based inverse predictive model for AFP based thermoplastic composites." Journal of Industrial Information Integration 22 (2021): 100197. DOI: https://doi.org/10.1016/j.jii.2020.100197

[32] Statistical optimization, soft computing prediction, mechanistic and empirical evaluation for fundamental appraisal of copper, lead and malachite green adsorption. DOI: https://doi.org/10.1016/j.jii.2021.100219

[33] Deepu, T. S., and V. Ravi. "A conceptual framework for supply chain digitalization using integrated systems model approach and DIKW hierarchy." Intelligent Systems with Applications 10 (2021): 200048. DOI: https://doi.org/10.1016/j.iswa.2021.200048

[34] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” Comput. netw., vol. 174, no. 107247, p. 107247, 2020.

[35] Akintaro, Mojolaoluwa, Teddy Pare, and Akalanka Mailewa Dissanayaka. "DARKNET AND BLACK MARKET ACTIVITIES AGAINST THE CYBERSECURITY: A SURVEY.” In The Midwest Instruction and Computing Symposium. (MICS), North Dakota State University, Fargo, ND, April 5-6 2019.

[36] E. C. Blessie and E. Karthikeyan, “Sigmis: A feature selection algorithm using correlation based method,” J. Algorithm. Comput. Technol., vol. 6, no. 3, pp. 385–394, 2012.

[37] M. Savić, V. Kurbalija, M. Ivanović, and Z. Bosnić, “A feature selection method based on feature correlation networks,” in Model and Data Engineering, Cham: Springer International Publishing, 2017, pp. 248–261.

[38] Yang, Chen, Peng Liang, Liming Fu, Guorui Cui, Fei Huang, Feng Teng, and Yawar Abbas Bangash. "Using 5G in Smart Cities: A Systematic Mapping Study." Intelligent Systems with Applications (2022): 200065. DOI: https://doi.org/10.1016/j.iswa.2022.200065

[39] H. Dalianis, “Evaluation Metrics and Evaluation,” in Clinical Text Mining, Cham: Springer International Publishing, 2018, pp. 45–53.

[40] “Sklearn.Metrics.Precision_recall_fscore_support-scikit-learn 0.24.2 documentation,” Scikit-learn.org.

[41] Dissanayaka, A.M., Mengel, S., Gittner, L. et al. Security assurance of MongoDB in singularity LXCs: an elastic and convenient testbed using Linux containers to explore vulnerabilities. Cluster Comput 23, 1955–1971 (2020). https://doi.org/10.1007/s10586-020-03154-7

[42] M. C. Belavagi and B. Muniyal, “Performance evaluation of supervised machine learning algorithms for intrusion detection,” Procedia Comput. Sci., vol. 89, pp. 117–123, 2016.

[43] S.-H. Kang and K. J. Kim, “A feature selection approach to find optimal feature subsets for the network intrusion detection system,” Cluster Comput., vol. 19, no. 1, pp. 325–333, 2016.

[44] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Idris, A.M. Bamhdi, and R. Budiarto, “CICIDS-2017 dataset feature analysis with information gain for anomaly detection,” IEEE Access, vol. 8, pp. 132911–132921, 2020.

[45] Thapa, Suman, and Akalanka Mailewa. "The Role of Intrusion Detection/Prevention Systems in Modern Computer Networks: A Review." In Conference: Midwest Instruction and Computing Symposium (MICS), vol. 53, pp. 1-14. 2020.