International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I12P236

Volume 70 Issue 12, 370-391, December 2022

Original Article

Exploratory Analysis on Anomaly-based IDS Data Using DASK and Ensemble Learning: A Data Parallelization Approach

Abhijit Das[1], Pramod[2]

[1]*Department of CSE, VTU, PESITM, Shimoga,*
CSE, BNM Institute of Technology, Bangalore, Karnataka, India
[2]*PESITM, VTU, Shimoga, Department of ISE, Karnataka, INDIA*

[1]*Corresponding Author : abhijit.tec@gmail.com*

Received: 23 September 2022 Revised: 14 December 2022 Accepted: 17 December 2022 Published: 24 December 2022

Abstract *- Many scholars and practitioners have focused on anomaly detection because of its potential for identifying novel attacks. Unfortunately, due to system complexity, which necessitates extensive testing, assessment, and tuning before the deployment, its applicability to real-world applications has impeded to perform exploratory analysis on anomaly-based network intrusion detection systems (AIDS). The current study's goal was to get valuable insights into the data by applying machine learning techniques. The AIDS data considered for our research is massive and falls under the big data category; CSE-CIC-IDS2018 comprises around one crore sixty lakh samples 1,62,33,002; after Cleaning, 12,52,846 rows and 78 columns were obtained. NSL KDD raw dataset has 1,50,000 after processing 1,35,684 rows with 44 features, and the UNSW-NB15 dataset with 2,5,40,044 rows with 44 features; all these datasets are the benchmark and cover a wide range of attack types. The work adopted an advanced data parallelism approach using DASK and machine learning algorithms. Data parallelism aims to increase processing throughput by partitioning the corpus into concurrent processing streams that all perform the same activities. As a result, widely used benchmark databases like NSL KDD, UNSW-NB-15, and CSECIC-IDS2018 were used in the proposed research work. The work combined Machine learning techniques and parallel execution of data intending to provide state-of-art technology in analyzing big AIDS data and finding relevant features from each.*

Keywords *- Anomaly-based Intrusion Detection System (AIDS), Exploratory Data Analysis (EDA), Machine learning, Statistical approach, IDS datasets.*

1. Introduction

Anomaly-based intrusion detection protects networks and data from unauthorized access. Cyber-attacks and network breaches have increased as technology-based services and devices become more reliant on the Internet and computers. Every organization's network security is at risk as cyber-crimes rise due to simple access to wireless networks via Bluetooth, Wi-Fi, Li-fi, Satellite Communication, WiMAX, Infrared, etc. Education, Healthcare, Banking, Financial Services and Insurance (BFSI), IT & Telecom, Defence, Energy & Power, Retail, Healthcare, and others have been seriously concerned with system security. A network system of any of the above organizations must manage security issues such as vulnerable credentials, policy violations, malware, data theft, distributed denial of service (DDoS) attacks, brute force attacks, insider threats, ransomware, and network intrusions. Intrusion detection systems safeguard computer networks against threats [1]. Such IDS identify malicious activity that could damage computer networks or systems. These systems can respond to attacks and identify policy violations and other security breaches.

This study combines the newest methodologies and machine learning algorithms to perform exploratory analysis on a vast corpus, such as NSL- KDD [2], UNSW-NB15 [3], and CSE-CIC-IDS2018 [4]. A critical analysis was built to help develop an efficient IDS; this involves multiple tests assessing various components of AIDS data and models using different checks. These checks apply multiple criteria and measures. They are the building elements of exploratory data analysis and address many data issues, including model flaws, label uncertainty, date overlap, and data leaks. Each test could give a visible display-ready result in tables, charts, or graphical output to verify predicted findings. This work also discusses research papers used in this study of machine learning algorithms on selected datasets for intrusion identification, along with their pros and cons. This study will help us gain more data insights and design more secure systems or IDSs.

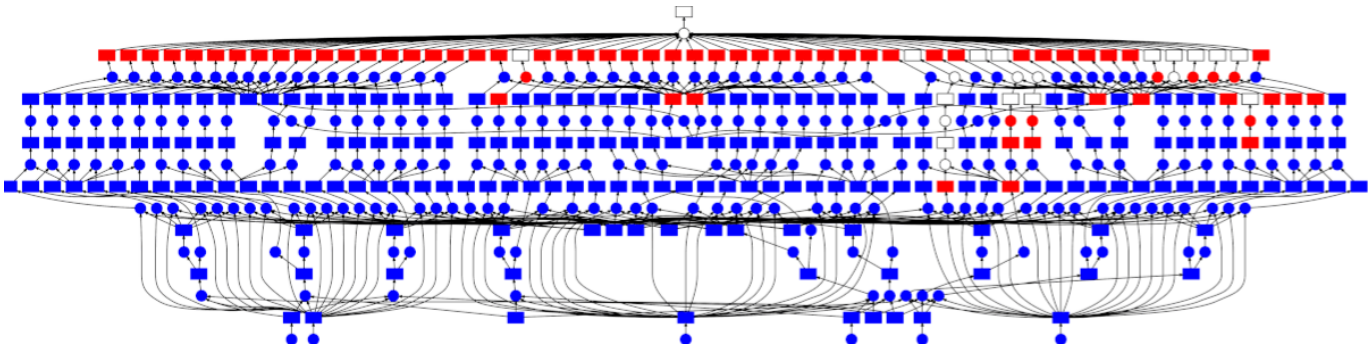


Fig. 1 Advanced Data Parallelism

EDA examines data sets to gain insights and understanding. EDA reveals hidden patterns and relationships in data sets. It can also be used to analyse data quality and identify errors in results. Figure 1 shows the Python-based DASK framework used for data analysis.

DASK is an open source. It uses parallel computing, which carries out multiple calculations or processes simultaneously. It serves as a platform for distributed app development [5]. Only the required data is used or displayed to the user; the data is not loaded instantly but is pointed. DASK is exceptionally rapid and effective with big data since it uses parallel computation and more than just a single-core processor. It stops errors brought on by memory overflow. DASK efficiently completes parallel operations on a single machine using multi-core CPUs.

AIDS using ML methods has grown in popularity recently [6]. These algorithms may be able to analyse data and identify patterns that point to criminal activity. This paper analyses network intrusion detection by comparing the accuracy and FPR of various machine learning (ML) methods. SVM [7], KNN [8], artificial neural networks [9], and decision trees [10] are the most popular ML algorithms for anomaly detection. Even though anomaly detection has been a popular study topic for a long time, it still presents a lot of obstacles due to its complexity and uniqueness, such as rare anomalies and a variety of anomaly categories, such as point, contextual, or group anomalies. Some anomalies are mysteries until they appear or occur. Analyzing anomaly-based network intrusion data with classical and machine learning methods is difficult.

The work suggests a new strategy for enhancing the performance of IDS. On the other hand, simple machine learning procedures are constrained, whereas intrusion detection techniques are evolving and becoming more complex. Therefore, an exploratory analysis was conducted before sending the data for the ML or DL model [11] and selecting relevant features for further processing. Intrusion detection data analysis, feature extraction, and feature selection require advanced learning algorithms. Exploratory analysis was used for the benchmark datasets to visualize and uncover patterns, such as associated characteristics, missing data, and outliers. EDAs are also required for developing hypotheses as to why these patterns arise. The analysis will explain tabular data, data highlights, and dashboards. These datasets can be used to evaluate the proposed system's intrusion detection ability. This work considers NSL-KDD, UNSW-NB15, and CICS-2018 datasets. These datasets have strengths and weaknesses, but all are good for training models and developing IDS [3].

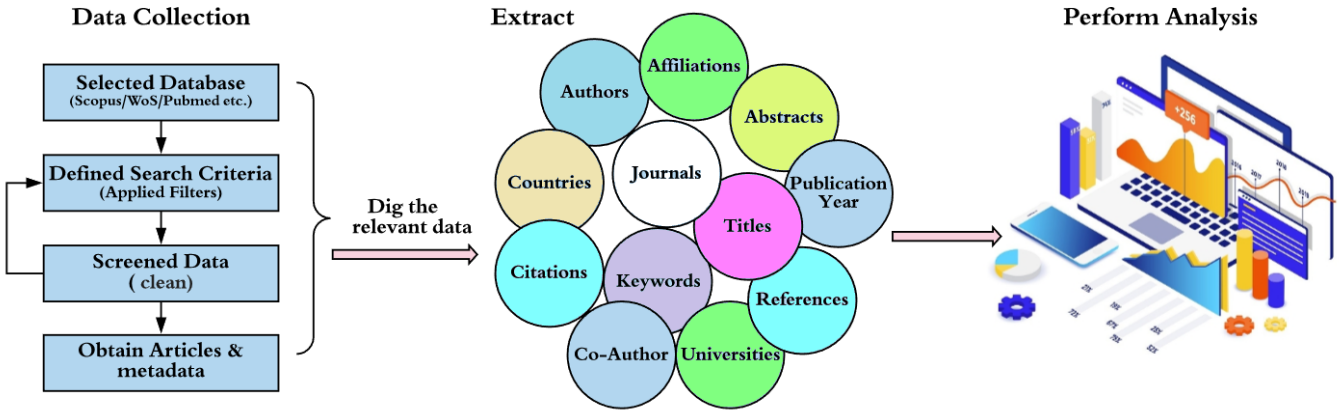


Fig. 2 Bibliometric Analysis Process

2. Background

2.1. Bibliometric Analysis to Analyse Benchmark Dataset

The bibliometric analysis identifies research documents and helps compare network anomaly detection algorithms. Figure 2 shows how Bibliometric Analysis helped to focus and map the study domain. This study extracted publications, removed download metadata, and analysed global development qualitatively and statistically to discover relevant publications. Figure 2 shows the method of gathering research material, geographic distribution, connections, countries, period of related publications, highly cited papers, self-citation, & cooperation. The bibliometric analysis emphasizes scientific discipline structure and links. Bibliometric data and indicators can help to articulate and communicate questions in research and healthcare, especially cardiovascular disease diagnosis and monitoring. Examining publication keywords, output, geographic distribution, and affiliation eliminates subjective biases, validates expert models and data results, highlights leading thinking, and reveals significant links. Quantitative analysis guides future research. Graphs and tables display analysis results.

A literature search utilizing the Scopus database and Google Scholar yielded several bibliometric studies on network intrusion and associated benchmark datasets. Table 1 indicates citation metrics considered for analysing the three-benchmark dataset for analysis. Citation metrics assess a paper's influence.

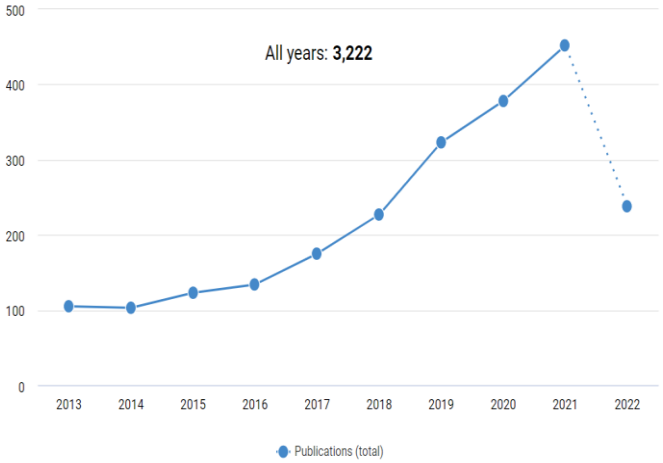


Fig. 3 Published articles considered for Initial Study

Figure 3 displays the annual volume of articles, revealing a steady and relatively rapid growth in published works. This trend predicts a similarly significant increase in the number of publications covering the topic over the next few years.

2.1.1. UNSW-NB 15

It was developed by the IXIA Storm tool in the Cyber Range Lab of the Australian Centre for Cyber Security [12]. Since late 2015, researchers have had access to this dataset to test NIDS. Scholars still use this dataset. Dataset packets were built with tools like IXIA Perfect-Storm. The dataset contains 2,540,044 class-labelled records with 49 attributes covering normal and attacked activities. The database's training set has been criticized for too many duplicates. Analysis, backdoor attack, DoS attacks, worms attack, generic exploits attack, reconnaissance, Fuzzers, & shellcode are attack types. Published Articles using the UNSW NB 15 dataset has depicted in figure 4.

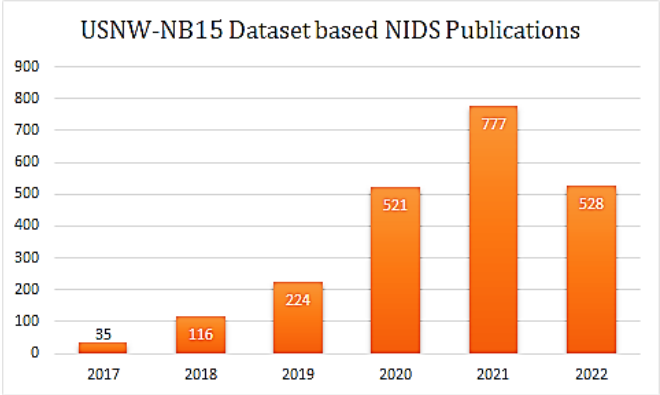


Fig. 4 Published Articles using the UNSW-NB15 dataset

2.1.2. CSE-CIC-IDS2018 Dataset

The University of New Brunswick designed CIC-IDS2018 to evaluate DDoS attack data. It was based on logs from their university's servers that stored DoS attacks over time. The dataset includes BF attacks, Heartbleed, Bot-net, DDoS, website threats, and internal system penetration. The attacker has 50 machines, while the victim has five departments, 420 machines, and 30 servers. Eighty collected features from each device's traffic and system records are contained in the dataset. It covers various attacks; the dataset has 16,233,002 instances, with 17% being attack activity. It's a good choice for testing new intrusion detection algorithms. Published Articles using the CSECIC IDS 2018 dataset has depicted in figure 5.

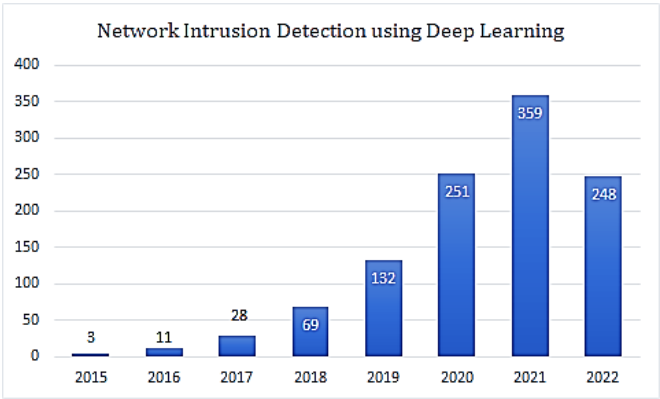


Fig. 5 Published Articles using the CSE-CIC-IDS2018 dataset

Table 1. Existing literature for Benchmark Dataset considered in the study

NSL KDD	UNSW-NB-15	CSECIC IDS 2018
Publication years: 2012-2022	Publication years: 2014-2022	Publication years: 2014-2022
Citation years: 10 (2012-2022)	Citation years: 8 (2014-2022)	Citation years: 8 (2014-2022)
Papers: 1000	Papers: 999	Papers: 425
Citations: 27516	Citations: 13734	Citations: 2447
Citations/year: 3057.33 (acc1=660, acc2=539, acc5=357, acc10=213, acc20=105)	Citations/year: 1962.00 (acc1=615, acc2=469, acc5=308, acc10=183, acc20=93)	Citations/year: 349.57 (acc1=198, acc2=150, acc5=85, acc10=48, acc20=27)
Citations/paper: 27.52	Citations/paper: 13.75	Citations/paper: 5.76
Authors/paper: 2.78/3.0/2 (mean / median/mode ;)	Author /paper: 3.070/3.00/3.0 (mean / median/mode ;)	Authors/ paper-2.98 / 3.00 / 3.0 (mean / median/mode ;)
Age-weighted reference rate: 8861.80 (sqrt = 94.14), 3141.82/author	Age-weighted reference rate: 7217.19 (sqrt=84.95), 2430.06/author	Age-weighted reference rate: 2078.83 (sqrt=45.59), 652.81/authors
Hirsch h - index: 79 (m = 4.41, n = 8.78, 17059 cite=62.0% coverages)	Hirsch h - index: 56 (m =4.38, n =8.00, 7728 cite=56.3% coverages)	Hirsch h - index: 26 (m =3.62, n =3.71, 1570 Cite =64.2% coverages)
Egghe g - index: 144 (g / h=1.82, 20745 cite=75.4% coverages)	Egghe g - index: 97 (g / h=1.73, 9566 Cite =69.7% coverages)	Egghe g - index: 43 (g / h=1.65, 1876 cite=76.7% coverages)
hI, norm: 46	hI, norm: 31	hI, norm: 13
hI, annual: 5.11	hI, annual: 4.43	hI, annual: 1.86
Fassin hA-index: 40	Fassin hA-index: 38	Fassin hA-index: 23

2.1.3. NSL-KDD

NSL KDD Cup database has been built to address KDD99's duplicate and redundant packet problems [13, 14]. After removing duplicate and extraneous records, the collection has 150,000. In KDD-CUP 99 dataset, there are 38 types of threats in the testing set, & 24 variants for the training set, and each connection has 41 features. As mentioned in author evaluations, KDD Cup 99 contains issues [15]. The dataset comprises duplicate and redundant entries and other errors that affect classifier performance. More frequent ones have inflated numbers. Connections are grouped by difficulty. It helps classifiers recognize challenging attacks during training. NSL KDD has similar issues as KDD-CUP 1999 [13]. The attacks in this dataset are old and don't reflect a modern network. Synthetic origin can't be fixed without a new recording—published Articles using NSL-KDD Dataset as shown in figure 6.

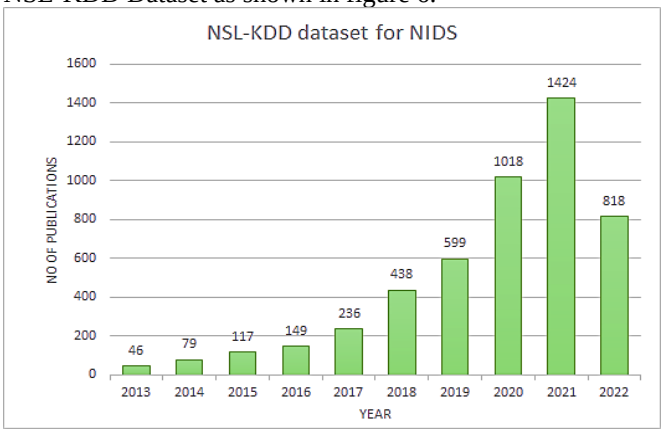


Fig. 6 Published Articles using the NSL-KDD dataset

In summarizing the section, it's crucial to highlight that research on AIDS is still popular, as indicated by the increase in publications from 2013 to 2022. With such a massive global output, it isn't easy to imagine studying the topic's literature without adequate tools and methodologies for interpreting the data in the available bibliographic corpus, especially since these databases are developing to become more comprehensive sources of information. This fact provides a significant obstacle for scholars and anyone seeking the truth about observable reality. A literature evaluation can't be based exclusively on numbers. Concerning the researcher's objective and purpose of the literature review, it is also necessary to conduct content analysis, which identifies articles with valuable content. This work delves into the literature published on three different datasets covered in this analysis from machine learning, big data, deep learning, and data research perspectives.

3. Literature Survey

3.1. Anomaly-Based Approaches

Anomaly-based IDS (AIDS) has been an active study issue for several decades. Due to its wide use in the military, national agencies, healthcare, forensics, risk management and assessments, compliance, security and privacy, financial surveillance, IoT, IoMT, and AI safety. Despite being researched extensively in advanced data analytics, pattern recognition, computer vision, NLP, statistics, and ML, it still has distinct issues and impediments that require advanced methodologies. Recent years have seen a rise in computational intelligence, transfer learning, and deep learning-based anomaly identification. However, scientific

advancement in this method is not thoroughly evaluated. This paper will address essential difficulties, cutting-edge techniques, how they tackle problems, and future potential. Figure 7 shows intrusion detection strategies. Many machine learning, cognition-based, statistical, and deep learning methods [16] may detect network intrusions. SVMs, decision trees, ANNs, and k-nearest neighbors are common (k-NN). Each method has strengths and drawbacks, so choosing the ideal one for data collection and application is crucial.

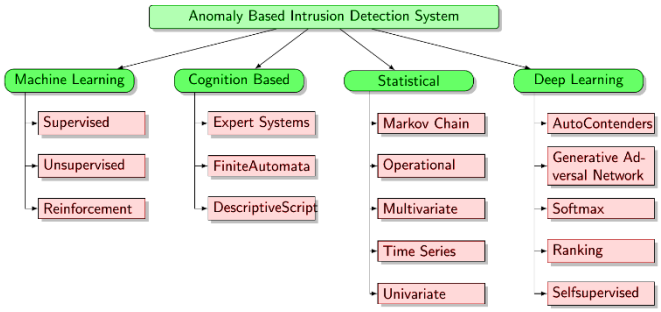


Fig. 7 Taxonomy of Anomaly-based IDS

ML algorithms for IDS frequently require a lot of training data. Most real-world networks don't have labeled datasets. Many machine learning algorithms are designed for static data sets, yet network traffic data is dynamic. This means typical machine learning models may not work. Even if an algorithm performs well on a dataset, it may not perform well on real-time traffic owing to changing network conditions.

Despite these challenges, machine learning algorithms can provide good intrusion detection. Semi-supervised or unsupervised Learning methods don't require as much training data as supervised approaches. Newer DL models, including LSTMs [16], outperform standard Machine Learning models on specific temporal/sequence-based prediction tasks. As always, it is important to experiment with multiple models to find the one that works best for a particular application and dataset.

3.2. Machine Learning Perspective

Machine learning for intrusion detection is an exciting research area. Machine learning can detect a range of attacks by automatically learning data patterns that signal malicious or unusual activity. This study reviews recent research on ML for IDS, focusing on current methods. SVMs have supervised learning algorithms that may detect anomalies in network traffic. They require a training dataset to learn malicious activity patterns. SVMs can handle high-dimensional data and nonlinear patterns but not complicated ones. They need a lot of training data and can't detect unusual anomalies. SVMs can detect anomaly-based network intrusions. The advantages of using SVMs for detecting anomaly-based network intrusions include: (1). SVMs process high-dimensional data. (2). SVMs recognize nonlinear patterns. (3). SVMs are considered powerful machine learning algorithms. The disadvantage of using SVMs' is they may struggle with high-dimensional data. SVM may not manage noise or outliers [17].

K-NN is good at detecting nonlinear patterns but can be computationally intensive. Decision trees can be good at both linear and nonlinear patterns, but they can be prone to overfitting [18]. Merits: KNN is a straightforward technique with a low learning curve. This algorithm is also called a lazy method. KNN is a versatile algorithm that can be used for various problems. Since KNN is a non-parametric method, it does not assume anything about the information processed. KNN can be used with a variety of distance measures. De-merits: KNN is a computationally expensive algorithm. KNN requires a large amount of data to be effective. KNN is sensitive to noisy data.

ANNs are also excellent at identifying anomalies in network traffic. They are computationally expensive and can't detect rare anomalies [18]. DTs detect network abnormalities. Decision Tree algorithms are a strong tool for AIDS in network data as they can handle nonlinear features and recognize complex behavior patterns. Decision trees might overfit if the training data doesn't indicate actual patterns. Decision trees are costly to train and demand a lot of memory.

SVMs are the best machine learning tool for network anomaly detection and need much training data. ANNs detect traffic irregularities. ANNs are highly suited for this task since they can learn complicated patterns from data and can be utilized for real-time detection. ANNs can learn complex patterns from data, which is one of their key benefits for anomaly detection. This is useful for anomaly identification since the ANN can learn normal behavior and mark changes as anomalies. Real-time detection is another benefit of ANNs. It means abnormalities can be detected immediately, not later. Table 2 depicts the use of ML algorithms to detect Intrusions. ANNs are computationally intensive, making them unsuitable for large networks. ANNs can also produce false positives, meaning typical behavior might be classified as abnormal.

3.3. Limitations of Reviewed Models

An IDS doesn't block or restrict malicious activity; it detects it. As a result, an IDS must be a complete plan comprising other security measures and personnel who understand how to respond effectively.

- Since IDS can't view encrypted packets, attackers can utilize them to access the network. An intrusion detection system (IDS) will not detect these interferences until they are further into the network, leaving the processes vulnerable until the interruption is found. Encoding will become increasingly common to protect privacy.

Table 2. Use of ML techniques by Authors to detect Intrusions

Year	Vulnerability Type	Detection Method	Features	Dataset	Accuracy	Ref
2020	Web Traffic	Random forest Algorithm, Gradient boosting machine	Generalizability	CSIC 2010 v2 and CICIDS2017, NSL KDD, UNSW NB 15	94.46%	[19], [20]
2020	Host or Network Attacks	Hybrid of Decision Tree and KNN	Network Security	NSLKDD, KDD-Cup99	99.33%	[21]
2019	Ping of death, denial of Service, Host Attack	Expectation Maximization (EM), SVM, NB, KNN, HNB	Improves the strength of contemporary method	NSI-KDD, KDD-Cup99, ADFA-LA and AFDA-WD	78%	[22]
2019	U2R and R2L Attacks	FAIS, FCAAIS	Scope of detecting novel attacks	NSL-KDD	88%	[20]
2018	Denial of Service, Distributed Dos	ANN, K-Means, SVM	Categorizing network attacks based on source	KDD-99, NSL-KDD	97.25%	23]
2020	Packet-Level Attack	LSTM, OHE	Packet-level feature extraction improves real-time speed	ISCX2012, USTC -TFC2016, CICIDS2017	99.74%	[24]
2019	U2R attack, R2L, DoS, Probe	LSTM - LR, NB, KNN, SVM	Enhanced Performance	KDD-CUP-1999, NSL KDD, UNSW NB 15, WSN DS, CICIDS2017, ADFA-LD, ADFA-WD	93.50%	[25]
2021	Network Traffic	Ensemble Approach with C4.5, Random Forest (Forest PA)	High Accuracy and Efficiency	NSL-KDD, AWID, CIC-IDS2017	99.52%	[26]
2018	Denial of Service, Distributed Dos	NB, Decision table and Stochastic gradient descent	High Accuracy	NSL-KDD	99.93%	[27]
2018	network Traffic	SVM, Random Forest Algorithm	Performance Improvement	NSL-KDD	99.33%	[28]
2017	Denial of Service, Distributed Dos	Modified KNN Classifier	Efficiency in classified instances	NSL-KDD	98.70%	[29]

- One major problem with IDS is that they frequently expose us to FP. In many circumstances, FP outnumbers real attacks. Although a well-adjusted IDS can significantly lower FP, it still requires human attention. True attacks may slip over or be neglected if they do not take care to check the false positives.
- Because NIDS examines protocols as they are collected, they are vulnerable to the same protocol-related vulnerabilities as web servers. Software faults and incorrect data can crash a NIDS.
- The classifier cannot accurately classify the same instance (events). This lowers the system's accuracy and detection rate.
- Alerts can be a benefit or a curse. Reviewing all the alerts and figuring out who tried to break in takes a lot of time and effort.

4. Methodologies: Exploratory Data Analysis

Data analysis has many dimensions and ways and is applied in business, research, and social science. The most common categories are shown in Figure 8. Exploratory Data Analysis (EDA) identifies patterns and relationships in data to test hypotheses [30]. EDA allows the analyst to understand data more efficiently.

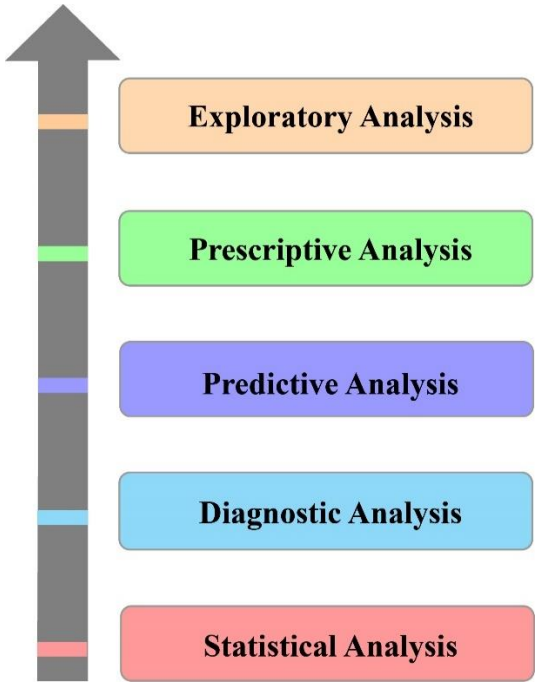


Fig. 8 Types of Analysis

EDA uses visual tools, including histograms, scatterplots, and boxplots. However, other methods, such as summary statistics, can also be used. EDA is essential for understanding the data and identifying potential problems. It is also a valuable tool for communicating the results of the data analysis to others. The work starts by looking at the data to find patterns and understand their meaning. Then, these patterns were used to make different ML models. The performance of models was analyzed, and the best detecting network intrusions techniques were chosen. Exploratory data analysis is an excellent way to do research in many ways. Here are some advantages: it improves data comprehension and identifies data variables and their relationships, finds data outliers and data mistakes, reveals data patterns, and improves experiment design and data accuracy.

4.1. Pre-Processing

Pre-processing removes inaccurate, corrupted, poorly formatted, redundant, or missed data from a dataset. Data duplication or labeling errors are unavoidable when data is acquired from several sources in different formats. Additionally, the algorithms and outputs are incoherent even if the information is correct. Data cleaning procedures differ from dataset to dataset, depending on the task. Therefore there's no one-size-fits-all approach. The preprocessing steps have shown in figure 9.

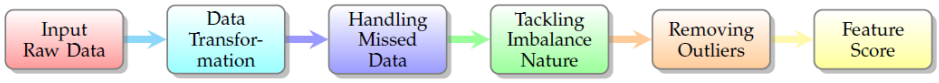


Fig. 9 Preprocessing

4.1.1. Raw Input Data

The benchmark dataset was gathered online. This data is generally in the form of unstructured nature with noise.

4.1.2. Data Transformation

It converts raw input data into a machine learning-friendly structure. This step contains feature engineering, which creates new features from raw data by replacing, renaming, dropping, adding, altering data types or converting data structures, label encoding for categorical data, etc.

4.1.3. Handling Missing Data

The preprocessing algorithms take care of the missing data in this step. They either impute the missing values or remove the data points that have missing values. Some columns from each dataset had missing cells, and the value was filled using the KNN imputation method.

4.1.4. Remove Data Imbalance

Data imbalance is a problem that occurs when the training data is not evenly distributed among the different classes. This problem can be solved by oversampling or under-sampling the data. Both techniques were applied to make the data free from an imbalance nature.

4.1.5. Remove Outliers from Data

Outliers are data points far from the rest of the data. They can be removed using various outlier detection algorithms, and three outlier algorithms were selected to remove such values.

4.2. Statistical Modelling of AIDS Data

Statistical data analysis inspects and models data to identify important statistical information, offer conclusions, and enhance decision-making. The entire dataset listed in table 3 was reviewed rapidly. It requires doing some elementary statistical analyses and making some elementary visuals. For instance, It was necessary to count the features, identify each feature's data type—text or numeric—count missing and duplicate rows, memory used to collect the information, create bar charts for all categorical data, etc. Statistical analysis helps to choose the sample from the overall population. Weighted, stratified, and random sampling techniques were employed at various levels of analysis.

Table 3. Statistical Overview of three Data sets

Dataset Statistics	CIC-2018	UNSW NB15	NSL-KDD
Number of Variables	79	45	42
Number of Rows	1.2528×106	175341	494020
Missing Cells	0	0	0
Missing Cells (%)	0.00%	0.00%	0.00%
Duplicate Rows	117437	0	348436
Duplicate Rows (%)	9.40%	0.00%	70.50%
Total Size in Memory	755.1 MB	95.6 MB	158.3 MB
Average RowSize in Memory	632.0 B	571.4 B	336.0 B
VarTypes Numerical - Nr	Nr: 60	Nr: 34	Ca: 14
Categorial -Ca	Ca: 19	Ca: 11	Nr: 28

4.3. Sampling of Population

Weighted sampling evaluates the importance of each population unit [31]. Each unit's weight is usually determined by size or location. This weighting guarantees that population-impacting units are given more consideration. Before sampling, CSE-CIC-IDS2018 had 1252846 rows and 78 columns; after sampling, 10000 were picked. A weighted sample accounts for biases like non-coverage, non-responses, and oversampling of particular observations with uneven probabilities [32]. If a biased data set is not rectified and a random sampling approach is used, population descriptors

(e.g., mean, median) will be skewed and not accurately represent the population's proportion. Weighted sampling eliminates sample bias by generating a sample based on population proportions. Weighted sampling produces a random and unbiased selection.

Missing value analysis can be done in a few ways, but the most popular is perceptual analysis [33]. It can impute them or eliminate the rows or columns if missing values are missing. Automation is another option. Check all columns using python's isnull, isnan, and is empty functions on the DASK data frame. Once discovered, use imputation to fill in missing values. Missing values can be handled by: 1) Remove missing rows or columns 2) Fill missing values forward or backwards 3) Substitute mean, median, or mode for missing data 4) Predict missing values using a model

The most straightforward solution for dealing with incomplete data is to remove rows or columns that have them, which means that data will be lost. A popular method for dealing with null data is to use the mean, median, or mode to fill in the gaps. However, this can introduce bias. Using a predictive model to infer missing values is accurate but difficult.

4.3.1. Descriptive Analysis

Descriptive analysis is the fundamental kind of data analysis [34]. It's simple and popular. The descriptive analysis explains what happened using historical data. It helps identify data trends, patterns, and distribution. Mean, standard deviation, variance, sum, skewness, kurtosis, and coefficient of variation help understand the data set. They show data distribution. Before doing statistical tests, assess the data.

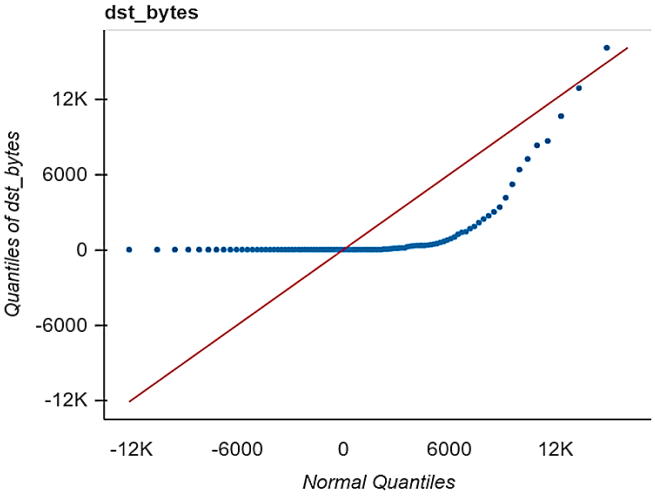
Descriptive Statistics-NSL-KDD Dataset

	Mean	Standard Deviation	Variance	Sum	Skewness	Kurtosis	Coefficient of Variation
Duration	314.1	2568.22	6.60E+06	314051	11.1978	142.8	8.1777
dstbytes	1392	5813.86	3.38E+07	1.39E+06	17.3155	414.6	4.1757
count	81.88	108.533	11779.429	81883	1.3162	1.169	1.3255
srvcount	23.04	61.1525	3739.634	23040	5.1992	31.08	2.6542
srcbytes	7946	176451	3.11E+10	7.95E+06	26.4235	732.9	22.2054
hot	0.164	1.9007	3.6127	164	13.7185	192.4	11.5897
serrorrate	0.296	0.4539	0.206	296.14	0.8978	-1.184	1.5328
srvserrorrate	0.295	0.454	0.2061	294.54	0.9041	-1.174	1.5413
rerrorrate	0.131	0.335	0.1122	131.28	2.1733	2.746	2.5518
samesrvrate	0.642	0.4435	0.1967	642.2	-0.4868	-1.691	0.6905
diffsrvrate	0.069	0.1893	0.03585	69.41	4.0494	15.94	2.728
srvdiffhostrate	0.091	0.2514	0.0632	91.04	2.9794	7.603	2.7614
dsthostcount	180.8	99.9448	9988.9623	180762	-0.7995	-1.126	0.5529
dsthostsrvcount	109.2	110.63	12238.903	109171	0.4001	-1.685	1.0134
dsthostsamesrvrate	0.498	0.4512	0.2035	497.58	0.0912	-1.88	0.9067
dsthostdiffsrvrate	0.08	0.178	0.03167	80.38	3.585	12.72	2.2142
dsthostsamesrcportrat	0.154	0.3155	0.09952	154.39	2.0306	2.502	2.0433
dsthostsrvdiffhostrate	0.034	0.1155	0.01334	34.22	5.4552	34.41	3.3755
dsthostserrorrate	0.297	0.4527	0.2049	297.21	0.8934	-1.189	1.523
dsthostsrvserrorrate	0.293	0.4528	0.205	292.79	0.915	-1.156	1.5464
dsthostrerrorrate	0.121	0.307	0.09426	120.89	2.3158	3.592	2.5396
dsthostsrvrerrorrate	0.131	0.3315	0.1099	130.59	2.1959	2.869	2.5383
level	19.54	2.1065	4.4371	19536	-2.2394	7.893	0.1078

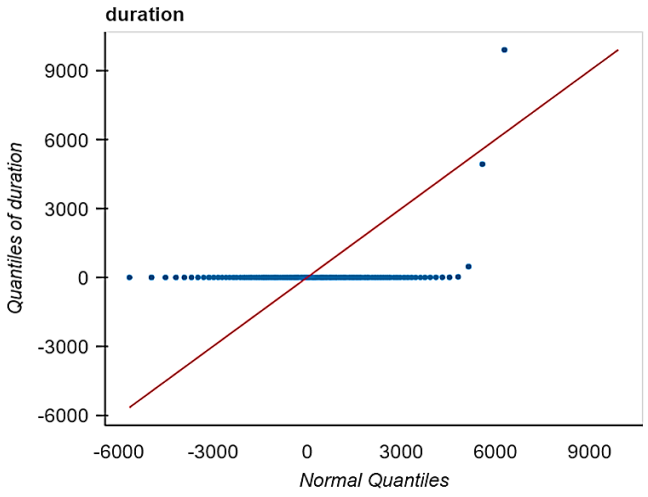
Fig. 10 Statistical Analysis of Features from NSL-KDD Dataset

Figure 10 shows that data in the last row is skewed -2.2, with a kurtosis of 7.89 and a coefficient of variation of 0.107. It means the data is less typical and has a higher peak. If there are few observations in each class, this could explain the low descriptive variables and high variance in duration, destination, and source bytes. The data may also be nonlinear. A linear model can't accurately learn the data's variable relationships. Data scatter plots can confirm this. A nonlinear model may be better if the data isn't linearly separable.

QQ plots check if a dataset is usually distributed. Data are plotted against a normal distribution. If data is allocated correctly, QQ plot points will be straight [35]. If data isn't properly distributed, points won't be straightforward. The data analysis included a QQ plot for numerous reasons. First, a QQ plot can provide a rapid visual check for regularity. Second, even if the data is not normal, a QQ plot can help identify data outliers. It can be useful information when the model tries to understand the data. Figure 11 shows the Q-Q plots from the KDD-NSL dataset.



(a)Destination Bytes



(b) Duration

Fig. 11 Q-Q plots from the KDD-NSL dataset

The p-value is the probability that the results from a statistical test are due to chance. It measures how near data points are to the fitted line. Low p-values indicate closeness to the fitted line. A p-value less than 5 suggests the finding is statistically significant and unlikely to be random. A p-value of 4.241697284117977e-25 suggests the chance is meager. Some P-values are listed below, which are calculated for the Q-Q plot.

- Feature hot is not normally distributed (p-value 4.241697284117977e-25)
- dst_host same srv_rate is not normally distributed (p-value 1.1145088708759414e-20)
- dst_host srv count is not normally distributed (p-value 1.82758255607032e-17)
- the count is not normally distributed (p-value 2.245324056361952e-24)
- dst_bytes is not normally distributed (p-value 5.1216506635126285e-25)
- duration is not normally distributed (p-value 4.23714067884298e-25)

Figure 12 shows a KDE plot that visualizes data distribution. The plot estimates data density at each place and creates a smooth curve to depict it. It helps to understand data distribution and spot outliers.

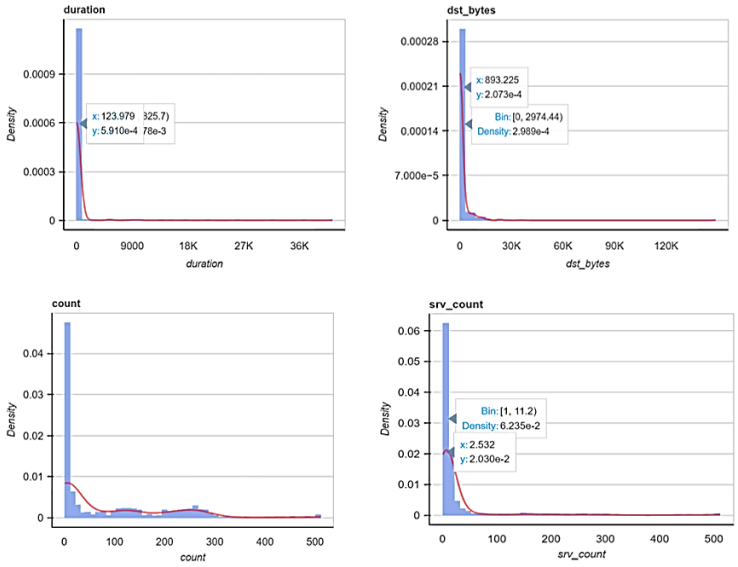


Fig. 12 KDE plots of NSL-KDD Dataset

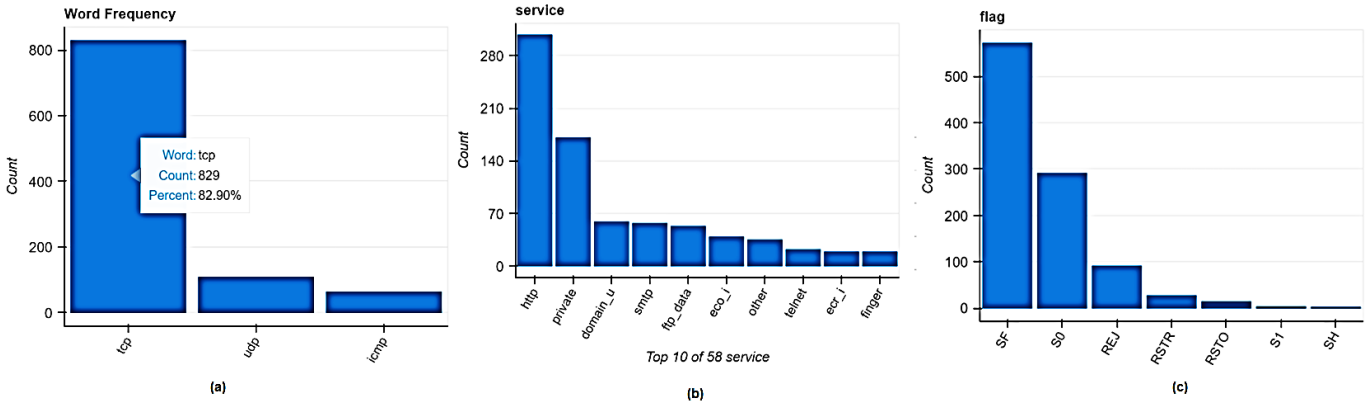


Fig. 13 Categorial Distribution plots from NSL-KDD Dataset

Figure 13 shows that the greatest value (http) is more than 1.8 times greater than the next biggest value (private), the highest value (SF) is about 1.97 times bigger than the next greatest value (S0), and the highest value (tcp) is more than 7.68 times greater than the next biggest value (s0) (udp). Checks for pairwise correlation between the features: Not more than nine pairs are correlated above 0.90, Correlation >0.90 for sets:

[(’FlowDuration’, ’FlowIATMean’),
(’FlowDuration’, ’FlowIATMax’),
(’TotBwdPkts’, ’TotLenB- wdPkts’),
(’TotLenBwdPkts’, ’TotLenFwdPkts’),
(’FwdPk-tLenMax’, ’TotLenFwdPkts’),
(’FwdPktLenMax’, ’FwdPk- tLenMean’),
(’FwdPktLenMax’, ’FwdPktLenStd’),
(’FwdPk- tLenMax’, ’Protocol’)]

Feature Selection using Random Forest Regres- sor: DstPort', 'Protocol', 'TotBwdPkts', 'FwdPktLen- Min', 'FlowByts/s', 'FwdIATTot', 'FwdPSHFlags', 'BwdURGFlags', 'FINFlagCnt', 'RSTFlagCnt', 'PSH-FlagCnt', 'CWEFlagCount', 'FwdByts/bAvg', 'Fwd-BlkRateAvg', 'BwdByts/bAvg', 'BwdPkts/bAvg', 'BwdBlkRateAvg', 'InitFwdWinByts', 'ActiveMean', 'ActiveStd','Label'

Correlation analysis measures two variables' relationship. The correlation coefficient measures strength from -1 to 1. +ve correlation suggests the two variables move in the same direction, while -ve means the reverse. Zero means the two features are unrelated. Using Pearson correlation, 37 of 79 features with a threshold between 0.8 and 1.0 was excluded because they were significantly associated and would not help with categorization. Correlation analysis helps examine variable-relationship ideas. It can detect correlations between variables to predict future behavior. Correlation analysis helps comprehend variable relationships and anticipate future behavior. Sequential Feature Selector (SFS) chose the best 20 features out of 42, as shown below.

Feature Score of selected features: Features: 1/20 – score: 0.5794692024195237

Features: 2/20 – score: 0.9445855230363396 Features: 3/20 – score: 0.9592631847605082 Features: 4/20 – score: 0.9617898219971565 Features: 5/20 – score: 0.9618853699586416 Features: 6/20 – score: 0.9613138927909466 Features: 7/20 – score: 0.9612358198266954 Features: 8/20 – score: 0.9633996926129814 Features: 9/20 – score: 0.9638854487420283 Features: 10/20 – score: 0.9629860592075955 Features: 11/20 – score: 0.9633483923948595 Features: 12/20 – score: 0.9638250542211247 Features: 13/20 – score: 0.961399730187546 Features: 14/20 – score: 0.9634849675074 11 Features: 15/20 – score: 0.96262578515393 Features: 16/20 – score: 0.9632772169200626 Features: 17/20 – score: 0.9633904981779301 Features: 18/20 – score: 0.9614960141226471 Features: 19/20 – score: 0.9624580073014549 Features: 20/20 – score:

Table 4. Classification Report

Class	Precision	Recall	F1-Score	Support
0	1.0	1.0	1.0	291153
1	1.0	1.0	1.0	11533
2	1.0	1.0	1.0	11310
3	1.0	1.0	1.0	41287
4	1.0	1.0	1.0	17184
5	1.0	1.0	1.0	2543
6	1.0	1.0	1.0	691
7	1.0	1.0	1.0	100
8	0.95	0.70	0.81	27
9	1.0	0.71	0.83	21
10	1.0	0.20	0.33	5
Accuracy	1			
Macro-Avg	1.0	0.87	0.91	
Weighted-Avg	1.0	1.00	1.00	

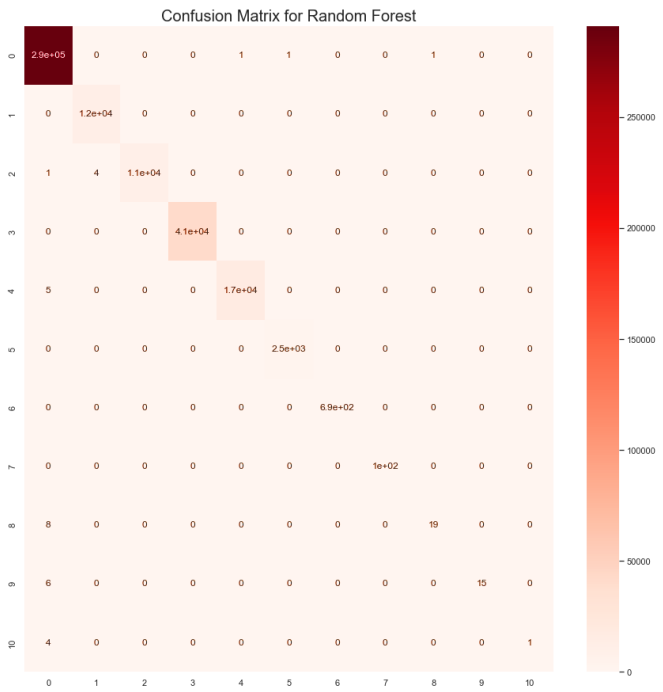


Fig. 14 Confusion Matrix for Random Forest Classifier.

Random Forest Classifier results: The accuracy is found to be Training Accuracy: 0.99999657921623 and Testing Accuracy: 0.9999175211651332. The Confusion Matrix for the RF method represents the classification report, as depicted in figure 14 and table 4.

4.3.2. Univariate Analysis

Univariate analysis analyses single-variable data. This analysis aims to understand one column better. It produces column-specific statistics and visualizations. The lowest and maximum values, distinct counts, median, or variance were calculated. A standard Q-Q plot can be made to contrast the data to the normal curve, and a box plot can be used to examine anomalies. Columns with unique values in all rows were checked. Found 10 out of 78 columns with a single value:

['Bwd_PUSH_Flags','Fwd_URGENT_Flags', 'Bwd_URGENT_Flags', 'CWE_Flag_Counts', 'Fwd_Bytes /b Average', 'Fwd_Packets /b Average', 'Fwd-Blk_RateAvg', 'Bwd_Bytes /b Avg', 'Bwd_Packets /bAverage', 'Bwd-Blk_Rate_Average']

4.3.3. Bivariate Analysis

Bivariate analysis is the statistical method used to determine the relationship between two features. This relationship can be linear or nonlinear. Bivariate analysis can be used to determine cause-and-effect relationships between the two variables. It can also forecast how the variable of interest will change in response to the separated variable's changes.

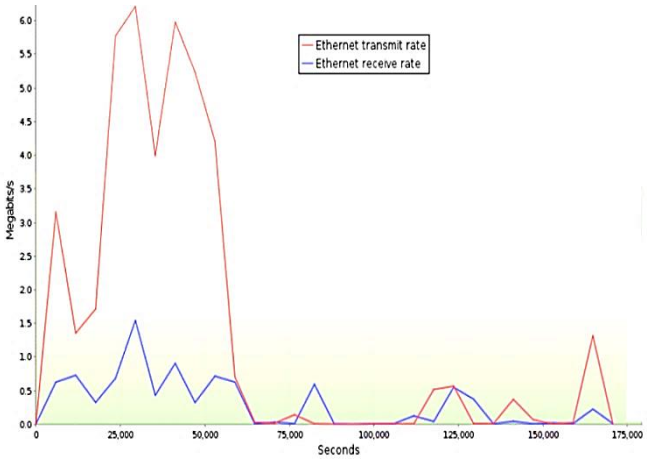


Fig. 15 Bivariate analysis of Attributes from UNSW-NB15 Dataset

Some examples of distribution-based insights discovered at CSE-CIC-IDS2018 include Constant Length (28), Similar Distribution (14), Total forward packets, Sub flow forward packets, Total Backward packets, Sub flow Backward packets, Total Length forward packets and Sub flow forward Bytes, Total Length Backward packets, and Sub flow Backward By Disparate (59) Destination Port, Flow Duration, Total Forward Packages, Total Reverse Packages, etc.

Negatives Init Forward Win Bytes has 287064 (22.91%) negatives; Init Backward Win Bytes has 619741 (49.47%) negatives. Bivariate analysis of Attributes from UNSW-NB15 Dataset as shown in figure 15.

4.3.4. Multivariate Analysis

Multivariate analysis studies numerous variables simultaneously by identifying variable correlations and data patterns. Multivariate data analysis uses regression, factor, and cluster analysis. These strategies can be used to study data from diverse perspectives and find relationships not apparent from the data alone. Statistical techniques were used to determine the association between the three dataset's properties. Graphs show the results. Multivariate data analysis can examine attack behavior, threats, and weaknesses. By reviewing numerous variables at once, more accurate predictions can be formed. UNSW-basic NB15's features are flow-based, time-related, content-based, additional generated characteristics, and labeled features. General Purpose Features are those with a number from 36 to 40. Features 41–47 are referred to as connections. Figure 16 shows the analysis of 3 features at a time from UNSW-NB15 Data. These patterns and trends can help to identify what is happening in the system and how it can be improved.

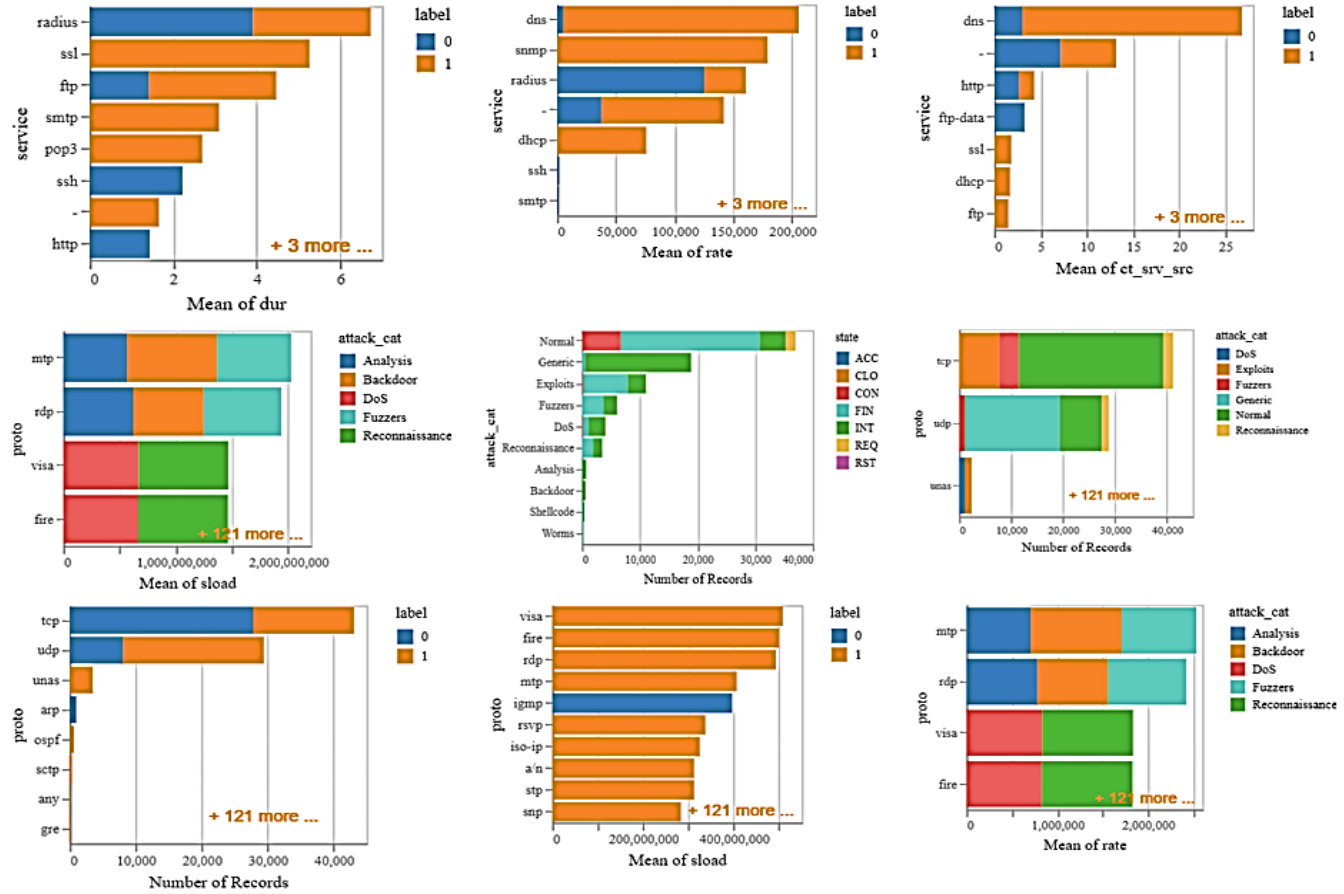


Fig. 16 Analyzing 3 features at a time from UNSW-NB15 Data

4.4. Diagnostic Data Analysis

The diagnostic analysis identifies, collects, and analyses network intrusion data to generate conclusions about a system's state. Data can come from system logs, network devices, application data, and user data, but it must be handled and evaluated consistently to be meaningful. After processing and analyzing the data, conclude. These results can improve system security. They can also identify areas needing more research.

4.4.1. CSE-CIC-IDS2018

- Null hypothesis H0 regarding data duplicates or repeated data ratio is <= 0% after the experiment observed 3.87% duplicate samples in CIC-IDS2018.
- H0, there were columns with only a single non-repetitive value in all rows. Obtained 10 out of 78 columns with a single value:' BwdPUSHFlags',' FwdURGENTFlags', 'BwdURGENTFlags', 'CWEFlagCounts',' Forward Bytes/bAvg',' ForwardP- kts/bAvg',' ForwardBlkRateAvg', 'BackwardByts/bAvg', 'BackwardPkts/bAvg',' BackwardBlkRateAvg.'
  . H0 sections with low-performance scores in training data: A section with an Accuracy of 0.2, compared to the average of 0.929 in the datasets, is shown in the study. There are 800 data samples used for the testing. Overall, the average Accuracy rating is 0.93.

4.4.2. UNSW-NB15

- Under the category mismatch: obtained 2 out of 4 features with a ratio of new category samples above threshold:' proto':' 5.47%',' attack cat':' 0.5%'
- One feature out of forty-four was found to have a PPS difference greater than the cut-off value of 0.24. Only two of the train dataset's forty-four attributes have PPS values over the cut-off in the feature-label correlation change: ' attack cat':' 1',' id':' 0.76'.
- "Mixed data" means a combination of different forms of information. No columns with even a trace of type mixing were rejected, and 45 were accepted.

4.4.3. NSL-KDD

- Saw 1 out of 18 features with the ratio of new category samples above threshold:'hot':' 1%'
- Got 2 out of 41 features with PPS difference above threshold:' diffsrvrate':' 0.27',' srcbytes':' 0.3'
- Observed 7 out of 42 columns with a single value: [' land',' urgent',' numfailedlogins',' rootshell',' suat-tempted',' numoutboundcmds',' ishostlogin'].
- Noted 11 out of 42 columns with a single value: [' land',' wrongfragment',' urgent',' num- failedlogins',' numcompromised',' rootshell',' suat- tempted',' numroot',' numshells',' numoutbound- cmds',' ishostlogin'].
- There are 11 high-variance new features.
- Checked pairwise Correlation which is greater than 0.9 for pairs [('serrorrate', 'srvserrorrate'), ('rerror- rate', 'srvrerrorrate'), ('dsthostserrorrate', 'flag'), ('dsthostsrvserrorrate', 'flag'), ('dsthostrerrorrate', 'flag'), ('dsthostsrvrerrorrate', 'flag'), ('flag', 'serror-rate'), ('flag', 'srvserrorrate'), ('flag', 'rerrorrate'), ('flag', 'srvrerrorrate')].
- Finding Mixed data type: In contrast to the 42 features with some type of mix, there are zero features with a minimal amount of type mix.

4.5. Predictive Analysis

Predictive analysis of network intrusion data can be used to identify and predict future attacks. This analysis can be used to understand past attacks and trends and identify potential future attacks. Organizations can be better prepared to defend against future attacks by understanding past attacks. Additionally, predictive analysis can help determine which systems and data are most at risk and which security measures are most effective.

Predictive analysis is a powerful tool for security, but it is not perfect. False positives can occur, and predicting future attacks is impossible. However, predictive analysis can be a valuable part of a security strategy and help organizations prepare for future attacks.

4.5.1. Methodology

The methodology studies the methods used in a particular area of research. Different techniques were combined in this predictive analysis to find answers to the hypothesis on data, training, testing, and performance evaluation and provide an interpretation of the obtained results. It is a way of thinking about and doing research that is based on a set of principles and methods. Similar preprocessing steps were followed, as discussed in the previous section. After preprocessing, different types of analysis were applied to the data mentioned in the previous section. Apart from these details, different check mechanisms were conducted to obtain more interesting insights into the three datasets.

Perform a Data integrity test: The first step is to check the integrity of the data. It includes checking for missing values, outliers, and other data anomalies such as new classes or labels, redundant data, dominant Frequency Change, presence of only single values in rows of the dataset, label ambiguity, mixed data types, mixed-nulls, presence of special characters, string length out-of-bounds, string mismatch and their comparisons. These kinds of checks are made for all three datasets. Information about these artifacts will make it easy for the researcher to remove them and apply rectification methods to prepare the data for the next step in the pipeline.

Train-test data validation: The next thing to do is to divide the information into a train-test set. The entire data was fragmented using the DASK library into several

partitions. A Block Partition is a simple partitioning of data into equal-sized blocks. This is the default partitioning used by DASK arrays, data frames, and bags. A HashPartition is a partitioning of data based on the hashing of the data. This is typically used when the data is not evenly sized or when the data is already in some order. Block partitioning was selected, as shown in Figure 17. The Seven partitions can be accessed and used for computations individually.

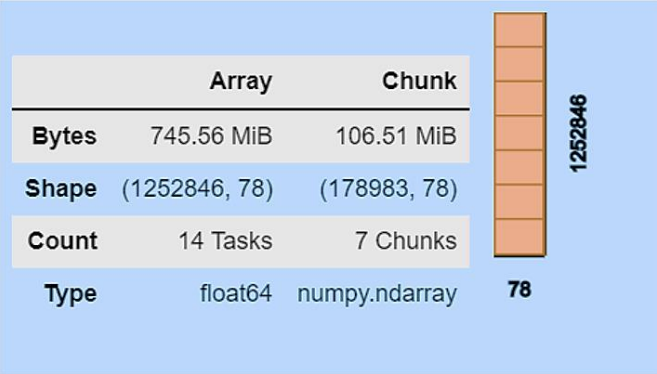


Fig. 17 Partitioning of Entire data.

Further, every partition can also be subdivided based on the need. Partitions Wall time: 380 ms, 19.1 ms, 02.12 ms, 02.59 ms, 5. 02 ms, 6. 02.05 ms, 7. 03.48 ms.

The four rectangles represent the data chunks, and the mean technique is applied to every segment as the data passes from left to right. DASK's calculations are slack. DASK will initially create a link between tasks as a computational graph. The graph is then optimized to decrease calculations. A network of related jobs was developed with data dependencies from the given array operations, then executed this graph concurrently using many threads. It is represented in Figure 18. Track of space and time for every transaction was kept. It shows the working Profile of every instance and how much time is taken to perform the task on massive data. DASK is a flexible parallel computing library for analytics. It easily scales code from a single machine to a cluster. It also works well with popular data analysis tools like NumPy, pandas, and Scikit-learn. Hence, different operations were performed; therefore, the model used this library to tackle the big data computation issue for analysing the data.

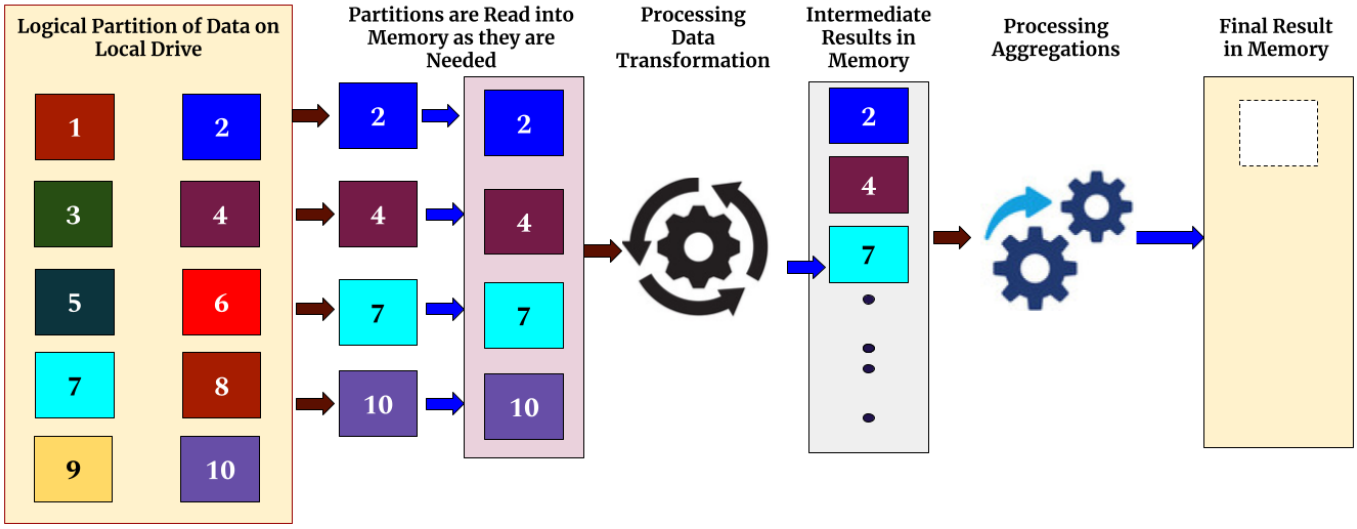


Fig. 18 Data Analysis

Model evaluation: The model is evaluated on the test set. It includes assessing accuracy, precision, recall, and other metrics.

4.5.2. Train-Test Validation

The integrity check and preprocessing steps are explained in previous sections. Before training the model, It is necessary to bi-furcate the entire data into three parts: train, test, and validation. A set of data was applied to fit the created model; this data set is called the training dataset; always the training is performed on a large partition of the data, usually in the ratio of 80:20 or 75:25, etc. Similar to how the validation dataset is kept apart from the training dataset, the testing set is utilized to provide a balanced analysis of the most recent model's capacity to select the models.

The validation set illustrates training data from the model typically used to gauge model competency while adjusting the model's hyperparameters. A subset of data called a validation set is utilized to employ an unbiased assessment of how well a method fits the train data by varying hyperparameters. The evaluation becomes increasingly biased as proficiency with the validation data becomes a necessary component of the model configuration. The test set is most generally used to explain model

assessment compared to other fully tuned models. In contrast, the validation set is most frequently used to represent model assessment during tuning hyperparameters and data preparation. The train test samples mix was checked, and it found 34.83% (109080 / 313212) of the testing database exists in the training database. The testing-Training ratio is 0.33. The training Size is 939634, and the test is 313212. The change in feature label correlation was calculated using a predictive power score (PPS). It returns a Score for each characteristic to determine how well it can predict a label. Predictive Power Metric, or PPS, is a form of a scoring system that can be used to determine whether; there are linear or nonlinear associations between two features in a given dataset. It is asymmetric in nature and data-type independent. The range of PPS values is 0 means no predictive power, to 1, indicating the highest predictive power. Since PPS normalizes the data, it is significantly more reliable. It allows us to determine how valuable a parameter would be in forecasting the values with another parameter in a particular dataset. A PPS score of 0.8 or less is generally regarded as good and indicates that a specific column X is likely to predict the values of a subsequent column Y accurately. In figure 19, it should be suspicious of data issues if:

- Train set PPS scores significantly high: This may suggest that data leaking, or the fact that the feature already has relevant information on the label, is the real cause of the feature's success in label prediction.
- There is a significant disparity between train and test PPS (training set PPS is higher): A feature that was strong in the train but weak in the test can be explained by data leaking in the train that is unrelated to a new dataset, which is an even
- There is a significant disparity between test and train PPS if the score is higher for test data: A peculiar value can be a sign of test sample drift that results in a coincidental correlation to the target label.

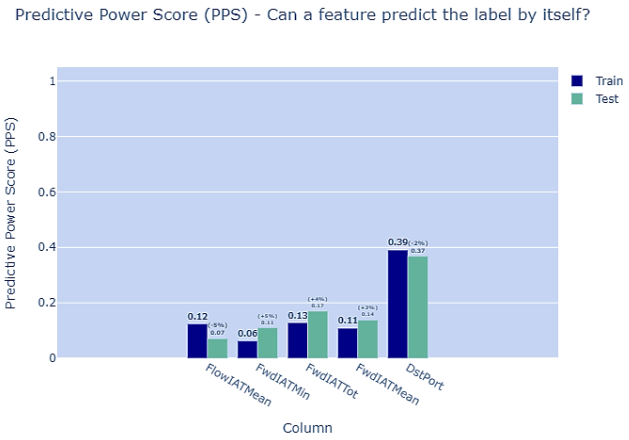


Fig. 19 Predictive Power Score.

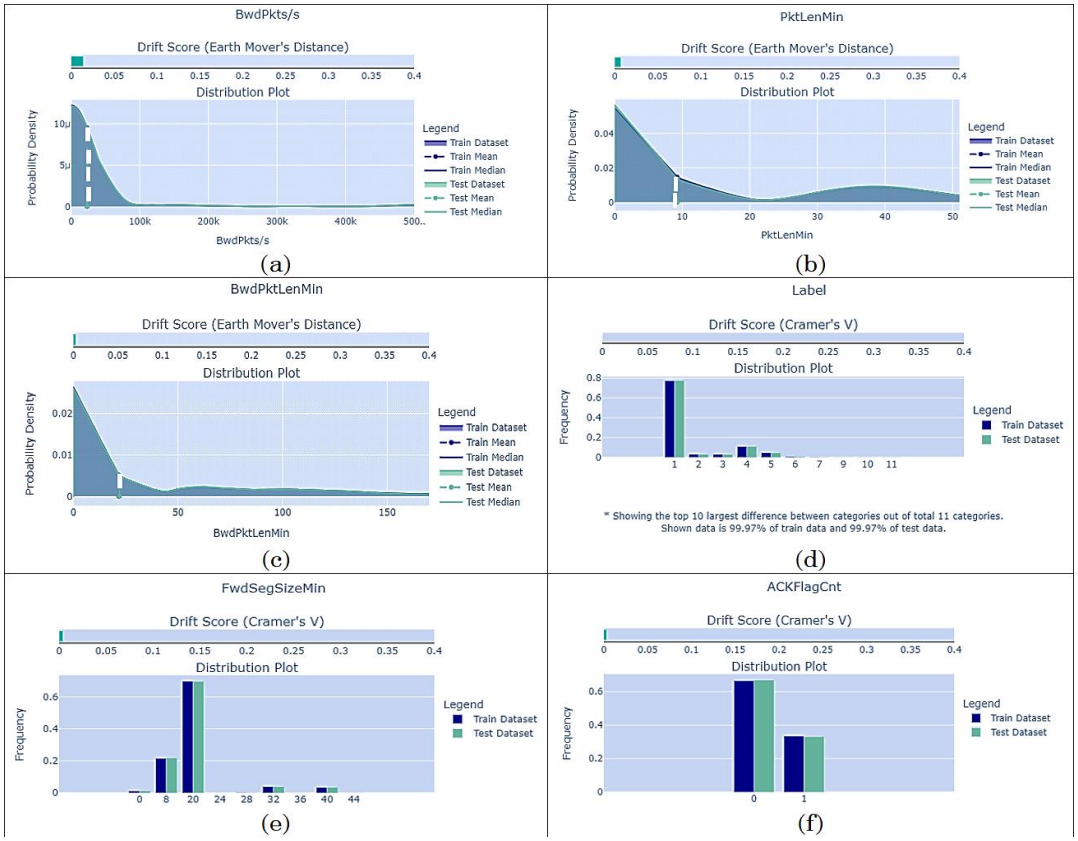


Fig. 20 Drifts in the Train-Test dataset.

4.5.3. Feature Drift

The feature drift of the training and testing dataset was calculated by applying statistical measures such as Earth movers' distance and creamer's rule. The earth movers' distance measure quantifies the amount of" work" needed to move one distribution of points to another. It is often used in machine learning to compare the similarity of two data sets. The Earth Mover's Distance (EMD) is a technique to assess dissimilarity between two multi-dimensional distributions. This distance is" lifted" by the EMD from discrete features to whole distributions. It is similar to the chi-square.

The change in an entity's position for a reference point is known as drift. Data drift refers to a change in the data's distribution, which drives model drift. In the context of operational ML models, this is the distinction between real-time production data and a baseline set of data, usually the training set, which is also representative of a task the model is supposed to execute. String data deviation score < 0.2 and numerical drift score < 0.1. Passed 77 columns out of 77 columns. Found column" FwdSegSizeMin" has the highest categorical drift-score: 4.08E-3, and the column" BwdPkts/s" has the highest numerical drift-score: 0.01. The difference between two distributions, in this case, the training and testing distribution values, are measured by the Drift score. Figure 20 shows the resulting drifts in the Train-Test dataset.

4.5.4. Train Test Performance

The relative drop in the Train-Test score is below 0.1. Provide a summary of the model's effectiveness on the training and testing samples using the chosen scorers. Figure 21 displays the assessment error calculations.

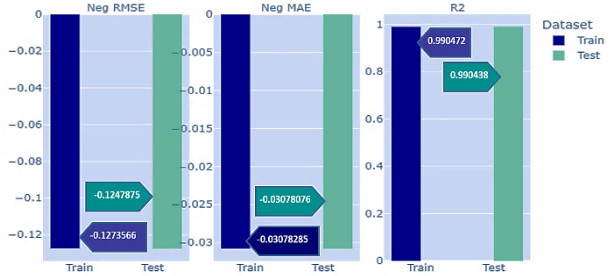


Fig. 21 Errors calculated for performance evaluation.

4.5.5. Train Test Prediction Drift

The categorical drift score is < 0.15, and the numerical drift score is < 0.075. Found model prediction Earth Mover's Distance drift-score of 6.07E-4 as shown in figure 22.

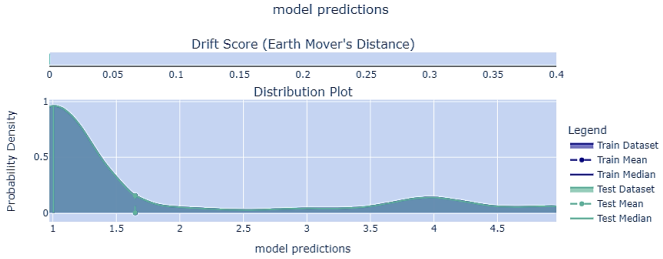


Fig. 22 Train Test Model Prediction Drift Distribution.

4.5.6. Weak Segments Performance

There are multiple segments of data, some segments pass the test, and few fail to specify conditions. Such segments are referred to as weak segments. In order to see where the model falls short, tests were run in those areas where it is known to be soft and present the resulting visualization. In Figure 23, the attributes above the line represent a selection of the most relevant characteristics. The characteristics below the line define the underutilized features with the most unpredictability.

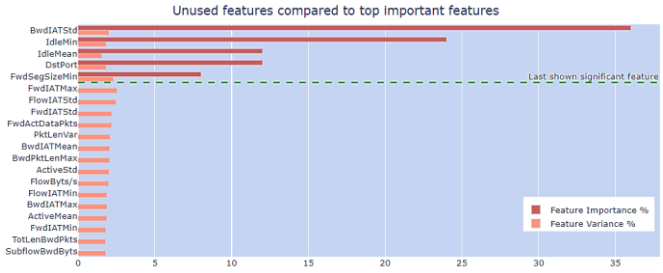


Fig. 23 High Variance features of the CSE-CIC-IDS2018 dataset.

The pre-analysis hypothesis for NSL-KDD was that less than five high-variance characteristics are not being used. Figure 24 displays the nearly 12 high-variance traits that were not used during the experiment. The attributes above the line represent a selection of the most crucial ones. In contrast, the features below represent the least-used features with the most significant volatility, as determined by the check parameters.

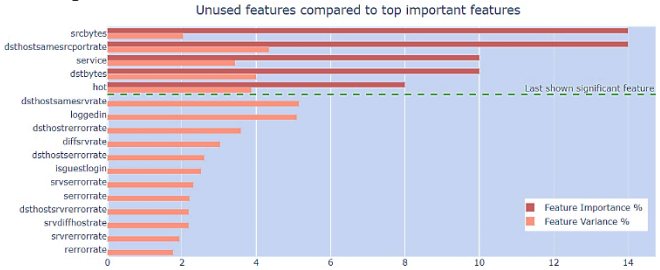


Fig. 24 High Variance features of NSL-KDD Dataset.

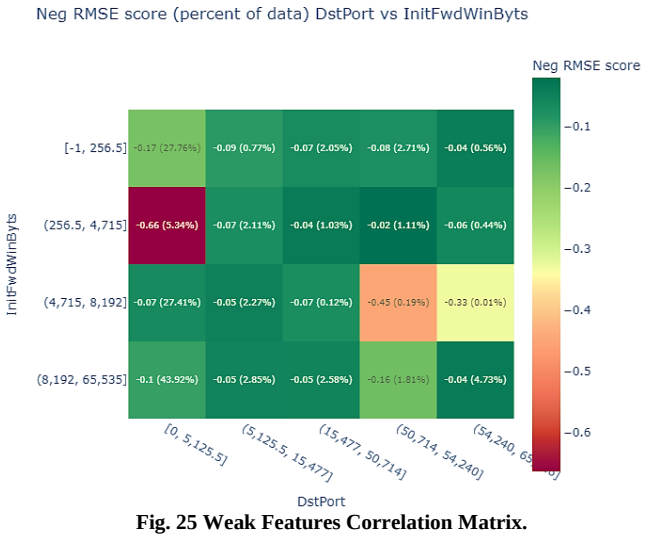


Fig. 25 Weak Features Correlation Matrix.

The regression Error Distribution of both the Train and test Datasets is almost similar. The kurtosis value is more significant than - 0.1. Found kurtosis value 1,725.35997. The systematic regression error was checked: The bias ratio is less than 0.01. Found bias ratio 2.30E-3. Increasing overfitting can occur if users run a gradient-boosted model with too many iterations. The average iterative test performance is less than 5 percent of the highest. Results in a 0% drop in the score were discovered. Figure 25 displays the Neg RMSE produced for each subgroup of forecasting models for both the train set and the testing set, as the check restricted the boosting strategy to use up to N predictions each time.

4.6. Ensemble Learning

Ensemble learning is a machine learning technique that combines multiple models to produce better results than any single model. It is more adaptable as well as fewer data are sensitive. The two popular ensemble learning procedures are bagging and boosting. In Figure 26, compare the selected model with a simple one - This ensures that this model somewhat performs better than the straightforward model. Indicating the proposed model is not good enough to complete the assigned task efficiently. In this case, the Gradient Boosting Regressor (GBR) model was chosen and compared with the baseline model. This GBR model is a part of the ensemble learning model. Ensemble learning is often used to improve the predictive accuracy of a model. Still, it can also be used to enhance a model's interpretability or reduce the computational cost of training a model. Regression, Gradient Boosting Regressor, and Decision tree were used in this analysis.

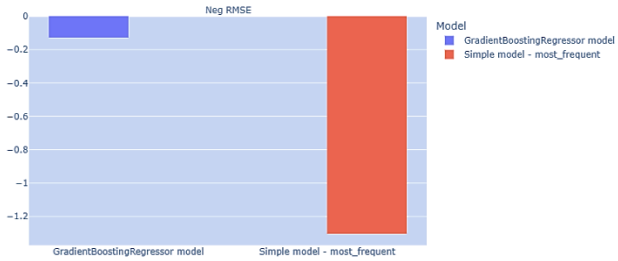


Fig. 26 Model Comparison

A gradient boost regressor method can improve predictive accuracy by reducing the error of a model [42]. It is a boosting algorithm that builds a model by combining the predictions of multiple weak models. The individual models are typically decision trees. Gradient boost regressor is a relatively simple algorithm, making it computationally efficient. Gradient boost regressor effectively detects network attacks since it can identify patterns indicative of malicious activity. Model Evaluation has listed below:

- Performance of Data: Training-Testing results in relative degradation is < 0.1
- Roc Related: area under the curve score for all the classes is > 0.7.
- Train Test Prediction- Drift: Textual-drift score < 0.15 and numerical-drift score < 0.075
- Simple Model Comparison: Model performance gain over simple model > 10%
- Weak Section Performance no-of-rows: The relative performance of the weakest segment > 80% of mean model performance.
- Regression Systematic-Error: Bias ratio < 0.01
- Regression Error Distribution: Kurtosis value > −0.1.
- Unused Features: Number of high variance unused features <= 5.
- Boosting-Overfit: Test score over iterations < 5% from the best score.
- Model Inference-Time: Mean model inference-time for 1 − sample < 0.001.

The non-zero mean of the error distribution indicated the systematic error in model predictions, as shown in figure 27.

Fig. 27 Model Prediction Error

4.6.1. Perspective Data Analysis

From a network perspective, data anomalies can be classified into three types: normal, suspicious, and malicious. Normal data is characterized by a lack of patterns or irregularities, while suspicious data is characterized by patterns deviating from expected behaviour. Malicious information is represented by patterns that indicate an attempt to exploit a system or network. When analysing data for anomalies, it is crucial to consider both the content of the data and the context in which it was collected. The content of the data can provide clues as to the nature of the anomaly, while the context can provide information on the potential impact of the anomaly. Anomaly-based network intrusion detection involves identifying suspicious or malicious patterns in network traffic data. These patterns may indicate an attack on a system or network, or they may simply be the result of normal activity. In order to determine which anomalies are worthy of further investigation, it is important to consider both the context in which the data was collected and the content of the data itself.

5. Results and Discussions

The null hypothesis is that the AdaBoosting regressor model has an Area Under the Curve (AUC) score higher than 0.7 across the board. All classes were successful after additional data and model analysis; class 1 had the lowest AUC.

An AUC of over 0.7 for every class on the NSL-KDD training and testing data was achieved. The minimal AUC found was for class 0, although all categories passed. The highlighted areas represent the best possible thresholds for making decisions. Youden's index [37], defined as sensitivity + specificity - 1, is used to calculate these values. The Youden indexes for the training set are 0.496, the testing set is 0.519, the TPR is 100%, and the false positive rate is 0%. Table 5 displays the results of the model's testing. Using the AdaBoost Algorithm, the receiver operating characteristics for the NSL-KDD dataset, the UNWS-NB15 dataset, and the CSE-CIV-IDS2018 dataset are depicted in Figures 28, 29, and 30, respectively.

Table 5. Performance of the Model

Class	Training Data			Testing Data		
	TPR (%)	FPR (%)	Youden's Index	TPR (%)	FPR (%)	Youden's Index
1	87.46	38.82	0.244	79.31	45.45	0.224
2	100	0.00	0.456	100	92.39	0.221
3	100	96.25	0.223	100	92.39	0.223
4	100	13.59	0.232	100	19.77	0.232
5	100	72.36	0.194	100	28.28	0.218
6	60	22.04	0.218	100	0	0.995

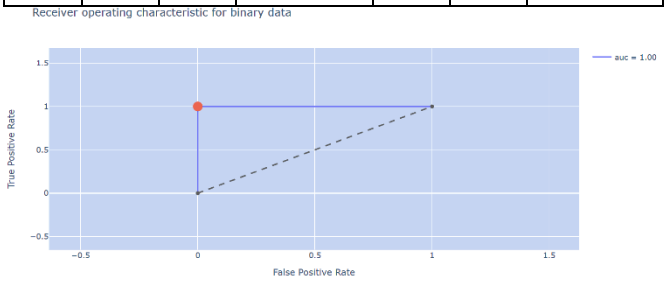


Fig. 28 Receiver Operating Characteristics of NSL-KDD Dataset Using AdaBoost Algorithm.

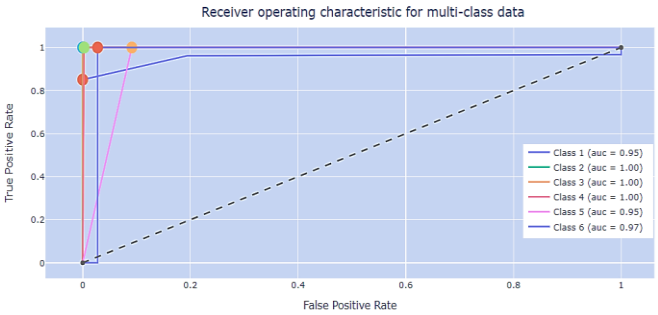


Fig. 29 Receiver Operating Characteristics of UNWS-NB15 Dataset Using AdaBoost Algorithm.

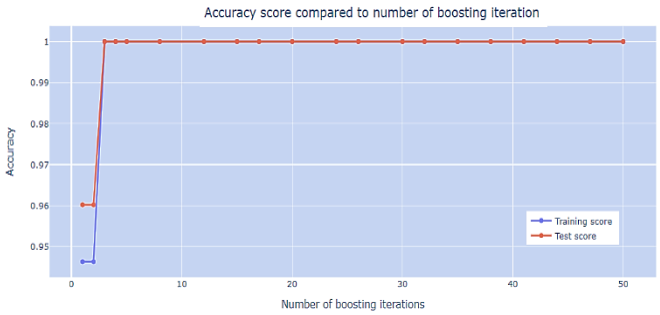


Fig. 30 Receiver Operating Characteristics of CSE-CIV-IDS2018 Dataset Using AdaBoost Algorithm.

The model assesses how well each feature can predict the Train-Test features by using the Confusion Matrix for NSL-KDD, UNSW-NB15 dataset (shown in Figures 31, 32), which returns the feature label correlation changes displayed in Figure 33 and the Predictive Power Score of all features. There is less than a 0.2-point disparity in predicted Power Scores. Only 3% of the 77 characteristics tested showed a PPS difference greater than 0.05.

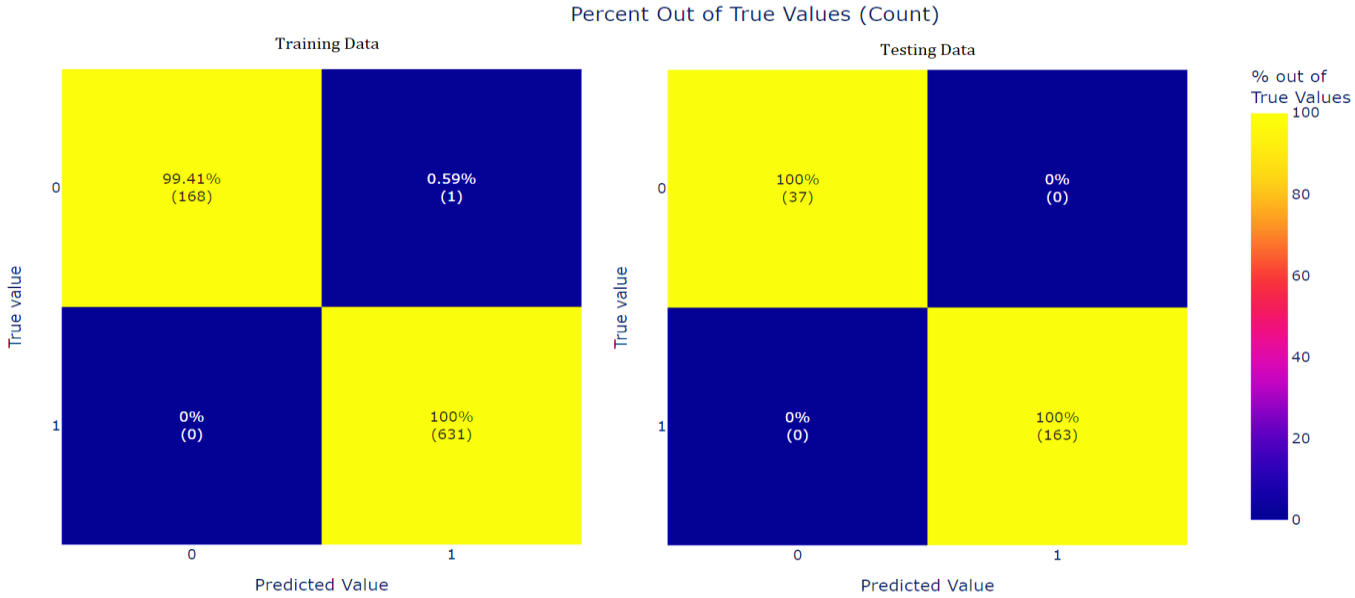


Fig. 31 Confusion Matrix of NSL-KDD Dataset Using AdaBoost Algorithm.

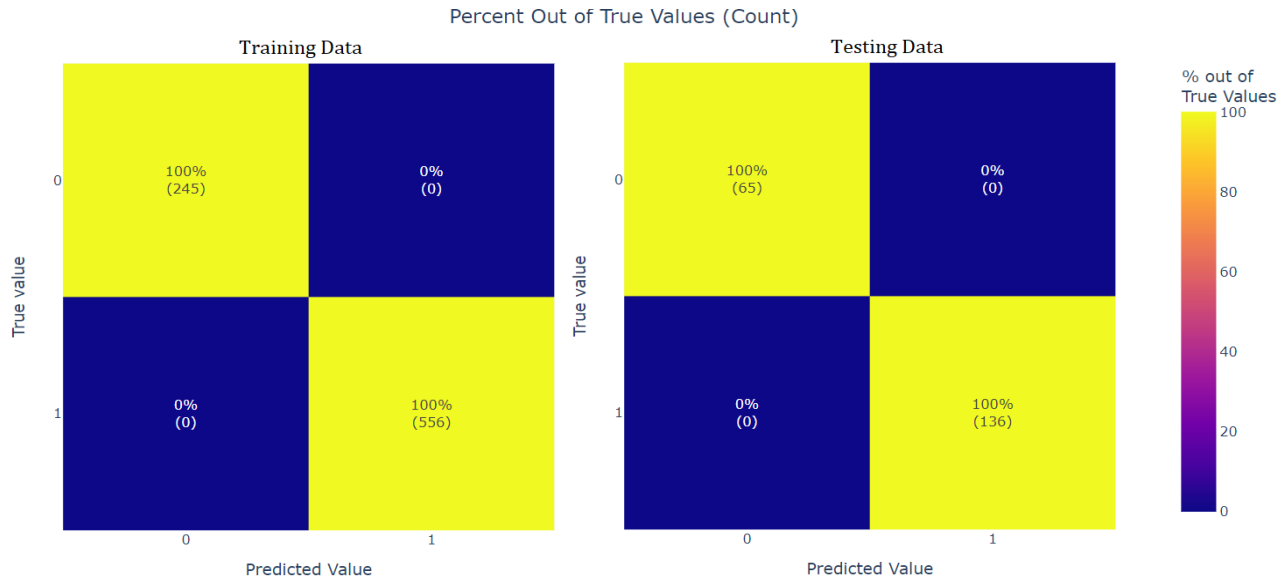


Fig. 32 Confusion Matrix of UNSW-NB15 Dataset Using AdaBoost Algorithm.

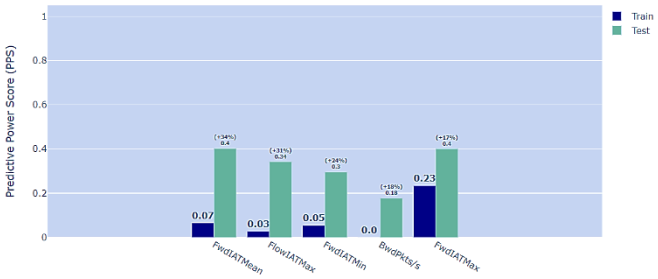


Fig. 33 Feature Label Correlation Change

' FlowIATMax':' 0.31',' FwdIATMean':' 0.34',' FwdIATMin':' 0.24'. Train features' Predictive Power Score is less than 0.7. Seventy-seven relevant columns were passed. The model average inference time (in seconds) per sample was calculated. For the training dataset, the average model inference time for one sample (in seconds) was: 5.918e−05; for test data, the average model inference time for one sample (in seconds) was 0.0001086. The confusion matrix of the model was calculated for training data, as shown in figure 34.

IdleMin, IdleMean, DstPort, and BwdIATStd were identified as the characteristics with poor performance in the preceding section. As seen in Figure 35, these overlaps highlight features that correspond with poorly placed segments. Comparing the accuracy of a classifier's probabilistic predictions is made possible via calibration curves [38]. For binned predictions, it compares the actual probability of the excellent label to the projected likelihood.

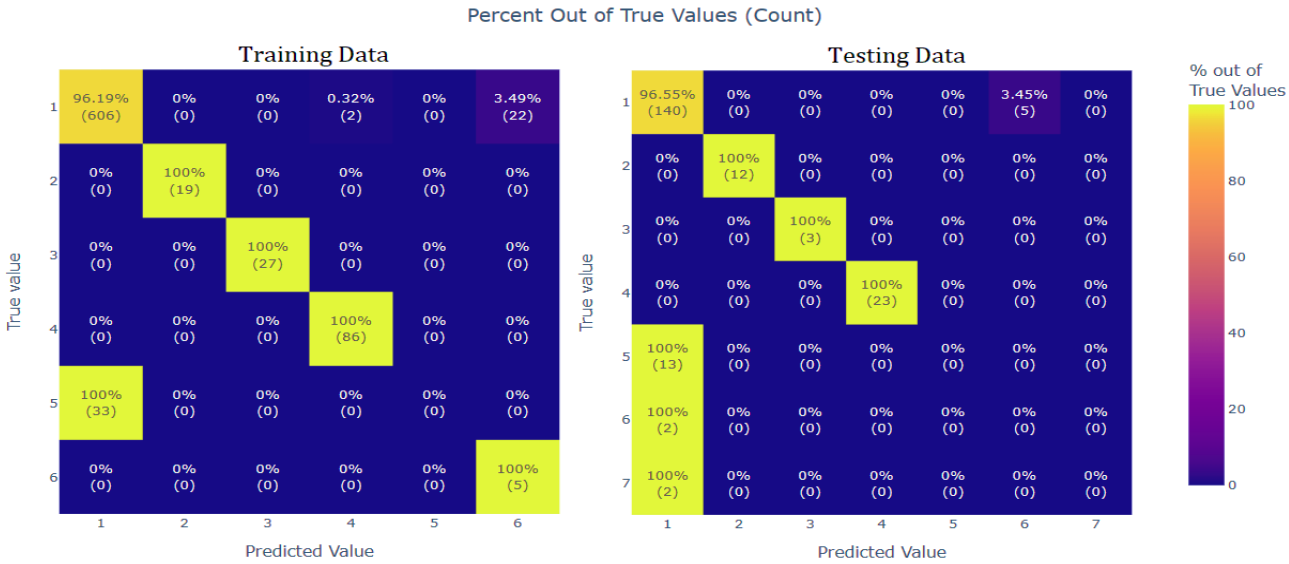


Fig. 34 Confusion Matrix of both training and testing data CSE-CIC-IDS2018

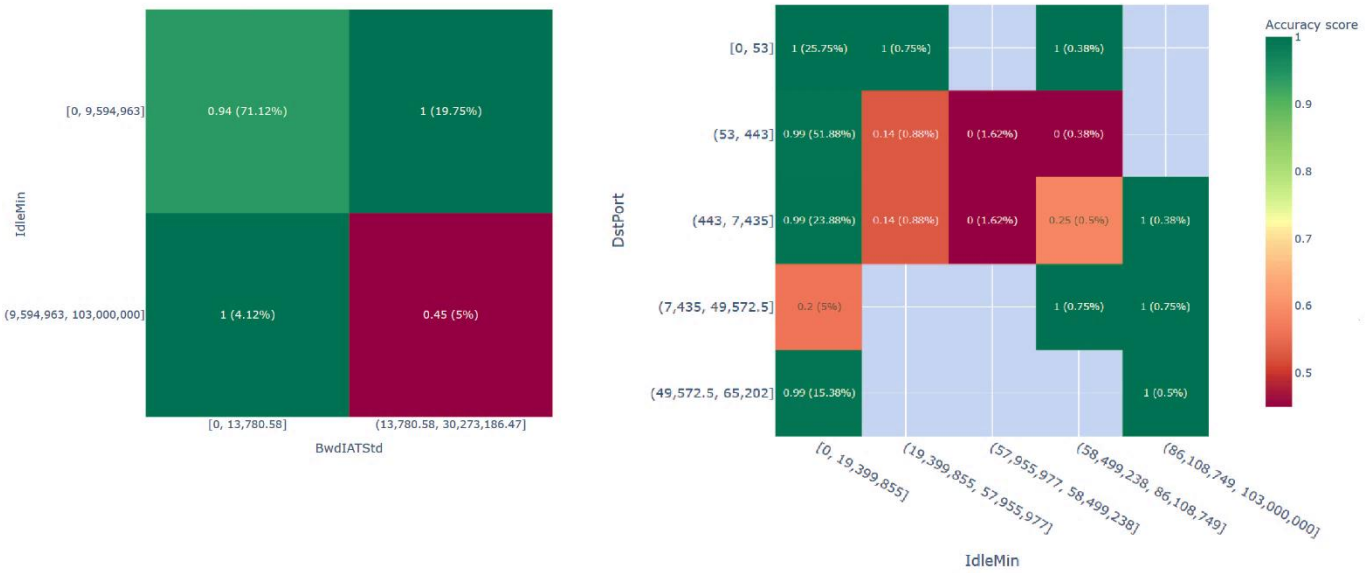


Fig. 35 Weak Section Performance CSE-CIC-IDS2018

Overfitting was tested in a gradient-boosted model due to excessive iterations [39]. The average iterative test performance is less than 5% of the best performance. The observed decrease in score of -3.78% forces the boosting model to employ no more than N estimators at a time and results in accuracy calculations for subsets of estimators plotted on both the train and test datasets. The average inference time for the training model per sample was 0.00014705 seconds, and the average inference time for the test model per sample was 0.00034344.

Brier scores are used to evaluate the reliability of probabilistic forecasts. The deviation between this estimate and the actual result is the basis for this method. The more the brier score approaches zero, the better. The Brier score is a useful metric for contrasting the precision of various prediction tools. It's also helpful in tracking one model's performance over time.

5.1. Model Performance

Multimodal using Random Forest, Decision tree, and AdaBoost algorithms on UNSW-NB15 Data is illustrated in Figures 36,37,38 [39]. Many authors have used these techniques to find anomalies in intrusion data. These algorithms were combined after a deep cleaning of raw data. The limitations of existing IDS models were identified by analysing their research work and found the essential criteria such as:

- The systems may not be able to accurately identify an anomaly if the pattern of activity is not well-defined or constantly changing.
- The systems may be unable to keep up with the rate of new attacks [40] or changes in the network.
- The systems may not be able to identify malicious activity that is not out of the ordinary for the network.
- The systems may be unable to identify all attacks, particularly those that are low-volume or targeted.

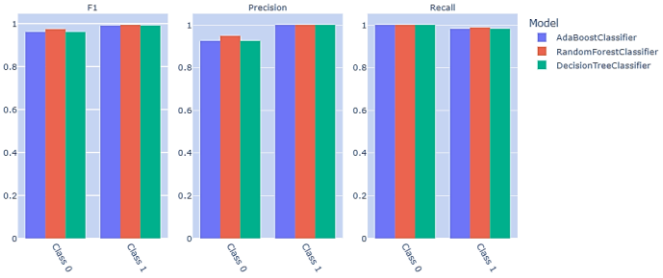


Fig. 36 Multi-model Performance (NSL-KDD)

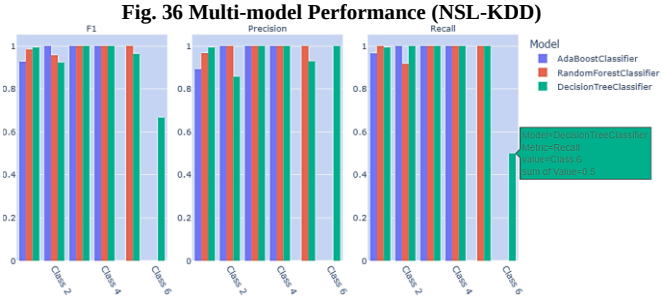


Fig. 37 Multi-model Performance (CIC-IDS2018)

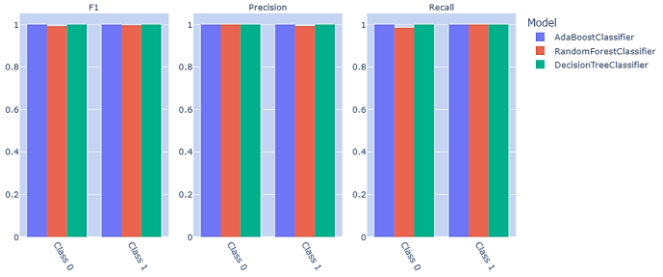


Fig. 38 Multi-model Performance (UNSW-NB15)

Table 6. Performance

	Model	Accuracy	Metric	Class	No of samples
0	AdaBoost	0.92	F1	1	145
1	AdaBoost	1	F1	2	12
2	AdaBoost	1	F1	3	3
3	AdaBoost	1	F1	4	23
4	AdaBoost	0	F1	5	13
...	...	...	...	...	...
58	DT	1	Recall	3	3
59	DT	1	Recall	4	23
60	DT	1	Recall	5	13
61	DT	0.5	Recall	6	2
62	DT	0	Recall	7	2

Table 7. Ensemble model Performance for CIC-2018 data

	Model	Accuracy	Metric	Class	No of samples
0	AdaBoost	.96	F1	0	37
1	AdaBoost	.99	F1	1	163
2	AdaBoost	.92	Precision	0	37
3	AdaBoost	1	Precision	1	163
4	AdaBoost	1	Recall	0	37
5	AdaBoost	.98	Recall	1	163
6	RF	.97	F1	0	37
7	RF	.99	F1	1	163
8	RF	.94	Precision	0	37
9	RF	1	Precision	1	163
10	RF	1	Recall	0	37
11	RF	.98	Recall	1	163
12	DT	.96	F1	0	37
13	DT	.99	F1	1	163
14	DT	.92	Precision	0	37
15	DT	1	Precision	1	163
16	DT	1	Recall	0	37
17	DT	.98	Recall	1	163

Table 8. Performance of Multimodal (NSL-KDD)

	Model	Accuracy	Metric	Class	No of samples
0	AdaBoost	1	F1	0	65
1	AdaBoost	1	F1	1	136
2	AdaBoost	1	Precision	0	65
3	AdaBoost	1	Precision	1	136
4	AdaBoost	1	Recall	0	65
5	AdaBoost	1	Recall	1	136
6	RF	.99	F1	0	65
7	RF	.99	F1	1	136
8	RF	1	Precision	0	65
9	RF	.99	Precision	1	136
10	RF	.98	Recall	0	65
11	RF	1	Recall	1	136
12	DT	1	F1	0	65
13	DT	1	F1	1	136
14	DT	1	Precision	0	65
15	DT	1	Precision	1	136

Several evaluation metrics can be used to assess the performance of an intrusion detection system [41]. The most common metric is the false positive rate (FPR). It is the percentage of times that the system incorrectly identifies an attack. A low FPR is desirable, as it means that the system is not generating a lot of false alarms. Another metric that can be used is the false negative rate (FNR). It is the percentage of times the system fails to detect an attack.

A low FNR is desirable, as the system is not missing any attacks. The accuracy of the system can also be measured. It is the percentage of times that the system correctly identifies an attack or correctly identifies a regular event. High accuracy is desirable. Other metrics can be used to assess the performance of an intrusion detection system, but these are some of the most common.

Model performance is shown in table 6. The ensemble model Performance for CIC-2018 data is shown in table 7, and the performance of Multimodal (NSL-KDD) has shown in table 8.

6. Conclusion

This paper discusses EDA's importance in the Data Science pipeline and how to do a good analysis. Incorrect or inadequate analysis can be misleading and impact machine learning model performance. Data from the NSL KDD, UNSW-NB15, and CSE-CIC-IDS-2018 datasets were analysed using machine learning methods for exploratory data analysis on intrusion detection presented in this research.

A brief overview of IDS and the different types of attacks that IDS can prevent was discussed in this work. Next, data preprocessing was explained to select features sing dimensionality reduction methods. Then this study compared the accuracy of several machine learning methods after applying them to all three datasets: AdaBoost, DT, KNN, NB, RF, and XGBoost.

While the area under the curve (AUC) improved after feature selection, several issues remained, including a low f1 score and inconsistent test recall. Different values were assigned to each class to address the data set imbalance and obtain satisfactory results.

Performance was Excellent from the feature-engineered model (tried random forest). In contrast to the DT and the RF, this is the most accurate value obtained, but it has lower scores for f1 and recall.

In the future, it may be possible to use hybrid machine learning approaches, such as combining all three techniques and other feature selection algorithms, to enhance the models' accuracy and precision.

References

[1] M. Souhail et al., "Network Based Intrusion Detection Using the UNSW-NB15 Dataset," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 477–487, 2019. *Crossref*, https://doi.org/10.12785/IJCDS/080505

[2] K. Cup, "Dataset," 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3] A. U.-N. Dataset, ADFA-NB15-Datasets/bot iot.php, "Next-Generation Network Intrusion Detection System (NG-NIDS)," 2015.

[4] A. Mahfouz et al., "Ensemble Classifiers for Network Intrusion Detection Using a Novel Network Attack Dataset," *Future Internet*, vol. 12, no. 11, pp. 180, 2020. *Crossref*, https://doi.org/10.3390/fi12110180

[5] M. Rocklin, "Dask: Parallel Computation with Blocked Algorithms and Task Scheduling," *Proceedings of the 14th Python in Science Conference*, pp. 126-132, 2015. *Crossref*, https://doi.org/10.25080/Majora-7b98e3ed-013

[6] P. Mishra et al., "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2018. *Crossref*, https://doi.org/10.1109/COMST.2018.2847722

[7] V. S. Manvith, R. V. Saraswathi, and R. Vasavi, "A Performance Comparison of Machine Learning Approaches on Intrusion Detection Dataset," *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), IEEE*, 2021, pp. 782–788, 2021. *Crossref*, https://doi.org/10.1109/ICICV50876.2021.9388502

[8] Z. H. Abdaljabar, O. N. Ucan, and K. M. A. Alheeti, "An Intrusion Detection System for IoT using KNN and Decision-Tree Based Classification," *2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI), IEEE*, pp. 1–5, 2021. *Crossref*, https://doi.org/10.1109/MTICTI53925.2021.9664772

[9] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for Intrusion Detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005. *Crossref*, https://doi.org/10.1016/j.cor.2004.03.019

[10] Z. A. Othman et al., "Improvement Anomaly Intrusion Detection using Fuzzy-ART Based on K-Means based on SNC labeling," *Jurnal Teknologi Maklumat & Multimedia*, vol. 10, pp. 1–11, 2011.

[11] T. Milo, and A. Somech, "Automating Exploratory Data Analysis via Machine Learning: An Overview," *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 2617–2622, 2020. *Crossref*, https://doi.org/10.1145/3318464.3383126

[12] R. Kaushik, V. Singh, and R. Kumari, "Multi-class SVM based Network Intrusion Detection with Attribute Selection using Infinite Feature Selection Technique," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 8, pp. 2137–2153, 2021. *Crossref*, https://doi.org/10.1080/09720529.2021.2009189

[13] B. M. Serinelli, A. Collen, and N. A. Nijdam, "Training guidance with KDD CUP 1999 and NSL-KDD Data Sets of ANIDINR: Anomalybased Network Intrusion Detection System," *Procedia Computer Science*, vol. 175, pp. 560–565, 2020. *Crossref*, https://doi.org/10.1016/j.procs.2020.07.080

[14] M. Tavallaee et al., "A Detailed Analysis of the KDD CUP 99 Data Set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, 2009. *Crossref*, https://doi.org/10.1109/CISDA.2009.5356528

[15] V. Kumar, A. Das, and D. Sinha, "UIDS: A Unified Intrusion Detection System for IoT Environment," *Evolutionary Intelligence*, vol.14, pp. 47-59, 2021. *Crossref*, https://doi.org/10.1007/s12065-019-00291-w

[16] Das, A.., and Pramod, "A Novel Deep Learning Model to Enhance Network Traffic Monitoring for Cybersecurity," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 1s, pp. 335-342, 2022.

[17] P.G.V. Suresh Kumar, and S. Akthar, "Execution Improvement of Intrusion Detection System Through Dimensionality Reduction for UNSW-NB$_{15}$ Information," *Mobile Computing and Sustainable Informatics, Springer*, pp. 385–396, 2022. *Crossref*, https://doi.org/10.1007/978-981-16-1866-6_28

[18] T. Acharya et al., "Efficacy of Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection," *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS), IEEE*, pp. 408–413, 2021. *Crossref*, https://doi.org/10.1109/I2CACIS52118.2021.9495864

[19] V. Kanimozhi, and T. Jacob, "Artificial Intelligence Outflanks all Other Machine Learning Classifiers in Network Intrusion Detection System on the Realistic Cyber Dataset CSE-CIC-IDS2018 using Cloud Computing," *ICT Express*, vol. 7, no. 3, 2021. *Crossref*, https://doi.org/10.1016/j.icte.2020.12.004

[20] T. S. Riera et al., "A New Multi-Label Dataset for Web Attacks CAPEC Classification using Machine Learning Techniques," *Computers & Security*, vol. 120, 2022. *Crossref*, https://doi.org/10.1016/j.cose.2022.102788

[21] B. A. Tama et al., "An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble," *IEEE Access*, vol. 8, pp. 24 120–24 134, 2020. *Crossref*, https://doi.org/10.10.1109/ACCESS.2020.2969428

[22] M. A. Umar, C. Zhanfang, and Y. Liu, "A Hybrid Intrusion Detection with Decision Tree for Feature Selection," *Cryptography and Security*, 2020. *Crossref*, https://doi.org/10.48550/arXiv.2009.13067

[23] A. Abdollahi, and M. Fathi, "An Intrusion Detection System on Ping of Death Attacks in IoT Networks," *Wireless Personal Communications*, vol. 112, pp. 2057–2070, 2020. *Crossref*, https://doi.org/10.1007/s11277-020-07139-y

[24] P. Kumar, G. P. Gupta, and R. Tripathi, "Toward Design of an Intelligent Cyber Attack Detection System using Hybrid Feature Reduced Approach for IoT Networks," *Arabian Journal for Science and Engineering*, vol. 46, pp. 3749–3778, 2021. *Crossref*, https://doi.org/10.1007/s13369-020-05181-3

[25] K. Adhikary et al., "Evaluating the Performance of Various SVM Kernel Functions based on Basic Features Extracted from KDDCUP'99 Dataset by Random Forest Method for Detecting DDoS Attacks," *Wireless Personal Communications*, vol.123, pp. 3127–3145, 2022. *Crossref*, https://doi.org/10.1007/s11277-021-09280-8

[26] Ming Li et al., "Design and Implementation of an Anomaly Network Traffic Detection Model Integrating Temporal and Spatial Features," *Security and Communication Networks*, vol. 2021, 2021. *Crossref*, https://doi.org/10.1155/2021/7045823

[27] R. Vinayakumar et al., "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019. *Crossref*, https://doi.org/10.1109/ACCESS.2019.2895334

[28] S. Einy, C. Oz, and Y. D. Navaei, "The Anomaly-and Signaturebased IDS For Network Security Using Hybrid Inference Systems," *Mathematical Problems in Engineering*, vol. 2021, 2021. *Crossref*, https://doi.org/10.1155/2021/6639714

[29] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, "Network Intrusion Detection using Deep Learning" *A Feature Learning Approach*, Springer, 2018.

[30] Y. Alnajjar, and J. Mounsef, "Next-Generation Network Intrusion Detection System (NG-NIDS)," *2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), IEEE*, pp. 411–416, 2021, *Crossref*, https://doi.org/10.1109/TELSIKS52058.2021.9606424

[31] M. Desquilbet et al., "Adequate Statistical Modelling and Data Selection are Essential when Analysing Abundance and Diversity Trends," *Nature Ecology & Evolution*, vol. 5, pp. 592–594, 2021. *Crossref*, https://doi.org/10.1038/s41559-021-01427-x

[32] K. R. M. Fernando, and C. P. Tsokos, "Dynamically Weighted Balanced Loss: Class Imbalanced Learning and Confidence Calibration of Deep Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2940-2951, 2021. *Crossref*, https://doi.org/10.1109/TNNLS.2020.3047335

[33] J. Liu, Y. Gao, and F. Hu, "A Fast Network Intrusion Detection System using Adaptive Synthetic Oversampling and Light GBM," *Computers & Security*, vol. 106, 2021. C*rossref*, https://doi.org/10.1016/j.cose.2021.102289

[34] M. K. Hasan et al., "Missing Value Imputation Affects the Performance of Machine Learning: A Review and Analysis of the Literature (2010–2021)," *Informatics in Medicine Unlocked*, vol. 27, 2021. *Crossref*, https://doi.org/10.1016/j.imu.2021.100799

[35] D. Chou, and M. Jiang, "A Survey on Data-Driven Network Intrusion Detection," *ACM Computing Surveys* (CSUR), vol. 54, no. 9, pp. 1–36, 2021. *Crossref*, https://doi.org/10.1145/3472753

[36] Mohammad Dawood Momand, Dr Vikas Thada, and Mr. Utpal Shrivastava, "Intrusion Detection System in IoT Network," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 4, pp. 11-15, 2020. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V7I4P104

[37] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving Adaboost-Based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset," *Journal of Physics*, vol. 1192, 2019. *Crossref*, https://doi.org/10.1088/1742-6596/1192/1/012018

[38] J. Feng, L. Yu, and R. Ma, "AGCN-T: A Traffic Flow Prediction Model for Spatial-Temporal Network Dynamics," *Journal of Advanced Transportation*, vol. 2022, 2022. *Crossref*, https://doi.org/10.1155/2022/1217588

[39] D. Levi et al., "Evaluating and Calibrating Uncertainty Prediction in Regression Tasks," *Sensors*, vol. 22, no. 15, 2022. *Crossref*, https://doi.org/10.3390/s22155540

[40] Abhijit Das, Pramod, and S. Praveen Kumar "An Enhanced Optimization Model with Ensemble Autoencoder for Zero-Day Attack Detection" *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 22, 2022.

[41] Giribabu Sadineni, M. Archana, and Rama Chaithanya Tanguturi, "Optimized Detector Generation Procedure for Wireless Sensor Networks based Intrusion Detection System," *International Journal of Engineering Trends and Technology*, vol. 70, no. 6, pp. 63-72, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I6P208

[42] Abhijit Das, and Pramod, "A Novel Ensemble Model Using Learning Classifiers to Enhance Malware Detection for Cyber Security Systems," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 10, no. 1s, pp. 31-43, *Crossref*, https://doi.org/10.17762/ijritcc.v10i1s.5793