Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 14, Issue. 2 : 2023
ISSN : **1906-9685**

REVIEW OF CHALLENGES, APPROACHES, DATASETS, AND TOOLS OF NETWORK IDS

T. Ruth Supriya, Research Scholar, Department of Computer Science, Shyam University, Dausa
Dr Gireesh Kumar Dixit Associate Prof, Shyam University
Satyanarayana Department of computer science, Principal, CMR institute of technology, Telangana

***Abstract*—** Because of the Internet's rapid expansion, intrusion detection system (IDS) research is one of the cybersecurity fields that holds great promise. Because of the huge volume of information, numerous intrusion detection systems (IDS) use arrangement calculations to distinguish network traffic, yet these calculations couldn't precisely recognize assaults. Notwithstanding, information can be successfully diminished and exact assault discovery can be accomplished by utilizing layered decrease. This paper's essential undertaking is to introduce an intensive investigation of the different IDS sorts, assault recognition methods, and advantages and downsides related with each kind. The creators likewise examine the 10 highlights of enormous information and the issues it presents for NIDSs, with a specific accentuation on the sort of Network Intrusion Detection System (NIDS). Moreover, we look at different AI based NIDS approaches and assess the viability of classifiers (Binary or Multiclass) across eight datasets and layered decrease techniques for every system. A comparison between the five tools used for Big Data analysis and a few machine learning methods is given. Discussions arose from our examination of recent studies. Lastly, we will conclude this paper by outlining our results and our plans for further research.

***Keywords*—** Big Data Techniques, Classification, Network Intrusion Detection System

I. INTRODUCTION

The expression "Big Data" alludes to the volume of information that has been produced as of late because of the expansion in Web clients, which has likewise brought about the making of various types of assaults [1]. Perhaps of the greatest test confronting designers of safety the board frameworks is securing and keeping up with the protection of Large Information. This is particularly evident given the rising utilization of web organizations and the outstanding development of information produced from various sources, which builds the space accessible for malevolent assaults by gatecrashers [2, 3].

The demonstration of uncovering exercises intended to risk an asset's accessibility, classification, or honesty is known as interruption discovery [4]. The most fundamental part of network safety is the intrusion detection system (IDS), which can distinguish interruptions either previously or after an assault. James Anderson authored the expression "IDS" in the last part of the 1970s and mid 1980s [5]. The interruption recognition framework, or IDS, is a strategy that is utilized to recognize examples in which security strategies are disregarded in PC organizations. It is regularly introduced inside the organization to watch out for all inner traffic [6].

IDS has been worked on after some time by different procedures, including factual, fluffy, Markov, bio-propelled, AI, and a lot more [7]. Artificial intelligence (AI) as programmed intrusion detection systems (IDSs) empowers PCs to learn and recognize interruptions. Up till now, researchers have made numerous intrusion detection system (IDS) that can distinguish assaults in various available

settings. The area of AI is gigantic, with many purposes, for example, discourse acknowledgment, normal language handling, web search tools, gaming, discourse conclusion, design distinguishing proof, and substantially more. Contingent upon regardless of whether a marked dataset is available, this assortment of calculations that advance by experience is classified as managed, unaided, or support learning [7, 8].

When an algorithm is trained using labelled datasets, it can predict an unlabeled dataset that is comparable by identifying a function to assign instances to classes. This process is known as supervised learning. In unsupervised learning, an unlabeled dataset is used to train the algorithm, which then applies the notion of clustering or grouping similar data to uncover the hidden design of the data [9]. Support learning centers around programming specialists that should answer in a manner that boosts the all out remuneration after some time. To recognize network attacks, scientists in this field utilize two sorts of AI draws near: directed and unaided. This study centers around the two strategies.

A. Big Data

Big Data is the term used to describe vast amounts of complex data that cannot be managed by conventional methods.

Big Data is explained in a variety of ways using Vs models. Volume, Velocity, and Variety, or the 3Vs, are the terminology typically used to describe big data. Doug Laney of Gartner first coined this concept in 2001 [10]. Volume is a measure of data amount and may be a sign of Big Data resistance. Speed alludes to the quick handling of information, which might create some issues while working with large information. Assortment is an indication of information intricacy and a major information rebellion when the information has testing issues such information with divergent information configurations or information from different sources [11, 12].

Big data is characterized by Zikopoulous as far as 5Vs [13], which supplement the ongoing 3Vs of volume, velocity, and variety with value and veracity. Big Veracity, also known as veracity. Huge Veracity, otherwise called veracity, is the term used to portray the precision of the information and can likewise include issues with information quality such clamor or missing data. An incentive for Huge Information alludes to the possibility that specific information is superfluous for Large Information examination in the event that it doesn't offer a critical worth. Ten factors are likewise used to characterize enormous information. To the 5Vs, the accompanying five ascribes are added: legitimacy, inconstancy, consistency, suitability, and instability [14, 15]. These 10Vs are the attributes of large information, but they are frequently alluded to as the 10 significant obstructions confronting huge information. Figure 1 shows the qualities of 10Vs.

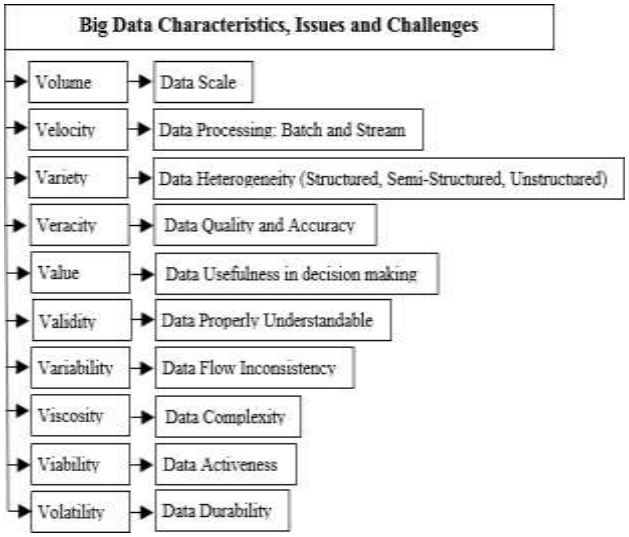

Figure 1. Features of Big Data in terms of 10Vs [16].

B. IDS types

Contingent upon where the occasions are put and the way in which they are examined, there are various types of IDS that can be characterized [5, 17]. Network-based IDS (NIDS), host-based IDS (HIDS), and hybrid or mixed IDS (MIDS) are the three types of IDS, contingent upon sending.

Be that as it may, IDS can be sorted into three structures concurring on the methodology utilized for location: half and half based IDS, inconsistency based IDS, and mark based IDS (in some cases called abuse based IDS) [18, 19]. Table 1 shows a depiction of IDS classifications relying upon area and identification techniques.

Table 1. IDS types based on detection method and location.

Classification Aspect	IDS Type	Description
Monitoring environment (placement)	HIDS	It examines any alterations made to hosts in order to identify any unlawful activity. In order to accomplish this, it keeps an eye on all network-related activities and events related to the host or device, counting logs, framework calls, record framework alterations, and approaching and active bundles to and from the host [20, 21]. It can detect intrusions by matching the operating system logs with a known pattern [19].
	NIDS	It monitors network communications for intrusions [21]. It looks just at network traffic to find unusual, unlawful, and unauthorised activity [18].
	MIDS	In a network, it integrates host-based (HIDS) and network-based (NIDS) techniques to enhance the efficacy and efficiency of cyberattack detection.
Detection method	Signature-based	is a phrase from anti-virus software that describes the process of finding certain data patterns to identify network attacks. These patterns are referred to as signatures.
	Anomaly-based	This is incompletely because of the fast improvement of malware and was implemented largely to identify unknown or zero-day attacks. Machine learning approaches compare the new behaviour to the model after training to create a model [22].
	Hybrid	Combining an anomaly-based IDS with a signature-based IDS

To give a basic idea of the benefits and drawbacks of each kind, this paper also lists the Advantages and Disadvantages of IDS types. Table 2 displays the Pros and Cons of Different IDS Types.

Table 2. Benefits and Drawbacks of Different IDS Types.

Classification Aspect	IDS Type	Advantages	Disadvantages
Monitoring environment (placement)	HIDS	It examines communications and encrypted data. It doesn't need any more hardware [23].	HIDS failure in the event that the attack causes the OS to crash. HIDS typically requires a lot of resources.
	NIDS	The operating environment is free, so NIDS won't influence have execution [5].	It makes no indication on the off chance that the strike was powerful or not. Investigation of scrambled traffic is preposterous. In this instance, it is challenging for NIDS to identify internal attacks [19].
	MIDS	more adaptable and productive. Mixed or hybrid IDSs (MIDS) capitalise on the	High overhead burden resulting from embedded techniques on the system under observation.

		advantages of the combined varieties.	
Detection method	Signature-based	The easiest and most reliable method for identifying known attacks [22]. More signatures are preferable than a single, set behavioural pattern.	This method is not effective for identifying unidentified attacks [17]. The number of zero-day attacks is rising [21]. Signatures must be updated [4].
	Anomaly-based	efficient in identifying fresh attacks. Less based on the operating system.	On the off chance that the obscure typical way of behaving has a high deception rate (FAR), it tends to be deciphered as malignant movement. Experiencing difficulty characterizing rules.
	Hybrid	It utilises the benefits of both approaches.	high asset utilization.

The result of the IDS type examination can help analysts in making reasonable sorts and designers in effectively understanding IDS types. This exposition will zero in on utilizing an oddity based intrusion detection system (IDS) in an organization interruption recognition framework.

C. Network Intrusion Detection System

These days, society is becoming more and more reliant on computers for a variety of tasks, including banking, security, and many daily activities. However, there are more and more dangers and assaults on the network. The capacity to take proactive measures to lessen or prevent attacks is examined in the cyber-security research area. The network intrusion detection system (NIDS) is situated decisively to screen all organization traffic and is responsible for investigating it to distinguish potential interruptions. To distinguish new attacks, AI strategies are utilized.

Abnormality based network interruption recognition and Mark based network interruption location are the two principal identification techniques that are normally utilized by NIDS. Cross breed approaches have additionally been proposed by various analysts; every technique discovery enjoys benefits and drawbacks.

Network level attack identification depends vigorously on anomaly-based intrusion detection systems (IDS). As far as recognizing new assaults, it performs better compared to Mark based IDS. To separate among ordinary and unusual movement, the AI model is prepared [4, 6].

D. Big Data in Network Intrusion Detection System

According to a Plain [24] study from 1994 for interference disclosure, "a client conventionally makes between 3 - 35 Megabytes of data in eight hours and it can require a couple of hours to separate a single hour of data." The review zeroed in on information decrease and order. That's what they added "if constant identification is wanted, separating, bunching, and include determination on the information are significant and can further develop recognition precision." This model shows that interruption discovery confronted large information gives well before the expression "big data" was utilized. The troubles and costs that Large Information presents for interruption discovery can be decreased with the utilization of big data approaches [25].

Owing to the intricacy of network data, big data approaches play a critical role in the analysis of network patterns and the discovery of network events. Furthermore, high dimensionality presents significant challenges for network data [11, 26]. Issues like high dimensionality and high traffic volumes should be handled by NIDS [27].

E. E. Issues with the Network Intrusion Detection System In spite of the improvement of various strategies, there are as yet various issues with NIDS that should be settled. Among these issues are:

1) Huge amount of data

For the NIDS to have the option to get familiar with the way of behaving of novel assaults, it must have minimal computing complexity during both testing and training [28].
Researchers employ two techniques to address this problem:
a) As opposed to using the whole preparation dataset, dynamic learning procedures can be utilized to find relevant information tests for preparing [27].
b) This issue can be resolved by utilising big data platforms [2, 29].

2) High false alarms

An important consideration for anomaly-based IDS is its false alarm rate [30]. A high false positive rate (FPR) in most NIDS can have disastrous effects for the network. An attacker can quickly take advantage of network weaknesses if the classifier produces more false positives. Even if the packet appears normal, an alarm is triggered in the event of false negatives. For the network managers, this results in a loss of time and effort [29].

Machine learning and probabilistic data mining approaches are needed to overcome this issue [31].

3) Imbalance Data

If the distribution of the classification class is not uniform, the dataset is unbalanced [31].

One technique for addressing inconsistent data is to apply a weighted extreme learning machine (ELM) to further develop execution. Another approach incorporates resampling the readiness dataset to address for class anomaly preceding learning [29].

This exposition's excess segments are coordinated as follows: In Section I, the ideas of Enormous Information, Interruption Recognition Framework types, NIDS, Large Information in NIDS, and issues in NID were introduced. Section II gives applicable work; Section III investigations related work; Section IV gives apparatuses and systems to Enormous Information; Section V presents conversations and suggestions; and Section VI finishes up future work.

II. RELATED WORK

Various methodologies have been proposed to resolve the issue of upgrading NIDS adequacy through AI techniques. In any case, there is a great deal of marginal exploration on large information accessible. In order to analyse and store data in NIDS and shorten calculation and training times, many researchers plan to employ Big Data tools and methodologies.

This paper presents a summary of several studies that employed machine learning algorithms to solve the classification challenges in NIDS utilising big data or conventional techniques.

The significant work is summed up in Table 3, which is organized by distribution year from most current to most established. Next, datasets are utilised to assess NIDS performance in chronological sequence. Additionally, algorithms for feature selection and classification, binary or multiclass problems, and performance indicators are displayed.

Table 3. Summary of Related work.

Study	Year	Classifier Algorithm	Feature selection	Dataset used	Classification problem	Tools	Performance metrics
[32]	2020	Support Vector Machine (SVM)	Feature selection used	KDD99	Multi	NS-3 simulation	Accuracy: 99
[33]	2020	DEGSA-HKELM	kernel principal component analysis (KPCA)	KDD99	Multi	Not available	Accuracy: 99.00 Training time: 13.204581 s Testing time: 0.012569
				UNSW-NB 15			Accuracy: 89.01 Training time: 43.306235 Testing time: 2.567050
[34]	2020	SVM	Intelligent Water Drop (IWD)	KDD99	Multi	Not available	Accuracy: 95.2 Detection rate: 95.1 Precision: 95.3
[35]	2020	SVM-based neighbor classification	Elman neural network	KDD99	Binary	Not available	Detection rate: 87.3 False alarm rate: 87.3
[36]	2020	Kernel Extreme Learning Machine (KELM)	Genetic Algorithms (GA)	KDD99	Multi	Not available	Detection rate: 97.88
				NSL-KDD			Detection rate: 94.01
[37]	2020	MeanShift	All feature used	KDD99	Multi	Python	Accuracy: 81.2 Detection rate: 79.1
[38]	2020	Weighted k-Nearest Neighbour WK-NN	Hyperbolic tangent function	Kyoto 2006+	Binary	Not available	Accuracy: (99.5%)
[39]	2020	RBF SVM	Information Gain Ratio	NSL-KDD	Binary	Not available	Accuracy: 96.24 Computation time: 4.90
[40]	2020	C5	Information Gain	UNSW-NB 15	Multi	IOT environment	Accuracy: 89.86 Detection rate: 99.32 False alarm rate: 0.72
[41]	2020	Fast kNN (FkNN)	variance function	CICIDS 2017	Binary	Java	Accuracy: 99.8 Precision: 99.91 Recall: 99.93 Computational time: 1,784
[42]	2020	AdaBoost	All features	CSE-CIC-IDS2018	Multi	Python	Accuracy: 95.49
[43]	2019	Random forest	All feature	KDD99	Multi	Weka	Accuracy:93.775 True positive rate TPR):93.8 False positive rate (FPR):0.1 Precision:99.1 ROC area:99.6

[44]	2019	Network Anomaly Detection Algorithm (NADA)	Relief-F	KDD99	Binary	MATLAB	Detection rate: 94.66 Accuracy: 97.02
							False alarm rate: 00.47 F-score: 83.31 MCC: 82.42
				Kyoto 2006+			Detection rate: 90.10 Accuracy: 98.22 False alarm rate: 01.13 F-score: 91.24 MCC: 90.26
[45]	2019	SVM	Principal component neural network (PCNN)	KDD99	Multi	Not available	Correct Rate: 97.42 False Alarm Rate: 1.48 Average Recognition Time (ms): 0.38
[46]	2019	k-means and Random Forest	Correlation coefficient	KDD99	Binary	Not available	Accuracy: 99.97 True Positives 1.000 False Positives 0.000 F-Measure 0.999 Training time: 235.52s Predict time: 4.29e-5
[47]	2019	SVM	kernel functions	NSL-KDD	Multi	MATLAB	Accuracy: 98.7 Error: 1.3
[48]	2019	Artificial Neural Network (ANN)	Correlation based	NSL-KDD	Binary	Weka	Detection Rate: 94.02
[49]	2019	SVM	Hyper Clique—Improved Binary Gravitational Search Algorithm (HC-IBGSA)	NSL-KDD	Binary	Python	Accuracy 98.85 Detection rate 98.72 False alarm rate 1.27
				UNSW-NB 15			Accuracy 94.11 Detection rate 98.47 False alarm rate 2.18
[50]	2019	distributed online averaged one dependence estimator (DOAODE)	Averaged One Dependence Estimator (AODE)	UNSW-NB 15	Multi	Not available	Accuracy: 83 Training time: less than 10 seconds.
[51]	2019	K-Nearest Neighbors (KNN)	Feature selection	CICIDS 2017	Multi	Not available	Precision: 99.53 Recall: 99.55 F1-Score: 99.50 Accuracy: 99.55
[52]	2019	Artificial Neural Networks.	All features	CSE-CIC-IDS2018	Binary	Anaconda	Training Accuracy: 0.99 Testing Accuracy: 0.99
[53]	2018	SVMwithSGD	Chi-Square	KDD99	Binary	Apache spark	AUROC: 99.55 AUPR: 96.24 Training time: 10.79 s Predict time: 1.21 s

[54]	2018	Decision Tree|	All feature	KDD99	Multi	Anaconda Fog computing	Calculation time: 0.655
[55]	2018	Logistic regression	All feature	KDD99	Multi	Apache spark	Accuracy: 99.1 Precision: 98.9 Recall: 99.5 F-Measure: 99.2 Prediction time (h): 0.089
[56]	2018	k-Means	All feature	KDD99	Binary	Apache Spark	Time computation.
[57]	2018	Kmeans++	PCA	KDD99	Binary	Anaconda	Time complexity
[58]	2018	Random forest	Information Gain (IG) (filter method)	NSL-KDD	Binary	MATLAB	FP: 0.001 TP: 0.993 Accuracy: 99.33 Precision: 0.993
[59]	2018	SVM	Multi-Linear Dimensionality Reduction (ML-DR)	NSL-KDD	Multi	MATLAB	Accuracy: 98.44 False Alarm Rate: 0.112
[60]	2018	k- nearest neighbor (k-NN)	Information Gain Ratio (IGR)	NSL-KDD	Binary	Weka	Accuracy: 99.07
[25]	2018	Random forest	All feature	UNSW-NB 15	Binary	Apache spark	Accuracy: 97.49 Sensitivity: 93.53 Specificity: 97.75 Training Time: 5.69
							Prediction Time: 0.08
[61]	2018	Random Tree	Linear Discriminant Analysis (LDA) (filter method)	UNSW-NB 15	Binary	Apache spark	Accuracy: 93.56 FPR: 0.025 Precision: 0.861 Recall: 0.865 ROC Area: 0.974 Training Time: 2.55
[62]	2018	Restricted Boltzmann Machine (RBM) Contrastive Divergence (CD)	Feature extraction	ISCX 2012	Binary	Weka	Accuracy: 88.6 True positive rate: 88.4 True negative rate: 88.8
		Persistent Contrastive Divergence (PCD)					Accuracy: 89 True positive rate: 84.2 True negative rate: 93.8
[63]	2018	K-Nearest Neighbors	Feature selection	CIDDS-001	Multi	Weka	Average accuracy: 99

[64]	2018	Random forest	decision tree	CICIDS 2017	Multi	Apache Spark	Precision: 96.4 Recall: 96.9 F1: 96.6 Build time (s): 0.03 Detect Time (s): 0.01
[65]	2017	Multi-Class SVM	Information Gain Feature Selection (IGFS)	KDD99	Multi	Not available	Accuracy:90.59 Calculation time: 52.25
[66]	2017	SVM	PCA	KDD99	Multi	Apache Spark	Accuracy: 92.48 Computation time.
[67]	2017	SVM	All feature	KDD99	Multi	Apache Storm	Accuracy: 98.03 Detection rate: 92. 60
[68]	2017	Random forest	Decision tree	NSL-KDD	Multi	Python	Accuracy: 94
[69]	2017	SVM	chi-square	NSL-KDD	Multi	MATLAB	Accuracy: 98 FAR: 0.13

Table 3 gives a rundown of the significant work. Both managed and solo learning procedures were applied in the advancement of anomaly-based intrusion detection systems (IDS) using AI calculations. The strategies that analysts have utilized in the important work will be analyzed in the accompanying segments.

III. ANALYSIS OF RELATED WORK

Since the late 1980s, numerous researchers have proposed various methods and strategies to increase the effectiveness of NIDS. They offered a variety of methods and strategies, which were compiled in the relevant paper. The general methodology that the analysts use in their connected work to find attacks is portrayed in Figure 2.

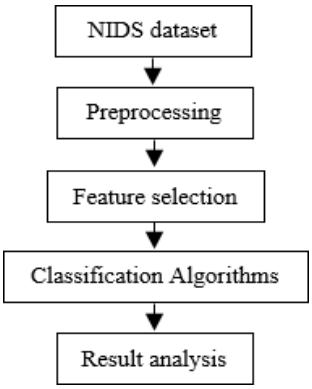

Figure 2. NIDS general methodology.

The majority of researchers have increased NIDS efficiency by doing similar actions. These actions are:

1. Choose the dataset: The NIDS dataset is essential for testing any NIDS technique since it enables researchers to assess how well the suggested model can identify incursion activity.
2. Preprocessing: In order for machine learning methods to work with the dataset, it needs to be processed.
3. Feature extraction and selection: To cut down on processing time and boost NIDS accuracy, the optimal set of features can be selected from all features using a variety of methods.
4. Classification algorithms: these identify whether an intrusion is an anomaly or a usual behaviour.
5. Lastly, outcomes are estimated using performance measures. The outline of the datasets utilized in the connected work, the processing of the datasets, the dimensionality reduction techniques, the classification issues in NIDS, and the performance metrics employed by the researchers to assess the

suggested model are the main topics of the ensuing subsections.

A. NIDS dataset

A number of publicly available datasets are used to evaluate the suggested models. Commercial items that are not easily accessible because of privacy concerns are analysed using datasets [70]. Nonetheless, public datasets like ADFA-LD, UNSW-NB15, Kyoto 2006+, KDD99, and the NSL-KDD are accessible [71, 72]. Eight publicly available datasets were employed in the linked work by the researchers: KDD99, NSL-KDD, KYOTO 2006+, ISCX2012, UNSW-NB 15, CICIDS2017, CIDDS-001, and CSE-CIC-IDS2018.

Table 3 shows how machine learning techniques were used to enhance NIDS. It is evident that the KDD99 dataset, NSL-KDD dataset, and UNSW-NB 15 dataset are the three that were used in the majority of studies (Table 3). To improve NIDS, alternative datasets can be employed. Table 4 displays these datasets in order of publication year, from earliest to most recent, and compares them according to various attributes and content.

Table 4. Comparing NIDS benchmark datasets.

Dataset	Year	Moder n attacks	Duration of data collected	Numb r of features	Number of attacks	Publicly available
KDD	1999	No	7 weeks	41	4	Yes
NSL-KDD	2009	No	16 h and 15 h	41	4	Yes
KYOTO 2006+	2009	No	3 years	24	3	Yes
ISCX2012	2012	Yes	7 days	14	7	Yes
UNSW-NB 15	2015	Yes	16 h and 15 h	49	9	Yes
CIDDS- 001	2017	Yes	4 weeks	14	5	Yes
CICIDS-2017	2017	Yes	5 days	86	14	Yes
CSE-CIC-IDS2018	2018	Yes	10 days	80	7	Yes

B. Preprocessing

To further develop the AI technique for the example grouping in NIDS, preprocessing is fundamental for the dataset. The enormous volume, overt repetitiveness, and assortment of information in this dataset give huge obstacles to information demonstrating and information revelation, prompting overfitting of the outcomes. At the point when a model is overfitted, it performs well on preparing information however inadequately on test information. In view of these elements of the information, preprocessing the information was expected prior to utilizing it to make the NIDS model [73, 74]. Coming up next are the preprocessing steps:

1) Transformation

The model's input data may include a variety of values (binary, numeric, and symbolic). Both numerical and non-numeric properties can be found in the NIDS databases. Since the training and testing inputs should be numerical, non-numerical features must be transformed into numerical features. [75, 76].

2) Standardization

Standardising datasets is crucial for machine learning techniques that employ Euclidean distance. Features that fall within a wider range of values could have been prioritised more if they are not standardised. Machine learning algorithms will suffer from features of varying sizes since not all features will be represented in the same measurement range. By standardising data and bringing it into the same range, this can be prevented [75, 77].

C. Dimensionality Reduction

Numerous widely-used feature selection/extraction algorithms and dimensional reduction approaches are available for high-dimensional data [77, 78]. All of these techniques seek to eliminate superfluous and pointless characteristics in order to improve the accuracy of classifying new instances and decrease complexity times by reducing the amount of features in the dataset [79]. The computing cost grows with the dimension, typically in an exponential fashion. There are two methods that are frequently employed:

1) Feature Extraction

By consolidating existing elements, highlight extraction lessens the dimensionality of the picked highlights from the first highlights by practical planning. This brings down highlight estimation costs, supports classifier viability, and upgrades arrangement exactness [80].

Many element extraction strategies, including Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA), have been used to lessen the dimensionality of information [81, 82].

2) Feature Selection

Since dimensionality decrease strategies attempt to pick a little subset of related highlights from the first elements by erasing excess, unessential, or loud data, include choice techniques are regularly used [83]. The essential qualification between highlight extraction and determination strategies is that the previous is utilized to diminish dimensionality by combining prior highlights, while the last option is utilized to get a subset of the most related highlights without copying them [84].

In machine learning, feature selection has the following benefits [77, 85]:

1. Lower feature space's dimensionality.
2. Accelerate an algorithm for learning.
3. Boost a classification algorithm's predicted accuracy.
4. Make the learning outcomes easier to understand.
5. Enhancing performance to increase the accuracy of predictions.
6. Eliminates data that is noisy, unnecessary, or duplicated.
7. Raising the calibre of the data.

Table 5. Distinction among Channel and Covering strategies.

Filter	Wrapper
The meaning of highlights are estimated by their affiliation with a reliant variable.	The best subset of the highlights is estimated by a truly preparing model on it.
Quicker.	Very expensive numerically.
Measurable ways are applied to assess a subset of highlights.	Utilized cross approval.
It could neglect to see as the best subset of elements.	It can constantly supply the best subset of elements.

Two techniques for selecting and reducing features are:

a) Filter Method

Using a features evaluator and a ranker approach, all features in the dataset are ranked in this method as opposed to just a subset of selected features [86].

Filter approaches typically follow a two-step approach, with feature selection occurring prior to classification and clustering tasks. First, each feature is given a rating based on a set of criteria. The final subset, which serves as the classifiers' input, may be a subset of the features that were chosen in the previous stage. One by one, features with lower ranks are eliminated in order to evaluate the classifier's accuracy at that particular moment [78, 87].

Numerous filter techniques, including relief, F-statistic, mRMR, and information gain, have been applied. The filter feature selection process is depicted in Figure 3.

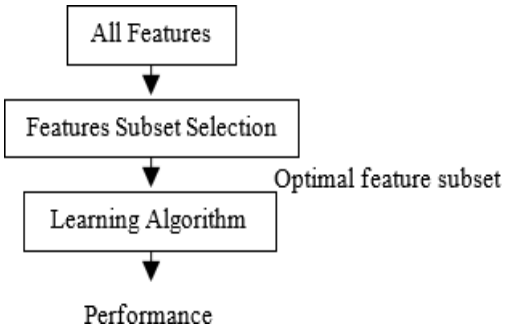

Figure 6. Method of Filtering Features Selection [87].

b) Wrapper Method

This approach uses a subset evaluator to assist with creating valuable subsets. To sort out which subset of features performs best with the request procedure, vacillated classifiers are created using a portrayal estimation and components of each and every subset [88]. The covering feature assurance approach is depicted in Figure 4.

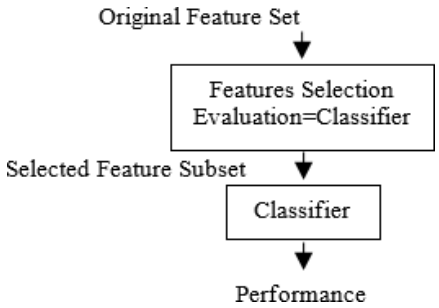

Figure 4: Wrapper Feature Selection Method [87].

Researchers have utilised both of the feature selection technologies approaches extensively; each approach has advantages and disadvantages. Table 6 lists the key distinctions between the filter and wrapper approaches for a few features.

D. Classification

One technique for managed learning is grouping [89]. Machine learning presents characterization issues in various spaces, for example, medication to distinguish a patient's disease or temperature to demonstrate a low, medium, or high temperature. Two of the main pressing concerns with NIDS are paired and multiclassification. The differentiation between the Twofold and Multi order issues in NIDS will be underlined in the resulting subsections.

1) Binary Classification

Distinguishing an interruption as either a typical or an assault on an example dataset is many times considered a paired grouping challenge [90]. Building a gaining model from a named preparing dataset is the point of regulated AI order draws near, which empower the classification of new models with obscure marks [91]. Various powerful techniques have been proposed to resolve the issue in the paired grouping situation.

Table 3 shows that machine learning approaches and interruption location research as often as possible utilize the Support Vector Machine algorithm to sort interruptions as typical or strange.

2) Multi Classification

Different names for interruption incorporate multiclassification issues, grouping an example dataset as run of the mill, or designated assaults [90]. strategies for AI have been proposed to address the test of double arrangement, and certain methods have been extended to address multiclass characterization issues [92].

Multiclass issues can be tackled in various ways. Strategies that can be extended from the Double state are remembered for the principal approach. Strategies for isolating a multiclassification issue into numerous twofold grouping issues are shrouded in the second [65]. Various leveled grouping strategies are depicted in the third strategy. In any case, a great deal of studies have utilized the half and half methodology. The most frequently involved strategies for settling multiclass issues are:

a) Extensible method

Some algorithms automatically expand their binary classification strategy to handle the multiclassification problem. Neural networks, decision trees, random forests, K-Nearest Neighbourhood, and Naive Bayes are a few of these [93].

b) Decomposing into binary classification

The most popular technique for multiclassification is breaking down into a binary classification. The process involves breaking down the issue into several two-class classification issues, which are then solved with the use of effective binary classifiers [93, 94]. The most often used techniques for breaking down a multiclassification problem into a binary classification are:

1. One-versus-all (OVA)

The OVA is the most direct strategy for separating the grouping issue into K paired issues, every one of what isolates a specific class from the excess K−1 classes.

2. All-versus-all (AVA)

The Parallel classifier in the AVA strategy is prepared to separate between each sets of classes while eliminating the excess classes. Each class is appeared differently in relation to each other.

3. Error-Correcting Output-Coding (ECOC)

To address multiclass issues, the Error Correcting Output Coding (ECOC) strategy utilizes paired (two-class) classifiers. The manner in which it works is by separating the K class order issue into a great deal of more modest two-class characterization issues. The ECOC technique doles out a particular code word to a class as opposed to a name.

c) Hierarchical Classification

In various leveled order, the classes are organized in a tree. The tree is made such that separates the classes at each parent hub into many groups, one for every youngster hub, etc, until there is just a single class present in each leaf hub. A fundamental classifier, regularly a Double classifier, recognizes the numerous youngster class groups at every hub of the tree.

Moreover, a huge group of exploration has involved machine learning procedures for interruption frameworks to classify assaults [89, 95].

Various machine learning calculations are utilized in NIDS examinations. Specialists likewise offered a correlation of AI approaches used in NIDS to address the Double or Multi grouping challenge. The advantages and downsides of a few calculations utilized in NIDS are displayed in Table 6 [28, 95, 96].

Table 6: An Extensive Analysis of NIDS Machine Learning Techniques.

Logistic regression (LR)	High accuracy.	just works with binary classifiers. Increase the duration of training.
Decision Tree (DT)	the capacity to work with large data collections. high precision.	It takes a lot of computing power to construct.
Bayesian Network (BN)	easy to understand and effectively computed. high precision.	It is possible that it does not have any good classifiers if prior knowledge is inaccurate. challenging to implement and expensive as well.
Genetic Algorithm (GA)	able to determine the best classification rules and choose the ideal parameters.	prone to overfitting. There is uncertainty about constant optimisation response times.
Hybrid methods	increased rate of attack detection.	high rate of false negatives.
Neural Networks (NN)	lack specialised expertise and are able to detect new or unidentified incursions.	It's possible to overfit while working out. Unsuitable for detection in real time.
Random Forest (RF)	It is lowering variance, increasing accuracy, and preventing overfitting.	There is a computational cost increase with Random Forest.
K Nearest Neighbor Artificial	Easy to put into practice. makes use of regional data. incredibly simple to implement in parallel	It is evident that the KNN takes a lot longer to train and test than other models. high storage needs.

Various markers, including deception rate, discovery rate, precision, review, F-measure, and model structure time, can be utilized to evaluate the exhibition of a framework. To survey and look at the presentation of different classifiers, execution measures are utilized. The connection between accurately sorted and inaccurately classed records is shown by the disarray grid [45].

The overall disarray network utilized in the appraisal is shown in Table 7. This is the manner by which the terms in the disarray network can be made sense of:

- Truly Positive (TP): The quantity of data accurately identified as belonging to a regular class.
- (FP): The quantity of records that were incorrectly identified as belonging to a regular class.
- (FN): The quantity of records that were incorrectly identified as belonging to the assault class.
- Number of entries accurately identified as belonging to the assault class (True Negative, TN).

Table 7. Confusion Matrix.2222

Algorithms	Advantage	Disadvantage
Support Vector Machine (SVM)	High training rate and accuracy. Ability to deal with high-dimensional data.	Limited to binary classifiers.

Confusion Matrix		Predicted value	
		Normal	Attack
Actual value	Normal	TP	FN
	Attack	FP	TN

The disarray lattice, which is utilized to survey execution, fills in as the establishment for most execution pointers. The forecast calculation's exhibition is shown by the qualities in the disarray framework. Table 8 gives a full outline of the numerous exhibition measures used to assess NIDS.

Table 8. Performance metrics for assessing NIDS.

Measure	Description
Accuracy	The entries that were accurately classified across all rows in the dataset. $$Accuray = \frac{TP + TN}{TP + TN + FN + FP}$$
Precision/detection rate /positive prediction value	percentage of correctly identified labels compared to all labels. $$p = \frac{TP}{TP + FP}$$
Recall	percentage of accurately categorised labels over all positively labelled labels. $$R = \frac{TP}{TP + FN}$$
F-measure	Harmonic mean of accuracy and memory. $$FM = 2\frac{P * R}{P + R}$$
False alarm rate	The percentage of regular data that is mistakenly identified as an attack is known as the false positive rate (FPR), often called the false alarm rate (FAR). $$FAR = \frac{FP}{FP + TN}$$

IV. TOOLS AND TECHNIQUES

Quick development in information have delivered conventional AI apparatuses lacking. With such countless conceivable outcomes accessible, choosing AI apparatuses for Huge Information may be testing [97]. Aces and drawbacks can be found in the accessible Large Information devices, a considerable lot of which have covering applications. This part gives a rundown of the AI instruments used to investigate Enormous Information, like MapReduce, Flash, Flink, Tempest, and H2O, alongside an examination of the motors that utilize them.

A. Hadoop ecosystem

Albeit numerous people befuddle the terms Hadoop and MapReduce, this isn't absolutely evident. In 2007, the Hadoop ecosystem was unveiled as an open-source solution that connected a distributed file system to MapReduce processing [98]. It has grown into an extensive network of initiatives covering all aspects of the Big Data workflow, such as gathering, storing, and processing data, among many other things.

YARN, or "Yet Another Resource Negotiator," the Hadoop distributed file system (HDFS), the MapReduce information handling motor, and Normal, an assortment of normal utilities expected by the other Hadoop modules, make up the Hadoop project itself as of right now [99]. An intensive comprehension of Hadoop stage requires looking at both the task and the general climate.

Data processing engines

Hadoop, which was significant in introducing the Enormous Information age, was made conceivable

by the MapReduce idea [100, 101]. MapReduce has begun to lose favor lately. Especially in the field of AI, because of its gradualness, costly above, and the way that a ton of AI undertakings don't effortlessly squeeze into the MapReduce structure.

Various drives have been sent off as of late with an end goal to resolve crucial issues with MapReduce. The following is a rundown of probably the most famous instruments for huge information investigation.

1) MapReduce

The group learning strategy utilized in the MapReduce way to deal with AI includes perusing the preparation dataset in its entire to make a learning model. The central concern with the bunch approach is its absence of speed and computational asset proficiency [97]. Huge scope information serious applications have been executed on item groups with incredible achievement because of MapReduce [102].

2) Spark

MapReduce fills in as the establishment for an undeniable level Apache project called Flash, which was first evolved at the College of California, Berkeley. By involving in-memory estimation, it works with iterative figuring and improves execution and asset issues [98]. Flexible Distributed Data Sets (RDD), which store information in memory and deal adaptation to non-critical failure without replication, are the fundamental reflections used in this task [103].

3) Storm

It is utilized to handle information progressively and was first intended to address weaknesses of different processors in get-together and assessing virtual entertainment takes care of. Storm was created at Twitter and BackType, two web-based entertainment examination organizations [104]. Ongoing handling is turning out to be increasingly more significant in the AI people group. Storm dependence is thusly filling in search settings.

4) Flink

Under the name Stratosphere, Flink was made at the Specialized College of Berlin. It empowers the execution of Lambda Design by offering the ability to oversee bunches and stream handling. Instead of being developed on top of MapReduce, it has its own runtime [96].

5) H2O

H2O is an open-source system that offers instruments for information pretreatment and assessment, as well as an equal motor for handling, investigation, math, and AI libraries. Furthermore, it offers a web UI, which helps analysts and experts who probably won't have a ton of programming experience with learning undertakings. It offers help for Java, R, Python, and Scala for the individuals who wish to change executions [97].

Table 9 gives an outline of the numerous significant variables to be considered while assessing these instruments, including adaptation to non-critical failure methods, effectiveness, versatility, interface language, and ease of use.

Table 9. Hadoop data processing engines [97].

Engine	Execution model	Supported language	Associated ML tools	In-memory processing	Low latency	Fault tolerance	Enterprise support
MapReduce	Batch	Java	Mahout	x	x	✓	x
Spark	Batch, Streaming	Java, Python, R, Scala	MLlib, Mahout, H,0	✓	✓	✓	✓
Fink	Batch, Streaming	Java, Scala	Flink-ML, SAMOA	✓	✓	✓	x
Storm	Streaming	Any	SAMOA	✓	✓	✓	x
H2O	Batch	Java, Python, R, Scala	H2O, Mahout, MLlib	✓	✓	✓	✓

v. DISSCUSION AND RECOMMENDATION

Various researchers use computer based intelligence approaches to their greatest advantage for a compelling model for NIDS. The makers give a layout of huge data in NIDS in this circulation. It moreover gives a far reaching diagram of the various IDS sorts, close by the benefits and impediments of every sort. The various methods that have been used to fabricate the ampleness of NIDS utilizing simulated intelligence estimations and straightforwardly accessible NIDS datasets are acquainted with assistance experts in perceiving new troubles and staying aware of exploration time to address NIDS concerns.

The journalists covered the different IDS types and gave a diagram in area I. IDS types were summed up in Table 1. Table 2 listed the advantages and disadvantages of different IDS types as indicated by the environmental elements and accessible discovery strategies. Moreover, Figure 1 gave an outline of the ten Major Information attributes. It likewise brought new hardships for NIDS.

Despite the fact that NIDS productivity has been the subject of various examinations, there are still a ton of issues and troubles. A few researchers attempt to distinguish valuable models for NIDS in segment II. Table 3 recorded the important writing in sequential request from freshest to most seasoned distributions, as well as the datasets utilized, organized from most seasoned to most up to date. The examinations from 2017 to 2020 that were the subject of related concentrate on in segment II were remembered for Table 3. The level of canvassed papers in the pertinent work across the distribution year is displayed in Figure 5.

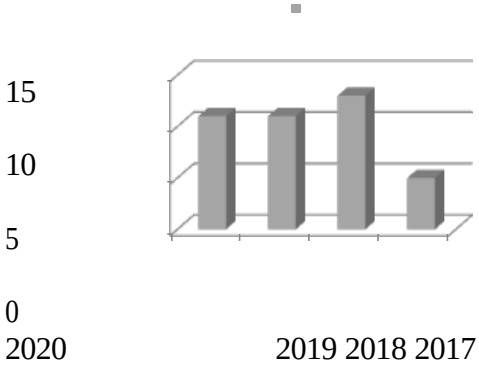

Figure 5. Relevant studies conducted after the year of publication.

Table 4 presents an examination of the datasets arranged by distribution year, from most established to latest. The creators broke down a few methodologies under eight datasets that were utilized by the scientist in the pertinent review gave in segment III. AI requires preprocessing strategies; two methods for planning datasets were illustrated.

This part introduced the element choice/extraction strategies for dimensionality decrease. The distinctions between the channel and covering approaches for a couple of elements were gathered in Table 5. Also, the authors tended to the Double and Multi order issues in NIDS and introduced various ways to deal with address Multi arrangement issues. NIDS utilizes various AI calculations, each with upsides and downsides. Table 6 showed the advantages and downsides of a couple of AI methods that have been applied to NIDS. Moreover, this part incorporated the exhibition measures.

Segment IV included the introduction of Enormous Information innovations, and Table 9 showed a correlation of Hadoop's information handling motors. Also, this segment presents our examination of the few exploration that were introduced in the connected work, alongside perspectives and suggestions.

Specialists utilized various techniques and procedures to expand the viability of NIDS, as displayed in Table 3, some of which enjoy benefits and disservices. The creators' perceptions and ideas for upgrading NIDS proficiency depend on Table 3:

- The best methods for recognizing both known and new organization level dangers are peculiarity based intrusion detection systems (IDS).
- However viable for NIDS, AI strategies are not as reasonable for taking care of a lot of organization information.
- In spite of the way that various freely available benchmark NIDS datasets exist, large numbers of them contain obsolete, incomplete, unbending, and irreproducible obstruction. Notwithstanding, these datasets are excessively old and fragmented to precisely address genuine organization attack situations.
- The making of NIDS relies upon three key components: highlights decrease, preprocessing, and classifier strategies.
- From Table 3 the creators mention the accompanying observable facts and suggestions about working on the proficiency of NIDS:
- Anomaly-based IDS identification strategies are prime in identifying network-level assaults, known and obscure assaults in networks.
- Machine learning methods are helpful in NIDS, however they have restrictions in managing Enormous Information on the organization.
- In spite of the fact that there are a few benchmark NIDS datasets openly accessible, large numbers of them contain outdated, deficient, resolute, and irreproducible interruption. Then again, these datasets are obsolete and lacking to reflect genuine organization assault situations.
- Three significant variables for improvement NIDS are preprocessing, highlights decrease, and calculations utilized for classifier.
- The creators of this paper exhort utilizing numerous datasets to survey the proposed model while assessing its exhibition for improving NIDS. This is on the grounds that obsolete datasets may not precisely reflect current assaults, and programmers are continuously developing their strategies.

VI. CONCLUSION AND FUTURE SCOPE

This paper surveyed IDS sorts as a general rule, introduced examinations among them, and talked about the advantages and disadvantages of each kind. The creators likewise gave the hardships NIDS. This paper analyzed different AI calculation based strategies used in network interruption discovery frameworks. We analyzed each model's presentation in two grouping classes (Paired and Multiclass) utilizing eight datasets and a few dimensionality decrease strategies. Moreover, an examination of a couple of AI techniques was given in this work. The creators likewise gave a presentation of the Huge Information examination advances, including MapReduce, Flash, Flink, Tempest, and H2O. The audit's discoveries will support appreciating the troubles introduced by large information in NIDS. Using Large Information approaches and ebb and flow datasets was likewise prompted by this examination. To get more exact classifiers in the Enormous Information setting, various multiclassification systems might be explored in future review.

REFERENCES

[1] D. Gaurav, J. K. P. S. Yadav, R. K. Kaliyar, and A. Goyal, "An Outline on Big Data and Big Data Analytics," pp. **74-79**.

[2] R. Devakunchari, "Analysis on big data over the years," International Journal of Scientific and Research Publications, vol. **4,** pp. **1-7, 2014.**

[3] A. Ju, Y. Guo, Z. Ye, T. Li, and J. Ma, "HeteMSD: A Big Data Analytics Framework for Targeted Cyber-Attacks Detection Using Heterogeneous Multisource Data," Security and Communication Networks, vol. **2019**, **2019.**

[4] L. Wang and R. Jones, "Big data analytics for network intrusion detection: A survey," International Journal of Networks and Communications, vol. **7**, pp. **24-31**, **2017.**

[5] S. M. Othman, N. T. Alsohybe, F. M. Ba-Alwi, and A. T. Zahary, "Survey on Intrusion Detection System Types," International Journal of Cyber-Security and Digital Forensics, vol. **7**, pp.

444-463, **2018.**
[6] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, Network Intrusion Detection Using Deep Learning: A Feature Learning Approach: Springer, **2018.**
[7] S. Gulghane, V. Shingate, S. Bondgulwar, G. Awari, and P. Sagar, "A Survey on Intrusion Detection System Using Machine Learning Algorithms," pp. **670-675.**
[8] Y. Hamid, M. Sugumaran, and L. Journaux, "Machine learning techniques for intrusion detection: a comparative analysis," pp. **1-6.**
[9] P. Dehariya, "An Artificial Immune System and Neural Network to Improve the Detection Rate in Intrusion Detection System," International Journal of Scientific Research in Network Security and Communication, vol. **4**, pp. **1-4**, **2016.**
[10] E. Guerra, J. de Lara, A. Malizia, and P. Díaz, "Supporting user-oriented analysis for multi-view domain-specific visual
[11] languages," Information and Software Technology, vol. **51**, pp. **769-784**, **2009.**
[12] P. Adluru, S. S. Datla, and X. Zhang, "Hadoop eco system for big data security and privacy," pp. **1-6.**
[13] M. Kaur and A. M. Aslam, "Big Data Analytics on IOT: Challenges, Open Research Issues and Tools," International Journal of Scientific Research in Computer Science and Engineering, vol. **6**, pp. **81-85**, **2018.**
[14] P. Zikopoulos, D. deRoos, K. Parasuraman, T. Deutsch, D. Corrigan, J. Giles, et al., "Harness the Power of Big Data—The IBM Big Data Platform. 2011," www-01. ibm. com/software/data/bigdata (letzter Zugriff am 31.03. 2018),
2011.
[15] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey," Journal of Big Data, vol. **2**, pp. **3-3**, **2015.**
[16] Z. Sun, "10 Bigs: Big data and its ten big characteristics," PNG UoT BAIS, vol. **3**, pp. **1-1**0, **2018.**
[17] N. Khan, M. Alsaqer, H. Shah, G. Badsha, A. A. Abbasi, and
S. Salehian, "The 10 Vs, issues and challenges of big data," pp.
52-56, 2018.
[18] C. Zouhair, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A Review of Intrusion Detection Systems in Cloud Computing," ed: IGI Global, , pp. **253-283, 2018.**
[19] K. Siddique, Z. Akhtar, M. A. Khan, Y.-H. Jung, and Y. Kim, "Developing an Intrusion Detection Framework for High- Speed Big Data Networks: A Comprehensive Approach," KSII Transactions on Internet & Information Systems, vol. **12, 2018.**
[20] F. A. B. H. Ali and Y. Y. Len, "Development of host based intrusion detection system for log files," pp. **281-285.**
[21] A. K. Saxena, S. Sinha, and P. Shukla, "General study of intrusion detection system and survey of agent based intrusion detection system," pp. **421-471.**
[22] M. Liu, Z. Xue, X. Xu, C. Zhong, and J. Chen, "Host-based intrusion detection system with system calls: Review and future trends," ACM Computing Surveys (CSUR), vol. 51, pp. **1-36**, **2018.**
[23] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, pp. 16-24, 2013.
[24] L. Wang, "Big Data in intrusion detection systems and intrusion prevention systems," J Comput Netw, vol. **4**, pp. **48- 55**, **2017.**
[25] J. Frank, "Artificial intelligence and intrusion detection: Current and future directions," pp. **1-12.**
[26] M. Belouch, S. El Hadaj, and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using Apache Spark," Procedia Computer Science, vol. **127**,
pp. **1-6**, **2018.**
[27] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," pp. **86-93.**
[28] R. Chapaneri and S. Shah, "A comprehensive survey of machine learning-based network

intrusion detection," ed: Springer, **2019**, pp. **345-356.**
[29] R. Patel, A. Thakkar, and A. Ganatra, "A survey and comparative analysis of data mining techniques for network intrusion detection systems," International Journal of Soft Computing and Engineering (IJSCE), vol. **2**, pp. **260-265**, **2012.**
[30] S. Suthaharan, "A single-domain, representation-learning model for big data classification of network intrusion," pp. **296-310.**
[31] P. Singh, S. Krishnamoorthy, A. Nayyar, A. K. Luhach, and A. Kaur, "Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system," International Journal of Distributed Sensor Networks, vol. **15**, **2019.**
[32] K. K. Wankhade and K. C. Jondhale, "An ensemble clustering method for intrusion detection," International Journal of Intelligent Engineering Informatics, vol. **7**, pp. **112-140**, **2019**.
[33] M. U. Farooq, H. Xiaoli, and S. A. Rauf, "Big Data Security Analysis in Network Intrusion Detection System," International Journal of Computer Applications, vol. **975**, pp. **8887-8887**, **2020.**
[34] L. Lv, W. Wang, Z. Zhang, and X. Liu, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," Knowledge-Based Systems, pp. **105648- 105648**, **2020.**
[35] N. Hariyale, M. S. Rathore, R. Prasad, and P. Saurabh, "A Hybrid Approach for Intrusion Detection System," ed: Springer, 2020, pp. **391-403.**
[36] W. Fang, X. Tan, and D. Wilbur, "Application of intrusion detection technology in network safety based on machine learning," Safety Science, vol. 124, pp. **104604-104604, 2020.**
[37] J. Ghasemi, J. Esmaily, and R. Moradinezhad, "Intrusion detection system using an optimized kernel extreme learning machine and efficient features," Sādhanā, vol. **45**, pp. **1-9**, **2020.**
[38] A. Kumar, W. Glisson, and H. Cho, "Network Attack Detection using an Unsupervised Machine Learning Algorithm."
[39] D. Protić and M. Stanković, "Detection of Anomalies in the Computer Network Behaviour," European Journal of Engineering and Formal Sciences, vol. **4**, pp. **7-13**, **2020.**
[40] S. Krishnaveni, P. Vigneshwar, S. Kishore, B. Jothi, and S. Sivamohan, "Anomaly-Based Intrusion Detection System Using Support Vector Machine," ed: Springer, **2020**, pp. **723- 731.**
[41] V. Kumar, A. K. Das, and D. Sinha, "Statistical analysis of the UNSW-NB15 dataset for intrusion detection," ed: Springer, **2020,** pp. **279-294.**
[42] K. V. Krishna, K. Swathi, and B. B. Rao, "A Novel Framework for NIDS through Fast kNN Classifier on CICIDS2017 Dataset," **2020.**
[43] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," IEEE Access, vol. **8**, pp. **32150-32162**, **2020.**
[44] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and
M. AlZubi, "Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques," **2019.**
[45] D. A. Kumar and S. R. Venugopalan, "A design of a parallel network anomaly detection algorithm based on classification," International Journal of Information Technology, pp. **1-14**, **2019.**
[46] K. Ye, "Key feature recognition algorithm of network intrusion signal based on neural network and support vector machine," Symmetry, vol. **11**, pp. **380-380**, **2019.**
[47] N. Kaja, A. Shaout, and D. Ma, "An intelligent intrusion detection system," Applied Intelligence, vol. **49**, pp. **3235- 3247, 2019.**
[48] B. S. Bhati and C. S. Rai, "Analysis of Support Vector Machine-based Intrusion Detection Techniques," Arabian Journal for Science and Engineering, pp. **1-13**, **2019.**
[49] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," pp. **643-646.**
[50] M. R. G. Raman, N. Somu, S. Jagarapu, T. Manghnani, T. Selvam, K. Krithivasan, et al., "An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm," Artificial Intelligence Review, pp. **1-32**, **2019.**
[51] M. Nawir, A. Amir, N. Yaakob, A. R. Badlishah, A. M. Safar,
M. N. M. Warip, et al., "Distributed Online Averaged One Dependence Estimator (DOAODE) Algorithm for Multi-class Classification of Network Anomaly Detection System," pp. **12015-12015.**

[52] M. Alrowaily, F. Alenezi, and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," pp. **277- 288.**
[53] V. Kanimozhi and T. P. Jacob, "Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC- IDS2018 using cloud computing," pp. **33-36.**
[54] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al- Hashida, "Intrusion detection model using machine learning algorithm on Big Data environment," Journal of Big Data, vol. **5**, pp. **34-3**4, **2018.**
[55] K. Peng, V. Leung, L. Zheng, S. Wang, C. Huang, and T. Lin, "Intrusion detection system based on decision tree over big data in fog environment," Wireless Communications and Mobile Computing, vol. **2018**, **2018.**
[56] E. M. Kurt and Y. Becerikli, "Network Intrusion Detection on Apache Spark with Machine Learning Algorithms," pp. **130- 141.**
[57] F. Karataş and S. A. Korkmaz, "Big Data: controlling fraud by using machine learning libraries on Spark," International Journal of Applied Mathematics Electronics and Computers, vol. **6**, pp. **1-5**, **2018.**
[58] K. Peng, V. C. M. Leung, and Q. Huang, "Clustering approach based on mini batch kmeans for intrusion detection system over big data," IEEE Access, vol. **6**, pp. **11897-11906**, **2018**.
[59] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly- based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. **25**, pp. **152-160**, **2018.**
[60] S. K. Biswas, "Intrusion detection using machine learning: A comparison study," International Journal of Pure and Applied Mathematics, vol. **118**, pp. **101-114**, **2018.**
[61] B. N. Kumar, M. S. V. S. B. Raju, and B. V. Vardhan, "Enhancing the performance of an intrusion detection system through multi-linear dimensionality reduction and Multi-class SVM," International Journal of Intelligent Engineering and Systems, vol. **11**, pp. **181-19**2, **2018.**
[62] P. Dahiya and D. K. Srivastava, "Network intrusion detection in big dataset using Spark," Procedia Computer Science, vol. **132**, pp. **253-262**, **2018.**
[63] T. Aldwairi, D. Perera, and M. A. Novotny, "An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection," Computer Networks, vol. **144**, pp. **111-119, 2018.**
[64] A. Verma and V. Ranga, "Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance- based machine learning," Procedia Computer Science, vol. **125**, pp. **709-716**, **2018.**
[65] H. Zhang, S. Dai, Y. Li, and W. Zhang, "Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark," pp. **1-7.**
[66] J. Maharani and Z. Rustam, "The Application of Multi-Class Support Vector Machines on Intrusion Detection System with the Feature Selection using Information Gain."
[67] H. Wang, Y. Xiao, and Y. Long, "Research of intrusion detection algorithm based on parallel SVM on spark," pp. **153- 156.**
[68] M. A. Manzoor and Y. Morgan, "Network intrusion detection system using apache storm," Probe, vol. **4107**, pp. **4166-4166**, **2017**.
[69] M. C. Belavagi and B. Muniyal, "Multi Class Machine Learning Algorithms for Intrusion Detection-A Performance Study," pp. **170-178.**
[70] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," Journal of King Saud University-Computer and Information Sciences, vol. **29**, pp. **462-472**, **2017.**
[71] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set IEEE Symp,"Comput. Intell. Secur. Def. Appl. CISDA 2009, no. Cisda, pp.

1-6, **2009**.
[72] M. K. Siddiqui and S. Naahid, "Analysis of KDD CUP 99 dataset using clustering based data mining," International Journal of Database Theory and Application, vol. **6**, pp. **23-34**, **2013.**
[73] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, **2019.**
[74] P. Kar, S. Banerjee, K. C. Mondal, G. Mahapatra, and S. Chattopadhyay, "A Hybrid Intrusion Detection System for Hierarchical Filtration of Anomalies," ed: Springer, **2019**, pp. **417-426.**
[75] C. Azad, A. K. Mehta, and V. K. Jha, "Evolutionary Decision Tree-Based Intrusion Detection System," pp. **271-282.**
[76] T. Ahmad and M. N. Aziz, "Data Preprocessing and Feature Selection for Machine Learning Intrusion Detection Systems," ICIC Express Letter, vol. **13**, pp. **93-101**, **2019.**
[77] J.-h. Woo, J.-Y. Song, and Y.-J. Choi, "Performance Enhancement of Deep Neural Network Using Feature Selection and Preprocessing for Intrusion Detection," pp. **415- 417.**
[78] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," pp**. 372-378.**
[79] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," Advances in bioinformatics, vol. **2015, 2015.**
[80] A. Subasi, Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques: Academic Press, **2019.**
[81] H. Motoda and H. Liu, "Feature selection, extraction and construction," Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol, vol. **5**,
pp**. 2-2, 2002.**
[82] A. A. Aburomman and M. B. I. Reaz, "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection," pp. **636-640.**
[83] G. Karatas, O. Demir, and O. K. Sahingoz, "Deep learning in intrusion detection systems," pp. **113-116.**
[84] J. Miao and L. Niu, "A survey on feature selection," Procedia Computer Science, vol. **91**, pp. **919-92**6, **2016.**
[85] M. Ziaye, S. Khalid, and Y. Mehmood, "Survey of Feature Selection/Extraction Methods used in Biomedical Imaging," International Journal of Computer Science and Information Security (IJCSIS), vol. **16**, **2018.**
[86] L. Ladha and T. Deepa, "Feature selection methods and algorithms," International journal on computer science and engineering, vol. **3**, pp. **1787-1797, 2011.**
[87] P. Kumbhar and M. Mali, "A survey on feature selection techniques and classification algorithms for efficient text classification," International Journal of Science and Research, vol. **5**, pp. **9-9**, **2016.**
[88] B. Sahu, S. Dehuri, and A. Jagadev, "A Study on the Relevance of Feature Selection Methods in Microarray Data," The Open Bioinformatics Journal, vol. **11**, **2018.**
[89] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine learning, vol. **46**, pp. **389-422**, **2002.**
[90] M. Abdulrazaq and A. Salih, "Combination of multi classification algorithms for intrusion detection system," Int. J. Sci. Eng. Res., vol. **6,** pp. **1364-1371**, **2015.**
[91] D. S. Kim and J. S. Park, "Network-based intrusion detection with support vector machines," pp. **747-756.**
[92] M. Praveena and V. Jaiganesh, "A literature review on supervised machine learning algorithms and boosting process," International Journal of Computer Applications, vol. **169**, pp. **32-35**, **2017.**
[93] M. Aly, "Survey on multiclass classification methods," Neural Netw, vol. **19**, pp. **1-9, 2005.**

[94] M. Topczewska, "Multiclass classification strategy based on dipoles," Zeszyty Naukowe Politechniki Białostockiej. Informatyka, pp. **79-90**, **2011.**

[95] S. A. Mulay, P. R. Devale, and G. V. Garje, "Intrusion detection system using support vector machine and decision tree," International Journal of Computer Applications, vol. 3, pp. **40-4**3, **2010.**

[96] I. A. Solomon, A. Jatain, and S. B. Bajaj, "Neural Network Based Intrusion Detection: State of the Art," Available at SSRN 3356505, **2019.**

[97] S. Ewen, S. Schelter, K. Tzoumas, D. Warneke, and V. Markl, "Iterative parallel data processing with stratosphere: an inside look," pp. **1053-1056.**

[98] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," Journal of Big Data, vol. **2**, pp. **24-24, 2015.**

[99] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," **2004.**

[100] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, et al., "Apache hadoop yarn: Yet another resource negotiator," pp. **1-16.**

[101] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. **51**, pp. **107-113, 2008.**

[102] A. K. Gupta and S. Gupta, "Security issues in big data with cloud computing," Int J Sci Res Comput Sci Eng, vol. **5**, pp. **27-32, 2017.**

[103] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," HotCloud, vol. **10,** pp. **95-95, 2010.**

[104] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, et al., "Fast and interactive analytics over Hadoop data with Spark," Usenix Login, vol. **37**, pp. **45-51, 2012.**

[105] N. Marz, "History of Apache Storm and lessons learned," Thoughts from the Red Planet, vol. **10**, **2014.**