Section: Research Paper
ISSN 2063-5346

INTRUSION DETECTION SYSTEM BASED ON SUPERVISED MACHINE LEARNING APPROACH

[1]Mr.Thamraj Narendra Ghorsad, [2]Dr. Rais Abdul Hamid Khan
[1] Phd Scholar, Computer Science & Engineering Department, G. H. Raisoni University Amravati,
[2] Professor, Computer Science & Engineering Department, G. H. Raisoni University Amravati
[1] raj.ghorsad@gmail.com, [2] rais.khan@ghru.edu.in

Abstract: The use of machine learning to identify DDOS attacks is suggested in this research. Python has been used to simulate a DDOS attack. Wireshark has been used to record the data communication, and it was exported as a comma-separated values format. The various classifiers such as K-Nearest Neighbors, MLP, Random Forest, Gaussian Naive Bayes, Xgboost, SVM model from the Python sci-kit learn package were input for the pre-processed datasets. The model did well in terms of categorization accuracy, precision, recall, F1_score, and matthew correlation coefficient.

Keywords: Machine Learning, Supervised Learning, Intrusion Detection, Distributed Denial of Service

Introduction

The various resources used by end individuals, servers, and systems are severely hampered by DDoS threats. In order to execute slave programmes on many hosts with poor security defences, other than the target, an intruder might get access them. The malware is then told to run concurrently on these hosts, attacking the attacker's systems and preventing them from using their capabilities. In addition to being freely accessible, DDoS techniques are frequently misused to carry out such assaults. The source IP addresses of data packets are faked in addition to the data originating from randomly affected machines, making it difficult to track down DDoS threats. The utilization of Machine Learning in identifying such threats is a modern solution to those problems [1]. Vulnerability assessment, domain specific packet analysis, and intrusion detection are the three subcategories of attack detection in online protection. The final one, that is of relevance in this research, is statistically based and looks for deviations in performance parameters from predicted typical behaviour, that is dependent on how the network architecture is typically used.

Machine learning has a subfield called abnormality identification. The identification of aberrant trends in learning material and the consequent inference of inferences from its foundation. The primary method for detecting anomalies is categorization as compared to analysis. Identifying a class label (Normal or Attack) for a particular data. The identification of potentially abnormal behavior in a network is made possible by building a machine learning classifier on a security intelligence database that comprises network behaviour (time, source IP, destination IP, port numbers, etc.).

This paper examines a machine learning method for enhancing intrusion detection that involves capturing data packets and sending it to a various classifiers model as inputs. Its objective is as below,

- Provide description of data security and associated security investigation study that has been performed in this field.
- To investigate existing methods for extending intrusion detection across baseline information.
- To create models that will protect the data from different kinds of attack.
- To verify the suggested models and measure the various model's effectiveness.
- To evaluate the effectiveness of the suggested methods by contrasting them with the current ones.

Section: Research Paper
ISSN 2063-5346

Related Work

It's often suggested to undergo a study main objective of previous study in order to build or carry out any good research. This might enable an in-depth study of research work and allow for the formulation of objectives based on gaps among current needs and current strategies.

In supporting of intrusion identification, Seraphim et al. (2021) concentrated on machine learning approaches for cyberspace protection. It employs the Naive Bayes classifier, Hoeffding tree classifier, and ensemble classifier, three separate techniques. The investigation is conducted using novel techniques and contrasts streaming and non-streaming environments. This study uses the well NSL KDD databases to offer the discussion on leveraging developing methodologies and obstacles. When employing the SEA generator, the idea of drift is introduced into the static streaming. The ensemble classifier is ultimately discovered to be better effective in contexts both with and without idea drifting [1].

In their study published in 2020, T. Saranya et al. analyse the comparative analysis of various ML techniques utilized in IDS for a number of areas, including cloud computing, Internet of Things (IoT), smart cities, and 5G networks. The categorization of intrusions is another goal of this effort, which will be accomplished utilising ML techniques such Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest. The effectiveness of the study was evaluated using the KDD-CUP database, and it was evaluated to the most recent study [4].

The defence of end user internet of things (IoT) equipment towards Assaults has also been examined using Artificial frameworks. In that setting, every technique that was examined had a testing sample performance of up to 99.9 percent [4], which is consistent with the findings of this research. The Deep Belief Network, a family of neural networks utilized for unsupervised learning, was one such method. It relies on learning the weights of the network's connections and then fine-tuning it appropriately to obtain the best results. A testing dataset performance of 92.84 percent was achieved using this method [5]. Comparable study was done by 2021 paper [6], which applied the identical KNN classifiers to a DDoS dataset and attained a 98.3 percent accuracy score.

Proposed Methodology

There are three forms of intrusion detection. The primary is misuse identification, which uses a signature-based method to analyse data based on a set of individually programmed pre-defined parameters. In that instance, only when data transmission is acting in contravention of such rules is it highlighted. It is clear why this is a problem: it is nearly difficult to identify unidentified assaults [3]. This also takes a lot of human effort, which is expensive in terms of both effort and expense. The stateful protocol examination is the next. It compares the reported conditions to pre-established standardized profiles offered by different companies [6], being informed of the conditions of the protocols [5]. Contrary to stateful networks, vulnerability assessment just analyzes recorded activity to a set of rules, but misusing and stateful protocol assessment appear to be identical. And hence have a thorough concept of when the protocols ought to operate and react in light of their various conditions [6]. Such comprehension does, although, arrive at a steep cost: significant workload. Abnormality identification is the final sort of recognition, and its pattern can be either variable or static. A static pattern remains unchanged once it has been created. A variable pattern picks up information when monitoring the data and adjusts it appropriately, making the system vulnerable to threats that happen throughout time because its behaviors will fit the predicted pattern.

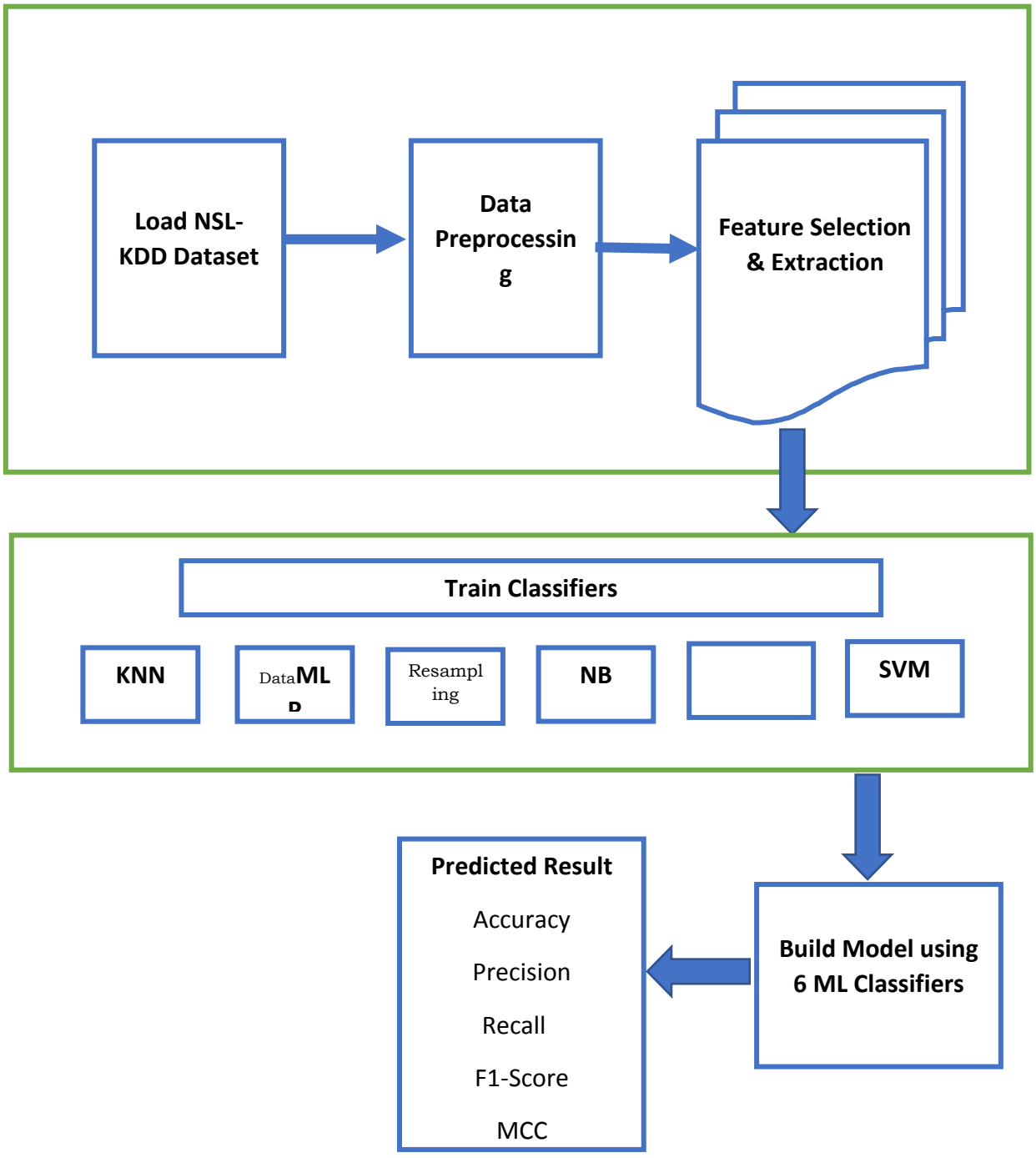


Figure 1: Proposed Machine learning model of intrusion detection system

Figure 1 shows the Architecture of Proposed System for intrusion detection. Following are subsequent steps of proposed methodology as described as follows:

Section: Research Paper
ISSN 2063-5346

Dataset Description

The publicly available dataset called as NSL-KDD is utilised in the field of intrusion detection over the internet of things environment. Total Six different machine learning algorithms KNN, MLP, RF, NB, Xgboost, and support vector machines are used to train the experimental models that examine the NSL-KDD datasets to detect the attacks.

```
        Time    Source Destination  Protocol  Length  Source port  \
0   0.000259    100215    17220101         4      97        52139
1   0.000268    100215    17220101         4      97        52139
2   0.135737  17220101      100215         4     361           53
3   0.136912  17220101      100215         4     201           53
4   1.086104    100215    17220101         4      73        52303
5   1.175460    100215    17220101         4      73        55408
6   1.175486    100215    17220101         4      73        55408
7   1.175897    100215    17220101         4      86        39476
8   1.190423  17220101      100215         4     121           53
9   1.190423  17220101      100215         4     121           53

   Destination port  Target
0                53       0
1                53       0
2             52139       0
3             52139       0
4                53       0
5                53       0
6                53       0
7                53       0
8             55408       0
9             52303       0

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 197342 entries, 0 to 197341
Data columns (total 8 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Time              197342 non-null  float64
 1   Source            197342 non-null  object
 2   Destination       197342 non-null  object
 3   Protocol          197342 non-null  int64
 4   Length            197342 non-null  int64
 5   Source port       197342 non-null  int64
 6   Destination port  197342 non-null  int64
 7   Target            197342 non-null  int64
dtypes: float64(1), int64(5), object(2)
memory usage: 12.0+ MB
None
------------------------------------------------------------------------
0    4
1    4
2    4
3    4
4    4
5    4
6    4
7    4
8    4
9    4
Name: Protocol, dtype: int64
```

Figure 2: Sample Dataset

Section: Research Paper
ISSN 2063-5346

As observed from the comparisons in Table 2 of four ML classifiers for multi-class classification, the RF classifier achieved high accuracy for Normal, DOS, Probe classes, the GB classifier achieved high accuracy for R2L, and the KNN classifier achieved high accuracy for U2R class. It finds a complete number of instances.

Table 1: Attack Classification

Attacks Taxonomy	Subclasses
DOS	Teardrop , Mailbomb , Smurf , Apache2 , Neptune , Back , Pod , Land
Probing	Saint, Portsweep, Ipsweep, Nmap, Satan
U2R	Guess_passwd, Buffer_overflow, Perl, Rootkit, loadmodule
R2L	Snmpget attack imap, Multihop, Spy, Phf, Warezmaster, Warezclient, Xlock, Ftp_ write

Table 2. Integral of Instances

Type of Attack	Training Instances	Total Testing Instances
Normal	7500	10000
Probing	1391	2579
R2L	216	2338
U2R	70	105
Dos	1807	3168
Integral	10984	18200

Feature Selection & Extraction

The Wireshark API was used to choose for features selection and extraction. In this step, choose packet number, time, source IP address, destination IP address, protocol, length, source port, and destination port as columns (according to the acquisition: will be removed). Furthermore, packet filtering is possible with Wireshark. Next will remove every packet including IPv6 addresses and every packet relating to the ARP protocol utilizing "! ARP and! IPv6" in order to correctly match six models. It will make it more difficult for models to transform all characteristics to decimals if it were n't removed. Then load all the collected dataset into Microsoft Excel after exporting it into a comma-separated values (.csv) format. Then individually add a new column called Attack that corresponds to the target attribute. Both "Normal" or "Attack" are assigned to every data. Typically, significantly fewer unusual data than normal ones are needed for anomalies identification. As a result, we kept 2.5 million packets overall after cutting the database. 2100 out of 33950 are considered abnormal, or roughly a 1:100 proportion.

Both extraction and selection of features acquired in this methodology fig. 2 shows a point wise approach. It contains information type operation, updation in lost information, information equalization and divide step.

Section: Research Paper
ISSN 2063-5346

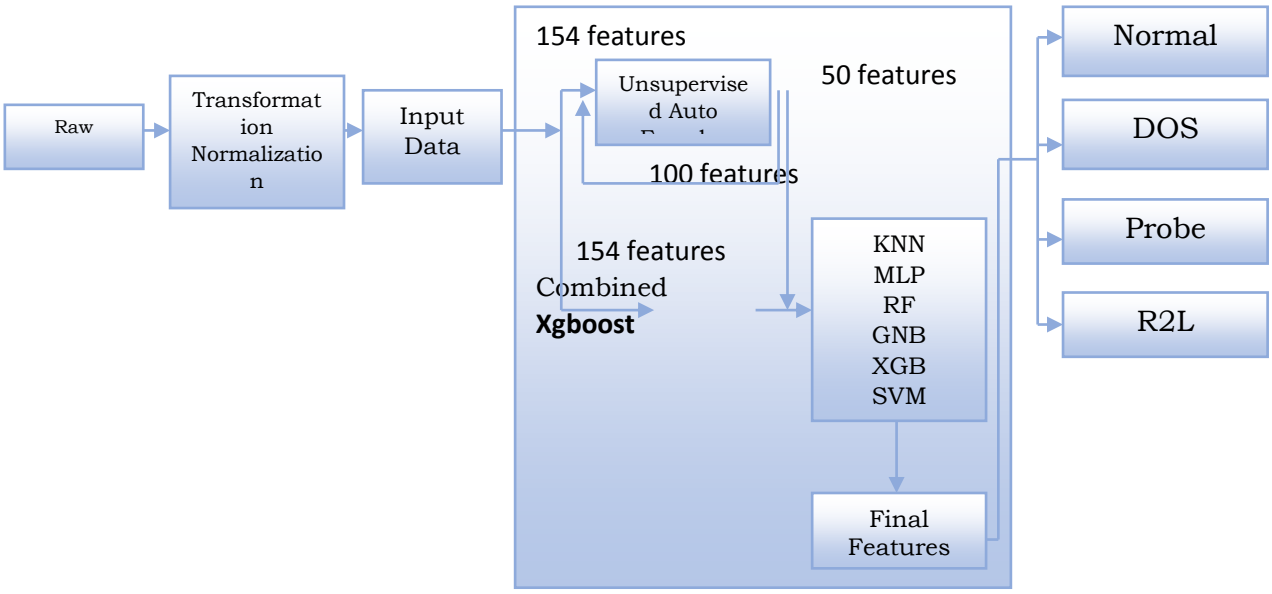


Figure 3: flow of Feature Selection & Extraction

Method 1
Representation of Method
plan of method
Purpose information gathering
get back Input Dataset
exit

Feature Extraction
for i = 1 to h do
calculate encoder function yi
calculate decoder function zi
minimize objective function Ei
obtain optimal Wi and bi
w tends to wi
exit for return end function

Feature Selection
switch AFES do
step KNN back to feature
step MLP back to feature
step RF back to feature
step GNB back to feature
step XGB back to feature
step SVM back to feature
exit function
exit procedure

Feature Extraction : Networks used to extract features and auto encoder provide propagation , input and target value same . Fig. 4 illustrates an auto encoder.
For example, Input attributes change, it will discover the relation between this algorithm. It shows actual attributes in compact format are more unresponsive to next stages.

Section: Research Paper
ISSN 2063-5346

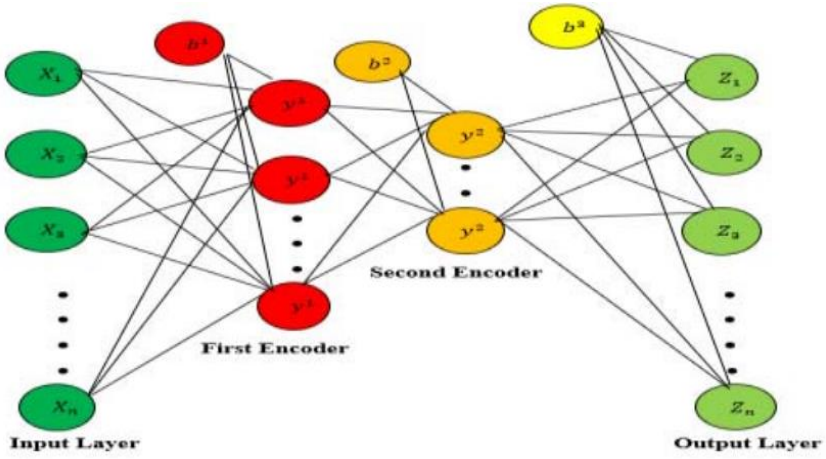


Figure 4: Autoencoder Network

Encoder decoder model extracts new feature with auto encoder. It recovers the original signal using the decoder function but it is close to the original input not exact as input. It searches for the best parameters values of attributes and bias encoding as well as decoding. It rectifies the difference between reconstruction output and input information. It uses two layers which are hidden by the encoder auto . It is also known as stack auto encoder.

Model training and testing

The machine learning model is powered by Python 3.7.12 and was configured using Jupyter Notebooks. The Scikit library was used to build the model's framework. In terms of pre-processing, let's start by removing the packet id column because it is related to the Wireshark captures and has no bearing on the ML models. To substitute a blank character ('') for every dot ('.') in source and destination IP addresses because all relevant features must be of a specific numeric kind. After that, it can allocate a value to every protocol and change the data in the Targets column to 0 or 1, which represent Normal or Attack, accordingly. The database is then input into a variety of supervised ML models, including such K-Nearest Neighbors (KNN), MLP, Random Forest, Naive Bayes, Xgboost, and Support Vector Machine, which is utilised in this case for categorization. A supervised model is one that utilises labelled data and has access to the target variables for each data size. Instead of producing a continuous result, a classifier produces a specific value that corresponds to each class (normal or Attack).

Attack

Attack are 39 types and it divided in 4 group

Probing Attack (Probe):

To collect data attacker scan network and quality of host . It search control of security and utilise using map as well as network services . Computer effected on PC features . It is a high priority attack and class of attack e.gNmap ,Portsweep , Ipsweep and satan.

Denial of Service (DoS):

It calculates storage devices which are found busy and full to store data so it causes an attack opponent . It is divided as back , land and pod , smurf , teardrop and neptune.

User to Root Attack (U2R): It access system and provide benefits on security issues. It tries to access the user account on the system (it uses passwords , attack). It takes benefits of security for accessing systems . e.g. rootkit ,perl , load module and buffer overflow.

Remote to Local Attack (R2L): It takes benefits when packets sent by an attacker on a network without security protocols but does not have a login credential machine for access to this machine. e.g. spy ,multihop , phf , ftp_write, guess_passwd, warezmaster.

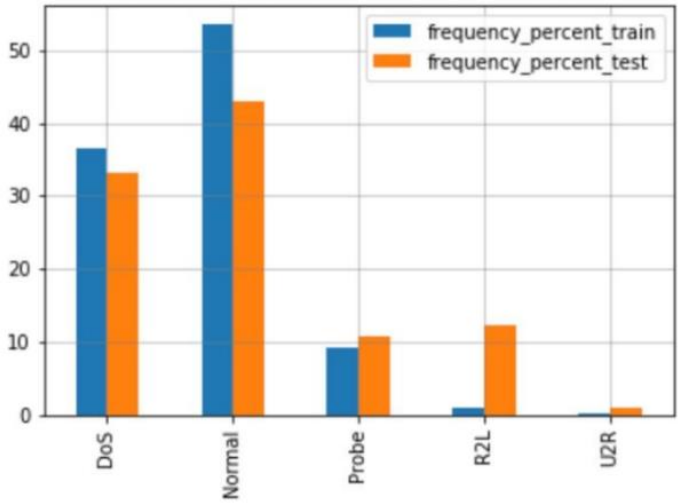


Figure 5: Training and Testing data distribution

It shows data occurrence frequency using testing information and attack distribution in practice. It also contains normal classes which show information on attacks detected by the system.

Packet Sniffing

Wireshark was initialised when the virtual machine started. A popular network protocol analysis called Wireshark enables users to record both incoming and exiting data frames from a network connection. Accessing multiple various websites, comprising stream, audio, instructional, media, and entertainment services, first served to imitate typical daily network load.

The assault was carried out after around 2.5 million packets had been recorded. 34950 packets with randomly generated faked source IP addresses were observed travelling over the networks to the targeted server after the assault had been carried out.

Classifiers used for the Model

In this study the six different machine learning classifiers for building models. The reason behind selecting this is to analyze the performance of various classifiers for attack detection to incorporate machine learning.

K Nearest Neighbor (KNN) :

It works on different domains , presiding over learning problems. Nearest node calculated classification by collecting votes of k. It's simple to construct , distort datta , and work on large training datasets [19].

Multi-layer perceptron (MLP):

Its type of ANN , which is a ML method and it takes inspiration from human brain work. The result is to map human brain properties to making decisions and collecting new data. As the human brain is made up of structured nerve cells, ANN is made up of joincells[18][1].

Random Forest (RF):

To utilize a decision tree it uses a machine learning method. with the combination of different decision tree forests created.

Section: Research Paper
ISSN 2063-5346

Naive Bayes (NB):

Its type of probabilistic classifying techniques which help probability theory and the Bayes Theorem for predictive modeling, all features are independent. That calculated probabilities with all to generate highest priority for single out the result [17][1] .

$$P(H/X) = \frac{P(X/H)\,.P(H)}{P(X)}$$

Support Vector Machine classifier [19][13]used for prediction and classification. The SVM uses a portion of the data to train the system.

XG Boost:

Gradient boost advance version called XG Boost. It uses pattern matching , recognition , and security with the help of algorithms. It solves ML problems and test models for under and overfitting in train test models . XG Boost is faster than the gradient boosting technique so it is suitable to perform big data, parallel, and Hadoop environments [21]. It allows cross-validation at each utterance of the boosting process and thus it is easy to get the precise best number of boosting iterations in a single run [2][22, 23].

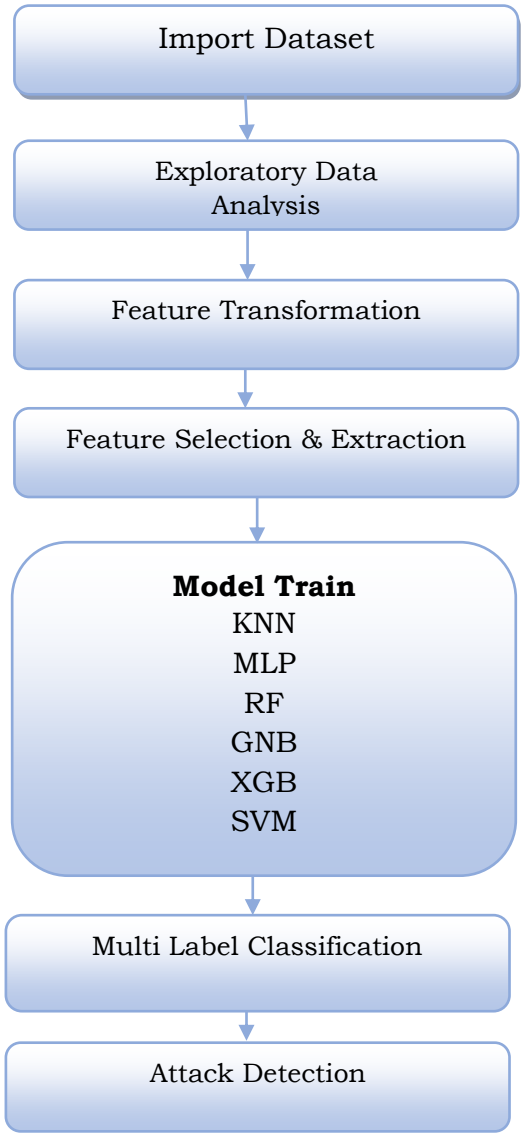


Figure 6 Work flow of Machine Learning Classifiers

Section: Research Paper
ISSN 2063-5346

RESULT ANALYSIS

The environment was set up on a single machine using python core programming and Jupyter notebook. The chosen operating software was Windows 10 (64-bit), ran on 8GB of RAM, 2 processors.

Performance Evaluation

Result and performance metrics for different models. Accuracy is one of the common performance metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is True-positive, TN is True-negative, FP is False-positive and FN is False-negative. The Seven models have yielded different accuracy scores, which are displayed in table 1.

Table 3: Accuracy Score of various Models

Sr. No.	Classifiers	Accuracy in %
1	KNN	99.83
2	MLP	99.82
3	Random Forest	100
4	Gaussian Naive Bayes	98.95
5	Xgboost	100
6	SVM	98.96

The test set accuracy, which is the accuracy of a model on samples it hasn't seen, using Scikit-Learn's accuracy_score, was shown to be 99.8%. The confusion matrix reported 58572 true negatives, 3 false negatives, 94 true positives, and 534 false positives. The classification scores are found.

These six different machine learning models are tested similar to other machine learning models by evaluating the several parameters as below.

Accuracy

$$ACC = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}$$

Precision: Is the measure of exactness

$$Precision = \frac{TP}{TP + FP}$$

Recall: Measures what percent of positive instances are positive

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: It measures accuracy using harmonic mean of precision and recall.

$$\mathrm{F}1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Section: Research Paper
ISSN 2063-5346

Table 4: Analysis and Performance Measure of Machine learning classifiers

Sr. No.	Classifiers	Precision	Recall	F1-Score	MCC
1	KNN	1.00	1.00	1.00	0.91
2	MLP	0.993	0.837	0.910	1.0
3	Random Forest	1.00	1.00	1.00	1.0
4	Gaussian Naive Bayes	1.00	0.0111	0.022	0.10
5	Xgboost	1.00	1.00	1.00	1.0
6	SVM	1.000	0.0239	0.0467	0.15

Table 4 shows the analysis and performance measure of various machine learning classifiers. Random forest and Xboost classifiers provide better result than other classifiers a use accuracy score, and other evaluation parameters like precision, recall and F1-score has 100% score. And also its Matthew correlation coefficient (MCC) score is 100%, so that Both the random forest and Xgboost models are acceptable. It is deemed to be a trustworthy metric yielding higher scores if the prediction yields a favourable score for each of these 6 variables.

In this section, we show the results of ML algorithms that are used for attack classification. These algorithms were implemented on Windows 7 and using a personal computer and the main programming language was Python using Jupyter Notebook on Anaconda platform [20]. The dataset of the experiment contained 125973 instances; 60% of instances. The distributions of instances in different attacks for both instruction and examining information that have been used in the experiment were shown in Figure 2. After training the classifiers, the evaluated performance was performed on testing data to classify data into one of five classes (Normal, Probe, R2L, and U2R). Moreover, the authors compared the performance of classifiers for each type of attack based on accuracy. Table 5 illustrates the accuracy for each individual class using four ML classifiers.

Attacks Taxonomy	Normal	DOS	Probe	R2L	U2R
KNN	99.86	99.86	98.87	90.66	42.85
MLP	99.23	99.87	98.62	90.98	41.25
Random Forest	99.97	99.97	99.61	95.45	38.09
Gaussian Naive Bayes	99.84	99.97	98.68	96.88	14.28
Xgboost	99.92	99.96	99.40	93.30	38.09
SVM	99.45	99.87	99.05	92.21	39.28

Table 5 Accuracy For Each Attack

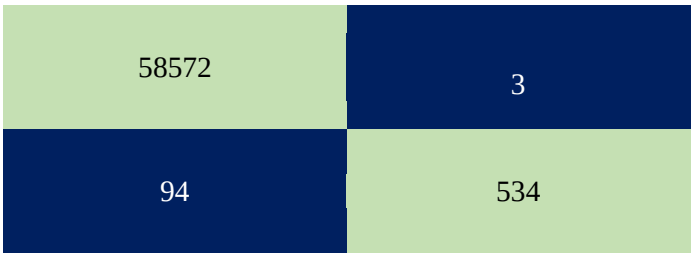
Confusion Matrix

One of the greatest techniques for evaluating IDS is the confusion matrix. The effectiveness of the suggested models, each column in this matrix indicates the predicted classes and each row indicates the actual classes, is dependent on a number of measures. Measuring the proportion of predicted information that are successfully identified and the proportion of mistakenly identified data allows one to assess the effectiveness of the suggested classifiers. The four fundamental components that define the confusion matrix's composition are listed in Table 4 [19] are as below.

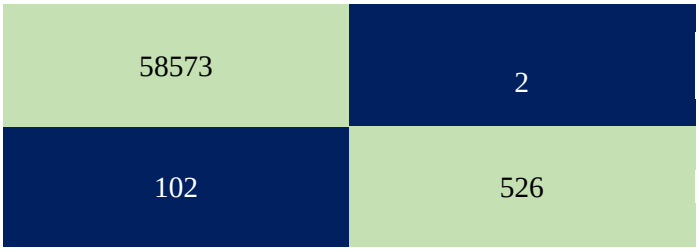
Actual		**Predicate Class**	
		Positive	**Negative**
Class	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 6: Layout of Confusion Matrix

Confusion Matrix of KNN



Confusion Matrix of MLP



Confusion Matrix of Random Forest

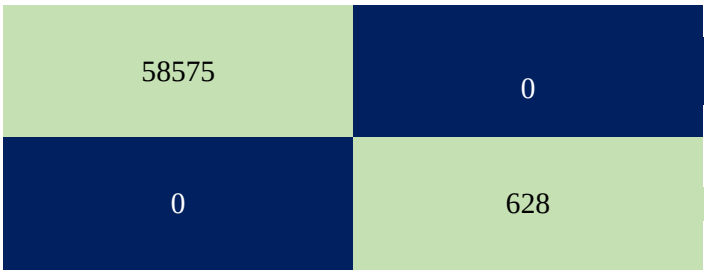


Figure 7: Confusion Matrix of MLP

Section: Research Paper
ISSN 2063-5346

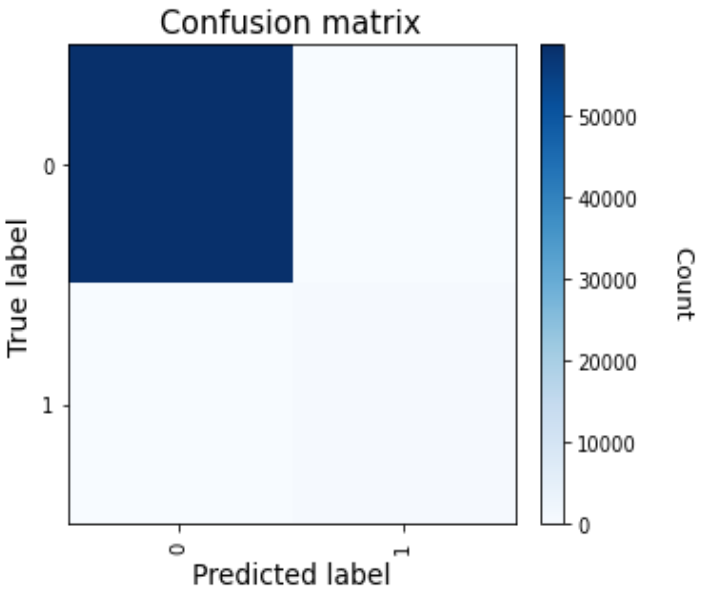
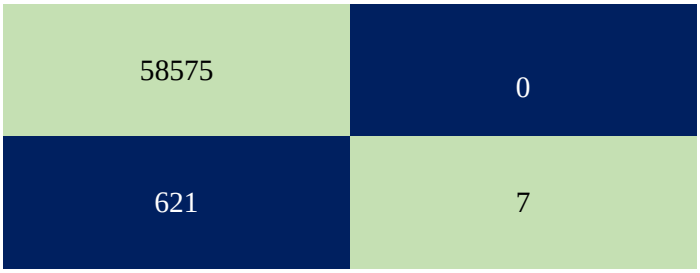


Figure 8: Confusion Matrix of Random Forest

Confusion Matrix of Gaussian Naive Bayes



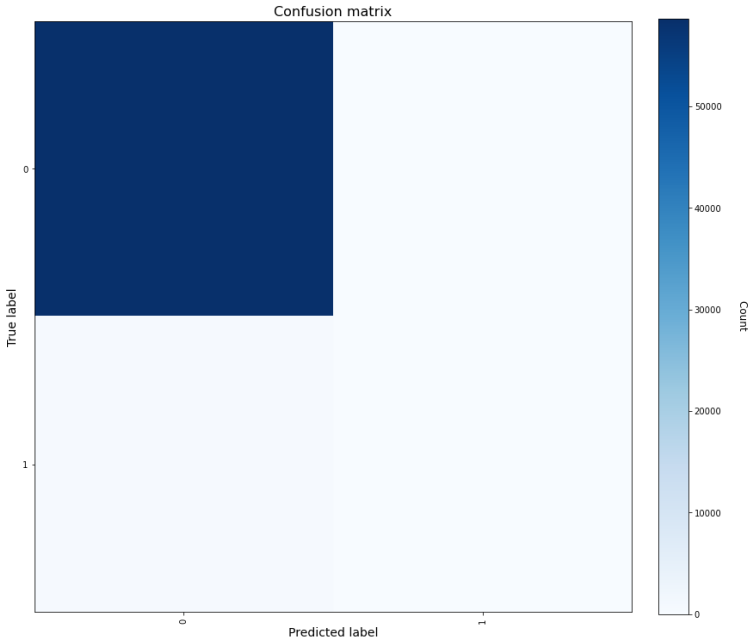


Figure 9: Confusion Matrix of Gaussian Naive Bayes

Confusion Matrix of Gaussian XGboost

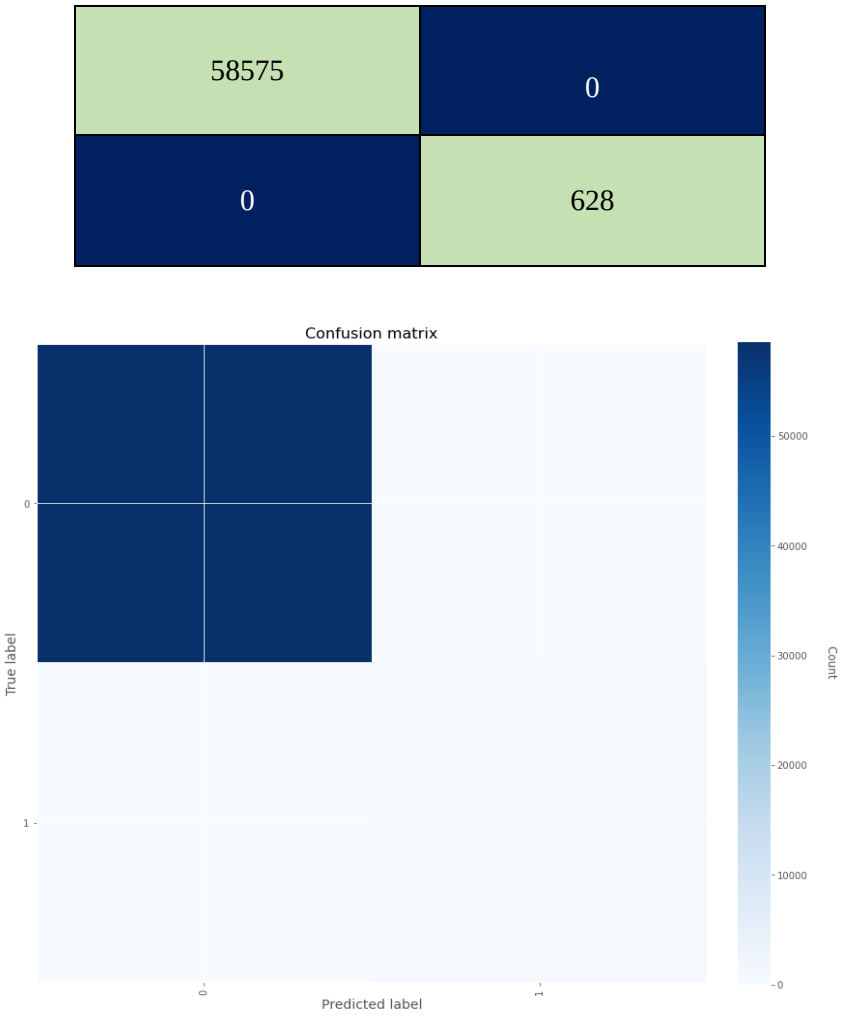
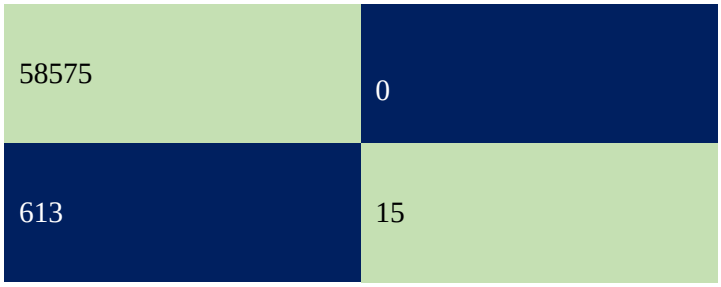


Figure 10: Confusion Matrix of XGboost

Confusion Matrix of Gaussian SVM



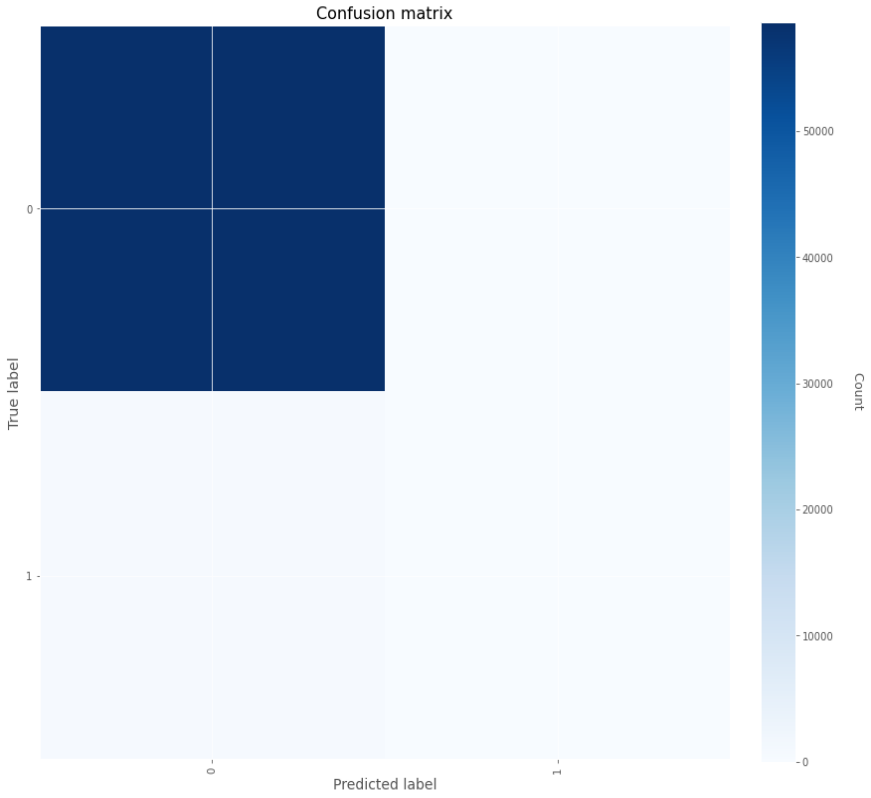


Figure 11: Confusion Matrix of Gaussian SVM

Conclusion

A strong intrusion detection mechanism is needed to safeguard the network perimeter given the exponential development in computer security. The construction of a major intrusion framework is the goal of this research. Firstly, from 2018 to 2022, examined over 25 articles on intrusion detection methods. We discovered that various detection methods, including DL, ML, and blockchain technologies, are essential to building these existence technologies. A study of the previous research gives background information on these methods' uses and restrictions in the field of attack detection. Furthermore, we looked into more than 18 articles, suggested an intrusion detection approach employing six classifiers, and assessed how well it performed in comparison to methods utilising the NSL-KDD database.

In this research, we demonstrated that, with high precision over vast data, various ML models can be employed for abnormality detection methods in intrusion detection systems for Assaults. Wireshark was used to collect the packets as we used a hping3 DoS attack to spoof the packets in order to mimic a DDoS attack. We then chose pertinent features for the IDS. After performing the required pre-processing and adding targeted variables, we downloaded the captures as a dataset. With an accuracy rate of almost 100%, ML has once again demonstrated its significance and effectiveness, which motivates additional research into the fusion of the domains of artificial intelligence and internet security. Further findings can be enhanced by using well-liked deep learning methods.

References

[1] Seraphim, Ida & Eswaran, Poovammal. (2021). Analysis of Intrusion Detection System Using Machine Learning Techniques. 10.1007/978-981-16-0965-7_34.

[2] Zhong, Wei & Yu, Ning & Ai, Chunyu. (2020). Applying a big data-based deep learning system to intrusion detection. Big Data Mining and Analytics. 3. 181-195. 10.26599/BDMA.2020.9020003.

[3] E. Viegas, A. O. Santin, and V. Abreu Jr, "Machine Learning Intrusion Detection in Big Data Era: A Multi-Objective Approach for Longer Model Lifespans," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 1, pp. 366-376, 1 Jan.-March 2021. DOI: 10.1109/TNSE.2020.3038618.

[4] T. Saranya, S. Sridevi, C. Deisy, Tran Duc Chung, M. K. A. Ahamed Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review", Procedia Computer Science, Volume 171, 2020, Pages 1251-1260.

[5] Ahlem Abid, Farah Jemili, "Intrusion Detection based on Graph oriented Big Data Analytics", Procedia Computer Science, Volume 176, 2020, Pages 572-581.

[6] Chen, Rung-Ching, et al. "Using rough set and support vector machine for network intrusion detection system." Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on. IEEE, 2009.

[7] Hadi, TheyaznHassn, and Manish R. Joshi. "Handling ambiguous packets in intrusion detection." Signal Processing, Communication, and Networking (ICSCN), 2015 3rd International Conference on. IEEE, 2015.

[8] Takkellapati, Venkata Suneetha, and G. V. S. N. R. V. Prasad. "Network intrusion detection system based on feature selection and triangle area support vector machine." International Journal of Engineering Trends and Technology3.4 (2012).

[9] Jemili, F., & Bouras, H. (2021). Intrusion Detection Based on Big Data Fuzzy Analytics. In (Ed.), Open Data. Intech Open. https://doi.org/10.5772/intechopen.99636

[10] Chongzhen Zhang, Yanli Chen, Yang Meng, Fangming Ruan, Runze Chen, Yidan Li, Yaru Yang, "A Novel Framework Design of Network Intrusion Detection Based on Machine Learning Techniques", Security and Communication Networks, vol. 2021, Article ID 6610675, 15 pages, 2021. https://doi.org/10.1155/2021/6610675

[11] Alzahrani, A.O.; Alenazi, M.J.F. Designing a Network Intrusion Detection System Based on Machine Learning for Software-Defined Networks. Future Internet 2021, 13, 111. https://doi.org/10.3390/fi13050111

[12] M. Zang and Y. Yan, "Machine Learning-Based Intrusion Detection System for Big Data Analytics in VANET," 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), 2021, pp. 1-5, doi: 10.1109/VTC2021-Spring51267.2021.9448878.

[13] Hongyu Liu, and Bo Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey", Appl. Sci. 2019, 9, 4396; doi:10.3390/app9204396

[14] Azad, Chandrashekhar, and Vijay Kumar Jha. "Fuzzy min-max neural network and particle swarm optimization-based intrusion detection system." Microsystem Technologies 23.4 (2017): 907-918.

[15] Singh, Shubhangi, and Rajendra Singh Kushwah. "A Study on Intrusion Detection in Wireless Networks by Using Genetic Algorithm Applications." Computational Intelligence and Communication Networks (CICN), 2014 International Conference on. IEEE, 2014.

[16] Benaicha, Salah Eddine, et al. "Intrusion detection system using genetic algorithm." Science and Information Conference (SAI), 2014. IEEE, 2014.

[17] Almansob, Saqr Mohammed, and Santosh ShivajiraoLomte. "Addressing challenges for intrusion detection system using naive Bayes and PCA algorithm." Convergence in Technology (I2CT), 2017 2nd International Conference for. IEEE, 2017.

[18] Altwaijry, Hesham, and Saeed Algarny. "Bayesian-based intrusion detection system." Journal of King Saud University-Computer and Information Sciences 24.1 (2012): 1-6.

[19]. M.Akhil Jabbar, B.L Deekshatulu, Priti Chandra," Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm", ScienceDirect Procedia Technology 10 85 – 94,2013

[20]. Chris, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance Problem", IEEE transactions on systems, man, and cybernetics part a: systems and humans, vol. 40, pp.185,197, 2010.

[21]. V. KrishnaVeni,T. Sobha Rani," On the Classification of Imbalanced Datasets" IJCST Vol. 2, SP 1, December 2011

[22] Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A.: Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 6, 33789–33795 (2018)

[23] Aksu, D., Ustebay, S., Aydin, M.A., Atmaca, T.: Intrusion detection with compar- ¨ ative analysis of supervised learning techniques and fisher score feature selection algorithm. In: International Symposium on Computer and Information Sciences. pp. 141–149. Springer (2018)

[24] Bansal, A., Kaur, S.: Extreme gradient boosting based tuning for classification in intrusion detection systems. In: International Conference on Advances in Computing and Data Sciences. pp. 372–380. Springer (2018)

[25] Boukhamla, A., Gaviro, J.C.: Cicids2017 dataset: performance improvements and validation as a robust intrusion detection system testbed

[26] Hou, J., Fu, P., Cao, Z., Xu, A.: Machine learning based ddos detection through netflow analysis. In: MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM). pp. 1–6. IEEE (2018)

[27] Papamartzivanos, D., M´armol, F.G., Kambourakis, G.: Introducing deep learning self-adaptive misuse network intrusion detection systems. IEEE Access (2019)

[28] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP. pp. 108–116 (2018)

[29] Mohammadreza Ektefa, Sara Memar, Fatimah Sidi and Lilly Suriani Affendey, "Intrusion Detection Using Data Mining Techniques", pp 200-203, IEEE, 2010