Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 8226 - 8235

Research Article

Network Intrusion Detection System Using KNN and Naive Bayes Classifiers

Rupa Devi T[a], Dr. Srinivasu Badugu[b]

[a]Assistant Professor, Department of Computer Science and Engineering, Keshav Memorial Institute of Technology
[b]Professor Department of Computer Science and Engineering Stanley College of Engineering and Technology for Women, Hyderabad, India - 500 001

***Corresponding author:** [a]rupa.devi179@gmail.com, [b]srinivasucse@gmail.com

Abstract

For every second after digitization, enormous quantities of data are generated from different networks. The value of providing this data with protection has therefore increased. The need to automate this protection framework has become necessary as the data is really massive. Intrusion detection systems are known as a great approach for intrusion detection. The system for detecting intrusions serves as an important mechanism to detect web security attacks. A proven technique for detecting network-based attacks, the Intrusion Detection System is still inexperienced in monitoring and recognising attacks, but efficiency remains unchanged. A large number of techniques that are based on machine learning approaches have been developed.

In this paper, the CIDDS-001 dataset [12] is analyzed and the observations have been marked. Two supervised machine learning techniques such as K-Nearest Neighbour and Naive Bayes are implemented on this dataset. KNN is implemented with different K – values. KNN is executed by changing the number of testing records. K – Value is also decided by making keen observations. KNN algorithm gives on an average 92.3% accuracy. Naive Bayes algorithm is also executed by changing the number of testing records. Naive Bayes algorithm gives on average 70.66% accuracy. Time complexity of NB algorithm is less than KNN. Comparison of both the methods is presented by comparing their evaluation metrics like Accuracy, Precision, Recall, Specificity and F-Measure. This paper is concluded by identifying the pros and cons of both the algorithms and by providing the future scope of this paper.

Keywords: Anomaly, K-Nearest Neighbour, CIDDS – 001 dataset, Naive Bayes, Accuracy, Precision Recall, Specificity and F-Measure.

1. Introduction

With substantial growth in internet use, network protection has become one of the most serious challenges for internet users and service providers [1,2,3]. In contrast to various kinds of intrusions, a secure network is specified in terms of the security of its software and hardware. By applying rigorous observation, analysis, and defence mechanisms, a network is safe. Computer networks are more vulnerable to malicious attacks as the world has become more linked through the Internet[1, 2, 4].

An intrusion can be defined as an unauthorized entry into the property or region of another, but in terms of computer science, it is the activities that threaten the fundamental security objectives of the computer network, such as confidentiality, integrity, and privacy. Detection of intrusion is the process of tracking and evaluating the activities that

occur in a computer system or network for indicators of potential instances of attacks and breaches of computer security standards, appropriate usage policies, or standard security policies.

"**Intrusion** is an attempt to compromise CIA (Confidentiality, Integrity, Availability), or to bypass the security mechanisms of a computer or network" [2, 5, 6]. "**Intrusion detection** is the process of monitoring the events occurring in a computer system or network, and analysing them for signs of intrusion" [2 , 5, 6]. NIDS is one of the main tools used to report network attacks.

"The network traffic is monitored by a network intrusion detection system (NIDS) via the identification of unusual activity that may constitute an attack or unauthorized access. NIDS are instruments that enforce these protocols in order to secure a network from intrusions that may occur inside or outside the network. These systems track a network's inbound and outbound traffic, regularly conduct analysis and report when an intrusion is detected. The network traffic collector, network traffic analysis engine, signature database, and alarm storage are the key components of this system"[4,5,6].

"The role of every component is important in intrusion detection. Network traffic is sniffed by traffic collectors which are in the form of packet traces, then analysis engine performs a profound analysis of the sniffed traffic data and directs the alarm signals to alarm storage once intrusion is identified. The patterns or signatures of known intruders are stored in signature database, and then matching is done using these signatures.[2]"

"NIDS is classified into Misuse detection (MD) [1, 7], Anomaly Detection (AD) [1, 8]". " In order to detect intrusions, MD based NIDS uses patterns or signatures of already living attacks. Whereas AD based NIDS tests and reports it as an assault on firm deviations from standard network traffic profiles. The Detection Rate (DR) of MD related NIDS is high, while the False Positive Rate (FPR) is low compared to AD. But AD-based NIDS detects new network attacks, so this property overtakes them from NIDS based on MD. On offline data, MD works better, while AD works better on online data. [2]".

"In creating a better NIDS, Machine Learning (ML) [1, 9] plays a major role. It allows a framework to learn and behave appropriately for the next traffic patterns from the already current traffic patterns or signatures. The two significant roles in the ML are training and testing. ML requires large and complex datasets consisting of distinct types of normal and abnormal traffic patterns. For enhanced learning, there is also a need to use ML algorithms for NIDS that are low in computational time and space complexity. We have used some popular NIDS assessment metrics such as DR, FPR, Accuracy, Precision and F-measure [2, 10] to evaluate the CIDDS-001 dataset [12] in this work. Due to its better DR and Naive Bayes classification, we have used ML models such as the KNN classification algorithm [1,11] as it can often outperform most advanced classification methods despite its simplicity"[2].

2. Related Work

Abhishek V et al.[1] used the new "CIDDS-001 dataset" [12] to detect the intrusion by using "machine learning algorithms" [9] based on distance. Luke et al.[13] shows that "for the NIDS, which is a sniffer in a network, a non-stationary model (PHAD) achieves more than 35 times greater efficiency than the simple stationary model (GMM)" . Alsalla Mutaz et. Al.[14] helps researchers "prepare computers for attacks such as zero-day attacks to be detected". Different classification algorithms such as " Multilayer perceptron, SMO, SVM, FT, Naive Bayes, J48, Bayesian network, and Multinomial logistic regression" were used. "A new semi-supervised anomaly detection method using the k-means clustering algorithm" was introduced by E. KarsligEl et al.[15] and regular samples were divided into clusters in the training process.They obtained a precision of 80.12 percent. "Minimum redundancy maximum- significance" was used by Biswas et al.[16], "CFS, PCA and IGR feature selection techniques and SVM, KNN, NN, DT and NB" are the classifiers used by them. They noticed that the "KNN classifier" works better than other classifiers. "anomaly-based network intrusion techniques" used by [7], "Machine learning based network intrusion detection" used by [13, 14, 15]. Using the "association rule mining algorithm", Moustafa et al. [17] Using classifiers, "accuracy and false alarm rate (FAR)" were measured. The findings show that the "UNSW-NB15" features are far more effective than "the KDD 99 dataset". "Random Forest" used by Hasan et al [18] here and improves accuracy by reducing time complexity.Janarthanan et al.[19] "kappa statistics" were found to be improved here due to classification using selected features. For feature reduction in the dataset, Aminanto et al. [20] used "deep feature selection and extraction". They was achieved 99.92 accuracy and 0.012 percent "false alarm rate (FAR)"

3. Proposed Method

We used two machine learning algorithms, such as the classification algorithm for KNN and the classification for Naive Bayes. For training and testing the models, we used the CIDDS-001 dataset.

CIDDS-001 Dataset: "CIDDS-001 ("CIDDS-001", 2017) [12] is a labeled flow-based dataset (Ring, Wunderlich et al. 2017). It was developed mainly to evaluate AD based NIDS. The dataset contains traffic from both OpenStack and External Servers. CIDDS-001 dataset comprises of13 features and a class attribute. Out of them

11 features were used for this study. The features like Attack ID and Attack Description are ignored because they just give elaborated information of the executed attacks. So, these attributes did not contribute to the analysis significantly. Almost 153,026 instances of external servers and 172,839 instances of OpenStack server were gathered for analysis. Every instance was labeled as classes namely, normal, victim, attacker, suspicious and unknown. “

3.1 Proposed Architecture

This section focuses on the system architecture of the proposed system, Network Intrusion Detection System using supervised machine learning algorithms like KNN and Naive Bayes classification algorithms. The main goal of the proposed system is to detect whether a given packet is a normal or suspicious or attacker or victim or unknown packet.

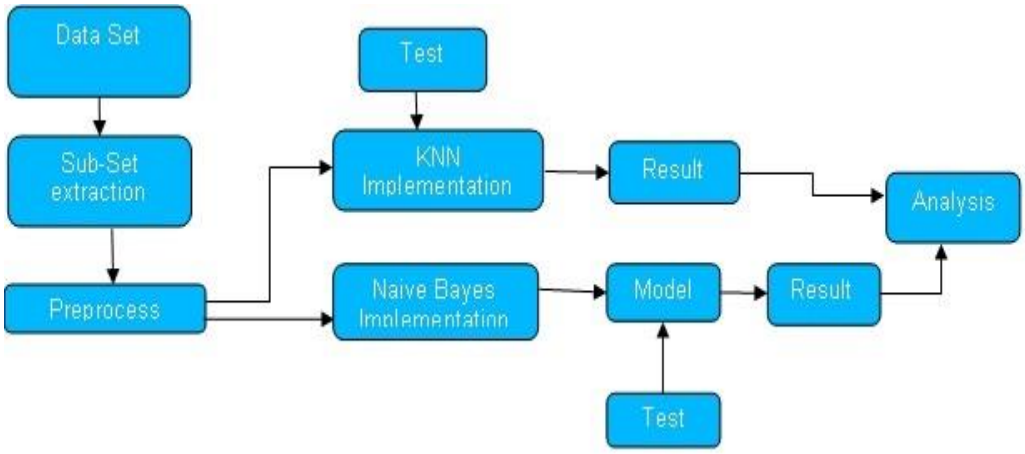

Fig 1 System Architecture

The following are the steps included in the system architecture shown in Figure 1

1. **Dataset:** CIDDS 001 dataset which consists of traffic from External Server and Open Stack server is taken. This consists of 4 weeks data each in External and Open Stack servers. In External Server each week file consists of 1, 50,000 records to 1,90,000 records. In Open Stack Server each week file consists of 10,48,575 records. “The CIDDS-001 has 14 attributes out of which 12 have been used in this empirical study”.
2. **Sub-Set Extraction:** As data is very huge, for computation purpose we have extracted subsets of datasets. These subsets are generated by randomly picking the records from a given file so that consistency is maintained in these subset files.
3. **Pre-process:** We have pre-processed data by removing the empty spaces appended left and right to each value in the comma separated file.
4. **KNN Implementation:** This module implements the KNN algorithm on the dataset.
5. **Naive Bayes Implementation:** This module implements the Naive Bayes algorithm on the dataset.
6. **Model Building:** Naive Bayes Algorithm when provided with training data from the dataset it will build a model (by calculating likelihood, prior probability of class and predictors) and be ready to use this model while testing a record. Whereas, KNN is a lazy learner so when a record is given for testing it will then compute the distances between the test record to all the training records.
7. **Test:** This module tests the respective algorithms by providing a test record.
8. **Result:** This module computes the results and classifies a test record.
9. **Analysis:** Both KNN and Naive Bayes results are computed for same training and testing records and these results are compared and analysed.

4. Implementation

As the dataset is very huge we have tried to extract some subsets of variable size, so that the computation process is not complicated. These subsets are generated by randomly picking the records from a given file so that consistency is maintained in these subset files.

Flow Chart of KNN

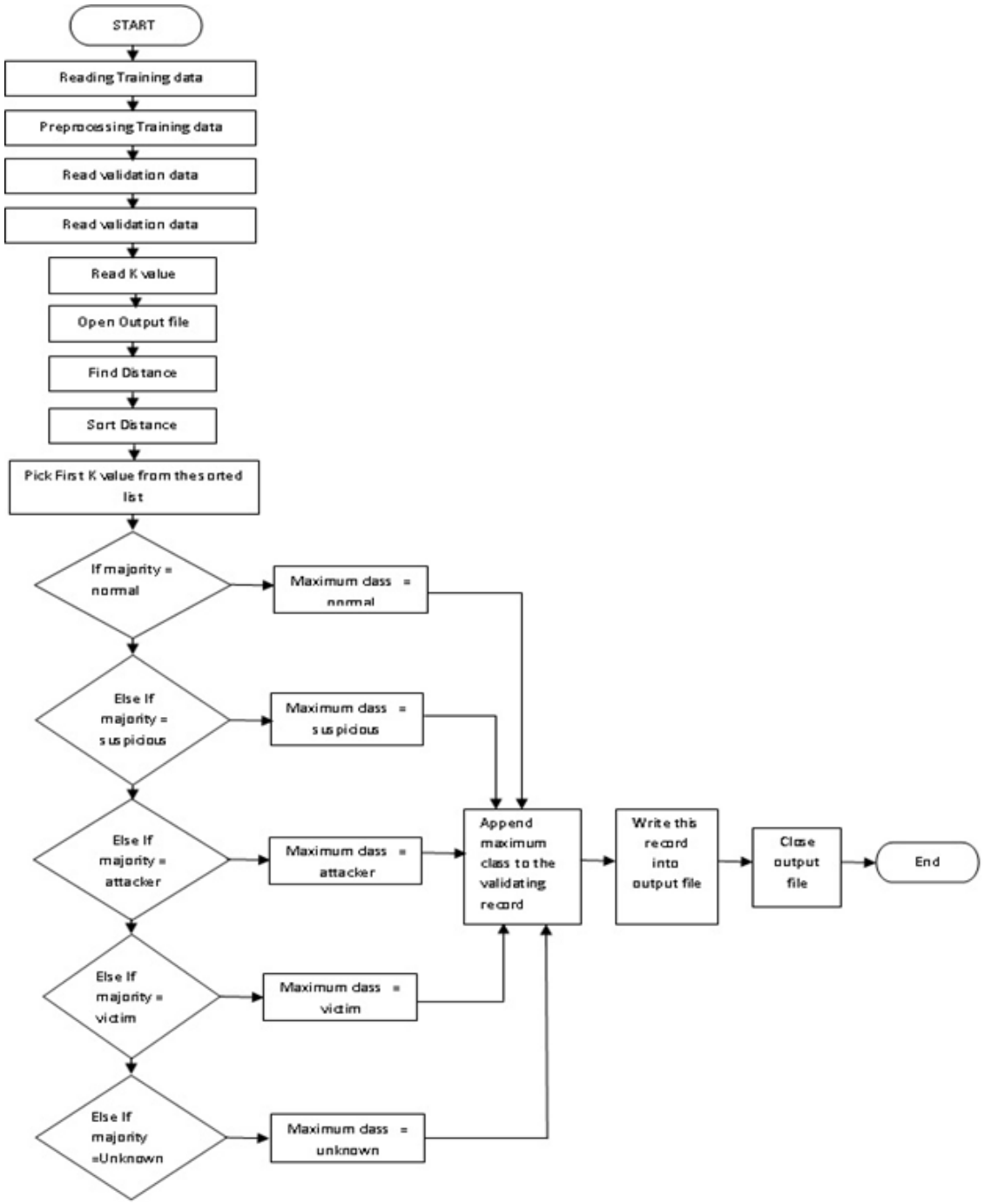

Fig 2 KNN Flow Chart

The KNN classifier is an example-based algorithm for learning and classification, also known as the lazy classifier. It is the distance-based classifier most frequently used and uses each training instance as a prototypical example. This classifier is based on a function of distance that essentially tests the similarity or distinction between two instances. The standard measure of distance, i.e. For numerical data, Euclidean distance is used but because our dataset consists of mixed variables like some variables are categorical and some variables are numerical, we have used Jaccard Similarity.

- $J(d1,d2) = (d1 \wedge d2)/(d1 \vee d2)$ Eq. 1

Where d1 and d2 are the records whose distance is being calculated.

The output file generated will have the test records appended with the class generated after implementing the algorithm. Firstly, we preprocess the data given for training and testing. Read K value. Then, find the distances of each test record with every training record. Sort these distances and pick first K distances. By applying simple majority method, we append the majority class to the test record.

Flow chart of Naive Bayes Algorithm

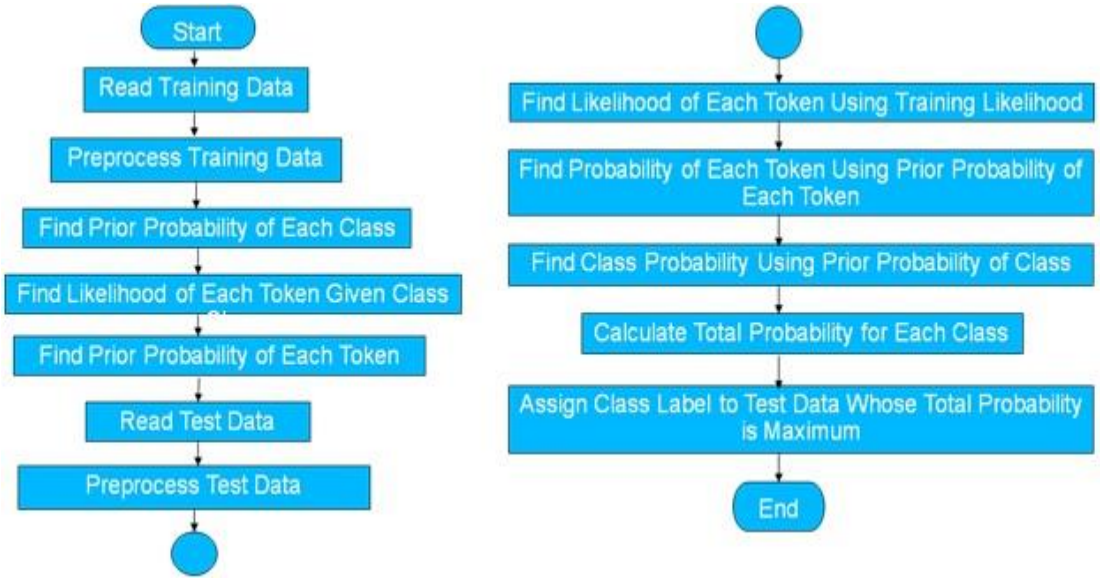

Fig 3 Naïve Bayes Flow Chart

Naive Bayes Algorithm reads and preprocesses training data. It calculates likelihood, prior probability of class and prior probability of predictors and stores in dictionaries. It preprocesses testing data. Using the dictionaries of likelihood, prior probabilities of classes and prior probabilities of predictors from training module, the test record will be evaluated and is assigned a class which has maximum probability. Then we generate confusion matrix and calculates Accuracy, Precision, Recall, Specificity and F-Measure.

5. Result Analysis

5.1 KNN

5.1.1Accuracy for Different K Values

KNN is implemented using 1,59,373 records of week2.csv from CIDDS 001 dataset are used to train the KNN model. A subset of 500, 1000, 1500 records taken from week3.csv from CIDDS 001 dataset to test the model. For different K values accuracy is recorded and analyzed as shown in Table 1.

Table 1: Accuracy of KNN using Different K Values

No. of Testing Records	**K=1**	**K=3**	**K=5**	**K=11**	**K=15**	**K=21**	**K=25**	**K=51**
500	91%	91.6%	91.4%	91.4%	90.8%	90.8%	90.8%	90.6%
1000	93.2%	93.7%	93.7%	93.5%	93.5%	93.7%	93.6%	93.6%
1500	92.5%	92.9%	92.9%	92.5%	92.6%	92.5%	92.6%	92.5%

Deciding K Value

From the Fig 4 we can say that for K=3 and K=5, we got highest accuracy for different number of testing records. Hence we try to fix K=5 as we have 5 different classes in the dataset. If we take K=3, the classification can be sometimes unique and it's not possible to get a majority class. But if for K=5 also, the classification is unique then we say that dataset is ambiguous which is not possible.

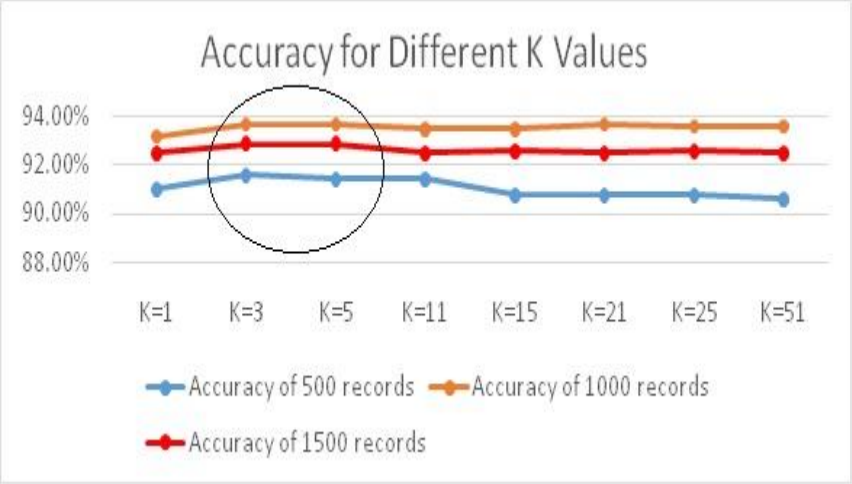

Fig 4 Deciding K Value Using Accuracy

5.1.2 Evaluation Metrics For KNN

Fixing K=5 and by increasing number of test records, we got on an average of 92.4125% accuracy with KNN classification algorithm

Confusion Matrix For KNN

For 5000 testing records, the table below shows the confusion matrix:

Table 2: Confusion Matrix for KNN

	NORMAL	SUSPICIOUS	ATTACKER	VICTIM	UNKNOWN
NORMAL	860	0	0	0	0
SUSPICIOUS	1	3612	0	0	13
ATTACKER	0	0	77	0	1
VICTIM	0	0	0	62	0
UNKNOWN	1	6	0	0	288

5.2 Naive Bayes

5.2.1 Accuracy for Different Size of Training Records

For a subset of 500 records from week2.csv which is used as testing data, the following Table 3 shows the correctly classified data and wrongly classified data for different number of training records. 500 to 20000 records taken from week2.csv of CIDDS 001 dataset are used to train the model. A subset of 500 records taken from week3.csv of CIDDS 001 dataset is used to test the model.

Table 3 Classification for Different Size of Training Records

No. of Training Records	Correctly Classified	Wrongly Classified
500	495	5
1000	479	21
5000	457	43
10000	305	195
15000	230	270
20000	242	258

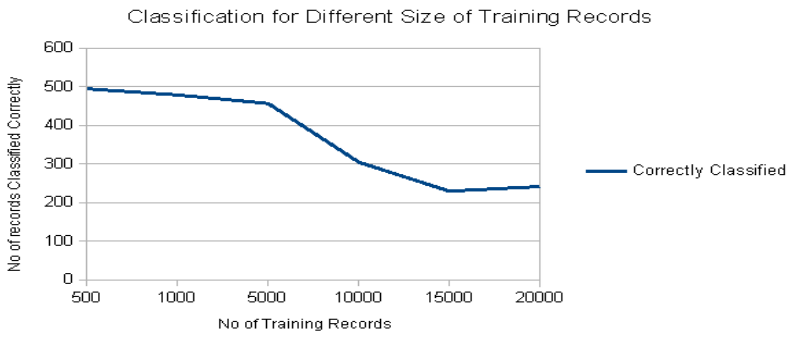

Fig 5 Graph for 500 testing records from week2.csv

5.2.2 Evaluation Metrics For Naive Bayes

Confusion Matrix Of Naive Bayes

For 5000 testing records, the table below shows the confusion matrix:

Table 4: Confusion Matrix for Naive Bayes

	NORMAL	SUSPICIOUS	ATTACKER	VICTIM	UNKNOWN
NORMAL	58	802	0	0	0
SUSPICIOUS	0	3626	0	0	0
ATTACKER	0	78	0	0	0
VICTIM	0	61	0	0	61
UNKNOWN	0	283	0	0	12

5.3 Comparative Study

Following table compares the evaluation metrics such as Accuracy, Precision, Recall, Specificity and F-Measure of KNN and Naive Bayes Algorithms. The evaluation metrics are compared by increasing the number of testing records in each iteration. From these comparisons we can say that performance of KNN algorithm is better than Naive Bayes algorithm.

Table 5: Comparison of KNN with Naive Bayes

Variations in testing records	Accuracy	Precision	Recall	Specificity	F-Measure
KNN_500	0.908	0.6811	0.9721	0.9801	0.8019
NB_500	0.736	0.4356	0.7156	0.9418	0.541549
KNN_1000	0.92	0.7355	0.9763	0.9828	0.8390
NB_1000	0.699	0.3943	0.6743	0.9310	0.497616
KNN_1500	0.9213	0.7250	0.9720	0.9841	0.8305
NB_1500	0.7190	0.4364	0.914	0.9377	0.590743
KNN_5000	0.9798	0.9919	0.9899	0.9975	0.9909
NB_5000	0.7392	0.2216	0.5341	0.9502	0.313237
KNN_10000	0.8851	0.7171	0.9690	0.9822	0.8242

NB_10000	0.6839	0.4124	0.9176	0.9361	0.56905
KNN_20000	0.8585	0.7202	0.9718	0.9820	0.8273
NB_20000	0.6625	0.4113	0.9162	0.9360	0.567733

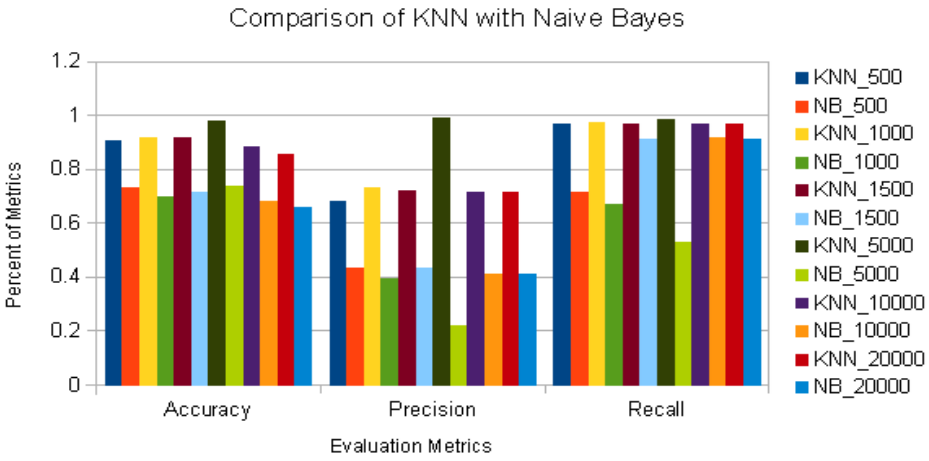

Fig 6 performance comparing between KNN and Naive Bayes using CIDDS-001

6. Conclusion& Future Scope

Numerous methods for the effective advancement of security systems are documented in the literature. However, study on the enhancement of datasets used for training and testing purposes of such security systems is also important. In this thesis, in depth analysis of CIDDS-001 dataset is shown and the sightings are presented. This thesis provides the detailed overview about the working procedure of multiple research techniques along with the merits and demerits. From the comparison of the research papers it is proved that the most recent method by Abhishek Verma et al. [2] of using machine learning algorithms on CIDDS-001 dataset provide the better results than the existing research method. Weka tool is used in base paper to implement the project.

With the rapid growth in the past two decades of information technology. Computer networks are commonly used in manufacturing, industry and different aspects of human life. The rapid advancement of information technology, on the other hand, has posed many problems in the construction of secure networks, which is a very difficult task. The security, integrity and confidentiality of computer networks are compromised by several kinds of attacks. Cyber threats are evolving rapidly and there's little hope about the cyber security situation. Therefore the framework for intrusion detection serves as a protective mechanism to detect web security attacks. We used two machine learning algorithms in this paper to detect the intrusion using the dataset CIDDS-001. We have writing the code in python to implement this project without using any in built packages. We also tried to overcome some of the issues of existing approaches by using latest datasets like CIDDS-001. KNN algorithm gives on an average 92.3% accuracy. Naive Bayes algorithm gives on an average 70.66% accuracy. Time complexity of NB algorithm is less than KNN. But NB algorithm is not suitable for this dataset as data is more unique.

In the future, we will implement Deep Learning Algorithms on the latest datasets such as CIDDS-002 to enhance computational time and cost. The results will be more precise if we can train the machine sing live data or online data by collecting them from real time networks.

References

[1] Verma, A., & Ranga, V. (2018). Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance-based machine learning. Procedia Computer Science, 125, 709-716.

[2] Devi, T. R., & Badugu, S. (2020). A Review on Network Intrusion Detection System Using Machine Learning. In Advances in Decision Sciences, Image Processing, Security and Computer Vision (pp. 598-607). Springer, Cham.

[3] Medaglia, C. M., & Serbanati, A. (2010). An overview of privacy and security issues in the internet of things. In The internet of things (pp. 389-395). Springer, New York, NY.

[4] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials, 21(1), 686-728.

[5] Sangkatsanee, P., Wattanapongsakorn, N., & Charnsripinyo, C. (2011). Practical real-time intrusion detection using machine learning approaches. Computer Communications, 34(18), 2227-2235.

[6] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24.

[7] Garcia-Teodoro, P., Diaz-Verdejo, J., MaciÃ¸a-FernÃ¸andez, G., & VÃ¸azquez, E. (2009). Anomaly-based network intrusion detection: Tech-niques, systems and challenges. computers & security, 28(1), 18-28.

[8] Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 305-316). IEEE.

[9] Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D.,& Zissman, M. A. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings (Vol. 2, pp. 12-26). IEEE.

[10] Flach, P. (2012). Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press.

[11] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[12] Ring, M., Wunderlich, S., Gruedl, D., Landes, D., Hotho, A.: "Flow-based benchmark data sets for intrusion detection." In: Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS), pp. 361-369. ACPI (2017)

[13] Lee, C. H., Su, Y. Y., Lin, Y. C., & Lee, S. J. (2017, September). Machine learning based network intrusion detection. In 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA) (pp. 79-83). IEEE.

[14] Wahyudi, B., Ramli, K., & Murfi, H. (2018). Implementation and analysis of combined machine learning method for intrusion detection system. International Journal of Communication Networks and Information Security, 10(2), 295-304.

[15] M. E. KarsligEl, A. G. Yavuz, M. A. Güvensan, K. Hanifi and H. Bank, "Network intrusion detection using machine learning anomaly detection algorithms,"2017 25th Signal Processing and Communications Applications Conference(SIU), Antalya, 2017.

[16] Biswas, Saroj. (2018). Intrusion Detection Using Machine Learning: A Comparison Study. International Journal of Pure and Applied Mathematics. 118. 101-114.

[17] Moustafa, N., & Slay, J. (2015, November). The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems. In Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on (pp. 25-31). IEEE

[18] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. Journal of Information Security, 7(03), 129-140. [19] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, Secondquarter 2016.

[19] Janarthanan, T., & Zargari, S. (2017). Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In Proceedings of 26th International Symposium on Industrial Electronics (ISIE) (pp. 1881-1886). Edinburgh, UK: IEEE.

[20] Aminanto, M. E., Choi, R., Tanuwidjaja, H. C., Yoo, P. D., & Kim, K. (2018). Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. IEEE Transactions on Information Forensics and Security, 13(3), 621-636.

[21] Zhang, Y., Yang, Q., Lambotharan, S., Kyriakopoulos, K., Ghafir, I., & AsSadhan, B. (2019, October). Anomaly-Based Network Intrusion Detection Using SVM. In 2019 11th International Conference on Wireless Communications