Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/ 21660
DOI URL: http://dx.doi.org/10.21474/IJAR01/21660

RESEARCH ARTICLE

ANALYSE THE EFFECTS OF NORMALIZATION ON NSL-KDD INTRUSION DETECTION SYSTEM

MD Mahmudul Hasan[1], Tauhid Uddin Mahmood[2] and Nudrat Fariha[2]

1. Department of Technology Management, University of Bridgeport, 126 Park Avenue, Bridgeport, 06604, Connecticut, USA.
2. Department of Business Analytics and Systems School of Business University of Bridgeport 126 Park Avenue, Bridgeport, 06604, Connecticut, USA.

Manuscript Info

Manuscript History
Received: 20 June 2025
Final Accepted: 23 July 2025
Published: August 2025

Key words:-
Intrusion Detection System, NSL-KDD, Machine Learning, Data Normalization, Classification Accuracy

Abstract

Protecting unauthorized access is critical in the realm of information and network ecurity for everyone's well-being. The Intrusion Detection System (IDS) is a type of intrusion detection system that playsone of the most significant roles here. It's a classifier that determines if the data is normal or malicious. In this paper, the author used several Machine Learning Techniques for Intrusion Detection to investigate the effects of normalization on the NSL-KDD dataset. In this experiment, WEKA is an open-source data mining tool that we utilized. The Decision Tree, Naïve Bayes, Random Forest, and One R algorithms are applied in the context of with and without normalization. Different normalization techniques are used, such as, Min-Max, Z-Score, Log Scaling, and Mean Centered Scaling. The obtained result shows that the Random Forest has a greater accuracy rate than others. In both cases, i.e., with and without normalization.

Introduction:-

In today's world, the internet plays an unavoidable part in our life. We are making our life effortless, simple and doing our works rapidly by using internet appropriately. It gives us information and knowledge for our personal, economic and also for social development. A monitoring system called Intrusion Detection System (IDS) is a powerful tool which spots suspicious activities and acknowledges when they have been detected. Based on this information network administrators can review the issues and take proper actions to deal with this threat. Generally, IDS is classified as an Anomaly Intrusion Detection System (AIDS) and Misuse Intrusion Detection System (MIDS) [1]. User's activities are matched with known attack types in MIDS. Any action is considered an attack in any match.

Any deviation from usual behavior is regarded as an attack [2]. AIDS has lower detection rate than MIDS as they are unable to detect new attack types. As lack of performance, misjudgment and false detection are challenges in IDS, technologies like Machine Learning (ML) are used due to the power of computing. ML classifiers increase the

Corresponding Author:-MdMahmudul Hasan
Address:-Department of Technology Management University of Bridgeport 126 Park Avenue, Bridgeport, 06604, Connecticut USA.

power and accuracy of the system. Today researchers use several ML approaches such as a support vector machine (SVM), a decision tree (DT), a random forest (RF), or a K-Nearest Neighbor (K-Nearest Neighbor) (KNN)for classification of network data and feature selection. Before the deployment in real world, datasets are needed to analyze the performance of the IDS. Only some datasets are accessible for public and from these some are even lack of completeness and comprehensiveness. In intrusion detection some frequently used datasets are NSL-KDD, KDD Cup 1999 Data, UNSW_NB15 and BETH dataset and so on. [3]

E-commerce, payment systems, and banks are very important to people because of their transaction processes. So, the security of these systems has to be ensured. As the use of the web has increased, security has become a prime concern. Criminal attack using single or multiple computers opposed to a single or multiple computers or system and loot data, damage computers or adopt an affected computer for other attacks. Some network security systems, like encryption and firewalls facility are applied even though various attacks omit the security of the system. So, it needs to have acknowledged them at the beginning of the damage to secure the resources. Therefore, relevant steps can be taken to eliminate intrusion. Some research has been done before to make IDS.

As a computer science student, it is our responsibility to take part in research to secure systems. We want to secure people by spotting doubtful activities and acknowledging the network administrators' sending of emails, text messages, etc. Both hardware and software level security exist. Hardware security secures systems physically with devices. Every hardware security needs software to work on the other hand software security can skip hardware. This is why hardware security is more costly than software security. Hardware security is more secure than software security. For example, TPM (Trusted Platform Module) is a chip that is installed on new motherboards of computers.

It uses both software and hardware to secure any kind of important passwords or encryption keys if they are sent in unencrypted form. Our objective is to observe different accuracy rate. Firstly, without normalization techniques and then with normalization techniques of algorithms on the dataset NSL-KDD. The precision of Decision Tree, Naïve Bayes, Random Forest, and One R algorithms will be observed here. The normalizations will be Min-Max, Z-Score, Log Scaling, and Mean Centered Scaling. NSL-KDD dataset will be used here. Canadian Institute for Cybersecurity is the provider of this dataset. In the suggested test sets, there are no duplicate records. As a result, the learners' performance is not influenced by approaches that have higher detection rates on large datasets. For both with and without normalization, our every algorithm and normalization technique have maximum accuracy in Random Forest then sequentially Decision Tree, One R and Naïve Bays.

Literature Review:-

There are many security systems available in the current world to protect information from hijacking [21, 22]. Still, malicious activities are taking place on computer systems and networks. As security becomes a major concern on computer networks, a large amount of work has been done in the area of Intrusion Detection System (IDS). In [4], this research work has been done to detect network intrusion.

Researchers used ML classifiers such as SVM, KNN, LR, NB, MLP, RF, ETC, and DT, with the findings analysed using NSL-KDD. Similarly, in [5], an ANN-based Intrusion Detection System was developed using the NSL-KDD dataset. The Levenberg-Marquardt (LM) and BFGS quasi-Newton Backpropagation algorithms were used to implement it. Also, in [6], work has been done for network intrusion detection, where a deep-learning-based technique was used. The NSL-KDD dataset was used to train a Logistic Regression method with a sparse auto-encoder. On an unbalanced NSL-KDD dataset, an Intrusion detection using a hybrid data mining technique was created [7]. A combination of the CANN 19 algorithm and the technique of synthetic minority oversampling was applied to detect the accuracy. In [8] they used to detect IoT assaults, a Deep Neural Network (DNN) was used. The Intelligent Intrusion Detection System can only be built if an effective dataset is provided.

Methodology:-

For intrusion detection, there are numerous datasets available. KDD Cup 1999 Data, UNSW_NB15 and CICIDS2017 are just a few examples. However, each has its own set of limitations. In this research, we have selected the NSL-KDD dataset for implementation. It can be used as a useful benchmark dataset to test different intrusion detection systems. In addition, the number of records in the NSL-KDD train and test sets is acceptable. Normal and malicious connections are included in the dataset. DoS, Probe, R2L, and U2R are the four types of

attacks. There are 22 attacks in the training set. The testing set features 38 attacks, including 16 that are not seen in the training set.

AttackClass	AttackType	Some Key Characteristics
DoS	Worm, Neptune, Smurf, Land, Udpstorm, Teardrop, Apache2, Back, Pod, Process table, Worm, Neptune, Smurf, Land, Udpstorm, Teardrop	Error-prone packets as a percentage of total packets -source bytes
Probe	Ip-sweep, Port-sweep, Satan, Nmap, Mscan, Saint	The duration of the connection is measured in bytes from the source.
R2L	Httptunnel, Spy, Xsnoop, Xlock, Sendmail, Warezmaster, Multihop, Phf, Ftp_write, Warezclient, Imap, Guess_Password, Snmpguess, Snmpgetattack	-the number of file creations -The number of shell prompts invoked
U2R	Buffer_overflow, Rootkit, Ps, Loadmodule, Perl, SQL attack, Xterm	Requested service – connection time – number of failed login attempts

Table 1: Attack types
Table 1: Types of attacksand characteristics [9]

The authors have used NSL-KDD Datasets for our experiment. We have chosen this version because it has many advantages. NSL-KDD is an updated version of the KDD cup99 data set that suggests solutions to some of the previous version's issues. This data collection serves as a useful benchmark for researchers comparing various types of intrusion detection system (IDS) methodologies, as well as building an IDS. There are other data sets available in this sector. The NSL-KDD data collection has a number of advantages, including:

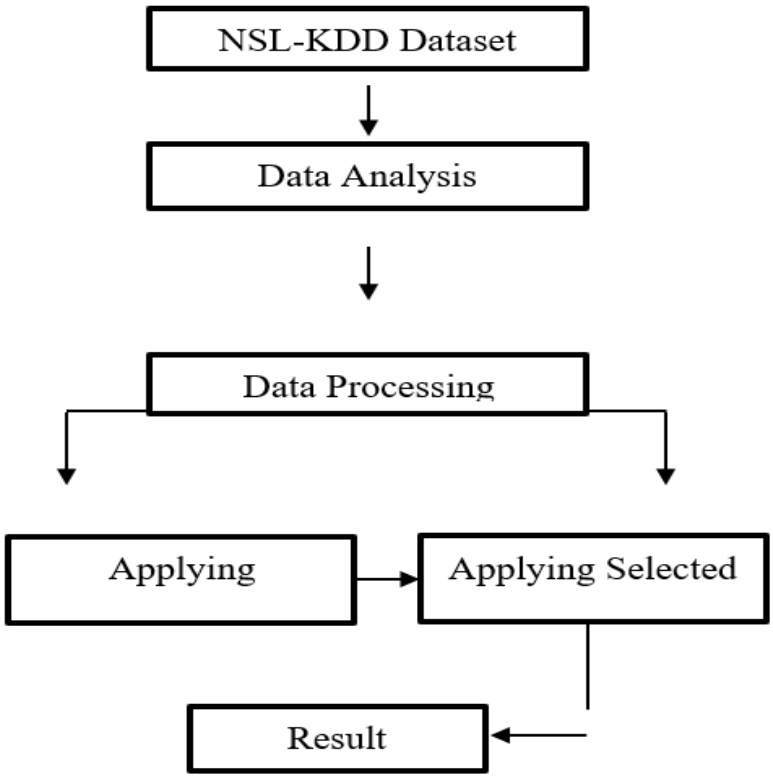
Figure 1: Methodology of the analysis of the NSL-KDD datasets

1. The test sets contain no duplicate records.

2. The train and test data sets have a sufficient number of records, and the quantity of chosen records from each tough level gathering corresponds to the level of records in the original KDD data set [10].

From figure 1, we can see the process of how the accuracy of the result will be found out. First, we will select a dataset from the NSL-KDD dataset collection. Then the dataset will be analyzed and processed. After that, we will implement the data with some algorithms through the WEKA tool and we will see some results. Then again, we will implement the dataset with some normalization techniques and algorithms to see what the data results will be like. Naïve Bays, Random Forest, Decision Tree and One R Algorithms will be used to measure the performance. The tests will be run in WEKA, and the efficiency of the classification algorithms in classifying the NSL-KDD data set will be evaluated. There are eight types of NSL-KDD data types, but the KDDTest-21.ARFF file has been chosen for implementation in WEKA with some algorithms and also with some normalization techniques. This dataset file means Records with a difficulty level of 21 out of 21 are excluded from this portion of the KDDTest+.arff file [11].

The KDDTest-21 dataset contains 11850 network traffic samples. There are 41 features, 1 class label, and 1 difficulty label on this album.



Data Analysis:-

Among many algorithms, Naive Bayes, Decision Tree, Random Forest and the One R algorithms have been chosen to apply to the NSL-KDD dataset. The reason behind choosing the Naive Bayes algorithm is that it is a quicker and uncomplicated machine learning approach for predicting a class of datasets. The Decision Tree algorithm is preferred because it is capable of processing data with a large number of dimensions. The decision tree algorithm makes it simple to understand [13]. The Random Forest algorithm was chosen because it takes less time to train than other algorithms. It accurately predicts output and runs rapidly, even with a large dataset. The One R approach was chosen because it is a straightforward technique that simply predicts a sample's class by determining the most common class for the feature values. In some real-world datasets, this basic algorithm has been proven to perform well [14]. These algorithms have been chosen to see what changes occur in accuracy results in NSL-KDD datasets by applying these algorithms.

Normalization Techniques:

Here are some normalization techniques that we are going to use along with the algorithms.

Z-Score Normalization

The essence of this technique is the transformation of data to a common scale using the values conversion, where the average number equals zero and the standard deviation equals one. [15].

The formula for Z-Score Normalization is-

$$z = \frac{v - mean}{sd} \dots\dots\dots\dots\dots\dots (i)$$

Here,

- Z is the final answer after doing the Z-Score normalization.
- V is the value of the data that we are going to work on.

The average value of the given data is the mean. sd means standard deviation.

Here is another rule to find standard deviation, which is-

$$sd = \sqrt{\frac{\sum|x - \bar{x}|^2}{n}} \dots\dots\dots\dots\dots\dots (ii)$$

The standard deviation is defined as x = a certain value.

- x is the average value.
- n = Number of values in total
- Outliers are taken into account by Z-Score normalization, a data processing method.

Min-Max Normalization:

One of the most prevalent methods of data normalization is normalization from minimum to maximum. The original data undergoes a linear transformation in this data normalization procedure. [16]. The data's minimum and maximum values are retrieved, and each value is replaced using the formula below. [16].

The formula for Min-max normalization is in the picture.

$$v' = \frac{v - \min(A)}{\max(A) - \min(A)} \left(new_{\max(A)} - new_{\min(A)}\right) + new_{\min(A)} \dots\dots\dots\dots\dots\dots (iii)$$

Here,

The final result is denoted by V' after normalization. Here, V means the value of the given data. A is the attribute value.

The absolute values of A's minimum (A) and maximum (A) are the smallest and biggest, respectively. The range's maximum and minimum values are represented by new _max (A) and new _min (B), respectively (A).

Min-max normalization has the advantage of ensuring that all of the features are on the same scale. The disadvantage of Min-max normalization is that, when it comes to dealing with outliers, it is not very efficient.

Log Scaling Normalization:

Log normalization is a method for standardizing anyone's data that can be useful when that person has a particular column with high variance [17]. To condense a vast range into a small one, log scaling calculates the log of a number person's data.

The formula for Log scaling normalization is given below.

$$X' = \log(x) \dots\dots\dots\dots\dots\dots (iv)$$

When a few of someone's values have a lot of points but the rest of their values have a lot of points, log scaling comes in handy [18]. The power law distribution is the name for this form of data distribution. A good example is movie reviews.

Mean Centered Scaling Normalization:

Mean centering means taking all the observed values and subtracting the mean value from the observed values. Mean centered has a mean of 0. This normalization can retain the unit of measure. It can help with multicollinearity.

Implementation and Result Analysis:

We used the Weka [19] tool for our implementation. Weka is a set of data mining-related machine learning techniques. It includes data preparation, categorization, regression, clustering, mining of association rules, and visualization tools [20]. There are many algorithms, such as- Linear regression, Logistic Regression, Decision Tree, SVM, Naive Bayes, KNN, K-Means, and Random Forest are examples of machine learning algorithms. These are the most widely used and well-known machine learning algorithms. Among these, we have used Decision Tree, Random Forest, One R and Naïve Bays algorithms

Table 2: Accuracy rate of the selected algorithms

	Naïve Bayes	Decision Tree	Random Forest	One R
Accuracy	65.6793%	97.1224%	97.7637%	91.7722%

In Table 2, we select four classifiers without normalizing the datasets, and random forest gives us the best accuracy rate.

Table 3: Accuracy rate of the selected algorithms after normalization

	Naïve Bayes	Decision Tree	Random Forest	One R
.Min-Max	65.6793%	97.6624%	91.7722%	97.1392%
Z-Score	65.6034 %	97.6709 %	91.7637 %	95.0886 %
Mean Centered Scaling	65.6034 %	97.6624 %	91.7722 %	96.4641 %
Log Scaling	65.6793 %	97.7637 %	91.7722 %	97.6118 %

Table 2 reveals the error rate of the four algorithm classifiers without applying any kind of normalization methods. Decision Tree algorithm proved to be very accurate with 97.12 percent accuracy whereas Random Forest did supersede all with the highest figure at 97.76 percent. One R algorithm was quite close with 91.77 percent, Naive Bayes performed dismally with 65.68 percent. This finding is in line with what was reported in the past (Yadav, 2015; Gurung et al., 2019) that the use of more advanced techniques is more effective in classifying or separating normal and attack messages in network traffic.

The influence of various normalization methods to algorithm performances is demonstrated by Table 3. As a whole, this promotes minor enhancement in the accuracy of normalization across-the-board. The Z-Score, Mean Centered, and Min-Max techniques produced almost maximum accuracy of Decision tree (97.67%%) and Random Forest (91.76%%). However, the most interesting finding is that the Random Forest algorithm did not fail to be the most accurate classifier when using any of the normalization techniques as well as its accuracy being more than 91%. Z-Score and Mean Centered gave a slight improvement in One R algorithm, whereas Min-Max scaling gave a major improvement to this algorithm and Naive Bayes almost behaved similarly in all the methods thus indicating its impairments to deal with complex patterns in the high dimension.

Conclusion:-

The paper used the NSL-KDD dataset in this study, which corrects some of the flaws of the KDD99 dataset. The NSL-KDD dataset, according to our findings, is ideal for comparing various intrusion detection techniques. We have used four different algorithms for our thesis implementation. The results are analyzed based on various performance of the algorithm. We found that the random forest method provides the best results. It measures the highest possible rate of finding accuracy. We have implemented our work applying Weka Tool. We have analyzed the performance in both way with normalization and without normalization. We can improve the accuracy of the NSL-KDD dataset in order to safeguard the datasets against unusual attack.

Reference:-

1. L. Lv, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," Knowledge-Based Systems, vol. 195, p. 105648, 2020.
2. H. Alazzam, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," Expert Systems with Applications, vol. 148, p. 113249, 2020.
3. "Find Open Datasets and Machine Learning Projects | Kaggle," [Online]. Available: https://www.kaggle.com/datasets?search=intrusion+detection+systems&datasetsOnly=true.
4. Z. A. a. F. M. Iram Abrar, "A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset," Proceedings of the International Conference on Smart Electronics and Communication, 2020.
5. B. I. Anamika Yadav, "Performance analysis of NSL-KDD dataset using ANN," 2015 International Conference on Signal Processing And Communication Engineering Systems (SPACES), January 2015.
6. M. K. G. A. S. Sandeep Gurung, "Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset," International Journal of Computer Network and Information Security, vol. 11, no. 3, pp. 8-14, 2019.
7. "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, 2016.
8. N. K. Sarika Choudhary, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT," International Conference on Computational Intelligence and Data Science (ICCIDS 2019), vol. 167, pp. 1561-1573, 2020.

9. D. V. S. S. Ahmed Mahfouz, "Comparative Analysis of ML Classifiers for Network Intrusion Detection," 27 08 2019.
10. L. Hiranya, "DataDrivenInvestor," 09 march 2020. [Online]. Available: https://medium.datadriveninvestor.com/did-you-know-the-famous-data-set-called-nsl-kdd-293b39420c74.
11. Unb.ca, "NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB," [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.
12. K. E. Y. M. L. H. Safaa LAQTIB, "A Deep Learning Methods for Intrusion Detection Systems based Machine Learning in MANET," International Journal of Future Generation Communication and Networking, vol. 12, no. 2, pp. 55-70, October 2019.
13. "Learning Data Mining with Python - Second Edition," [Online]. Available: https://www.oreilly.com/library/view/learning-data-mining/9781787126787/dbbf3003-dcd4-46d0-b35e-253f12220d23.xhtml.
14. "Data mining normalization | Galaktikasoft," [Online]. Available: https://galaktika-soft.com/blog/data-mining-normalization.html.
15. "Data Normalization in Data Mining - GeeksforGeeks," [Online]. Available: https://www.geeksforgeeks.org/data-normalization-in-data-mining/.
16. "Log normalization | Python," [Online]. Available: https://campus.datacamp.com/courses/preprocessing-for-machine-learning-in-python/standardizing-data?ex=4.
17. "Data Transformation for Numeric features," [Online]. Available: https://medium.com/analytics-vidhya/data-transformation-for-numeric-features-fb16757382c0.
18. Wikipedia,"Weka(machine learning),"[Online]. Available: https://en.wikipedia.org/wiki/Weka_(machine_learning).
19. "Weka 3:Machine Learning Software in Java,"2020.[Online].Available: https://www.cs.waikato.ac.nz/ml/weka/.
20. L. Lv, Knowledge-Based Systems, vol. 195, 2020.
21. Z. Nayeem, T. U. Mahmood, and D. Tenney, “An educational approach to best practices for improving operational analytical data integration success,” in 2025 Northeast Section Conference, Mar. 2025.
22. Z. Nayeem, D. Tenney, and T. U. Mahmood, “The impact of supply chain analytics and artificial intelligence on supply chain management education,” in ASEE North East Section Conference, Apr. 2024.