IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A REVIEW ON INCREASEING THE ACCURACY OF INTRUSION DETECTION SYSTEM (IDS)

[1]Prof. Pritam Ahire, [2]Abhijeet Mowade, [3]Nikhil Bankar

[1]Professor, [2]Student, [3]Student
[1]Computer Engineering,
[1]D.Y. Patil Institute of Engineering and Technology Ambi, Pune, India

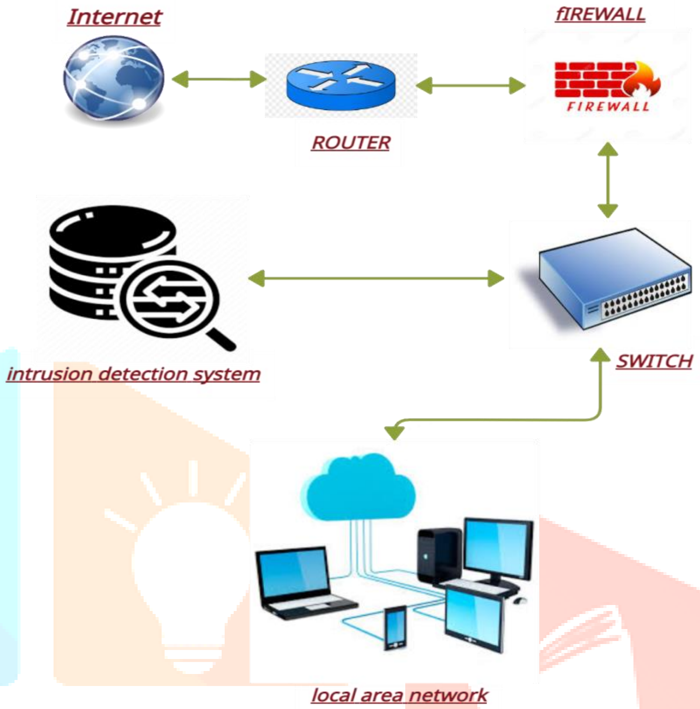
Abstract: In more recent times, there has been an increase in the number of people using computers, as a result of which there is a widespread use of the Internet. The use of the Internet enables hackers to access computers using new, more sophisticated, and more complex forms of attacks, to protect computers from them Intrusion Detection System (IDS) is used, which is trained with few machine learning algorithms along with datasets. The datasets used are collected over a period of time in some networks and usually contain up-to-date data. Furthermore, they are unbalanced and incapable of storing adequate data for all forms of attacks. These uneven and outdated datasets reduce the effectiveness of current IDSs, especially for attacks that are rarely encountered. In this paper, Using Random Forest, Linear Discriminant Analysis, K-Nearest Neighbor, Decision-Tree, Adaboost, and Gradient Boosting algorithms, we suggest six machine-based IDSs. To make IDS more logical, an up-to-date security database, CSE-CIC-IDS2018, is being used in place of older and more widely used datasets. The selected database is also not balanced. As a result, the rate of inequality in the dataset is reduced using a data model called the Synthetic Minority Oversampling Technique (SMOTE) to improve the reliability of the system based on the types of attacks and to eliminate unreliable access and false alarms. Data processing is done for small classes, and their numbers increase to medium data size in this way. Experimental results have shown that the proposed method significantly increases the acquisition rate of the attacks that are rarely encountered.

Index Terms - **machine-learning, intrusion detection, IDS, SMOTE, CSE-CIC-IDS2018 dataset.**

I. INTRODUCTION

As a result of technological developments, much of the real-world transactions have been made online available through the internet in the cyber world. Therefore, banking, shopping, online examinations, electronic commerce, communication, and many such operations are widely used within the internet. With the widespread use of smartphones, people can connect to this global network and perform transactions at any-time and anywhere. While this digitalization facilitates daily human activities, due to the weakness of the servers and the newly emerged network intrusion techniques, networks are often attacked by the attackers who take advantage of the anonymous environment of the Internet not only to steal certain information or money but also to slow down the performance of network services. Security administrators traditionally choose password protection methods, encryption techniques, and access controls in addition to firewalls as a way to protect the network. However, those methods are insufficient for protecting the system. Therefore, many administrators or managers prefer the use of Intrusion Detection Systems (IDSs) to detect malicious attacks by monitoring network traffic, as shown in Fig. 1. Intrusion can be defined as any kind of unauthorized activity that causes damage to confidentiality, availability, or integrity of the data within the information system. IDSs are mostly preferred means of detecting this type of activity. IDS can be divided into three groups: Signature-based Intrusion Detection Systems (SIDS) Systems, Anomaly-based Intrusion Detection Systems (AIDS), and Hybrid Systems. SIDS keeps signatures of malicious activities on a database and attempts to gain access through pattern matching methods. At the moment, AIDS is trying to learn the normal work ethic and set some as suspicious. In this type of system, there is no need to use a signature-base, and the system can target zero-day attacks that have never been experienced before. Hybrid systems are built on a combination of SIDS and AIDS to increase the rate of acquisition of known risky activities by reducing the negative i.e. false positive rate of zero-day attacks. Because of the benefits of AIDS, many existing IDs directly apply or benefit from AIDS contained by a hybrid approach. These IDSs need to be trained by using a machine learning model to process databases. Many of the functions in this article have adopted older data sets, which contain unwanted details and uneven volumes of data types. While we may encounter some data sets containing up-to-date data, the unequal size of data types remains a challenge for researchers. The effectiveness of the IDS is directly related to the selected learning model and the quality of the data sets used. A good quality database can be defined as a database that develops better performance metrics in a real-world interaction. As mentioned in [1], In the case of unequal division, training for one category (minority) far exceeds the training set of another category (minority), where, a minority class is often the most popular category [2] the inequality database presents a problem for investigators. The database is said to be unequal when class allocations are unequal [3]. This is a common problem in many classification problems due to the data sets used. An unequal database result in a split that is used in the general category; however, for most of them, the goal is to try to find a minority class [4] this results in a large error in separating the samples of the minority class and the larger goals that can be missed. To maximize data quality, it should be measured in terms of data types. Therefore, in this paper, we aim to use up-to-date datasets to train IDS to develop a knowledge base for the detection of an anomaly. To improve the efficiency of the system, a comparison task was performed using six different machine learning algorithms. To increase the detection rate of low-sample attacks, a data-generating tool is used, and the results obtained from the current work are compared to those of previous tests.

FIGURE 1: Intrusion Detection System on Local Area Network.



II. Existing System

Intrusion Detection Systems are striking areas not only for cybersecurity research but also for case studies. In the last few years, many papers have been published on this subject. In this section, these notable pieces of research (especially related to unequal data) are briefly discussed. In 2019, Gao et al. used the NSL-KDD dataset to test and develop IDS using a consistent ensemble learning model [5]. They used four different algorithms K Nearest Neighbor, Deep Neural Networks, Random Forest, and Decision Tree. Also, they designed a compatible voting algorithm. They used the NSL-KDD-Test+ file to verify their path. The decision tree algorithm's accuracy is 84.2 % and the flexible algorithm's final accuracy is 85.2 %. Finally, they compared the relevant research papers and found that the consistency of the results is embedded in their integration model. An online Principal Component Analysis (PCA) study designed to address the problem of misdiagnosis is proposed in [6]. Their approach is focused on using online platforms for major issues. Going through the few categories of the target state by oversampling, their proposed algorithm allows them to determine the randomness of the target state. Comparison between PCA and other acquisition algorithms supported the efficiency, and accuracy of the proposed method. Also, their algorithm has reduced computational costs and memory requirements. Yueai and Junjie proposed a two-phase strategy with a load balancing model (such as an online and offline category) using IDS [7]. In the online section, the system has taken packets from the network and then received attacks. Currently, in the offline category, the training database has been used to create an offline model. They used SMOTE to sample and did their classification with AdaBoost and Random Forest algorithms. Their test results showed that SMOTE and AdaBoost were not working properly. Abdulhammed et al. (2019) used the CIDDS-001 database to manage unequal databases to create effective IDS for a variety of strategies [8]. They have successfully studied the CIDDS-001 sample methods and tested this database by voting, Deep Neural Networks, Variational Autoencoder, Random Forest, and stacking learning algorithms. This program received 99.99% accuracy when using unequal datasets.

The hybrid method of IDS with the NSL-KDD database was studied in [9]. The way they work includes a combination of SMOTE, cluster centers, and nearby neighbors. Select the key features using the single break method. K-Fold Cross-validation (K is 10) is used for measurement purposes. Test results showed that suggested the method achieved acceptable accuracy with a low false alarm level, with no significant difference. In 2019, Taher et al. a proposed machine-readable machine learning system for isolating network traffic [10]. They used the NSL-KDD database for testing and training because they wanted to find out if traffic was cruel or normal. For that purpose, they have used Support Vector Machine (SVM) and Artificial Neural Network (ANN) algorithms and feature selection methods. They found that ANN with feature options did better than SVM. Tesfahun and Bursari used SMOTE in the training database and method selection method based on Information Gain in NSL-KDD [11]. This study was conducted to address unequal data on IDS. A random forest algorithm was used to divide the proposed route. The Random Forest algorithm with SMOTE and selected feature selection details worked well in their studies. Chandra et al. improved the hybrid model for 2019 by using the KDD Cup99 database. [12]. They use Filter-Based

Qualification Selection to reduce the size of the database feature. K-Means and Sequential Minimal Optimization algorithms were used to detect data attacks. Their proposed method greatly improves the level of accuracy.

In 2012, Qazi and Raza studied the results of descriptive results to increase the effectiveness of classification [13]. Also, they used oversampling and undersampling to reduce the rate of data inequality. They used SMOTE used for oversampling. They found that in unequal databases, the sampling process was more accurate than SMOTE for subdividing small classes. It was also found that the Decision Tree and Naive Bayes algorithms are more accurate than other algorithms. Al-issa et al. (2019) used the Decision Tree (DT) and the Support Vector Machine (SVM) algorithm to obtain attack signatures using a specific database [14]. The database contained standard profiles and several DoS instances on wireless sensor networks. The results showed that DT received a wrong false positive and high truth positive level than SVM, with DT at 99.86% and SVM with an actual rate of 99.62%, and DT with 0.05%, while the SVM had 0.09%, which is false - negative portion. In 2018, Ahmad et al. conducted comparative research that solved the problems associated with accuracy and related metrics using Random Forest, Support Vector Machine, and Extreme Learning algorithms [15]. The NSL-KDD database was used, which is considered the IDS test standard. The results show that Extreme Learning Algorithm is better than other algorithms with recall, accuracy, and precision.

Table 1: Existing Result

REF	DATASET	ALGORITHMS	RESULT	OS/US
[5]	NSL-KDD	DT, KNN, RF, DNN	AVG = 85.2	NO
[6]	REAL WORLD DATASET	PCA	-	OS
[7]	ONLINE DATA	ADABOOST, RF	-	-
[8]	CIDDS-001	DNN, RF, AE	DNN US = 99.30, DNN OS = 94.27, RF US =99.99, RF OS= 99.99	OS AND US
[9]	NSL-KDD	CANN	CANN = 99.13	OS
[10]	NSL-KDD	SVM, ANN	SVM = 82.34, ANN =83.68	NO
[11]	NSL-KDD	RF	-	OS
[12]	KDD CUP99	K-MEANS+, SMO	AVG = 99.32	NO
[13]	KDD CUP99	DT, NB	-	OS AND US
[14]	KDD CUP99	DT, SVM	DT = 99.86, SVM = 99.62	NO

Comparisons of these related activities were performed in Table 1 by showing the data sets used, the efficiency achieved, and the use of the most widely used methods oversampling (OS) and undersampling (US). The numbers in parentheses are the reference numbers of related activities. It is evident that the subjects mentioned in Table 1 use older data sets such as KDD-Cup99, NSL-KDD, or their databases. This makes finding new attacks a challenge. Systems especially developed with older data sets like KDD-Cup99 and NSL-KDD are not suitable for current attacks. To use the IDS more effectively, up-to-date data is required for use. Besides, many previous IDS applications measure the overall accuracy of the system to show its effectiveness. However, this value does not provide optimal system performance, especially for unequal databases. Therefore, the average accuracy measurement, which gives the same weight to all category types, should be accepted as the main performance metrics.

III. DATASETS

Researchers can use public databases or they can use their own datasets. In the following paragraphs, several selected datasets are mentioned and compared with their content and properties.

A. KDD CUP99

KDD Cup99 was created in 1998 by DARPA to detect network volatility and was used in the 1999 KDD Cup Challenge to test IDS [16], This database is one of the most popular databases in the field of data mining and machine learning. There are about 5 million details in the standard database. About 80% of the data are details of the attack, and the remaining 20% are benign [17]. 41 areas in the database can be grouped under three headings; basic features, traffic features, and content features.

B. NSL-KDD

The NSL-KDD database was created in 2009 to solve problems related to unfamiliar data in the KDD Cup 99 database [18]. The reliability of the systems developed over the years was questioned, as there was no accurate IDS dataset.

C. CIC-IDS2017

CIC-IDS2017 was created in 2017 and features the most recent and practical attacks in the world that year. It was built by evaluating network traffic using time stamp information, IPs for source and destination, ports for source and destination, attacks, and protocols. [19]. 86 network-related features with IP addresses and forms of attacks are included. In accordance with the final database testing framework in 2016, the conditions for establishing a reliable database are determined. Prior to the construction of the CIC-IDS2017 database, no IDS dataset acquisition data met the process of building a reliable database, built-in 2016.

D. CSE-CIC-IDS2018

The outline concept has been used to create a CSE-CIC-IDS2018 database [20]. The most recent data available in 2018/2019 is the Canadian Institute for Cybersecurity. These profiles can be used by agents or individuals to create events on the network and can be used on different network protocols with different approaches. In addition, the database was developed by analyzing the standards used in constructing CIC-IDS2017. In addition to the basic procedures, it offers the following benefits:

This is one of the most recent databases right now. The two profiles were separated, using five methods of attack on the database. The numbers of the benign and attacks are shown in Table 2 below. Also, this table shows IDS Database and its features.

- The number of duplicate data is very low,
- Uncertain information is almost non-existent,
- The database is in CSV format, so it is ready for use without processing.

This is one of the most recent databases right now. The two profiles were separated, using five methods of attack on the database. The numbers of the benign and attacks are shown in Table 2 below. Also, this table shows IDS Database and its features.

TABLE 2: CSE-CIC-IDS2018 data distribution.

CLASS LABEL	NUMBER	VOLUME (%)
Benign	2,856,035	63.111
Bot	286,191	6.324
Brute Force	513	0.011
DoS	1,289,544	28.497
Infiltration	93,063	2.056
SQL injection	53	0.001
TOTAL	4,525,399	100

E. IMBALANCED RATIO OF KNOWN DATASETS

Table 3 shows the number of most popular and preferred datasets, categorized by their classes. As can be seen, these data sets are not equally balanced. In order to accurately calculate the efficiency of the system, this unequal structure is required to be articulated. The imbalanced measure that can be calculated in Equation 1 can be used as a metric.

$$\boldsymbol{Imbalanced\ Ratio = \rho = \frac{max_i\{C_i\}}{min_i\{C_i\}}} \qquad \text{Equation 1}$$

When C_i shows the size of the data in class i. In other words, the rate of inequality can be defined as the fraction between the number of events in the minority (min) class and majority (max) class. According to this statistical estimate, the inequalities (imbalanced ratio) of the most popular and recent databases are listed in Table 4. There is a large gap between data classes affecting system performance. In addition, sophisticated hackers focus on building small data types to achieve their goals. Therefore, to maximize system efficiency, this level of inequality should be reduced.

Table 3: Data sizes of datasets

Dataset	Class-1	Class-2	Class-3	Class-4	Class-5	Class-6
KDD CUP99	4,113,233	553,301	45,268	18,599	112	-
NSL-KDD	77,054	53,387	14,.077	4,833	119	-
CIC-IDS2017	2,358,036	453,438	15,967	1,966	36	21
CSE-CIC-IDS2018	2,856,035	1,289,544	286,191	93,063	513	53

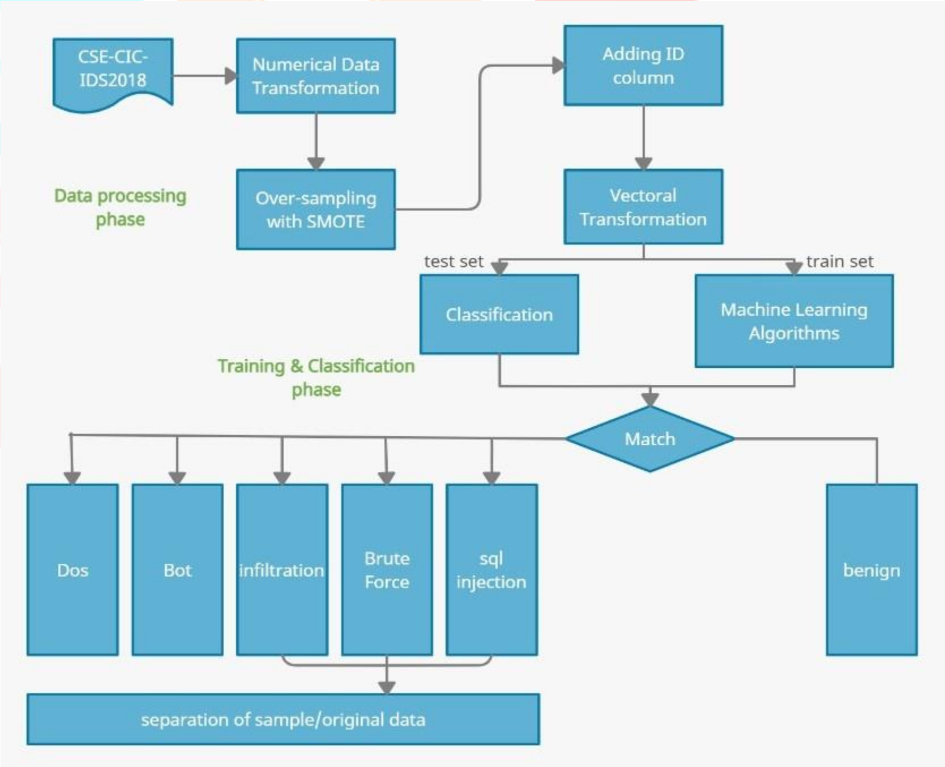
Table 4: Imbalanced Ratio of known Datasets

Dataset	Imbalance Ratio
KDD CUP99	***36,725***
NSL-KDD	***684***
CIC-IDS2017	***112,287***
CSE-CIC-IDS2018	***53,887***

IV. PROPOSED SYSTEM

Many IDS development studies have been conducted over the years, and increasing the accuracy of the acquisition is a very important matrix for developers. However, if the database is unbalanced and a particular category forms the most important part of the database, then the use of accuracy as a single metric is not very acceptable. If there is a large gap between the size of the data within the majority and minority classes, fragmented attackers can focus on the types of attacks of minority types to increase their efficiency. As a result, the focus of this paper is on eliminating the effect of asymmetry between classes in the database by increasing the accuracy of the system. As mentioned earlier, most IDSs are created with the discovery of Anomaly by identifying general data using six machine learning algorithms. Therefore, many useful tools have been developed in the last few decades, and in the meantime, the Python programming language, as one of the most popular areas of development, has become increasingly important for the use of new learning programs. Not only for system improvement but also for testing, present libraries such as Scikit-Learn (Sklearn) provide excellent flexibility and ease of use.

FIGURE 2: Flowchart of the proposed system



V. MACHINE LEARNING ALGORITHMS

A. ADABOOST ALGORITHM

Adaptive Boosting (AdaBoost) is a community learning algorithm, used for classification problems [22]. `` boost-ing '' is the process of achieving a strong result by combining weak results from data. It transmits the information evenly in the first step and then separates. At this stage, it detects a weak separator and restores weight. It focuses on the worst outcome during the recovery process. After a while, it involves several bad classifiers to make a successful separation. Its purpose is to increase its success in classification.

B. DECISION TREE ALGORITHM

Decision Tree (DT) is one of the supervised learning algorithms used for classifying the numerical and class data. It has a pre-defined goal description. It also has leaf nodes that are supported by decision-making steps in order to achieve one of the algorithm structure's topdown goals [23]. It takes advantage of its easy nature to rapidly process large volumes of data. In some cases, the most complex trees have to deal with database fragmentation. In such cases, decision trees become more difficult, and it is difficult to achieve any goals. Another problem in the decision tree algorithms is overfitting. Some leaf nodes are extracted from the decision tree to solve this problem. Information gain and entropy should be calculated for decisive trees.

C. RANDOM FOREST ALGORITHM

Random Forrest (RF) is a type of built-in surveillance system that can be used for regression problems and classification [24]. It is painless to use, and it creates a decision forest through Decision Making and solves a problem in this way. With this, it creates a random collection of trees. During the process, more than one Decision tree is trained to provide the most accurate classes. Most of the time, without using a parameter, it can give really good results. It is one of the most popular methods because it provides instant and accurate results even in mixed, incomplete, and noisy databases.

D. K NEAREST NEIGHBOR ALGORITHM

K Nearest Neighbors (KNN) is a supervised learning algorithm. Unlike other supervised learning algorithms, it does not have a training phase [25]. KNN is implemented using data from the first phase of the sample. The K data is selected, which is the closest neighbor to the new data which must be determined by which sample class to be added. The range of new data to be added to any original sample groups taken from the data showing the K near neighbor property.

E. GRADIENT BOOSTING ALGORITHM

Gradient Boosting Algorithm (GB) is used for classification and regression problems [26]. Similar to the Adaboosting algorithm, models of a decision tree is created by combining weak classification model. The purpose is to increase the gradient by updating the ratings according to the reading level, to achieve lower error values.

F. LINEAR DESCRIMINANT ANALYSIS

LDA is used to reduce the size of the dimensions because it simplifies the calculation, takes steps to classify data in the best possible way, and reduces the absence = overfill problems [27]. Also, LDA can be used to process data before segmentation. It examines the distribution of classification classes and finds differences between average values to create spaces.

G. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

The Synthetic Minority Oversampling Technique (SMOTE) method has been used for the production of synthetic sample data. This method uses the K-nearest neighbor algorithm [21] to create new samples. Two related ADASYN and RandomOverSampler approaches are referred to in the papers. The first, ADASYN, also produces sample data using the KNN algorithm. However, the data generated by ADASYN is not separated from the immediate neighbors, while SMOTE makes no difference. Therefore, it takes longer to produce sample data by ADASYN. In the first test, the gain rate obtained by ADASYN was almost 5% lower than that created by SMOTE. In addition, data processing time was also very long. The ADASYN method was therefore not selected for this study.

VI. EXPERIMENTAL RESULTS

In this study, the effectiveness of machine learning algorithms in testing access procedures is tested, tests were conducted on the most recent dataset available (CSE-CIC-IDS2018). Automatically selected parameters for all algorithms used except KNN. In the KNN algorithm, the number of classes was set to six (one for non-attack, and 5 for attack types). To reduce the variability of performance results due to the random production of trains and test certificates, the K-Fold Cross-Validation method was used in the test. THE selected K value was 5, with training and testing data divided into 80% to 20%. Suggested programs were launched at Keras / Tensorfiow using Python programming language, and Scikit learn libraries. To calculate the effectiveness of the proposed systems; Precision rates, Accuracy, Recall, F1-Score, and Error Rate [28] are used. These metrics are calculated according to Equations 2- 8.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \qquad \textit{Equation 2}$$

$$Accuracy_i = \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i} \qquad \textit{Equation 3}$$

$$Average\ Accuracy = \frac{\sum_{i=1}^{l} \frac{TP_i + TN_i}{TP_I + FN_I + FP_i + TN_i}}{l} \qquad \textit{Equation 4}$$

$$Error\ Rate = 1 - Accuracy \qquad \textit{Equation 5}$$

$$Precision = \sum_{i=1}^{l} \frac{TP_i}{TP_i + FP_i} \qquad \text{Equation 6}$$

$$Recall = \sum_{i=1}^{l} \frac{TP_i}{TP_i + FN_i} \qquad \text{Equation 7}$$

$$F1 - Score = \frac{(\beta^2 + 1)\, Precision * Recall}{\beta^2 Precision + recall} \qquad \text{Equation 8}$$

where TPi is ith True Positive, FPi is ith False Positive, FNi is ith False Negative, l is a multiclass number, and β is a balancing factor. The most common option for fi is 1, which is a harmonic method of recall and precision. The accurately used definition is important because accuracy is the most important metric used to measure the performance of predictive systems. Accuracy often refers to the complete accuracy of the system, however, Accuracyi can also refer to the conscience of each class i. In an unequal database, the final definition of accuracy - which is the average of individual data - is very important to investigators. In this paper, we have used six different machine learning methods such as Random Forest, K Nearest Neighbor, Gradient Boosting, Adaboost, Decision Tree, and Linear Discriminant Analysis. Performance metrics are derived from the original database and the expanded database with sample data on the types of attacks. Like the first metric, accuracy is measured.

As mentioned above, there are some comparisons of the proposed algorithms such as accuracy, time, precision, recall, f1-score. However, to measure the effectiveness of the system, comparing current research and recent work, (published in 2018) its results are shown in Table 5. The current study and comparative study [15] have a single machine - Learning the same algorithm (random forest). The use of sample data leads to a significant increase in the accuracy system, as the 99.34% accuracy rate is measured. A notable difference between the papers is that in this paper instead of using the NSL-KDD dataset, which is not recent, we use CSE-CIC-IDS2018 [21]. Further comparisons with other machine learning algorithms (e.g. SVM, RBF, and ELM), show that trained IDSs work better than other algorithms.

Table 5: Comparison table (*accuracy values are written approximately depending on the referenced paper).

Reference [15]		Normal		Sampled	
Algorithm	Accuracy (%)	Algorithm	Accuracy (%)	Algorithm	Accuracy (%)
		ADA	99.69	ADA	99.60
SVM Lin	98.8	DT	99.66	DT	99.57
SVM RBF	98.3	RF	99.21	RF	99.35
RF	97.7	KNN	98.52	KNN	98.58
ELM	99.5	GB	99.11	GB	99.29
		LDA	90.80	LDA	91.18

VII. CONCLUSION

In recent years, due to the extensive use of the Internet, computer devices can connect to the global network anytime and anywhere. However, an anonymous Internet path leads to many security breaches in the network, leading to hacking. In addition, the current attackers have greatly improved, and with the help of automated production tools, they can create new malwares depending on the weakness of Intrusion Detection Systems (IDSs). IDSs are usually trained using pre-collected data sets. However, almost all of these data sets cannot be measured by different values of inequality, ranging from 648 and 112,287. Unequal data retention leads to bias and classism, and in some rare cases, small classes are overlooked. However, these sub-categories are usually positive categories. Therefore, the level of inequality should be reduced to increase the efficiency of the system and reduce the accuracy. In this paper, the learning models of six different machines (Decision Tree, Random Forest, K Nearest Neighbor, Adaboost, Gradient Boosting, and Linear Discriminant Analysis) were used using the latest database (CSE-CIC-IDS2018). To reduce the inequality rate, a sample data model was used by increasing the data size of the smaller groups. Test results have shown that the models used have a very good level of accuracy compared to the latest literature. The use of sample data resulted in intermediate accuracy of the models increasing between 4.01% and 30.59%. Nowadays, due to the high efficiency of big data, many machine learning programs are being transferred to in-depth learning models. This paper was the study to look at the effectiveness of in-depth learning methods in detecting small sample attacks on timely data sets. Therefore, in-depth learning algorithms should be used in future work. By using a different design approach, the performance of the system is expected to increase.

VIII. REFERENCES

[1] J. M. Johnson and T. M. Khoshgoftaar, ``Survey on deep learning with class imbalance,'' *J. Big Data*, vol. 6, no. 1, p. 27, 2019.

[2] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, ``Classification with class imbalance problem: A review,'' *Int. J. Adv. Soft Comput. Appl.*, vol. 7, no. 3, pp. 176_204, 2015.

[3] F. Provost, ``Machine learning from imbalanced data sets 101,'' in *Proc. AAAI Workshop Imbalanced Data Sets*, Menlo Park, CA, USA: AAAI Press, 2000.

[4] S. Barua, M. M. Islam, X. Yao, and K. Murase, ``MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning,'' *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405_425, Feb. 2014.

[5] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, ``An adaptive ensemble machine learning model for intrusion detection,'' *IEEE Access*, vol. 7, pp. 82512_82521, 2019.

[6] Y.-J. Lee, Y.-R. Yeh, and Y.-C.-F. Wang, ``Anomaly detection via online oversampling principal component analysis,'' *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1460_1470, Jul. 2013.

[7] Z. Yueai and C. Junjie, ``Application of unbalanced data approach to network intrusion detection,'' in *Proc. 1st Int.Workshop Database Technol. Appl.*, Apr. 2009, pp. 140_143.

[8] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Abumallouh, ``Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,'' *IEEE Sens. Lett.*, vol. 3, no. 1, pp. 1_4, Jan. 2019.

[9] M. R. Parsaei, S. M. Rostami, and R. Javidan, ``A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 6, pp. 20_25, 2016.

[10] K. A. Taher, B. Mohammed Yasin Jisan, and M. M. Rahman, ``Network intrusion detection using supervised machine learning technique with feature selection,'' in *Proc. Int. Conf. Robot., Electr. Signal Process. Techn. (ICREST)*, Jan. 2019, pp. 643_646.

[11] A. Tesfahun and D. L. Bhaskari, ``Intrusion detection using random forests classifier with SMOTE and feature reduction,'' in *Proc. Int. Conf. Cloud Ubiquitous Comput. Emerg. Technol.*, Nov. 2013, p. 127_132.

[12] A. Chandra, S. K. Khatri, and R. Simon, ``Filter-based attribute selection approach for intrusion detection using k-means clustering and sequential minimal optimization techniq,'' in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 0_745.

[13] N. Qazi and K. Raza, ``Effect of feature selection, SMOTE and under sampling on class imbalance classification,'' in *Proc. UKSim 14th Int. Conf. Comput. Modeling Simulation*, Mar. 2012, pp. 145_150.

[14] A. I. Al-issa, M. Al-Akhras, M. S. Alsahli, and M. Alawairdhi, ``Using machine learning to detect DoS attacks in wireless sensor networks,'' in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 107_112.

[15] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, ``Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection,'' *IEEE Access*, vol. 6, pp. 33789_33795, 2018.

[16] G. Karatas, O. Demir, and O. Koray Sahingoz, ``Deep learning in intrusion detection systems,'' in *Proc. Int. Congr. Big Data, Deep Learn. Fighting Cyber Terrorism (IBIGDELFT)*, Dec. 2018, pp. 113_116.

[17] G. Karatas and O. K. Sahingoz, ``Neural network based intrusion detection systems with different training functions,'' in *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, Mar. 2018, pp. 1_6.

[18] M. Tavallaee, E. Bagheri,W. Lu, and A. A. Ghorbani, ``A detailed analysis of the KDD CUP 99 data set,'' in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1_6.

[19] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ``An evaluation framework for intrusion detection dataset,'' in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, Dec. 2016, pp. 1_6.

[20] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, ``Toward generating a new intrusion detection dataset and intrusion traffic characterization,'' in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108_116.

[21] R. Sharma, R. K. Singla, and A. Guleria, ``A new labeled ow-based dns dataset for anomaly detection: PUF dataset,'' Procedia Comput. Sci., vol. 132, pp. 1458-1466, 2018.

[22] A. J.Wyner, M. Olson, J. Bleich, and D. Mease, ``Explaining the success of adaboost and random forests as interpolating classifiers,'' *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1558_1590, 2017.

[23] N. Frosst and G. Hinton, ``Distilling a neural network into a soft decision tree,'' 2017, *arXiv:1711.09784*. [Online]. Available: http://arxiv.org/abs/1711.09784

[24] M. Belgiu and L. Dr guµ, ``Random forest in remote sensing: A review of applications and future directions,'' *ISPRS J. Photogram. Remote Sens.*, vol. 114, pp. 24_31, Apr. 2016.

[25] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, ``Efficient kNN classification with different numbers of nearest neighbors,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774_1785, May 2018.

[26] G. Ke, Q. Meng, T. Finley, T.Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, ``LightGBM: A highly efficient gradient boosting decision tree,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146_3154.

[27] L. Wu, C. Shen, and A. V. D. Hengel, ``Deep linear discriminant analysis on Fisher networks: A hybrid architecture for person re-identification,'' *Pattern Recognit.*, vol. 65, pp. 238_250, May 2017.

[28] M. Sokolova and G. Lapalme, ``A systematic analysis of performance measures for classification tasks,'' *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427_437, Jul. 2009.