Efficient Intrusion Detection of Imbalanced Network Traffic using Deep Learning

Mr.Nayan kumar V[1], Mr.Punith kumar M [2], Mr.Tejas Kumar K[3], Mr.Vinod kumar S[4], Dr.Aruna M.G[5]

[1,2,3,4] Students, Department of Computer Science and Engineering ,M.S Engineerinng College, Bangalore, India

[5]Associate Professor, Department of Computer Science and Engineering ,M.S Engineerinng College, Bangalore, India

***Abstract*— In imbalanced network traffic, malicious cyber-attacks can often hide in large amounts of normal data. At Cyberspace, using a high level of encryption and making it difficult for NIDS to ensure accuracy and timeliness. Despite decades of development, IDSs still face challenges in improving in detection accuracy. Deep Learning is a branch of Machine learning, whose performance is remarkable and as a hotspot in field of research. This paper involves both machine learning and Deep learning for intrusion detection in imbalanced network traffic. This process a DSSTE algorithm to avoid imbalance problems. Initially the training set is preprocessed to modify imbalanced data and features are extracted. Then newly obtained training set is fed to a various classification model to evaluate that proposed DSSTE algorithm outperforms other methods.**

Keywords-Random Forest, Alxenett, Lstm, DSSTE algorithm

I.INTRODUCTION

With the rapid development and wide application of 5G, IoT, Cloud Computing, and other technologies, network scale, and real-time traffic become more complex and massive, cyber-attacks have also become complex and diverse, bringing significant challenges to cyberspace security. As the second line of defence behind the firewall, the Network Intrusion Detection System (NIDS) needs to accurately identify malicious network attacks, provide real-time monitoring and dynamic protection measures, and formulate strategies.

The concept of intrusion detection was developed in 1980[1]. However due to limitation of computer storage at that time machine learning failed to attract attention. Thus, many have focused on developing IDSs with higher detection rates and reduce false prediction. Another problem with the existing is they lack ability to detect unknown attacks. Thus, it's necessary to develop IDSs that can detect unknown attacks.

Faced with imbalanced network data, we propose the (DSSTE) Difficult set sampling Technique algorithm to tackle imbalance problem.

This method effectively reduces imbalance and makes classification model learning difficult samples more effectively. We use Machine Learning and Deep Learning models to verify on two benchmark datasets.

II.EXISTING SYSTEM

Pervez proposed a new method for feature selection and classification merging of multi- class NSL-KDD Cup99 dataset using Support Vector Machine (SVM) and discussed the classification accuracy of classifiers under different dimension features[12]. Shiraz studied some new technologies to improve CANN intrusion detection methods' classification performance and evaluated their performance on the NSL-KDD Cup99 dataset. He used the K Farthest Neighbor (KFN) and the K Nearest Neighbor(KNN) to classify the data and used the Second Nearest Neighbor(SNN) of the data when the nearest and farthest neighbors have the same class label. The result shows the CANN detection rate and reduces the failure the alert rate is improved or provides the same performance[13].

Bhattacharya proposed a machine learning model based on hybrid Principal Component Analysis (PCA)-Firefly. The dataset used was the open dataset collected from Kaggle. Firstly, the model performs one key coding for transforming the IDS dataset, then uses the hybrid PCA-Firefly algorithm to reduce the dimension, and the XGBoost algorithm classifies the reduced dataset [14].

A. Class Balancing Methods:

In the realm of machine learning, the topic of category imbalance has long been a source of concern. As a result, intrusion has increased. Detection faces enormous challenges in network traffic as well. Piyasak proposed a method for improving minority classification accuracy .Synthetic Minority Over-sampling Technique (SMOTE) with the Complementary Neural Network to handle unequal data sharing (CMTNN) integrates by this method.

III. PROPOSED SYSTEM

We propose a novel Difficult Set Sampling Technique (DSSTE) algorithm to tackle the class imbalance problem. First, use the Edited Nearest Neighbor (ENN) algorithm to divide the imbalanced training set into the difficult set and the easy set. Next, use the KMeans algorithm to compress the majority samples in the difficult set to reduce the majority. Zoom in and out the minority samples' continuous attributes in the difficult set synthesize new samples to increase the minority number. Finally, the easy set, the compressed set of majority in the difficult, and the minority in the difficult set are combined with its augmentation samples to make up a new training set. The algorithm reduces the imbalance of the original training set and provides targeted data augment for the minority class that needs to learn. It enables the classifier to learn the differences in the training stage better and improve classification performance.

IV.SYSTEM ARCHITECTURE

The system architecture provides a holistic view of the system to be built. It depicts the structure and organization of software components, their properties and the connections between them. The architectural design process is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communications between these components

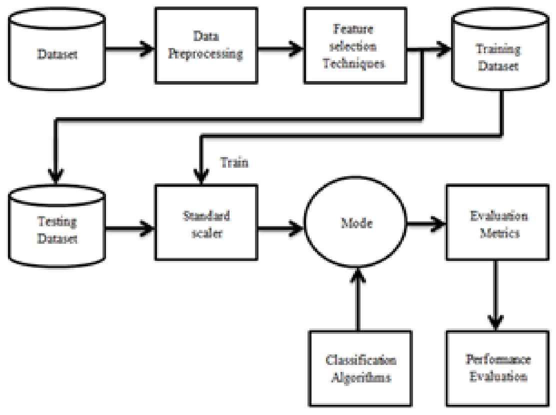

Fig 1: System Architecture

This illustrates the steps in intrusion detection system, Data pre- processing first performed in our intrusion detection structure, including duplicate, outlier, and missing value processing. Then, partitioning the test set and the training set, and the training set processed for data balancing using our proposed DSSTE algorithm. Before modeling, to increase the speed of the convergence, we use Standard Scaler to standardize the data and digitize the sample labels. Finally, the processed training set is used to train the classification model, and then the model is evaluated by the test set.

B. Sequence Diagram.

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think the sequence diagrams are meant exclusively for them.

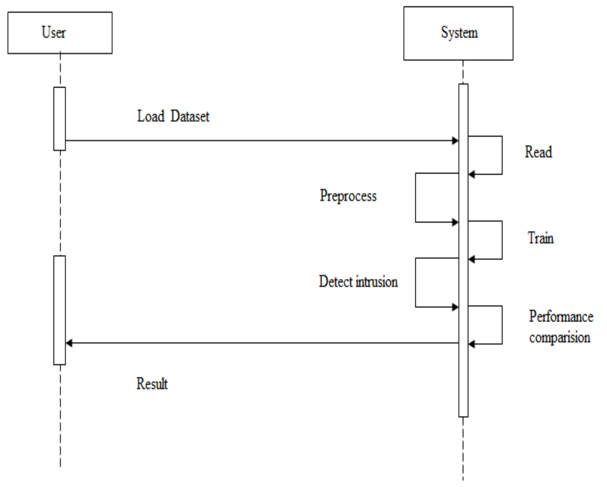

Fig 2: Sequence Diagram.

V.METHODOLOGY

A.DSSTE ALGORITHM:

In unequal network traffic, different types of traffic data have similar presentations, making it difficult for a class divider to understand the differences between them during the training phase. Minority attacks can hide among a large amount of normal traffic, making it difficult for the classifier to learn the differences between them. The majority class in the imbalanced training set's similar samples is redundant noise data. Because the number is much greater than the minority class, the classifier is unable to learn the minority class distribution, so we compress the majority class.

```
Algorithm Difficult Set Sampling(Training set S)
{
//Input: Imbalanced training set S,
//Output: New training set SN
Step1: START
Step2: Distinguish easy set and difficult set
Compute its K nearest neighbors
Remove whose most K nearest neighbors
            SD = S-SE
Loop
Step3: Compress the majority samples in difficult set
Step4: end loop
Step5: Zoom augmentation
           XD1 = XD
           XC1 = XC x (1- 1/n)
           XD2 = XD
           XC2 = XC  (1 + 1/n)
                                           )
SZ append [concat(XD1; XC1; Y ), concat(XD2;  XC2;
Y )]
Step6: return  processed dataset
          }
```

A. Machine Learning And Deeplearning Algorithms.

In the classifier's design, we use Random Forest, Adaboost, XGBoost, LSTM, AlexNet, and KNN to train and test, which are detailed in the following part.

1. RANDOM FOREST: Leo Breiman suggested a random forest in 2001. Random Forest is a supervised learning approach that may be used to model the model to forecast which classification results for a certain type of sample belong to base on data set assets and classification results.

```
Algorithm Random Forest:(Preprocessed data S)
{
//Input: Preprocessed data set
//Output: Accuracy rate
Step 1:Start
Step 2: Importing library from sklearn and tensorflow
Step 3: Build Random Forest
Step 4: Split preprocessed dataset into train set and test
set Step5: Train the modules using fit( )
Step 6: Save training model
}
```

2. XGBOOST: XGBoost is a compatible retreat tree model developed by Chen and Guestrin that combines Boosting with a descent decision tree.

```
Algorithm XGBoost (Preprocessed data S)
{
//Input: Preprocessed data set
//Output: Accuracy rate
Step 1:Start
Step 2: Importing library from sklearn and tensorflow
Step 3: Build XGBoost
Step 4: Split preprocessed dataset into train set and test
set Step5: Train the modules using fit( )
Step 6: Save training model
}
```

3. LONG SHORT-TERM MEMORY(LSTM): The LSTM network is ubiquitous because it can calculate any part of the network that can be calculated with a standard computer, as long as there is a sufficient weight matrix. If there is a decrease in time and an unknown boundary between critical events, the timeline can be split, analyzed, and predicted.

```
Algorithm Long Short-term Memory (Preprocessed data
S) {
//Input: Preprocessed data set
//Output: Accuracy rate
Step 1:Start
Step 2: Importing library from sklearn and tensorflow
Step 3: Build LSTM
 Step 4: Split preprocessed dataset into train set and test
set Step5: Train the modules using fit( )
Step 6: Save training model
}
```

4. ALEXNET: One of the most important learning networks best known is AlexNet. It was first introduced in 2012 by Hinton and his student Alex Krizhevsky. It has an 8-layer deep neural network with 5 layers of conversion and 3 fully connected layers not included in activation and integration layers. Instead of the Sigmoid function, which was commonly used on previous networks, the ReLU function is used as the activation function in the AlexNet convolutional layer. When the neural network is deep, the introduction of ReLU activity overcomes the problem of gradient dispersion. In AlexNet's neural network flexibility, the Max integration method is used for sample reduction.

```
Algorithm Alexnet(Preprocessed data S)
{
//Input: Preprocessed data set
//Output: Accuracy rate
Step 1:Start
Step 2: Importing library from sklearn and tensorflow
```

Step 3: Build Alexnet
Step 4: Split preprocessed dataset into train set and test set Step5: Train the modules using fit( )
Step 6: Save training model
}

VI.EXPERIMENT

In this experiment random forest, Knearest neighbours, XGboost, Long-short term memory, Alexnet, Adaboost. Algorithm is used.

Dataset : We choose NSL-KDD as the benchmark dataset for experiments. NSL-KDD is the most classic dataset in the field of intrusion detection . It is an improvement based on the KDD99 dataset and is reasonably divided into different difficulty levels in the test set. Although it still has some problems and is not a perfect representation of the existing real network, it can still be used as an effective benchmark dataset to help researchers compare different intrusion detection methods. Each sample in NSL-KDD includes 41 features listed in Table 1.

Table 1: Dataset description

Attributes	Description
01-09	Basic features of network connections
10-22	Content-related traffic features
23-31	Time-related traffic features
32-41	Host-based traffic features

B.Equations

The data is suspended, i.e., the Z-Score method, so that the average value of each element is 0 and the standard deviation is 1, converted to a standard deviation distribution, related to the total sample distribution, and each sample point can influence the setting to eliminate the influence of medium size. of indicators and accelerated gradient decline and model integration.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

TP=True positive,

TN= True Negative
FP=False positive,

FN= False Negative
To test the performance of the test model, we use metrics for Accuracy. These test methods show the level of accuracy of the flow recognition and the level of false alarm of the access detection system. The true label and predictive results of the model are divided into four categories: False Negative (FN) negative sample negatively assumed as negative sample. Negative samples are not as well understood as those in False Positive (FP). True Negatives (TN), which is a very negative sample, is so accurately tested. True Positive (TP) positive samples were tested as such.

C. Experimental Result:

In our tests, we first looked at the performance of the editor in a training set that had different features for the reduction of material used in it. The suggested DSSTE method has a K parameter measuring function. When K rises within a certain distance, the number of hard samples also increases, but when K rises above the width, the number of hard samples remains unchanged. However, as K changes, more pressure and less addition to hard samples will increase. In NSL-KDD, we employed different scaling factors K to handle the training set. The suggested six classifiers were tested, and their performance was assessed using the average accuracy of each classifier.
The below Fig 3 shows Metrics value of Random Forest Algorithm.

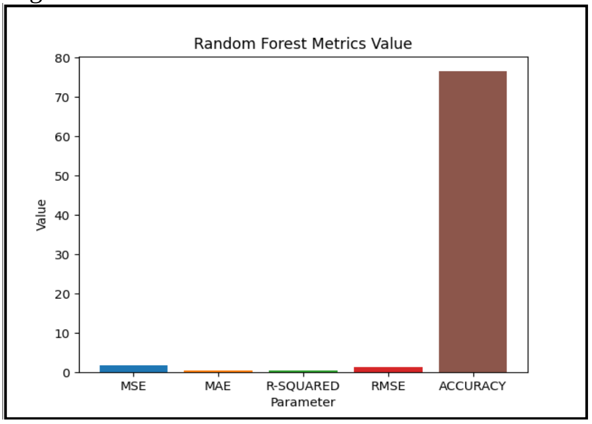

Fig 3: Accuracy of Random algorithm

The below Fig 4 shows Metrics value of XgBoost Algorithm.

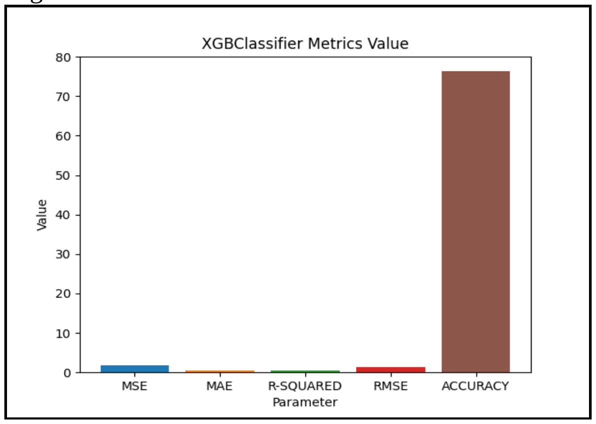

Fig 4: Accuracy of XGboost algorithm

The below Fig 5 shows Metrics value of Alexnett Algorithm.

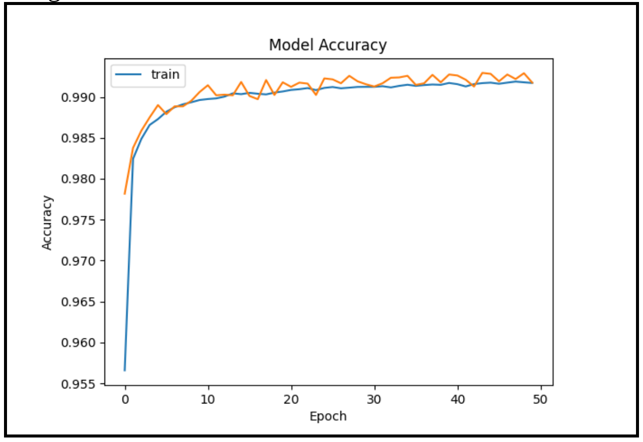

Fig 5: Resultant Graph AlexNett

The below Fig 6 shows Metrics value of LSTM Algorithm.

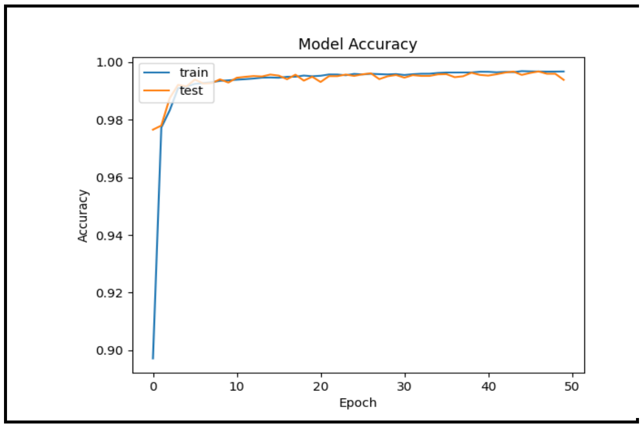

Fig 6 : Resultant Graph LSTM

The below Fig 7 shows Detection Rate of attacks.

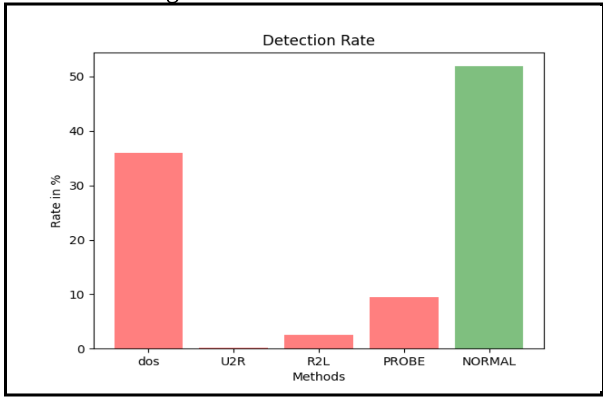

Fig 7: Detection rate of attacks in a data.

Table 2: Developent Environment

Project	Properties
OS	Widows 7 or above
Processor	Intel core i5 @3.3GHZ
Disk	120 GB
Memory	8GB
Framework	Tkinter+Tensorflow

VIII PERFORMANCE EVALUATION

This phase involves the performance and comparative evaluation of Intrusion detection algorithms on imbalanced network traffic. The performance is measured on the accuracy of the algorithm.

Accuracy of the algorithms of Machine learning and deep learning has been plot in the graph. As we can see the deep learning algorithms have upper hand when compared to machine learning algorithms. Alexnet and long short-term memory(LSTM) are deep learning algorithms which gives better accuracy compared to all the other. These evaluation criteria reflect the performance of the intrusion detection system's flow recognition accuracy rate, and false alarm rate. This evaluation shows and helps us understand better how the accuracy can be differ from one to another.

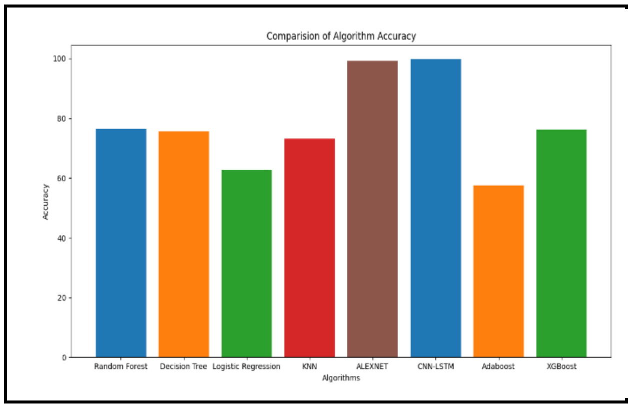

Fig 4:Performance Evaluation.

IX. CONCLUSION

This study introduced a new Difficult Set Sampling Technique (DSSTE) system that allows a separation model to learn from unbalanced network data more effectively. A systematic increase in the number of small samples to be taught can improve class accuracy by reducing network traffic inequalities and enhancing the learning of fewer under the required samples. In machine learning and in-depth reading, we have combined six common classification algorithms with various sample techniques. Experiments show that our approach can accurately measure samples that need to be expanded to unbalanced network traffic. Using the DSSTE approach, a sample of the uneven training set was take., we found that in-depth learning was more effective than a study machine in research. Although neural networks improve data exposure, current public data sets already release data features in advance, making the in-depth reading ability to read pre- processed features and use the default output feature even more limited.

As a result we plan to use methods to speed up inference of pre-trained models, as well developing application for desktop and mobile.

REFERENCE

[1] D. E. Denning, ‘‘An intrusion-detection model,’’ IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. . 1987.

[2] N. B. Amor, S. Benferhat, and Z. Elouedi, ‘‘Naive Bayes vs decision trees in intrusion detection systems,’’ in Proc. ACM Symp. Appl. Comput. (SAC), 2004, pp. 420–424.

[3] M. Panda and M. R. Patra, ‘‘Network intrusion detection using Naive Bayes,’’ Int. J. Comput. Sci. Netw. Secur., vol. 7, no. 12, pp. 258–263, 2007.

[4] M. A. M. Hasan, M. Nasser, B. Pal, and S. Ahmad, ‘‘Support vector machine and random forest modeling for intrusion detection system (IDS),’’ J. Intell. Learn. Syst. Appl., vol. 6, no. 1, pp. 45–52, 2014.

[5] N. Japkowicz, ‘‘The class imbalance problem: Significance and strate gies,’’ in Proc. Int. Conf. Artif. Intell., vol. 56, 2000, pp. 111–117.

[6] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[7] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, ‘‘Deep learning for visual understanding: A review,’’ Neurocomputing, vol. 187, pp. 27–48, Apr. 2016.

[8] T. Young, D. Hazarika, S. Poria, and E. Cambria, ‘‘Recent trends in deep learning based natural language processing [review article],’’ IEEE Comput. Intell. Mag., vol. 13, no. 3, pp. 55–75, Aug. 2018.

[9] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, ‘‘A deep learning approach to network intrusion detection,’’ IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Feb. 2018.

[10] D. A. Cieslak, N. V. Chawla, and A. Striegel, ‘‘Combating imbalance in network intrusion datasets,’’ in Proc. IEEE Int. Conf. Granular Comput., May 2006, pp. 732–737.

[11] M. Zamani and M. Movahedi, ‘‘Machine learning techniques for intru sion detection,’’ 2013, arXiv:1312.2177. [Online]. Available: http://arxiv. org/abs/1312.2177

[12] M. S. Pervez and D. M. Farid, ‘‘Feature selection and intrusion classifi cation in NSL-KDD cup 99 dataset employing SVMs,’’ in Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA), Dec. 2014, pp. 1–6.

[13] H. Shapoorifard and P. Shamsinejad, ‘‘Intrusion detection using a novel hybrid method incorporating an improved KNN,’’ Int. J. Comput. Appl., vol. 173, no. 1, pp. 5–9, Sep. 2017.

[14] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, ‘‘A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU,’’ Electronics, vol. 9, no. 2, p. 219, Jan. 2020.

[15] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, ‘‘A deep learning approach for network intrusion detection system,’’ in Proc. 9th EAI Int. Conf. Bioinspired Inf. Commun. Technol. (Formerly BIONETICS), 2016, pp. 21–26.

[16] P. Torres, C. Catania, S. Garcia, and C. G. Garino, ‘‘An analysis of recurrent neural networks for botnet detection behavior,’’ in Proc. IEEE Biennial Congr. Argentina (ARGENCON), Jun. 2016, pp. 1–6.

[17] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, ‘‘Malware traffic classification using convolutional neural network for representation learning,’’ in Proc. Int. Conf. Inf. Netw. (ICOIN), 2017, pp. 712–717.

[18] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, ‘‘A survey of deep learning-based network anomaly detection,’’ Cluster Comput., vol. 22, pp. 949–961, 2019.

[19] B. A. Tama, M. Comuzzi, and K.-H. Rhee, ‘‘TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system,’’ IEEE Access, vol. 7, pp. 94497–94507, 2019.

[20] P. Jeatrakul, K. W. Wong, and C. C. Fung, ‘‘Classification of imbalaanced data by combining the complementary neural network and smote algorithm,’’ in Proc. Int. Conf. Neural Inf. Process. Springer, 2010, pp. 152–159.

[21] B. Yan and G. Han, ‘‘LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network,’’ Secur. Commun. Netw., vol. 2018, pp. 1–13, Aug. 2018.

[22] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, ‘‘Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,’’ IEEE sensors Lett., vol. 3, no. 1, Jan. 2019, Art. no. 7101404.

[23] P.-J. Chuang and D.-Y. Wu, ‘‘Applying deep learning to balancing network intrusion detection datasets,’’ in Proc. IEEE 11th Int. Conf. Adv. Infocomm Technol. (ICAIT), Oct. 2019, pp. 213–217.

[24] P. Bedi, N. Gupta, and V. Jindal, ‘‘Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network,’’ Procedia Comput. Sci., vol. 171, pp. 780–789, 2020.

[25] L. Breiman, ‘‘Random forests,’’ Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[26] C. Cortes and V. Vapnik, ‘‘Support vector machine,’’ Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[27] L. Feng, L. Yu, L. Xueqiang, and L. Zhuo, ‘‘Research on query topic classification method,’’ Data Anal. Knowl. Discovery, vol. 31, no. 4, pp. 10–17, 2015.

[28] T. Chen and C. Guestrin, ‘‘XGBoost: A scalable tree BOOSTING system,’’ in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

[29] X. Lei and Y. Xie, ‘‘Improved XGBoost model based on genetic algorithm for hypertension recipe recognition,’’ Comput. Sci, vol. 45, pp. 476–481, 2018.

[30] S. Hochreiter and J. Schmidhuber, ‘‘Long short-term memory,’’ Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[31] A. Raghavan, F. D. Troia, and M. Stamp, ‘‘Hidden Markov models with random restarts versus boosting for malware detection,’’ J. Comput. Virol. Hacking Techn., vol. 15, no. 2, pp. 97–107, Jun. 2019.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘Imagenet classification with deep convolutional neural networks,’’ in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.