International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 09 | Sep 2023 www.irjet.net p-ISSN: 2395-0072

Intrusion Detection System using K-Means Clustering and SMOTE

Shrinivas Khedkar[1], Madhura Babhulgaonkar[2]

[1]*Assistant Professor, Dept. of Computer Engineering and IT, VJTI Mumbai, India*
[2]*Student, Dept. of Computer Engineering and IT, VJTI Mumbai, India*

Abstract - *Network intrusion detection serves as a critical line of defense against cyber threats, enabling organizations to safeguard their networks, protect sensitive information, and mitigate financial and reputational risks. Due to the large number of normal samples and only few malicious samples, there is a serious issue of data imbalance in network intrusion detection systems. To solve this problem this work proposes a hybrid sampling technique combining K-Means and Synthetic Minority Oversampling Technique (SMOTE). First, K-Means clustering is applied to handle the outliers, and then SMOTE is used for generation of minority samples. Thus, we get a balanced dataset for training our model. The Random Forest (RF), Convolutional Neural Network (CNN) and other classification models are used for performance analysis. Also the algorithm is compared with DBSCAN + SMOTE hybrid sampling algorithm. The proposed model was implemented on an benchmark NSL-KDD dataset and got an accuracy of 95.53% with RF classifier and of 94.97% with CNN classifier.*

Key Words: Network intrusion detection system, Clustering, K-Means, RF, CNN, SMOTE, DBSCAN.

1.INTRODUCTION

An Intrusion Detection System (IDS) functions as a security solution intended to oversee activities within a network or system, identifying instances of unauthorized or harmful actions, and providing notifications to administrators or security staff about potential risks. Implementing an IDS is vital for upholding the security and reliability of computer networks and systems. The two primary categories of IDS are Network-based Intrusion Detection Systems (NIDS) and Host-based Intrusion Detection Systems (HIDS).

NIDS are deployed at strategic points within a network to monitor and analyze network traffic. They inspect data packets flowing through the network and compare them against known attack patterns or signatures. NIDS are effective in detecting a wide range of network-based attacks, such as DoS (Denial of Service) attacks and network scanning. HIDS operate on individual hosts or endpoints and monitor activities at the system level. They analyze log files, system calls, and other host-related data to detect unauthorized access, file modifications, and abnormal behaviors that could indicate a compromise or attack.

Intrusion Detection Systems (IDS) offer several advantages, including the ability to detect known attack patterns and deviations from normal behavior, facilitating real-time alerts for prompt threat response, enhancing network visibility to identify potential vulnerabilities, and aiding organizations in meeting regulatory compliance. However, IDS are susceptible to generating false positive alerts, potentially leading to alert fatigue and resource wastage. They primarily rely on known attack signatures, limiting their effectiveness against novel threats, and their resource-intensive nature can impact overall system performance. Additionally, skilled attackers can employ evasion techniques to circumvent IDS, and the complexity of implementation and management demands expertise in security and network administration.

This study seeks to enhance the resolution of data imbalance in network intrusion detection with the goal of improving threat detection accuracy, minimizing instances of false negatives, accommodating emerging attack patterns, strengthening overall network security, and pushing forward the advancements in the field of intrusion detection.

To fulfill the above purpose, we proposed a Network Intrusion Detection System combining K-Means Clustering with SMOTE to improve the detection rate. First K-Means Clustering algorithm is used which helps in removing outliers. Then SMOTE is applied to generate minority samples. Thus, a balanced dataset is generated.

The work makes two contributions, outlined as follows:

- This work has proposed a hybrid sampling method combining clustering techniques with SMOTE. Applying SMOTE within clusters solves the problem of outliers and generates a balanced dataset.
- Both Euclidean and Manhattan distance matrices are used for calculating the distances in clustering techniques.

2. RELATED WORK

A deep hierarchical network model that leverages power of CNNs to capture spatial characteristics and BiLSTM to extracts temporal features [1]. However, OSS under sampling technique leads to a loss of important attack data such as U2R and R2L. Novel SMOTE is also prone to oversampling noisy data due to the presence of outliers. Proposed SGM (Synthetic Gaussian Mixture) which is a mixture of SMOTE with Gaussian Mixture model to address data imbalance [2] integrated SGM with CNN to give a flow-based Intrusion Detection System. But SMOTE+GMM achieves similar results

to SMOTE+K-Means while being significantly more computationally expensive, especially when dealing with high-dimensional data or large-scale datasets. Hybrid algorithm that combines the K-means algorithm along with SMOTE oversampling technique [3] proposes an enhanced version of random forest that uses a similarity matrix of network attacks and applying voting processing. An intrusion detection system which incorporates SMOTE oversampling technique which is used for balancing the data distribution at data level [4] and Multiple CNN-IDSs are trained through Bagging and integrated together using the idea of voting at algorithm level. However, Use of novel SMOTE is prone to oversampling noisy data due to presence of outliers. Ensemble of CNNs is extremely computationally expensive and inefficient.

The authors presented idea of utilizing the DBSCAN clustering algorithm for creation of clusters and then the application of oversampling and under-sampling techniques to solve the issue of dataset imbalance problem for classification task [5]. But the determination of the right amount of imbalance ratio of each dataset is very difficult and also it is a difficult task to determine "the best matching classes" for creation of the optimal subset. Also under-sampling is not preferred for balancing network attack data.

The study [6] is the proposal of a novel GAN-based oversampling model for NIDS, addressing the issue of imbalanced normal and malicious network traffic. The model uses a three-step approach of which first is feature extraction, second is data clustering, and third is data generation. Experimental results demonstrate that the model achieves better or comparable results compared to the conventional oversampling method, SMOTE, across multiple datasets and NIDS classifiers. However it is still susceptible to the problem of noise oversampling. The contribution of the paper [7] is the development of the CKSMOTE algorithm, which demonstrates its effectiveness in experiments with 8 datasets from the KEEL repository, although it acknowledges the shortcomings in the selection of distance thresholds.

Paper [8], suggest machine learning methods like rule learning using RIPPER, to improve intrusion detection rates to reduce false positive rates using under-sampling, replication, and synthetic generation and oversampling techniques. This work supports clustering and efficacy of oversampling utilising synthetic generation. A NIDS based on LightGBM and ADASYN oversampling technology based on decision trees to solve the problem of dataset imbalance [9]. Results show a higher detection accuracy rate in comparison to other algorithms. This model reduces training and detection time.

A work in [10], devised new clustering algorithm overcoming limitations of hierarchical clustering based IDS such as Agglomerative and Divisive. Proposed model requires less training time than the latter, and only has to calculate the distance between two samples once. The contribution of the paper [11] is the development of the ECO-Ensemble framework, which effectively optimizes the parameters of CSO and creates a diverse and accurate ensemble of models, resulting in superior performance compared to existing methods.

A work [12], centers on the application of deep learning through a bi-directional LSTM-based IDS model. The efficacy of this system was evaluated using the KDD CUP-99 and UNSW-NB15 datasets. The performance of the bi-directional LSTM model was notably impressive. Additionally, experimentation involved the variation of activation functions within the network. Noteworthy outcomes were observed with softmax and relu activations. When juxtaposed with contemporary methods, the bi-directional LSTM exhibited superior performance, surpassing analogous studies present in the existing literature. In the work by [13], authors introduced an innovative hierarchical neural network named LuNet, which integrates convolutional neural networks (CNN) and recurrent neural networks (RNN). LuNet is designed to process input traffic data in a coordinated way, progressively enhancing its analysis from a broader perspective to more intricate details. This strategy facilitates a more efficient extraction of spatial and temporal features from the data. A research [14] investigated the performance of various resampling techniques when dealing with imbalanced datasets in the realm of network intrusion detection. Their study encompassed six datasets, with three of them (KDD99, UNSW-NB15, and UNSW-NB18) exhibiting high levels of imbalance, while the remaining three (UNSW-NB17) were comparatively less imbalanced. The findings highlighted that oversampling extended the training duration, while undersampling shortened it. This is attributed to the fact that undersampling reduces the instances within the training data, whereas oversampling increases them. Notably, when dealing with extremely imbalanced data, both oversampling and undersampling substantially improved recall. However, in cases where data imbalance was less pronounced, resampling demonstrated minimal influence. The application of resampling, primarily through oversampling, notably led to enhanced detection of minority data (attacks).

In the research conducted [15], they introduced an effective intrusion detection system leveraging the Ensemble Core Vector Machine (CVM) technology. CVMs are founded on the concept of the Minimum Enclosing Ball and possess the capability to identify diverse attack types, including U2R, R2L, Probe, and DoS attacks. The study involves the development of distinct CVM classifiers for each attack category, all of which are trained and evaluated using the KDD CUP-99 dataset. To streamline the process, relevant features for each attack are selected through the Chi-square test, followed by the implementation of a weighted function to reduce dimensionality. [16] Employ clustering techniques such as K-means and Fuzzy C Mean (FCM) to tackle the challenge of a high false positive rate in network intrusion detection

systems. Their objective is to discern false alerts, minimize inaccurate notifications, and enhance the examination of alerts. The dataset utilized for this investigation was generated through interactions with DARPA2000 LLDOS 1.0 via Snort, subsequently enriched with manual annotations.

The above studies NIDS revealed significant gaps, particularly in addressing data imbalance, where imbalanced datasets with skewed class distributions pose challenges. While few studies tackled this, oversampling techniques like SMOTE were found promising, enhancing NIDS sensitivity to rare attacks and reducing false negatives. Combining oversampling with clustering algorithms showed potential benefits, improving dataset balance and the separation of normal and malicious instances, resultantly leading to better detection performance with increased accuracy and reduced false positives. Our objective is to develop an effective technique for solving data imbalance problems in Network Intrusion Detection Systems.

To overcome the limitations of existing approach the proposed work makes use of clustering, sampling, and classifiers to detect intrusions.

3. PROPOSED METHODOLOGY

NIDS is crucial for network security, but insufficient samples can lead to high false detection rates. A proposed algorithm combines network intrusion detection with oversampling using clustering, increasing minority samples for classification. The algorithm classifies attack types. This network intrusion detection model balances traffic data and feature complexity using numerical normalization, oversampling, clustering, and uses detection algorithms. Figure 1 depicts the general framework of the proposed NIDS.

Following models are used in proposed NIDS framework:

SMOTE: To alleviate the class imbalance in a dataset, the SMOTE (Synthetic Minority Over-sampling Technique) is frequently utilized. By combining existing samples, it creates synthetic samples for the minority class.

The fundamental procedures of the SMOTE algorithm can be elucidated in mathematical terms as outlined below:

Step1: Let's assume we have a dataset with two classes: the majority class Cmaj and the minority class Cmin. Calculate the count of synthetic samples to be produced using the parameter Nsynth.

Step 2: For every instance xi within the minority class Cmin :

Identify k nearest neighbors of xi from the minority class (k is user-defined parameter).

Randomly pick one from the k nearest neighbors, denoted as Xnn.

linearly interpolating between xi and xnn. The interpolation is done by selecting a random value between 0 and 1, denoted as λ.

$$Xsynth \leftarrow Xi + \lambda * (Xnn - Xi) \quad (1)$$

Step 3: Repeat steps 2a-2c Nsynth times to generate the desired number of synthetic samples.

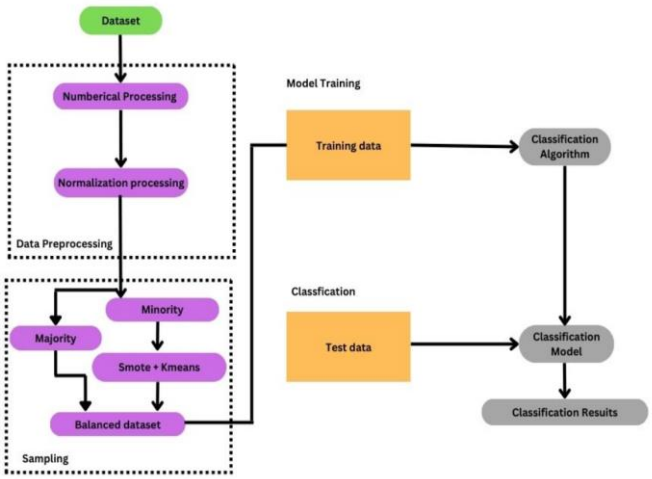

Fig 1: Framework for Network Intrusion Detection Model

K-MEANS: The k-means algorithm, a widely used clustering technique, aims to partition a provided dataset into k clusters, with "k" determined by the user. Through iterative processes, the algorithm minimizes the sum of squared distances between data points and the respective cluster centroids.

The fundamental steps of the k-means algorithm can be summarized as follows:

Step 1: Initialization: Randomly initialize the cluster centroids c.

Step 2: Assignment Step: For every data point xi in the dataset x:

Compute the Euclidean distance between xi and each centroid cj, with j varying from 1 to k.

Assign xi to the cluster associated with the nearest centroid:

$$ri \leftarrow \min(||xi - cj||2) \quad (2)$$

Here, ||xi - cj||2 represents the squared Euclidean distance between xi and cj .

Step 3: For each cluster j from 1 to k: Update the centroid cj by calculating the mean of all data points assigned to cluster j:

$$cj \leftarrow \Sigma(xi , ri \leftarrow j) / |cj| \quad (3)$$

Here, |cj| represents the number of data points assigned to cluster j, and Σ(xi , ri ← j) denotes the sum over all data points xi for which ri equals j.

Step 4: Iterate Steps 2 and 3 until convergence is achieved: Continue the process of performing assignment and updating steps iteratively until either the cluster assignments and centroids exhibit minimal alteration or a predetermined maximum iteration count is attained.

DBSCAN: Data points that are closely packed in high-density areas are grouped by the density-based clustering technique known as DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Amid the noise, it is very useful for locating clusters of any shape.

The mathematical representation of the DBSCAN algorithm can be outlined as follows:

Let's define the following variables:

X: The dataset comprises N data points distributed across a feature space with d dimensions.

ε: Denotes the maximum radius defining the neighborhood around a specific data point.

minPts: Represents the minimum count of data points necessary for the formation of a dense region.

The steps of the DBSCAN algorithm can be summarized as follows:

Step 1: Core Point Identification: For every data point Xi within dataset X: Compute the distance between Xi and all other data points. Count the number of data points within a distance of eps from Xi . If this count is greater than or equal to minPts, mark Xi as a core point.

Step 2: Density-Reachability: For each core point Xi : Find all data points that are within a distance of eps from Xi and form a neighborhood set Ni. If Ni contains fewer than minPts points, mark Xi as a border point.

Step 3: Cluster Formation: For each core point Xi : Create a new cluster C. Add Xi to C. Recursively expand the cluster by adding all density-reachable points from Ni to C.

Step 4: Noise Identification: Assign all border points and unvisited points as noise.

RANDOM FOREST: Random Forest is an ensemble learning technique that amalgamates predictions from numerous decision trees to yield precise forecasts. This algorithm is applicable to both classification and regression assignments. The subsequent equation embodies the essence of the random forest algorithm:

RF fii ← [(Σjεalltrees norm fiij) / T] (4)

Here, RF fii represents the significance of feature i derived from all trees within the Random Forest model. norm fiij signifies the normalized importance of feature i within tree j. T denotes the overall count of trees in the ensemble.

CNN: Convolutional Neural Networks (CNNs) are a category of deep learning architectures widely employed in tasks such as image classification and object detection within the realm of computer vision.

The mathematical expressions that underpin CNNs can be depicted as follows:

Convolution Operation: Given an input image I and a filter F, the convolution operation can be defined as:

$$C_{i,j} \leftarrow \Sigma(F(k, l) * I(i + k, j + l)) \quad (5)$$

for all values of k and l within the filter size.

Activation Function: The Rectified Linear Unit (ReLU) stands as the prevailing activation function utilized in CNNs. Its definition can be expressed as follows:

$$ReLU(x) \leftarrow \max(0, x) \quad (6)$$

However, we used Softmax Activation Function for multiclass classification.

Pooling Operation: Given a feature map F, the max pooling operation can be defined as:

$$P_{i,j} \leftarrow \max(F_{k,l}) \quad (7)$$

for all values of k and l within the pooling window.

Fully Connected Layers: Given an input vector x and weight matrix W, the output of a fully connected layer can be calculated as:

$$Y = W * x + b \quad (8)$$

Where Y is the output vector and b is the bias vector.

Loss Function and Optimization: The optimization procedure entails computing gradients of the loss function concerning the network parameters and subsequently updating these parameters through iterative steps.

Features of Proposed System:

Clustering: First K-Means clustering is employed to cluster the training data so that outliers can be removed.

Oversampling: SMOTE is used to generate the minority samples within each cluster. Clustered-SMOTE prevents the generation of noisy data i.e. outliers and we get a balanced dataset.

Classification: Different classification techniques such as Random Forest, CNN, CNN-LSTM are used to evaluate the results.

4. EXPERIMENTAL SETUP

4.1 Dataset Description

We take into account the widely used NSL-KDD intrusion detection dataset, which is well-known in the intrusion detection industry and has shown to be excellent for evaluating various intrusion detection algorithms. Each intrusion record in the data set includes a feature consisting of 42 dimensions that is broken down into a digital feature of 38 dimensions, a 3-dimensional symbol feature, and a label of type traffic . The label mostly includes normal information as well as four other attack data kinds (Dos, U2R, R2L, Probe). The NSL-KDD dataset includes both the training set (KDDTrain +) and the test set (KDDTest +) which are utilised as the model's training set and test set, respectively, in the experiments of this work. Table 1 shows the numbers related to each category in train, validation and test data and the training data distribution among different categories before and after performing oversampling.

Table 1: NSL-KDD Dataset classes with number of samples before and after sampling

Categories	Original Train data samples	After sampling train data	Validation	Test
Normal	58748	58748	8135	10349
DoS	40154	50042	6090	7143
U2R	46	55939	52	21
R2L	870	22403	2249	583
Probe	10182	50042	1991	1904

4.2 Data Preprocessing

4.2.1 Numerical Processing

Considering that the model takes in a digital matrix as input, the approach of one-hot encoding is employed to convert data with symbolic attributes within the dataset into digital feature vectors. This encoding strategy is mainly directed at three specific attributes in the dataset: service, protocol_type, and flag. These attributes are independently subjected to one-hot encoding and encompass 70, 3, and 11 symbol properties respectively. To illustrate, in the case of the "protocol_type" attribute, TCP, UDP, and ICMP are represented as binary vectors (1,0,0), (0,1,0), and (0,0,1) respectively within the NSL-KDD dataset.

4.2.2 Normalization Processing

The continuous feature data within the dataset exhibits a significantly broader range of values compared to other data types. For example, the "num shells" feature encompasses a value range of [0, 5], whereas the "num root" feature within the NSL-KDD dataset spans a value range of [0, 7468]. To facilitate computations and streamline dimensionality, a normalization technique is adopted. This approach ensures uniform and linear transformation of the range for each feature, effectively constraining it within the range of [0, 1]. The normalization calculation is carried out using the following formula:

$$\chi\, norm = \frac{\chi - \chi min}{\chi max - \chi min} \quad (9)$$

Here, the variables are defined as follows:

- χnorm denotes the normalized value of χ,

- χ represents the original value,

- χmin stands for the minimum value within the range of χ,

- χmax corresponds to the maximum value within the range of χ.

4.2.3 Oversampling with Clustering

The issue of imbalanced dataset where one class is much less represented than another class is addressed by employing oversampling. In our experiment, we used a popular oversampling technique called SMOTE. SMOTE helps address the issue by creating artificial samples of the minority class. However, SMOTE can encounter problems when there are outliers or noisy data points in the dataset. To overcome this, SMOTE can be combined with clustering algorithms like k-means or DBSCAN. These algorithms group similar instances together based on their features or distances. Outliers, which are significantly different from the majority of instances, are often placed in separate clusters or considered as noise. By using clustering algorithms, we can identify secure areas where oversampling can be applied effectively, while avoiding the generation of noisy samples. This helps in improving the quality of the oversampling process and ensures better representation of the minority class.

4.2.4 Classification

By assisting in the identification of attacks, classification models play a significant role in network intrusion detection systems (NIDS). These models are trained with the intention of monitoring network traffic data and classifying

occurrences as either intrusive or normal based on patterns and traits they have acquired. These models are trained using labelled training data, which teaches them the patterns and correlations between input characteristics and the associated class labels. Advanced classification models like Random Forest, Convolutional Neural Networks (CNN), and an integration of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) have been employed in our methodology. Given their success in identifying and categorising network intrusions, these models are very well suited for developing network intrusion detection systems.

4.2.5 Evaluation Metrics

This research employs vital metrics, including Accuracy, Recall, Precision, and F1-Measure, to assess the model's performance. These metrics are determined based on the confusion matrix's four core characteristics.

Data that the model properly identifies as an attack is referred to as True Positive (TP).

False Positive (FP) describes regular data that the model mistakenly interprets as an assault.

True Negative (TN) this form of negative pertains to normal data that the model accurately recognizes as normal.

False Negative (FN) is a term used to describe attack data that the model mistakenly interprets as normal.

These techniques will be used to evaluate the effectiveness of our suggested approach in properly identifying and categorising assaults.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

$$F1-Measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (13)$$

5 RESULTS AND DISCUSSIONS

The results segment of this paper offers an extensive examination and juxtaposition between the suggested model and prevailing models in the domain of network intrusion detection. The primary objective of this section is to appraise and quantify the proposed model's ability, efficacy, and efficiency in identifying and categorizing network attacks, relative to well-established methodologies. Through a rigorous assessment, this endeavor aims to uncover the advantages and limitations of the proposed model and provide insights into its viability for real-world implementation.

Table 2: Different classifiers comparison on NSL-KDD for the proposed model

Category	Precision				Recall				F1-Score			
	RF	CNN	CNN-BiLSTM	CNN-LSTM	RF	CNN	CNN-BiLSTM	CNN-LSTM	RF	CNN	CNN-BiLSTM	CNN-LSTM
Normal	0.93	0.94	0.94	0.94	**0.99**	0.98	0.95	0.98	0.96	0.96	0.94	0.96
DoS	0.99	0.99	0.97	0.99	**0.97**	0.96	0.96	0.95	**0.98**	0.97	0.96	0.97
Probe	**0.98**	0.91	0.86	0.87	0.90	0.92	0.90	**0.93**	**0.94**	0.91	0.88	0.90
R2L	**0.97**	0.85	0.47	0.77	0.30	0.32	0.29	**0.38**	0.46	0.47	0.36	**0.51**
U2R	**0.86**	0.43	0.20	0.13	0.29	**0.71**	0.76	0.52	0.43	**0.54**	0.31	0.21

Table 2 summarizes the performance of classification algorithms which are RF, CNN, CNN-BiLSTM and CNN-LSTM on the proposed model. In IDS recall is important as it considers false negatives. The recall and F1-Score of CNN is highest for the minority classes. From the results, we can say that deep learning techniques have lower false detection rates than machine learning techniques. For normal traffic, RF has a precision of 93% and other algorithms perform equally well. For DOS attack, Except CNN-BiLSTM others has a precision of 99%, and CNN-BiLSTM has a precision of 97%. For Probe attack, RF outperforms other with a precision of 98%. For U2R and R2L attack, Random Forest has a precision of 97% and86% respectively whereas other algorithms are not even close to RF.

For recall value that is false negative rate of normal traffic, DOS attack, and Probe attacks is considerably similar and above 90% in all algorithms however the minority attacks R2L and U2R false negative rate is 38% and 71% respectively using CNN-LSTM and CNN only.

The significantly import performance metric F1-score for normal traffic with all algorithms gives more than 94% accuracy. For DOS attacks highest F1-score is using Random Forest of 98% and lowest using CNN-BiLSTM of 96%. For Probe attacks, Random Forest has an F1-Score of 94%, CNN

has an F1-Score of 91%, CNN-BiLSTM has an F1-Score of 88% and CNN-LSTM has an F1-Score of 90%. For R2L attack, CNN-BiLSTM has a F1-Score of 36%, lowest among all and CNN-LSTM has a F1-Score of 51% which is recorded as higher among all. Similarly for U2R attack, Random Forest has a F1-Score of 43%, CNN has an F1-Score of 54%, CNN-LSTM has an F1-Score of 31% and CNN-LSTM has an F1-Score of 21%. From these results we can understand that the detection rate in R2L and U2R is significantly lower than the detection rate of other attack classes.

To improve the detection rate of classification algorithms we applied K-means and DBSCAN clustering mechanism using Euclidean and manhattan distance metrics to find the best clusters. The clustered data samples are augmented using Synthetic Minority oversampling Technique on minority classes. From table 3, clearly the performance of the proposed model i.e. K-Means clustering + SMOTE using Euclidean distance is better than other techniques. The accuracy and F1-Score of K-Means + SMOTE is higher than the accuracy and F1-Score of DBSCAN + SMOTE and only SMOTE. Using Random Forest classification on the result of K-Means+SMOTE with Euclidean distance gives best performance with 95.53% accuracy. For CNN, K-Means+SMOTE with Euclidean distance, K-Means+SMOTE with Euclidean distance gives best results with 94.97% accuracy. Using CNN-LSTM, K-Means+SMOTE with Euclidean distance gives best performance with 94.45% accuracy.

Table 3: Performance of classification and clustering techniques on Euclidean distance and Manhattan distance measures in clusters (Acc: Accuracy, F1Score)

Classification Algorithms	K-Means+SMOTE (Euclidean)		K-Means+SMOTE (Manhattan)		DBSCAN+SMOTE (Euclidean)		DBSCAN+SMOTE (Manhattan)		SMOTE	
	Acc	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc	F1-Score
RF	**95.53**	**95.01**	95.40	94.89	95.34	94.81	95.14	94.58	95.04	94.48
CNN	**94.97**	94.55	94.91	94.57	94.92	**94.64**	94.56	94.52	94.75	94.44
CNN-LSTM	**94.45**	**94.27**	93.97	93.69	94.20	94.09	93.96	94.02	93.58	93.55

As shown in Figure 2, Random Forest, K-Means+SMOTE with Euclidean distance and K-Means+SMOTE with Manhattan distance gives best performance with 95.6% precision. For CNN, K-Means+SMOTE with Euclidean distance gives best results with 94.9% precision. CNN-LSTM with K-Means+SMOTE using Euclidean distance gives best performance of 94.5% precision. Random Forest on K-Means+SMOTE using Euclidean distance gives best performance with 95% F1-Score. For CNN, K-Means+SMOTE with Euclidean distance give best results with 94.5% F1-Score.

CNN-LSTM on K-Means+SMOTE using Euclidean distance gives best performance with 94.2% F1-Score. For Random Forest on K-Means+SMOTE using Euclidean distance gives best performance with 95.5% recall. For CNN, K-Means + SMOTE with Euclidean distance, K-Means+SMOTE with Manhattan distance, DBSCAN+SMOTE with Euclidean distance give best results with 94.9% recall. For CNN-LSTM on K-Means+SMOTE using Euclidean distance gives best performance with 94.4% recall.

6 CONCLUSION

This work introduces a Network Intrusion Detection System that incorporates the SMOTE oversampling technique in conjunction with the K-means clustering method to mitigate outlier issues. This approach proves significantly potent in addressing data imbalance concerns, and the outcomes have been truly impressive. The proposed model achieves an accuracy rate of 95.53% with the RF classifier, 94.97% with CNN, and 94.45% with CNN-LSTM. Assuming the balanced parameter to identify the attacks due to minority samples present in the imbalanced dataset is as F1-score, the models are evaluated. The F1-score 95.01 of Random Forest classifier is highest among all other algorithms when dataset is clustered using K-Means and sampled using SMOTE. Experimentation indicates that the CNN performs well after RF when we cluster the dataset using DBSCAN and apply sampling using SMOTE with 94.64%.

The future scope of this work includes a few potential areas of improvement and expansion such as developing strategies

for real-time implementation of the proposed model. Explore learning techniques that allow the model to adapt and update itself continuously as new data becomes available and acquiring and incorporating more diverse and comprehensive datasets for training and evaluation purposes

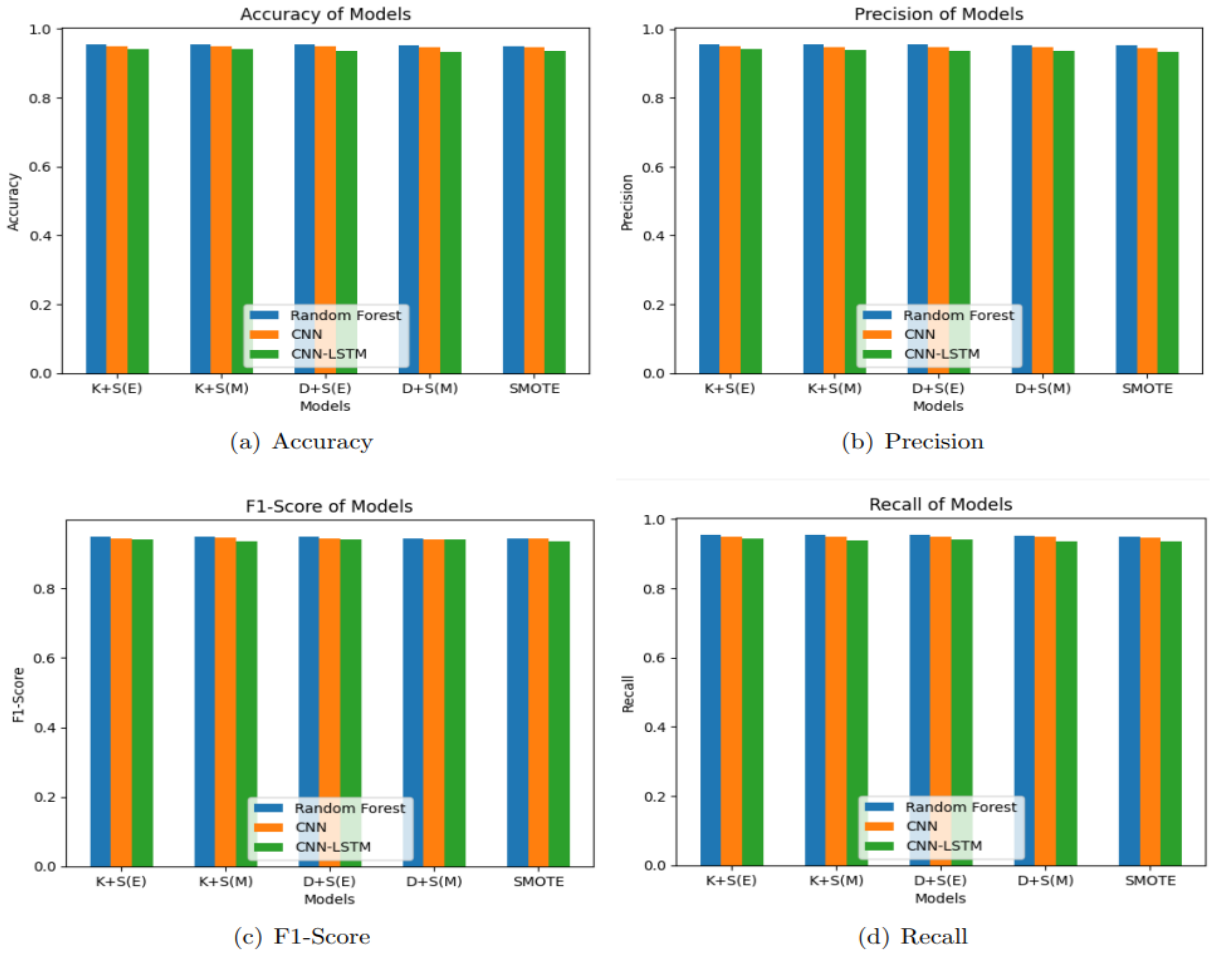

Fig 2: Classification using K-Means+SMOTE and DBSCAN+SMOTE on Euclidean distance and Manhattan distance on NSL KDD data

REFERENCES

[1] Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. Computers & Security, 106, 102289.

[2] Zhang, H., Huang, L., Wu, C. Q., & Li, Z. (2020). An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. Computer Networks, 177, 107315.

[3] Wu, T., Fan, H., Zhu, H., You, C., Zhou, H., & Huang, X. (2022). Intrusion detection system combined enhanced random forest with SMOTE algorithm. EURASIP Journal on Advances in Signal Processing, 2022(1), 1-20.

[4] Tian, L., & Lu, Y. (2021, February). An intrusion detection model based on SMOTE and convolutional neural network ensemble. In Journal of Physics: Conference Series (Vol. 1828, No. 1, p. 012024). IOP Publishing.

[5] Verma, M. K., Xaxa, D. K., & Verma, S. (2017, April). DBCS: density based cluster sampling for solving imbalanced classification problem. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (Vol. 1, pp. 156-161). IEEE.

[6] Li, D., Kotani, D., & Okabe, Y. (2020, July). Improving attack detection performance in NIDS using GAN. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 817-825). IEEE.

[7] Guo, C., Ma, Y., Xu, Z., Cao, M., & Yao, Q. (2019, November). An improved oversampling method for imbalanced data–SMOTE based on Canopy and K-means.

In 2019 Chinese automation congress (CAC) (pp. 1467-1469). IEEE.

[8] Cieslak, D. A., Chawla, N. V., & Striegel, A. (2006, May). Combating imbalance in network intrusion datasets. In GrC (pp. 732-737).

[9] Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. Computers & Security, 106, 102289.

[10] Wei, L., Zhong-Ming, Y., Ya-Ping, C., & Bin, Z. (2017, July). A clustering algorithm oriented to intrusion detection. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (Vol. 1, pp. 862-865). IEEE.

[11] Lim, P., Goh, C. K., & Tan, K. C. (2016). Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning. IEEE transactions on cybernetics, 47(9), 2850-2861.

[12] Pooja, T. S., & Shrinivasacharya, P. (2021). Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cyber security. Global Transitions Proceedings, 2(2), 448-454.

[13] Wu, P., & Guo, H. (2022, December). Holmes: An efficient and lightweight semantic based anomalous email detector. In 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (pp. 1360-1367). IEEE.

[14] Bagui, S., & Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data, 8(1), 1-41.

[15] Divyasree, T. H., & Sherly, K. K. (2018). A network intrusion detection system based on ensemble CVM using efficient feature selection approach. Procedia computer science, 143, 442-449.

[16] Hu, L., Li, T., Xie, N., & Hu, J. (2015, August). False positive elimination in intrusion detection based on clustering. In 2015 12th International conference on fuzzy systems and knowledge discovery (FSKD) (pp. 519-523). IEEE