Harnessing Decision Tree-guided Dynamic Oversampling for Intrusion Detection

Ritinder Kaur
SCA, Manav Rachna International Institute of Research & Studies, India
ritinderkaur.sgtbimit@gmail.com (corresponding author)

Neha Gupta
SCA, Manav Rachna International Institute of Research & Studies, India
neha.sca@mriu.edu.in

Received: 27 June 2024 | Revised: 26 July 2024 and 12 August 2024 | Accepted: 18 August 2024

 DOI: https://doi.org/10.48084/etasr.8244

ABSTRACT

Imbalanced datasets present a significant challenge in the realm of intrusion detection, as the rare attacks are often overshadowed by the normal instances. To tackle this issue, it is essential to utilize the various strategies of imbalanced learning that aim to mitigate the effects of class imbalance and improve the performance of intrusion detection systems. One effective approach for dealing with class imbalance is through data augmentation methods like the Synthetic Minority Oversampling Technique (SMOTE). This research presents a novel data resampling approach that performs adaptive synthetic sampling on rare and complex samples by using decision boundaries. The benchmark dataset NSL-KDD was used to evaluate and validate the effectiveness of this approach. The experimental results demonstrated a significant improvement in the detection accuracy of rare classes, achieving 42% for u2r instances and 83% for r2l instances.

Keywords-intrusion detection; imbalanced learning; oversampling; NSL-KDD

I. INTRODUCTION

The rapid advancements in technology and the increasing reliance on networked computer systems have led to a surge in cyber threats and intrusions. To counter these evolving attacks, Intrusion Detection Systems (IDSs) play a crucial role in safeguarding computer networks. However, traditional IDS methods face challenges in detecting novel and sophisticated attacks, particularly when dealing with imbalanced network traffic. Data imbalance is a prevalent and critical issue in intrusion detection, posing significant challenges for Machine Learning algorithms. The imbalanced distribution of classes in intrusion detection datasets makes it difficult for ML algorithms to accurately classify and detect potential intrusions [1]. As a result, ML algorithms may struggle to identify and predict the minority class accurately, leading to a higher false-negative rate and decreased overall performance. This issue is particularly problematic in real-world applications, where the minority class represents rare events such as network attacks or unauthorized access attempts. Imbalanced learning in IDSs has emerged as a promising approach to address this issue by improving the detection rate of minority classes [2]. To address the problem of data imbalance in intrusion detection and improve the performance of ML models, several approaches can be employed. They can be categorized as:

A. Data-driven Approaches

These techniques focus on modifying the training data to rebalance the class distribution. It can be implemented either by generating synthetic samples through oversampling or reducing the majority class instances through undersampling. Some popular oversampling approaches are Random Oversampling, SMOTE (Synthetic Minority Over-sampling Technique), or ADASYN (Adaptive Synthetic Sampling). Random Undersampling and Tomek Links are examples of undersampling techniques. Though these techniques are simple and easy to implement, they may lead to overfitting and redundancy if not applied judiciously [3].

B. Algorithm - based Approaches

These techniques modify the learning algorithm itself to handle class imbalance by making the model more sensitive to minority class instances. They can be implemented by assigning misclassification costs to classes through cost-sensitive learning or by adjusting the classification threshold to favor minority class instances. Ensemble methods are also popular [4] and blend multiple classifiers to improve classification performance, such as Bagging, Boosting, and Random Forest. This method is applicable to any algorithm and does not require data modification but it does not perform well if imbalance is severe and also suffers from high computational costs.

C. Hybrid Approaches

Hybrid approaches combine data-driven and algorithm-based techniques to leverage their respective advantages. Some examples of hybrid techniques include SMOTEBoost (SMOTE + Boosting) [5], RUSBoost (Random Undersampling + Boosting algorithm) [6] and SMOTE-ENN (Edited Nearest Neighbor undersampling + Random Sampling) [7]. Empirical studies prove that oversampling performs better than undersampling for classification [8-9]. It preserves information present in the rare class, avoids discarding valuable information, and increases model generalization over unseen data that leads to robust models prepared over diverse sets of examples. SMOTE [10] is the most computational demanding oversampling algorithm that initially identifies the minority class instances and then determines the required quantity of synthetic samples to mitigate imbalance. For each minority sample, the algorithm determines its k-nearest neighbors and then generates synthetic samples by interpolating between the original sample and its selected neighbors. The interpolation procedure integrates a random proportion of the distance between the instance and its neighbor into the original instance. This algorithm iterates until the desired number of synthetic samples is produced. SMOTE facilitates balancing of the class distribution, thereby diminishing the risk of overfitting and enhancing model performance, albeit with potential class overlap. While SMOTE is a widely used technique for addressing imbalanced data, it suffers from the following limitations [11]:

1. As synthetic samples are generated by interpolation between existing instances within the feature space, it may generate unrealistic samples.
2. Linear assumption of decision boundaries may not apply to complex datasets, hindering capture of intricate patterns and relationships within the data.
3. When synthetic samples are generated using nearest neighbors, noise from the nearest neighbors can also be introduced.

Over the years, SMOTE has been extensively researched and many variants of SMOTE have proposed to mitigate these limitations and improve the performance of the imbalanced data classification task [12]. Borderline-SMOTE [13] focuses on generating synthetic samples near the decision boundary of the minority class to address the issue of misclassification of borderline instances. The ADASYN (Adaptive Synthetic Sampling) variant also adjusts the distribution of synthetic samples based on the density of minority class instances emphasizing the regions with fewer instances [14]. Safe-Level SMOTE [15] is an enhanced version of SMOTE that considers safe-level ratio of minority instances to generate synthetic samples, aiming to avoid noisy and ambiguous samples. SVM-SMOTE [16] combines Support Vector Machines (SVMs) with SMOTE to generate synthetic samples along the decision boundary, improving classifier's ability to detect minority instances accurately. Authors in [17] proposed a novel oversampling approach based on decision boundary computation, utilizing boundary area and neighboring space to generate new synthetic data points, which performed better when tested against existing methods. Recent research in SMOTE variations incorporates neural and nature-inspired algorithms to improve effectiveness by utilizing attention mechanisms from neural networks to generate synthetic samples near minority class instances, such as attention-based SMOTE [18]. In addition, federal-based SMOTE [19] and meta-learning based SMOTE [20] have been researched in distributed databases to address class imbalances and enhance minority class representation from multiple sources to impact decision boundaries. Table I discusses the features and limitations of some decision-based smote variants.

TABLE I. DECISION-BASED SMOTE VARIANTS

Variant	Features	Limitations
Borderline SMOTE [13] (2005)	Addresses the class imbalance problem by generating synthetic samples near the decision boundary.	May generate noisy samples if the decision boundary is not well-defined.
Safe-Level SMOTE [15] (2009)	Generates synthetic samples based on the safety level of the majority class samples.	May not work well in datasets with complex decision boundaries.
ADASYN [14] (2008)	Increases the density of the minority class by generating synthetic samples in proportion to the degree of imbalance.	May generate noisy samples if the decision boundary is not well-defined.
MWMOTE [21] (2012)	Generates synthetic samples based on the density of the minority class and the distance to the majority class samples.	Variable selection required to avoid generating synthetic samples in irrelevant feature spaces
G-SMOTE [22] (2019)	Generates synthetic samples based on the density and gradient of the data distribution.	May generate noisy samples if the decision boundary is not well-defined.
KMeans-SMOTE [23] (2018)	Generates synthetic samples based on a clustering algorithm.	Low performance in datasets with complex decision boundaries.
SMOTEBoost [5] (2003)	Generates synthetic samples that improve classification accuracy adjusting the decision boundary to focus on correctly classifying minority samples.	Depends on underlying weak classifier, is sensitive to noisy data, and faces challenges with continuous data.
Smote-DL [24] (2021)	Focuses on decision boundaries within the deeper layers of neural networks to capture the complex decision boundaries.	Sensitive to deep learning model's architecture. Is computationally intensive.
SVM-Smote [16] (2012)	More informed oversampling with hyperplane information from SVM.	If the decision boundary is highly nonlinear and complex, intricate patterns may not be captured effectively.
Attention-based SMOTE [18] (2022)	Utilizes attention mechanisms to identify critical instances near the decision boundary of the minority class for generating high-quality synthetic samples.	Quality of attention mechanisms effects correct identification of crucial instances. Computationally intensive.
Federal-based SMOTE [19] (2022)	Implements SMOTE principles in a federated learning setting, generating synthetic samples collaboratively across multiple data sources, aiming to address class imbalance while maintaining data privacy and decentralization.	Synchronization and coordination among multiple entities might be complex. The efficacy might depend on the diversity and quality of participating data sources.

The effectiveness of decision-based SMOTE variants heavily relies on accurate decision boundaries. In some cases, decision-based SMOTE variants may require variable selection to be effective. For instance, SMOTE-DL [24] classifiers without variable selection can bias the classification towards the minority class. Therefore, it is important to consider variable selection in high-dimensional data scenarios.

II. GAPS IDENTIFIED

The studies in previous section indicate that decision-based SMOTE variants offer several advantages over traditional SMOTE. By incorporating decision boundaries and classification information, these variants can better capture the underlying structure of the data and generate more discriminative synthetic samples. However, implementing decision-based SMOTE variants comes with its set of challenges. The effectiveness of these variants heavily relies on the quality of the underlying classifier and the ability to accurately capture the decision boundaries. Additionally, the generation of synthetic samples based on the decision boundaries requires careful consideration to avoid introducing noise and artifacts into the dataset. In high dimensional spaces, these methods struggle with identifying meaningful decision boundaries. Also, since the synthetic samples are generated near the decision boundary these approaches might not effectively explore the full space of the minority class, possibly overlooking crucial patterns or variations within it.

Understanding these limitations, this research aims to propose an adaptive decision boundary based oversampling algorithm which covers the entire minority space while avoiding outliers. It intends to contribute to the development of more accurate and meaningful synthetic samples of the minority class to handle imbalanced classification.

III. RESEARCH METHODOLOGY

To address the problem of data imbalance in intrusion detection, this study proposes the Proximity-Adaptive Synthetic Minority Over-sampling Technique (PASMOTE) method. Figure 1 depicts the workflow of the proposed model in its entirety. The approach offers solutions to various limitations studied over SMOTE and its variants from the literature. Table II provides a description of all the inadequacies at different stages of modelling along with their provided solution in this study.

A. Dataset Description

For conducting this experiment, the widely used intrusion detection dataset NSL-KDD [25] was considered. It is an extended version of KDD cup 99 that addresses some inherent issues present in its predecessor [26]. It contains four classes of attacks: remote-to-user, user-to-root, denial of service, and probe. However, these classes suffer from class imbalance, where the number of instances in attack classes remote-to-user and user-to-root are significantly lower than the normal instances. Figure 2 illustrates the class distribution of the NSL-KDD training set emphasizing the proportion of rare attack classes. The training/testing ratio was 80:20. Table III displays the number of samples in both the data files to better understand the frequency distribution in different classes.

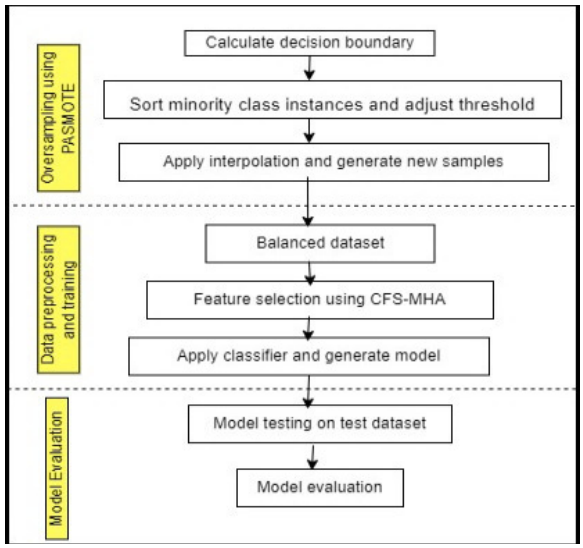

Fig. 1. Workflow of the proposed model.

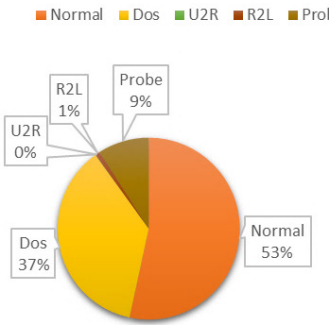

Fig. 2. Class distribution of the NSL-KDD 20% training set.

TABLE II. FACTORS CONSIDERED IN THE PROPOSED MODEL

Stage	Problem	Solution
Pre-processing	Uneven distribution of samples in training and testing	Use K-fold cross validation to ensure no class is left out while training.
Balancing the dataset	- Eliminating the bias towards majority class. Underlying classifier used for balancing overfitting and underfitting. - Insufficient samples for clustering (as in k-means SMOTE). - Works on only categorical features (as in SMOTENC). - Complex and diverse datasets	- Generate more samples of the minority class. - Decision trees with their inherent capability to segment feature space are used. - Early stopping with low number of iterations required and uses pruned decision tree. - Considers each sample individually. - Can work on both categorical and numerical features - Performs adaptive sampling with the targeted approach by adjusting the sampling strategy based on the decision boundary proximity
Feature selection	Corelated features, high-dimensional and diverse feature space	Uses a meta-heuristic approach for dimensionality reduction which is more flexible, adaptable. Performs global optimization.
Model building	Biased nature of learning models to overfit majority classes.	Training models are made adaptable by using cost-sensitive classifiers.

TABLE III. FREQUENCY DISTRIBUTION OF NSL- KDD

Dataset	Class				
	Normal	**Dos**	**Probe**	**u2r**	**r2l**
Train 20%	13449	9234	2289	11	209
Test+	9711	7458	2421	341	2754

B. Data Preprocessing

The dataset contains 42 features and 1 class label. Out of these, 3 features (protocol-type, service, and flag) are nominal in nature. To optimize the learning of models, these features are transformed to their numeric counterparts. Next, the numerical features are normalized to preserve the relative differences between data points, ensuring that the range of values remains consistent. Min-max normalization is a simple and intuitive method for scaling features to a predefined range usually (0-1). Equation (1) is used to rescale the data point X from its original range to a new range [a, b]:

$$X_{scaled} = \frac{X - Xmin}{Xmax - Xmin} \; x \; (b - a) + a \qquad (1)$$

where *Xmin* and *Xmax* represent the minimum and maximum value of the feature, *Xscaled* is the rescaled value, and *a* and *b* define the new range (commonly from 0 to 1).

C. Data Balancing

The proposed PASMOTE based on decision-boundary proximity method has been applied to balance the training dataset. Initially, PASMOTE extracts the minority class samples and normal samples and trains a classifier on this dataset to provide information about the decision boundaries and their distances, which are then used in the resampling process to generate synthetic samples. These synthetic samples aim to balance the class distribution, by taking into account their proximity to decision boundaries learned by the classifier. In this experiment, decision trees were used to learn the decision boundary as they have the following leverages in oversampling:

- They naturally create decision boundaries based on feature splits during training. Thus, they effectively segment the feature space and help in determining distances to these boundaries for each data point.
- They are easier to interpret and consider the features that are most important for classification.
- They calculate distances to decision boundaries for each data point and provide valuable information for understanding the proximity of instances to decision regions, which can be employed to guide oversampling strategies.
- These models inherently rank features based on their importance in the classification process. This information can be utilized to assess which features play a significant role in distinguishing minority class instances, potentially guiding feature engineering or selection processes.
- They are robust to irrelevant or redundant features. This can be advantageous in scenarios where the dataset includes a large number of features, some of which might not contribute significantly to the classification task.
- They can model nonlinear relationships between features and the target variable, allowing for more complex decision boundaries than linear classifiers.

The following algorithm outlines the procedure of utilizing decision boundaries to guide the generation of synthetic samples for the minority class in an imbalanced dataset:

```
1. initialize_classifier(random_state)
2. train_classifier(training_data,
   training_labels)
3. minority_indices =
   get_indices_of_minority_class(traini
   ng_labels, minority_class_label)
4. decision_distances =
   compute_distances_to_decision_bounda
   ry(training_data, minority_indices)
5. sorted_indices =
   sort_samples_by_distance(decision_di
   stances, minority_indices)
6. minority_df =
   extract_samples_by_indices(training_
   data, sorted_indices)
7. majority_df =
   remove_samples_by_indices(training_d
   ata, minority_indices)
8. combined_df =
   concatenate_dataframes(majority_df,
   minority_df)
9. threshold_distances =
   define_threshold_distances(decision_
   distances)
10.resampling_rates =
   calculate_resampling_rates(sorted_in
   dices, decision_distances,
   threshold_distances)
11.interpolators =
   fit_cubic_spline_interpolators(train
   ing_data)
12.unique_categorical_values =
   get_unique_categorical_values(minori
   ty_df)
13.resampled_minority_df =
   create_empty_dataframe()
14.for index, rate in
   resampling_rates.items():
15.nearest_neighbors =
   find_nearest_neighbors(training_data
   , index, rate)
16.for neighbor in nearest_neighbors:
interpolated_sample =
interpolate_features(interpolators,
training_data, index)
17.assign_random_categorical_values(int
   erpolated_sample,
   unique_categorical_values)
18.resampled_minority_df.append(interpo
   lated_sample)
```

```
19.final_training_data =
   combine_dataframes(resampled_minorit
   y_df, training_data)
20.final_training_labels =
   create_combined_labels(resampled_min
   ority_df, training_labels,
   minority_class_label)
21.final_training_data =
   join_labels(final_training_data,
   final_training_labels)
22.normalize_numerical_features(final_t
   raining_data, numerical_columns)
```

The distance from the boundary is calculated for each minority class sample and then they are sorted based on these distances. Based upon the range of distances, threshold values can be calculated to assign the resampling rates. The strategy is to assign higher resampling rates to the samples closer to the decision boundary. For this experiment, three resampling rates of 0.6, 0.4 and 0.2 were applied based on the distance threshold values. To generate the synthetic samples, the interpolation method of cubic-spline has been implemented. Cubic spline interpolation is a mathematical technique used to estimate the values of a function between known data points. It constructs a piecewise continuous curve composed of multiple cubic polynomials, ensuring smoothness and continuity. PASMOTE loops through each feature column and applies cubic spline interpolation to generate interpolated values between the known data points. To ensure diversity in the synthetic samples within the original class distribution, the categorical variables (protocol_type, flag and service) are randomly selected from the unique values of these columns in minority class samples. These interpolated numerical values along with the adjusted categorical variables represent the synthetic samples. Cubic spline is beneficial to handle nonlinear and complex data relationships as it preserves the shape and behavior of synthetic data between existing data points and effectively handles sparse or unevenly spaced datasets. To interpolate a data point between points (x_i, y_i) and (x_{i+1},y_{i+1}), the cubic spline general form is:

$$S_i(x) = a_i + b_i(x - x_i) + c_i\,(x - x_i)^2 + d_i(x - x_i)^3$$

which is valid for $x_i \leq x \leq x_{i+1}$ for $i = 1, \ldots, n-1$ where the coefficients a_i, b_i, c_i, and d_i must be determined for each of the cubic functions. Table IV shows the sample distribution of training 20% NSL-KDD dataset before and after resampling.

TABLE IV. NSL- KDD BEFORE AND AFTER RESAMPLING

Training 20%	Class				
	Normal	**Dos**	**Probe**	**u2r**	**r2l**
Before resampling	13449	9234	2289	11	209
After resampling	13449	9234	2289	842	2603

D. Feature Selection

Selecting the most relevant features from the dataset can reduce dimensionality and enhance the ML model's performance. This study employs the CFS-MHA technique to harness the strength of metaheuristic algorithms for feature selection. CFS-MHA [26] is a CfsSubsetEval ensemble algorithm that uses meta-heuristic algorithms for searching feature space. It is efficient in extracting most representative features from high dimensional intrusion detection datasets, thus reducing the computational time and complexity of models. After applying the technique on the resampled NSL-KDD dataset, 15 features out of 42 were selected. The selected features are described in Table V.

TABLE V. NSL- KDD SELECTED FEATURES

S. No	Feature name	Description
1	service	Destination network service used
2	flag	Status of the connection – Normal or Error
3	wrong_fragment	Total number of wrong fragments in this connection
4	hot	Number of "hot" indicators in the content such as: entering a system directory, creating programs and executing programs
5	logged_in	Login Status : 1 if successfully logged in; 0 otherwise
6	is_guest_login	1 if the login is a "guest" login; 0 otherwise
7	count	Number of connections to the same destination host as the current connection in the past two seconds
8	same_srv_rate	The percentage of connections to the same service, among the connections aggregated in count
9	diff_srv_rate	The percentage of connections that were to different services, among the connections aggregated in count
10	dst_host_srv_count	Number of connections having the same port number
11	dst_host_diff_srv_rate	The percentage of connections that were to different services, among the connections aggregated in dst_host_count
12	dst_host_same_src_port_rate	The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count
13	dst_host_srv_diff_host_rate	The percentage of connections that were to different destination machines, among the connections aggregated in dst_host_srv_count
14	dst_host_serror_rate	The percentage of connections that have activated the flag s0, s1, s2 or s3, among the connections aggregated in dst_host_count
15	dst_host_srv_serror_rate	The percent of connections that have activated the flag s0, s1, s2 or s3, among the connections aggregated in dst_host_srv_count

E. Modeling

The modeling of the reduced dataset was conducted by a cost-sensitive algorithmic adaptation of the random forest. Cost-sensitive classification is a customized approach to handle imbalanced datasets by incorporating domain-specific costs to adjust the model's ability to generalize while considering the true implications of misclassification errors [28]. Cost sensitive classifiers focus on disparities in class distributions. They incorporate a cost matrix or misclassification cost matrix that explicitly defines the costs associated with different types of classification errors (false positives and false negatives). While traditional classifiers intend to minimize the overall

classification error, the cost-sensitive classifiers intend to minimize a cost function derived from the misclassification costs specified in the cost matrix. This method modifies the underlying classifiers (Decision Trees, SVMs, or Neural Networks) to handle cost-sensitive learning explicitly by adjusting the training process to account for the specified costs. Random forest was used as the core classifier for model training and evaluation. It is a learning method prominently famous for its adaptability, robustness, and high predictive accuracy across various ML tasks [29]. It operates by constructing multiple decision trees on random subsets of the training data and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It applies additional randomness by considering only a subset of features at each split point in the decision tree, boosting diversity among the trees.

F. Model Evaluation

To evaluate the model classification, the following metrics were considered:

- Accuracy: It calculates the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset. It provides a general understanding of how well the model performs across all classes. It is calculated by:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3)$$

- Precision: It calculates the ratio of correctly predicted positive instances (true positives) to the total predicted positive instances (true positives + false positives). It indicates the model's ability to avoid false positive predictions.

$$\text{Precision} = \frac{(TP)}{(TP+FP)} \quad (4)$$

- Recall or sensitivity: It calculates the ratio of true positives to the total actual positive instances (true positives + false negatives). It highlights the model's capability to capture all positive instances, minimizing false negatives. It is calculated by:

$$\text{Recall} = \frac{(TP)}{(TP+FN)} \quad (5)$$

IV. RESULTS AND DISCUSSION

To assess the model performance, a separate test dataset needs to be implemented to calculate the model's accuracy and generalization capabilities. In this study, three approaches were implemented for model investigation. Approach I trained the model on the initial training dataset which contains imbalanced class distribution with all 42 features. Approach II performed feature selection on the imbalanced training set to extract 10 representative features by applying CFS-MHA. Approach III oversampled the training dataset using PASMOTE, applied feature selection to extract 15 attributes and then generated the model for evaluation. All three approaches were then evaluated using the NSL-KDD test+ dataset to assess the capabilities of the model. Table VI shows the performance of the three considered approaches on the minority classes u2r and r2l.

TABLE VI. RESULTS OF TESTED APPROACHES

Metric		Approach I	Approach II	Approach III
# of features		41	10	15
Overall accuracy		70.73	71.31	**78.76**
u2r	**Accuracy**	0.00	0.18	**0.42**
	Precision	0.00	0.80	0.53
	Recall	0.00	0.18	0.42
r2l	**Accuracy**	0.001	0.08	**0.83**
	Precision	1.00	0.60	0.48
	Recall	0.001	0.08	0.83

The results show that Approach I's overall accuracy is relatively high (70.73%), indicating good performance on the dataset as a whole. However, for the minority classes (u2r and r2l), the performance is very poor. Specifically, it fails to detect u2r attacks with 0% accuracy, precision, and recall, indicating that it is unable to correctly identify instances of this class. For r2l attacks, it has a slightly better precision (1.00) but extremely low recall (0.001%), suggesting it correctly identifies very few R2L instances while missing the majority. Approach II advocates that reducing the number of features to 10 marginally improves the overall accuracy to 71.31% compared to approach I. u2r detection sees a minor improvement but the values are still relatively low. Similarly, r2l detection improves slightly in precision and recall but the overall performance is still poor. Approach III achieves the highest overall accuracy (78.76%), indicating a significant improvement compared to the previous approaches. u2r detection shows noticeable enhancement in accuracy, precision, and recall compared to the previous approaches. r2l detection shows significant improvement in accuracy and recall, with a slightly decreased precision compared to Approach II.

Figures 3 and 4 present a graphical representation of the achieved results. Approach III outperformed the others in terms of overall accuracy and minority class (u2r and r2l attacks) detection.

Fig. 3. Results on u2r samples.

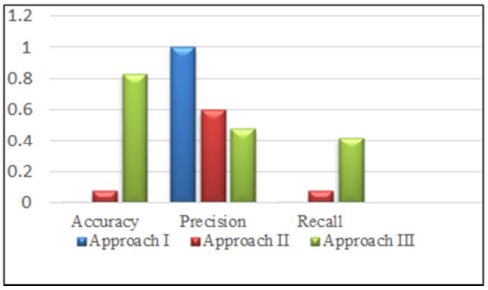

Fig. 4. Results on r2l samples.

To further evaluate the performance of this approach, the results were compared with state-of-art works. The results are presented in Table VII. Although some of the surveyed papers claim higher overall accuracies, they achieve so only due to bias towards the majority class. Studies [30, 31] report higher recall rates for u2r samples but their precision rates are very lower than the proposed approach. Alternatively, [32] has higher precision rates but the reported recall level is only 7% which is very low in comparison with the 42% of the proposed approach. This pattern is visually presented in Figure 5 which shows the tradeoff between precision and recall for the minority classes. Adaptive algorithms that adjust cost weights are heavily researched in handling imbalance but the proposed PASMOTE approach with cost-sensitive learning outperforms the contemporary approaches [31, 34].

TABLE VII. COMPARISON OF THE PROPOSED APPROACH WITH STATE-OF-ART RESEARCH WORKS

Author	Technique	Approach for handling imbalance	Classification approach	Results			
				u2r		r2l	
				Precision	Recall	Precision	Recall
[30] (2020)	Siam-IDS	Adaptive ANN algorithm that computes similarity between samples with Euclidian distance	DNN, CNN	10.11	56.72	57.94	33.25
[31] (2021)	GAN-based Oversampling	Uses generative adversial networks to generate synthetic samples	Used cost-sensitive ANN with three hidden layers	1	94	0	0
	KNN based oversampling	Uses KNNs to interpolate between existing minority samples		2	78	23	10
[33] (2022)	DLNID	Uses ADASYN for oversampling	Bi-LSTM	-	24	-	65.76
[34] (2022)	Enhanced RF	K-means with SMOTE	Enhanced RF with similarity matrix	26.50	26.50	30.63	30.63
[32] (2023)	SMOTE	Applied SMOTE to check its impact with feature reduction	RF	83	7	27	28
[35] (2023)	ROGONG-IDS	Uses a combination of oversampling SMOTE and undersampling near-miss to attain balance	XGBoost	-	10	-	39
PASMOTE (proposed)		Oversampling using decision-boundary proximity with varying threshold values for sampling	Cost-sensitive RF	53	42	48	83

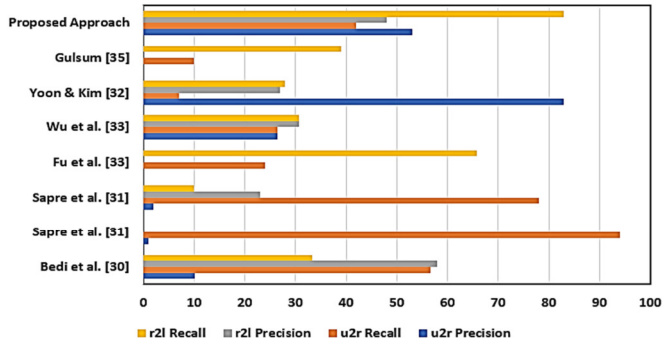

Fig. 5. Comparison of Precision and Recall of the proposed approach with contemporary works.

V. CONCLUSION AND FUTURE SCOPE

By addressing the issue of class imbalances in intrusion detection systems, this research aims to improve their accuracy, adaptability, and efficiency to bolster network security against cyber threats. Decision-based SMOTE variants have emerged as a powerful technique to rebalance data by incorporating decision boundaries and classification information into the sampling process. These variants offer a promising approach to addressing imbalanced data and enhancing the decision-making process in various domains.

The proposed hybrid approach demonstrated improvement in performance on the minority classes by incorporating the concept of decision-boundary proximity and oversampling. By considering decision boundary, it captured the intricacies and complexities of the minority class near the decision boundary, which is important for classification accuracy. The experiment explored hyperparameter optimization by working on different threshold values for resampling. The proposed hybrid algorithm is adaptive in nature as after every iteration of oversampling, it adjusts and selects new instances for sampling and prioritize the generation of synthetic samples in regions closer to the decision boundary, reducing the risk of overgeneralization and improving the separation between classes. Unlike SMOTE, it does not select the sample points randomly from the feature space leading to selection of noisy samples.

While the proposed hybrid algorithm shows satisfactory performance, there is room for improvement in identifying rare classes in the NSL-KDD dataset. Further tuning or alternative modeling approaches can enhance detection of minority classes. Continued research on decision-based SMOTE variants is expected to advance machine learning and improve predictive models. Practitioners can utilize these variants to elevate imbalanced data sampling and enhance machine learning performance in various applications.

REFERENCES

[1] F. Provost, "Machine Learning from Imbalanced Data Sets 101," presented at the AAAI'2000 Workshop on Imbalanced Data Sets, 2000.

[2] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Nov. 2016, https://doi.org/10.1007/s13748-016-0094-0.

[3] D. Elreedy and A. F. Atiya, "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance," *Information Sciences*, vol. 505, pp. 32–64, Dec. 2019, https://doi.org/10.1016/j.ins.2019.07.070.

[4] M. Machoke, J. Mbelwa, J. Agbinya, and A. E. Sam, "Performance Comparison of Ensemble Learning and Supervised Algorithms in Classifying Multi-label Network Traffic Flow," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8667–8674, Jun. 2022, https://doi.org/10.48084/etasr.4852.

[5] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," in *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Dubrovnik, Croatia, Sep. 2003, pp. 107–119, https://doi.org/10.1007/978-3-540-39804-2_12.

[6] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010, https://doi.org/10.1109/TSMCA.2009.2029559.

[7] M. Lamari *et al.*, "SMOTE–ENN-Based Data Sampling and Improved Dynamic Ensemble Selection for Imbalanced Medical Data Classification," in *Advances on Smart and Soft Computing*, F. Saeed, T. Al-Hadhrami, F. Mohammed, and E. Mohammed, Eds. New York, NY, USA: Springer, 2021, pp. 37–49.

[8] R. Kaur and N. Gupta, "An Empirical Study on Imbalanced Learning in Intrusion Detection Using Random Tree Classifier," in *International Conference on Augmented Intelligence and Sustainable Systems*, Trichy, India, Nov. 2022, pp. 944–949, https://doi.org/10.1109/ICAISS55157.2022.10010583.

[9] Y. Wang, M. M. Rosli, N. Musa, and F. Li, "Multi-Class Imbalanced Data Classification: A Systematic Mapping Study," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14183–14190, Jun. 2024, https://doi.org/10.48084/etasr.7206.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, https://doi.org/10.1613/jair.953.

[11] T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," *Information*, vol. 14, no. 1, Jan. 2023, Art. no. 54, https://doi.org/10.3390/info14010054.

[12] D. Bajer, B. Zonc, M. Dudjak, and G. Martinovic, "Performance Analysis of SMOTE-based Oversampling Techniques When Dealing with Data Imbalance," in *International Conference on Systems, Signals and Image Processing*, Osijek, Croatia, Jun. 2019, pp. 265–271, https://doi.org/10.1109/IWSSIP.2019.8787306.

[13] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *International Conference on Intelligent Computing*, Hefei, China, Aug. 2005, pp. 878–887, https://doi.org/10.1007/11538059_91.

[14] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong, China, Jun. 2008, pp. 1322–1328, https://doi.org/10.1109/IJCNN.2008.4633969.

[15] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Bangkok, Thailand, Apr. 2009, pp. 475–482, https://doi.org/10.1007/978-3-642-01307-2_43.

[16] M. A. H. Farquad and I. Bose, "Preprocessing unbalanced data using support vector machine," *Decision Support Systems*, vol. 53, no. 1, pp. 226–233, Apr. 2012, https://doi.org/10.1016/j.dss.2012.01.016.

[17] L. Sun, M. Li, W. Ding, E. Zhang, X. Mu, and J. Xu, "AFNFS: Adaptive fuzzy neighborhood-based feature selection with adaptive synthetic over-sampling for imbalanced data," *Information Sciences*, vol. 612, pp. 724–744, Oct. 2022, https://doi.org/10.1016/j.ins.2022.08.118.

[18] J. Li, Y. Liu, and Q. Li, "Intelligent fault diagnosis of rolling bearings under imbalanced data conditions using attention-based deep learning method," *Measurement*, vol. 189, Feb. 2022, Art. no. 110500, https://doi.org/10.1016/j.measurement.2021.110500.

[19] Y. Liu, G. Wu, W. Zhang, and J. Li, "Federated Learning-Based Intrusion Detection on Non-IID Data," in *International Conference on Algorithms and Architectures for Parallel Processing*, Copenhagen, Denmark, Oct. 2022, pp. 313–329, https://doi.org/10.1007/978-3-031-22677-9_17.

[20] K. A. ElDahshan, A. A. AlHabshy, and B. I. Hameed, "Meta-Heuristic Optimization Algorithm-Based Hierarchical Intrusion Detection System," *Computers*, vol. 11, no. 12, Dec. 2022, Art. no. 170, https://doi.org/10.3390/computers11120170.

[21] S. Barua, Md. M. Islam, X. Yao, and K. Murase, "MWMOTE–Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, Oct. 2014, https://doi.org/10.1109/TKDE.2012.232.

[22] G. Douzas and F. Bacao, "Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE," *Information Sciences*, vol. 501, pp. 118–135, Oct. 2019, https://doi.org/10.1016/j.ins.2019.06.007.

[23] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, Oct. 2018, https://doi.org/10.1016/j.ins.2018.06.056.

[24] S. Divakar, A. Bhattacharjee, and R. Priyadarshini, "Smote-DL: A Deep Learning Based Plant Disease Detection Method," in *6th International Conference for Convergence in Technology*, Maharashtra, India, Apr. 2021, https://doi.org/10.1109/I2CT51068.2021.9417920.

[25] H. Y. I. Khalid and N. B. I. Aldabagh, "A Survey on the Latest Intrusion Detection Datasets for Software Defined Networking Environments," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13190–13200, Apr. 2024, https://doi.org/10.48084/etasr.6756.

[26] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, Jul. 2009, https://doi.org/10.1109/CISDA.2009.5356528.

[27] R. Kaur and N. Gupta, "CFS-MHA: A Two-Stage Network Intrusion Detection Framework," *International Journal of Information Security and Privacy*, vol. 16, no. 1, pp. 1–27, Jan. 2022, https://doi.org/10.4018/IJISP.313663.

[28] R. Greiner, A. J. Grove, and D. Roth, "Learning cost-sensitive active classifiers," *Artificial Intelligence*, vol. 139, no. 2, pp. 137–174, Aug. 2002, https://doi.org/10.1016/S0004-3702(02)00209-6.

[29] A. Parmar, R. Katariya, and V. Patel, "A Review on Random Forest: An Ensemble Classifier," in *International Conference on Intelligent Data Communication Technologies and Internet of Things*, Coimbatore, India, Aug. 2018, pp. 758–763, https://doi.org/10.1007/978-3-030-03146-6_86.

[30] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in Intrusion Detection Systems using Siamese Neural Network," *Procedia Computer Science*, vol. 171, pp. 780–789, Jan. 2020, https://doi.org/10.1016/j.procs.2020.04.085.

[31] S. Sapre, K. Islam, and P. Ahmadi, "A Comprehensive Data Sampling Analysis Applied to the Classification of Rare IoT Network Intrusion Types," in *18th Annual Consumer Communications & Networking Conference*, Las Vegas, NV, USA, Jan. 2021, https://doi.org/10.1109/CCNC49032.2021.9369617.

[32] J.-E. Yoon and K. Kim, "Comparison of Dimensional Reduction and Oversampling Methods for Efficient Network Anomaly Detection," *Journal of Digital Contents Society*, vol. 24, no. 3, pp. 583–591, Mar. 2023, https://doi.org/10.9728/dcs.2023.24.3.583.

[33] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A Deep Learning Model for Network Intrusion Detection with Imbalanced Data," *Electronics*, vol. 11, no. 6, Jan. 2022, Art. no. 898, https://doi.org/10.3390/electronics11060898.

[34] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, May 2022, Art. no. 39, https://doi.org/10.1186/s13634-022-00871-6.

[35] A. O. Arik and G. C. Cavdaroglu, "An Intrusion Detection Approach based on the Combination of Oversampling and Undersampling Algorithms," *Acta Infologica*, vol. 7, no. 1, pp. 125–138, Jan. 2024, https://doi.org/10.26650/acin.1222890.