HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

# A COMPARATIVE EVALUATION OF CLASS-IMBALANCE HANDLING STRATEGIES FOR MACHINE LEARNING-BASED INTRUSION DETECTION

**X.K. Samarov**
**Z. A. Barotova**
**Davlatbek Abduvoxidov**
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan
E-mail: samarov07@gmail.com
State Institution "Inspectorate for Regulation of the Alcohol and Tobacco Market" under the Department for Combating Economic Crimes at the Prosecutor General's Office of the Republic of Uzbekistan.
E-mail: zahrobarotova09@gmail.com,
E-mail: davlatbek.abduvoxidov@gmail.com
doi:https://doi.org/10.5281/zenodo.24419429

**Abstract.** In machine learning-based network intrusion detection systems, class imbalance can negatively affect the effectiveness of detecting rare attack types. This article presents a comparative analysis grounded in existing literature of balancing strategies aimed at addressing this issue. Specifically, it examines the advantages and limitations of three approaches: the initial imbalanced state (baseline), the generation of synthetic samples for minority classes (SMOTE), and the reduction of samples from majority classes (Random Undersampling – RUS). The analysis utilizes the CIC-IDS2017 and UNSW-NB15 datasets, alongside classifiers such as Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). Findings from existing research indicate that the effectiveness of class balancing depends on the specific dataset, the classifier employed, and the evaluation conditions. Consequently, the study argues that system performance should not be assessed solely through overall accuracy; rather, it necessitates the use of metrics such as Macro-F1, Macro-Precision, class-specific Recall, and the false positive rate. The study establishes a methodological framework for comparing balancing strategies under identical experimental conditions, along with key criteria for future experimental research.

**Keywords:** network intrusion detection, machine learning, class imbalance, SMOTE, Random Undersampling, CIC-IDS2017, UNSW-NB15, Macro-F1

## Introduction.

The increasing complexity of network infrastructures and the constantly evolving nature of cyberattacks have heightened the urgency of detecting unauthorized access and suspicious behavior.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

While machine learning-based methods are widely explored as a solution to this challenge, relying solely on a machine learning algorithm is insufficient for evaluating system effectiveness. In the context of attack detection, factors such as training features, class distribution, and evaluation methodology play a critical role. The problem of class imbalance remains a significant concern.

In scenarios involving class imbalance, a high overall accuracy score does not necessarily mean that a model performs well in detecting all types of attacks. Therefore, when evaluating such systems, it is essential to look beyond simple accuracy and consider metrics such as the performance for each individual attack class, the macro F1-score, and the rate of false results.

There are several approaches to mitigating the problem of class imbalance; it is advisable to compare these methods within a consistent experimental environment and using the same training dataset. In this context, the objective of this study is to conduct a comparative experimental evaluation of the effectiveness of methods for mitigating class imbalance in machine learning-based intrusion detection systems. The study addresses the following research questions (RQs):

- RQ 1. How do various methods for mitigating class imbalance affect the overall and class-specific performance of machine learning-based attack detection systems?
- RQ 2. How do changes in the rate of false positives and computational time accompany the improvement in the detection rate of rare attacks?
- RQ 3. Under the experimental conditions being studied, which balancing method provides the optimal balance between the detection of rare attacks and computational costs?

The primary scientific contribution of the study lies in comparing selected methods for mitigating class imbalance within a unified experimental framework, evaluating their impact on the detection of rare attacks, and analyzing the trade-offs between detection quality, false positive rates, and computational costs. The findings provide a basis for drawing scientifically grounded conclusions that account for the advantages and limitations of various balancing techniques when selecting an appropriate method.

**Literature review.**

Class imbalance is considered a significant issue in modern Intrusion Detection Systems (IDS) and has been the subject of extensive research. In their study, Gopalan et al. [1] analyzed research conducted using the CSE-CICIDS dataset, highlighting that the issue of class imbalance is often not sufficiently addressed in many existing studies. Research by Ahsan et al. [2] focused on evaluating classical machine learning algorithms and data preprocessing techniques for IDS using the NSL-KDD dataset; furthermore, the study examined the impact of oversampling, undersampling, and hybrid sampling approaches on classification models. A key conclusion from this work is that, rather than selecting a balancing method based on a single general metric, evaluating such methods through a comparative analysis of their effects on different classes leads to improved performance. Bagui et al. [3] investigated the challenge of detecting network attacks belonging to minority classes and analyzed the ratios of oversampling and undersampling, using the Random Forest algorithm as a case study. The significance of this research

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

lies in the insight that managing class imbalance involves more than just equalizing the number of instances per class, the specific method used to select samples also influences the evaluation outcomes. In their study, Ahmed et al. [4] examined several machine learning algorithms using the UNSW-NB15 dataset. By providing a comparative analysis of the models' performance both before and after the application of the SMOTE technique, they clearly demonstrate the importance of data balancing. Widodo et al. [5] investigated the use of SMOTE, data normalization, and various machine learning algorithms on the NSL-KDD and CIC-IDS2017 datasets; their findings particularly highlight the advantages of the SMOTE technique. Shanmugam et al. [6] conducted a comprehensive comparative analysis of approaches aimed at addressing class imbalance, employing metrics such as the F1-score and G-mean, in addition to accuracy for evaluation. The results of this analysis also shed light on the specific advantages and limitations of each method.

Table 1. A brief analysis of the reviewed literature

| **Ref.** | **Dataset** | **Approach** | **Evaluation criteria** | **Limitation** |
|---|---|---|---|---|
| [1] | CSE-IDS | LR | - | It does not compare methods through a single, controlled experiment |
| [2] | NSL-KDD | CR | Acc, Rec, Prec, F-score | Provides a broad comparison, but the study focuses on a single dataset. |
| [3] | UNSW-NB15 | LR | Acc, Rec, Prec, F-score | the question of testing a consistent multi-class protocol across two datasets remains open for us |
| [4] | UNSW-NB15 | AR | Acc, Rec, Prec, F-score | Since SMOTE and feature selection are used in combination, the experimental design must be carefully considered when interpreting the specific impact of the balancing technique |
| [5] | NSL-KDD, CIC-IDS2017 | CR | Acc, Rec, Prec, F-score | SMOTE is the central approach; however, we must compare SMOTE against alternative balancing methods under identical conditions. |
| [6] | CIC-DDoS2019 | LR | F1-score, G-mean | A very similar study |

Based on the literature reviewed above, the primary focus of this research is to examine how balancing techniques affect the trade-off between the detection of rare attacks, the rate of false alarms, and computational costs. This approach allows for evaluation based on several criteria that are significant not only for classification but also for the practical implementation of Intrusion Detection Systems (IDS).

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

**Methodology.**

This research focuses on a comparative evaluation of the impact of class imbalance handling strategies on classifier performance within machine learning-based intrusion detection systems. The study analyzes overall classification accuracy, as well as the detection of rare attack classes, false positive rates, and computational costs. The experimental study utilizes two datasets, four classifiers, and three strategies for managing class imbalance. The overall research process is illustrated in Figure 1 and comprises the following stages:

1. Dataset selection;
2. Data preprocessing;
3. Selection of methods for handling class imbalance;
4. Selection of a classifier;
5. Defining evaluation criteria;
6. Conducting experiments and calculating results;
7. Comparing results and drawing conclusions.

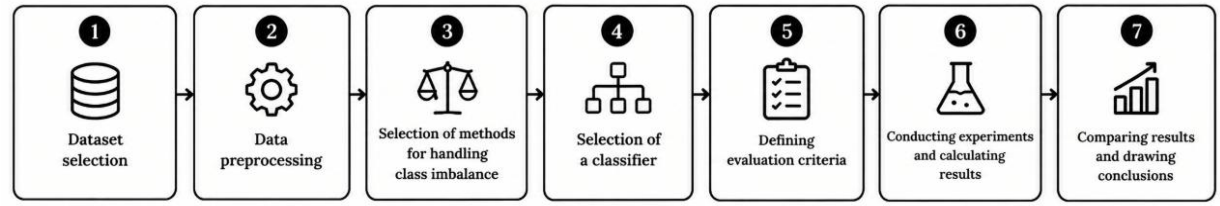


Figure 1. General stages of research methodology

Selection of Datasets

For the purpose of conducting experimental trials in this research, the CIC-IDS2017 and UNSW-NB15 datasets have been selected. The CIC-IDS2017 dataset includes normal traffic as well as various types of attacks, such as DoS, DDoS, Brute Force, and PortScan. The UNSW-NB15 dataset encompasses normal traffic along with nine distinct classes of attacks. These datasets facilitate the evaluation of various traffic features and strategies for balancing class distributions.

Table 2. General characteristics of the selected datasets

| **Dataset** | **Number of features** | **Notes** | **Types of attacks** | **Purpose** |
|---|---|---|---|---|
| UNSW-NB15 | 41 | 2540044 | 9 types of attacks, encompassing activities such as shellcode and reconnaissance | It is used to detect modern cyberattacks and to evaluate deep learning and feature selection algorithms. |

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

| | | | | |
|---|---|---|---|---|
| CIC-IDS2017 | 49 | 2.8 million | PortScan, DoS, DDoS, brute-force, botnet | Artificial intelligence-based IDS, built upon realistic network traffic, is used in the creation of models for detecting and classifying anomalies. |

Each dataset is treated as an independent multi-class classification task. Since the names and composition of attack classes may vary across datasets, the results for each are analyzed separately. For the UNSW-NB15 dataset, the official training and test sets will be used, while for the CIC-IDS2017 dataset, a splitting method that preserves the class distribution will be applied.

Classifiers

Various machine learning algorithms are employed for model training and attack detection. In this research, several algorithms were selected specifically to study and analyze the problem of class imbalance; brief descriptions of these are provided below.

The Decision Tree algorithm partitions input data features based on a sequence of conditions and classifies a sample into the appropriate class. Belonging to the category of supervised learning algorithms, it is used in network traffic analysis based on the statistical characteristics of packets or flows to distinguish between normal traffic and various classes of attacks [7].

Random Forest is an ensemble algorithm that combines multiple decision trees. Each tree is trained based on randomly selected training samples and features. The final class is determined by aggregating the predictions of the individual trees.

In multi-class classification, majority voting is expressed as follows [8]:

$$\hat{y}(x)=\text{mode}\{T_1(x), T_2(x), \ldots, T_B(x)\} \quad (1)$$

where:

- $x$ is a traffic sample to be classified;
- $T_B(x)$ is prediction of a tree;
- mode is an operator that selects the most frequently occurring class.

KNN is a distance-based algorithm that classifies a new sample based on the classes of the k samples in the training set that are closest to it. Rather than learning specific parameters, it relies on the positioning and proximity of the training samples.

The Euclidean distance between two m dimensional samples is:

$$d(x,z)=\sqrt{\sum_{j=1}^{m}(x_j-z_j)^2} \quad (2)$$

Here, x and z are two samples, and m is the number of features.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

An MLP is an artificial neural network consisting of an input layer, one or more hidden layers, and an output layer. It can classify normal traffic and various types of attacks by learning complex, non-linear relationships among the statistical features of network traffic.

The output of each neural layer is expressed as follows [9]:

$$\mathbf{h}^{(l)}=g(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)}+\mathbf{b}^{(l)}) \quad (3)$$

where:

- $\mathbf{h}^{(l)}$ is output of the layer;
- $\mathbf{W}^{(l)}$ is weight matrix of the layer.

Table 3. General description of the selected classifiers

| Type | Advantages | Limitation |
|---|---|---|
| Decision Tree | - The decision-making process is easy to explain and visualize.<br>- It can capture non-linear relationships.<br>- It does not require complex mathematical transformations of the data. | - If the tree depth is not controlled, it may overfit the training data.<br>- In cases of class imbalance, it may become biased toward the majority class.<br>- Small changes in the data can alter the tree structure. |
| Random Forest | - Combining the results of multiple trees can improve stability compared to a single tree.<br>- It identifies complex and non-linear relationships.<br>- It can handle various types of features.<br>- It allows for the assessment of feature importance. | - Increasing the number and depth of trees can lead to higher computational and memory costs.<br>- It is more complex to interpret than a single decision tree.<br>- The use of an ensemble does not automatically resolve class imbalance.<br>- In real-time IDS (Intrusion Detection Systems), it is necessary to specifically measure prediction latency. |
| KNN | - Simple structure and easy to understand.<br>- Utilizes the structure of the local data.<br>- Capable of representing non-linear class boundaries.<br>- Does not require retraining the model when classifying new samples. | - With large datasets, calculating the distance for each new sample can be time-consuming.<br>- If features are on different scales, distance calculations may yield inaccurate results.<br>- In high-dimensional space, the differences between distances may diminish. |

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

| | | |
|---|---|---|
| | | - Class imbalance and the chosen value of k can significantly affect classification results. |
| MLP | - Learns complex non-linear relationships.<br>- Suitable for multi-class classification tasks.<br>- Capable of identifying complex traffic patterns, provided there is sufficient data and properly tuned parameters. | - Training outcomes depend on the number of layers, number of neurons, learning rate, and other hyperparameters.<br>- Feature scaling is typically required.<br>- Training time and resource consumption can be high.<br>- If class imbalance is not managed, the ability to detect rare attacks may be compromised. |

These four classifiers encompass different algorithmic families. By comparing them using the same dataset, the same train/test split, and three different strategies for handling class imbalance, it is possible to examine the impact of class imbalance on the results of various models.

Class-Imbalance Handling Strategies

To investigate the effect of class imbalance on intrusion detection performance, three experimental strategies were selected: the baseline approach without resampling, Synthetic Minority Over-sampling Technique (SMOTE), and Random Undersampling (RUS). These strategies represent three different conditions: preserving the original class distribution, increasing minority-class representation, and reducing majority-class samples.

SMOTE generates synthetic minority-class samples using interpolation between neighboring observations [10]:

$$x_{new}=x_i+\lambda(x_{nn}-x_i),\ \lambda\in[0,1] \quad (4)$$

where:

- $x_{new}$ is created sample;
- $\lambda$ is a random number between 0 and 1.

In contrast, RUS reduces the number of majority-class samples, potentially decreasing computational costs but also removing informative observations. The baseline strategy serves as a reference for determining whether either resampling method provides measurable benefits.

The experimental design combines two datasets (CIC-IDS2017 and UNSW-NB15), four classifiers (DT, RF, KNN, and MLP), and three imbalance-handling strategies, resulting in 24 core experimental configurations. Resampling is applied exclusively to the training data, while the test set retains its original class distribution to ensure an unbiased evaluation.

Performance is assessed using Macro-F1, per-class Recall, Macro-Precision, false positive rate (FPR), and computational time. This design enables comparison of detection performance, minority-

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

class recognition, false alarms, and computational cost across classifiers and datasets. The benefits and limitations of each strategy are evaluated empirically rather than assumed in advance.

Data preparation and experimental procedure

First, the datasets are checked for missing, infinite, and incorrect values, and the number of samples for each class is determined. Then, the data is split into training and testing sets. Preprocessing steps, such as encoding categorical features and scaling numerical features, are fitted only on the training data.

Subsequently, the training data is prepared according to three strategies, and the selected classifiers are trained. The trained models are evaluated on the unmodified test set. This procedure helps prevent the test data from indirectly influencing the balancing or training process.

When repeated experiments or parameter selection are employed, balancing operations are performed within the scope of the respective training subset. The data splitting method, random seed, and preprocessing rules are kept consistent across all relevant configurations.

Evaluation criteria

The evaluation phase is crucial for determining the performance of a model developed for IDS systems. This evaluation is carried out based on several metrics. In this research, the following evaluation metrics are used to assess the performance of the classifiers.

Table 4. Selected evaluation metrics description

| **Metric** | **Definition** | **Formula** |
|---|---|---|
| Accuracy | Accuracy is defined as the number of correct predictions divided by the total number of predictions. | $Acc=\frac{TP+TN}{TP+TN+FP+FN}$ |
| Recall | Recall or sensitivity calculates the model's performance in identifying the attack samples against all the actual positives. | $Rec=\frac{TP}{TP+FN}$ |
| False Positive Rate (FPR) | The False Positive Rate represents the proportion of normal traffic that is incorrectly identified as an attack. | $FPR=\frac{FP}{FP+TN}$ |
| Macro-F1 | The F1-score is the harmonic mean of Precision and Recall, taking both metrics into account. Macro-F1, on the other hand, calculates the average of the F1 scores for all K classes, assigning equal weight to each: | $F1=2*\frac{Precision*\ Recall}{Precision+Recall}$ $Macro-F1=\frac{1}{K}\sum_{i=1}^{K} F1$ |

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

| | | |
|---|---|---|
| Macro-Precison | Macro-Precision is calculated as the arithmetic mean of the Precision values for all classes: | $Prec=\frac{TP}{TP+FP}$ $Macro-Prec=\frac{1}{K}\sum_{i=1}^{K} Prec$ |
| Time (train) | Training time refers to the time spent training a model based on the provided training data. | $T_{train}=t_{end}-t_{start}$ |
| Time (predict) | The time the trained model spends determining the class label for new samples. | $T_{sample}=\frac{T_{predict}}{N}$ |

Result Analysis and Reproducibility

The experiment is planned to be conducted in a Python environment using the scikit-learn and imbalanced-learn libraries. To ensure the reproducibility of the research, details regarding hardware and operating system specifications, library versions, model parameters, the random seed, data files, and preprocessing steps will be documented.

Results will be presented separately for each dataset, and the impact of balancing strategies on various classifiers will be compared. The analysis will focus on the relationships between changes in Macro-F1 and minority-class Recall metrics, false positive results, and computational costs. Final conclusions will be based solely on the results of the experiments actually conducted.

**Experimental Research and Discussion**

In the context of network attack detection, class imbalance affects not only the overall performance of classifiers but also their ability to identify rare types of attacks. Consequently, this analysis examines three approaches: the initial, unbalanced state (baseline); the artificial augmentation of minority classes using SMOTE; and the reduction of majority class samples via Random Undersampling (RUS).

Ahmed et al. evaluated several classifiers Random Forest (RF), Decision Tree (DT), Logistic Regression, K-Nearest Neighbors (KNN), and Artificial Neural Network (ANN) using the UNSW-NB15 dataset. In the unbalanced scenario, an accuracy of 89.29% was recorded for the Random Forest model. In a configuration utilizing SMOTE combined with feature selection (resulting in 24 selected features), an accuracy of 95.1% was achieved. However, this improvement cannot be attributed solely to the effect of SMOTE, as the feature selection process was also modified [11].

Widodo et al. investigated the application of SMOTE to the NSL-KDD and CIC-IDS2017 datasets. Their study considered not only class balancing but also data normalization, feature selection, and the use of various classifiers. The authors noted that the application of SMOTE improves results; however, the importance of separately evaluating the contribution of various preprocessing stages remains [12].

Zhang et al. proposed the SGM method, which combines SMOTE with cluster-based undersampling using a Gaussian Mixture Model (GMM), for the UNSW-NB15 and CICIDS2017 datasets. The SGM-CNN model achieved an accuracy of 99.74% for binary classification and 96.54%

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

for multi-class classification on the UNSW-NB15 dataset, and 99.85% accuracy for 15-class classification on the CICIDS2017 dataset. While these results demonstrate the potential of a hybrid balancing approach, they cannot be directly compared to results obtained using standard SMOTE or Random Undersampling (RUS) [13].

Balla et al. compared various balancing techniques—including Random Undersampling, One-Sided Selection, NearMiss, SMOTE, and ADASYN—using the CICIDS2017 dataset and data related to electrical power systems. In binary classification experiments on the CICIDS2017 dataset, Random Undersampling yielded an accuracy of 96.65% and an F1-score of 96%, while NearMiss achieved an accuracy of 99.25% and an F1-score of 99.34%. These results indicate that the effectiveness of undersampling methods depends on the chosen algorithm and the experimental conditions. However, the classification task in this study is not entirely suited for distinguishing between various types of multi-class attacks [14].

In another study focused on detecting rare attacks within the UNSW-NB15 dataset, BSMOTE and SVM-SMOTE methods were examined using various oversampling and undersampling configurations. This approach, which involves controlling the undersampling rate, demonstrates that the resampling ratio should be considered a distinct parameter when balancing minority classes [15].

In 2025, Li et al. compared SMOTE, Random Oversampling, and methods based on generative models using the NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018 datasets. The authors showed that, in certain cases, generative approaches improve the Macro-F1 score; however, the results vary depending on the dataset and the specific attack class. This finding confirms that evaluation based solely on overall accuracy is insufficient [16].

Table 5. Results obtained from the research

| **Ref.** | **Dataset** | **Classificator** | **Balancing method** | **Main result** |
|---|---|---|---|---|
| [11] | UNSW-NB15 | DT, RF, KNN, ANN | SMOTE, Baseline | RF accuracy is 89.29% in the initial state; it is 95.1% in the configuration where SMOTE and PCA are applied. |
| [12] | CIC-IDS2017, NSL-KDD | DT, RN, MLP | SMOTE | Balancing and pre-processing affect the results. |
| [13] | UNSW-NB15, CIC-IDS2017 | RF, MLP, CNN | SMOTE, GMM, undersampling | The hybrid balancing method showed high detection performance |
| [14] | CIC-IDS2017 | CNN, MLP | RUS, SMOTE, ADASYN | The results depend on the balancing method and the dataset. |

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

| | | | | |
|---|---|---|---|---|
| [15] | UNSW-NB15 | Classification with resampling combinations | BSMOTE, SVM-SMOTE, undersampling | The resampling ratio and the sequence of methods are significant. |
| [16] | UNSW-NB15, CIC-IDS2017, NSL-KDD | RF, DT and others | SMOTE, RUS, generative methods | Macro-F1 results vary depending on the dataset and the attack class. |

As shown in the table, while some studies have investigated the effectiveness of SMOTE, others have focused on undersampling or hybrid methods. However, the classifiers, data preprocessing techniques, class combinations, and evaluation criteria vary across these studies. Consequently, it would be methodologically unsound to present the percentages from different papers as a single, unified ranking.

Comparative Discussion of Baseline, SMOTE, and RUS Strategies

A baseline is necessary to establish the initial performance of a model when no balancing technique is applied. This approach reveals how the model performs given the original class distribution. However, in datasets where one class is dominant, a high accuracy score does not necessarily imply that rare attack instances are being effectively detected.

SMOTE generates synthetic examples based on existing samples from the minority class. This technique can assist classifiers, such as DT, RF, KNN, and MLP, in learning the characteristics associated with minority classes. At the same time, the quality of these synthetic samples and the complexity of the decision boundaries between classes can influence the final results. Specifically, when dealing with datasets like UNSW-NB15, which contain categorical features, it is essential to implement an appropriate preprocessing method for these categorical variables before applying standard SMOTE. RUS reduces the size of the dataset by randomly removing samples from majority classes. While this can shorten training time, there is a risk that samples representing important features might also be randomly discarded. Therefore, the results of RUS should be evaluated not only based on computational speed but also using the Macro-F1 score and the Recall metric for each attack class.

The effectiveness of these three strategies can vary depending on the type of classifier used. For instance, in the case of KNN, the additional samples generated by SMOTE alter the neighborhood structure; for DT and RF, synthetic samples may influence decision boundaries. The results for an MLP also depend on the data distribution and training parameters. Consequently, rather than selecting a balancing method based on the results from a single classifier, it is advisable to evaluate it across all intended classifiers using a consistent evaluation protocol.

**Conclusion**

In this article, strategies for managing class imbalance in machine learning-based network intrusion detection systems are comparatively analyzed based on existing scientific research. The results indicate that the effectiveness of a balancing method depends not only on the chosen strategy but also on the characteristics of the dataset, the type of classifier, and the evaluation methodology.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

The baseline approach serves as an initial benchmark for determining the impact of balancing techniques on model performance. SMOTE can facilitate the learning of minority classes by generating synthetic samples, while RUS can reduce the number of samples in majority classes, thereby decreasing data volume and training costs. However, SMOTE is subject to limitations regarding the quality of synthetic samples, and RUS carries the risk of losing potentially important samples. Consequently, there is insufficient evidence to designate any single strategy as the most effective for all scenarios.

The analysis suggests that comparing DT, RF, KNN, and MLP classifiers using Baseline, SMOTE, and RUS strategies under consistent conditions within the context of the CIC-IDS2017 and UNSW-NB15 datasets represents a significant avenue for research. In such a comparison, utilizing metrics like Macro-F1, Macro-Precision, as well as Recall and FPR for each attack class, allows for a more comprehensive assessment than relying solely on overall accuracy. Furthermore, it is crucial for the reliability of the results that balancing techniques are applied exclusively to the training data, leaving the test set unaltered.

Overall, this comparative analysis demonstrates that there is no single universal solution for managing class imbalance and underscores the necessity of conducting evaluations under consistent methodological conditions. Future experimental research could involve systematic testing across the two aforementioned datasets, four classifiers, and three balancing strategiesincorporating a class-specific analysis of results and an assessment of computational costs.

**References:**

1. Gopalan, S., et al. “Balancing Approaches towards ML for IDS: A Survey for the CSE-CIC IDS Dataset.” 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA), 2021. https://doi.org/10.1109/ICCSPA49915.2021.9385742.
2. Ahsan, R., Shi, W., and Corriveau, J.-P. “Network Intrusion Detection Using Machine Learning Approaches: Addressing Data Imbalance.” IET Cyber-Physical Systems: Theory & Applications, 7(1), 30–39, 2022. https://doi.org/10.1049/cps2.12013.
3. Bagui, S., Mink, D., Bagui, S., Subramaniam, S., and Wallace, D. “Resampling Imbalanced Network Intrusion Datasets to Identify Rare Attacks.” Future Internet, 15(4), 130, 2023. https://doi.org/10.3390/fi15040130.
4. Ahmed, H. A., Hameed, A., and Bawany, N. Z. “Network Intrusion Detection Using Oversampling Technique and Machine Learning Algorithms.” PeerJ Computer Science, 8, e820, 2022. https://doi.org/10.7717/peerj-cs.820.
5. “Machine Learning-Based Intrusion Detection on Multi-Class Imbalanced Dataset Using SMOTE.” Procedia Computer Science, 234, 578–583, 2024. https://doi.org/10.1016/j.procs.2024.03.042.
6. Shanmugam, V., Razavi-Far, R., and Hallaji, E. “Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches.” Electronics, 14(1), 69, 2025. https://doi.org/10.3390/electronics1401006

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

7. "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," IEEE Access, 2021. doi: 10.1109/ACCESS.2021.3056614.
8. [2] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," Journal of Big Data, vol. 7, Art. no. 105, 2020. doi: 10.1186/s40537-020-00379-6.
9. "Multi-layer Perceptron-Based Network Intrusion Detection for the UNSW-NB15," Lecture Notes in Networks and Systems, 2025. doi: 10.1007/978-981-96-4136-9_6.
10. Ahmed, M. M., Hameed, S., & Bawany, N. Z. "Network intrusion detection using oversampling technique and machine learning algorithms." PeerJ Computer Science, 8, e820, 2022. https://doi.org/10.7717/peerj-cs.820
11. [2] Widodo, A. A., Setiawan, E. B., & Indraswari, R. "Machine Learning-Based Intrusion Detection on Multi-Class Imbalanced Dataset Using SMOTE." Procedia Computer Science, 234, 578–583, 2024. https://doi.org/10.1016/j.procs.2024.03.042
12. [3] Zhang, Y., Huang, Y., Wu, Q., & Li, Z. "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset." Computer Networks, 177, 107315, 2020. https://doi.org/10.1016/j.comnet.2020.107315
13. [4] Balla, T. et al. "The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems." Sensors, 23(2), 758, 2023. https://doi.org/10.3390/s23020758
14. [5] "Determining Resampling Ratios Using BSMOTE and SVM-SMOTE for Identifying Rare Attacks in Imbalanced Cybersecurity Data." Computers, 12(10), 204, 2023. https://www.mdpi.com/2073-431X/12/10/204
15. [6] Li, Z. et al. "Mitigating Class Imbalance in Network Intrusion Detection with Feature-Regularized GANs." Future Internet, 17(5), 216, 2025. https://doi.org/10.3390/fi17050216
16. 7. 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, 2002. https://doi.org/10.1613/jair.953