Parameter-Free Negative Extreme Anomalous Undersampling Techniques on Class Imbalance Problems

Benjawan Jantamat, Krung Sinapiromsaran
Department of Mathematics and Computer Science, Faculty of Science,
Chulalongkorn University, Bangkok, Thailand, 10330

***Abstract*—This research addressed the critical challenge of class imbalance in classification, which is a prevalent issue in real-world applications. Standard classifiers often struggled with imbalanced datasets and frequently misclassified the minority class (positive instances) due to the overwhelming presence of the majority class (negative instances). The proposed Negative Extreme Anomalous Undersampling Technique (NEXUT) was introduced as a parameter-free approach. It leveraged the negative extreme anomalous score to strategically eliminate negative instances located in overlapping regions. This targeted removal was designed to improve the classifier's ability to effectively distinguish between the two classes. To evaluate the effectiveness of the proposed method, we conducted a comprehensive comparison with established undersampling techniques. The evaluation utilized both synthetic datasets and twelve datasets from the UCI repository. Six different classifiers were employed to ensure a diverse and unbiased performance assessment. Results from the Wilcoxon signed-rank test confirmed that the proposed method achieved significantly higher performance compared to existing techniques. These findings demonstrated the potential of NEXUT as a robust and valuable tool for addressing class imbalance problems.**

***Keywords*—Classification; class imbalance; imbalanced datasets; undersampling; parameter-free method; negative extreme anomalous score**

I. Introduction

Classification is an important topic in machine learning and data mining. This is especially true for class-imbalanced datasets, which frequently appear in many real-world applications. Examples include disease classification in the medical field [1], credit card fraud detection [2], satellite-based remote sensing [3], environmental pollution monitoring [4], and network intrusion detection [5]. A class imbalance problem [6] occurs when the number of instances in one class is significantly smaller than that of another. The smaller class is called the minority class, and the other is called the majority class. An instance in the minority class is referred to as a positive instance, whereas an instance in the majority class is referred to as a negative instance.

A classification algorithm [7] has difficulty when trained on class-imbalanced datasets because of the small proportion of minority class instances. A classifier generated by a standard classification algorithm tends to predict unknown instances as negative, which leads to the misclassification of most positive instances. However, correctly detecting positive instances is usually crucial. For example, fraud cases in fraud detection may cause significant financial losses, while missing cancer patients in cancer detection [8] can lead to life-threatening consequences.

To address the class imbalance problem, two methodologies have been proposed in the literature: data-level and non-data-level approaches [9]. Data-level algorithms aim to rebalance instances in a dataset so that standard classifiers can better recognize the decision boundary and classify instances from both classes correctly. In contrast, non-data-level algorithms modify the classification process itself to improve the detection of positive instances or assign higher penalty costs to misclassified positive instances. Data-level methods are generally more flexible because they can be applied to any classification algorithm and do not require deep knowledge of the classifier's internal design [10].

Two primary data-level methods for rebalancing datasets are oversampling and undersampling. Oversampling techniques balance data by adding or generating positive instances, which may not always represent valid cases in real-world situations. In contrast, undersampling techniques rebalance data by removing negative instances while preserving the characteristics of the minority class. However, oversampling increases the total number of instances, which leads to longer computational times for classification algorithms. In big data environments, oversampling may even require reimplementing algorithms to handle the larger datasets. Undersampling, on the other hand, reduces the number of instances and computational time [10], making it more suitable for large-scale data. Moreover, removing negative instances around the decision boundary of positive instances improves a classifier's ability to correctly predict positive instances. Therefore, this research focuses exclusively on undersampling techniques.

Several undersampling techniques have been proposed to address class imbalance, including Random Undersampling [11], Tomek Links [12], Cluster Centroids [13], and Near Miss [14]. However, these methods have certain limitations. The Random Undersampling removes negative instances randomly, which does not guarantee a clear decision boundary between classes. The Tomek Links technique removes negative instances from Tomek link pairs, which connect two instances from different classes as their nearest neighbors. As a result, it deletes only a small number of negative instances and rarely clarifies the decision boundary. The Cluster Centroid technique generates centroids of negative sub-clusters and retains them along with all positive instances. Although effective in reducing dataset size, this technique may also remove many negative instances near the boundary region, potentially reducing precision. The Near Miss technique retains only negative instances located close to positive instances. However, these selected negatives often come from boundary regions and may represent noisy or overlapping data that are difficult to classify.

In 2017, Chiamanusorn and Sinapiromsaran [15] proposed the negative extreme anomalous score (NaS) for use in oversampling techniques. The NaS of an instance is defined as the maximum possible radius of an open ball centered at that instance, such that no instances of the negative class are contained within the ball. The NaS represents the distance between the instance under consideration and its closest negative neighbor, without being influenced by positive instances.

The Negative Extreme Anomalous Undersampling Technique (NEXUT) is introduced in this study as a data-level approach that utilizes NaS to eliminate negative instances. It removes redundant negative instances and noise in the overlapping regions of both classes. This clarifies the boundary of the minority class, enabling classifiers to more effectively identify positive instances located near negative ones.

This study introduces a novel approach that:

- Offers flexibility by allowing any classifier to be applied to any dataset.
- Addresses the challenge of imbalanced data without requiring specific parameter settings.
- Enhances the decision boundary through undersampling techniques.

This study is organized into five sections: Section I presents the motivation for this study. Section II reviews related work. Section III describes the proposed NEXUT algorithms. Section IV reports the experimental results conducted on both synthetic datasets and twelve real-world datasets from the UCI repository. Finally, Section V concludes the study.

II. Related Work

Extensive research on undersampling techniques has focused on addressing class-imbalanced data. This effort has led to the development of methods that improved classification performance. In recent years, several surveys and systematic reviews have provided comprehensive discussions on the challenges of class imbalance and the evolution of related techniques [16], [17], [18], [19]. These studies highlight that early undersampling methods, such as Tomek Links [12], ENN [20], and CNN [21], remained important baselines. However, more recent approaches emphasized the need for robust and adaptive solutions, particularly in modern applications such as deep learning and remote sensing. Numerous undersampling techniques have been proposed and can be categorized into four main types.

A. Type I: Size-Based Undersampling

These techniques eliminated instances from the majority class until its size became equivalent to that of the minority class. Examples included Random Undersampling (RUS) [11], Cluster Centroids (ClusterCentroids) [13], and Clustering-Based Undersampling (CBU) [22].

- RUS was the simplest technique, randomly removing negative instances from the majority class. This approach tended to lose some important negatives.
- ClusterCentroids applied the K-means algorithm to represent negative instances with their centroids, while retaining all positive instances.
- CBU applied K-means clustering to select or generate negative instances that represented the majority class.

Limitation: These methods often removed informative negatives and ignored overlapping regions with the minority class.

B. Type II: Overlap-Based Undersampling

These techniques focused on eliminating negative instances in regions overlapping with positive instances. Examples included Tomek Links (TomekLinks) [12], One-Sided Selection (OSS) [23], Edited Nearest Neighbor (ENN) [20], Repeated ENN (RENN) [24], AllKNN [24], Neighborhood Cleaning Rule (NCL) [25], Majority Undersampling Technique (MUTE) [26], and Neighborhood-Based Undersampling Technique with Modified Tomek Link Search (NB-Tomek) [27].

- Tomek Links was a well-known method for addressing class overlap. It identified pairs of instances from different classes that were each other's nearest neighbors, and removed only the negative instances in these pairs.
- OSS applied the 1-NN rule to identify misclassified instances and then removed negative instances from Tomek links.
- ENN applied the k-NN rule, removing majority class instances if at least half of their k-nearest neighbors belonged to the minority class. By default, a 3-NN setting was used.
- RENN iteratively applied the ENN procedure until all remaining majority class instances had at least half of their k-nearest neighbors also belonging to the majority class.
- AllKNN used the k-NN rule across a range of k values (typically from 1 up to a specified k). It eliminated majority class instances if at least half of their neighbors were positive.
- NCL implemented a two-step cleaning process. First, it applied ENN to remove noise. Then, it used the k-NN rule to remove misclassified majority class instances. The default setting for k was 3.
- MUTE, proposed by Bunkhumpornpat et al. in 2011 [26], was later enhanced in 2014 with a safe-level graph [28]. The original version removed only noise, while the adaptive version could remove noise, borderline instances, or even reduce the majority class to core instances, depending on the safe-level configuration.
- NB-Tomek removed a majority class instance if it had a minority class neighbor and was also one of the k-nearest neighbors of that minority instance.

Limitation: These methods required parameter tuning (e.g., k in k-NN) and often failed to preserve the overall data distribution, discarding important negatives.

C. Type III: Neighbor-Based Undersampling

These techniques retained negative instances based on their neighbors. Examples included Near Miss (NearMiss) [14] and Condensed Nearest Neighbor (CNN) [21].

- NearMiss had three variants (NearMiss-1, 2, 3) that selected majority class instances based on different proximity criteria.
  - NearMiss-1 chose negative instances with the smallest average distance to their three nearest positive instances.
  - NearMiss-2 chose negatives closest to all positive instances, typically using the farthest three.
  - NearMiss-3 selected a fixed number of nearest negatives for each positive instance. However, this approach removed many negatives, often damaging the decision boundary of the minority class. Moreover, the selected negatives were difficult to classify.
- CNN used the nearest neighbor (NN) rule to identify misclassified instances located near the class boundary, which were retained as reference points for future classification.

Limitation: Neighbor-based methods often removed too many negatives, weakening the minority decision boundary.

D. Type IV: Anomaly-Based Undersampling

These techniques eliminated majority class instances by identifying them as anomalies or outliers relative to the data distribution. The assumption was that atypical majority instances were less representative and acted as noise in the learning process. Examples included Anomaly Scoring Based Ensemble (ASE) [29] and Geodesic Based Outlier Detection (GDLD) [30].

- ASE combined anomaly scores from multiple detectors to identify and remove unrepresentative majority class instances. Although effective, it required hyperparameter tuning for score normalization and ensemble weighting.
- GDLD leveraged geodesic distances and manifold properties to identify extreme negative outliers. While it preserved complex class boundaries, it often introduced significant computational overhead.

Limitation: Although anomaly-based undersampling was effective in detecting atypical negatives, most approaches depended heavily on predefined thresholds, scoring functions, or hyperparameters. This reliance reduced robustness across diverse datasets and complicated practical application.

Although a wide range of undersampling methods has been developed, several limitations remain [16], [17]. Many techniques required parameter tuning, and improper settings often led to the removal of informative negatives, degrading performance. Moreover, some methods failed to maintain the inherent data distribution of each class, particularly in overlapping regions. These weaknesses reduced the robustness and generalizability of current approaches, as emphasized in domain-specific contexts such as remote sensing and semi-supervised learning [19], [18].

To address these limitations, we propose the Negative Extreme Anomalous Undersampling Technique (NEXUT). Unlike existing anomaly-based methods, which relied heavily on hyperparameters or thresholds, NEXUT was entirely parameter-free. It automatically identified and removed extreme anomalous negatives while preserving all minority instances and maintaining the original class structure. This design ensured robustness across diverse datasets and avoided the performance degradation caused by improper parameter settings. Therefore, NEXUT introduced both a clear novelty and a practical advantage over existing anomaly-based approaches.

III. The Proposed Technique: NEXUT

This section describes the proposed Negative Extreme Anomalous Undersampling Technique (NEXUT). The technique aims to eliminate noisy or overlapping majority class instances while preserving all minority class instances. This ensures that the original distributions of both the majority and minority classes remain unchanged. NEXUT incorporates three distinct removal approaches, namely NEXUT-All, NEXUT-Plus, and NEXUT-PlusMinus.

Definition 3.1: Let D be a dataset with numeric attributes that contains a set of negative instances, Neg, representing the majority class, and a set of positive instances, Pos, representing the minority class. Let $\boldsymbol{x} \in D$ be an instance in the dataset. The extreme anomalous score of an instance $\boldsymbol{x}$, denoted as $EaS(\boldsymbol{x})$, is defined by the following notation:

$$EaS(\boldsymbol{x}) = sup\{d > 0 || B(\boldsymbol{x}, d) \cap (D \backslash \{\boldsymbol{x}\})| = 0\}$$

where, $B(\boldsymbol{x}, d)$ denotes an open ball of radius d centered at $\boldsymbol{x}$.

It should be noted that the negative extreme anomalous score, $NaS(\boldsymbol{x})$, for an instance $\boldsymbol{x} \in D$, is equivalent to $EaS(\boldsymbol{x})$ when it is calculated using only majority class instances:

$$NaS(\boldsymbol{x}) = sup\{d > 0 || B(\boldsymbol{x}, d) \cap (Neg \backslash \{\boldsymbol{x}\})| = 0\}.$$

NEXUT employs the negative extreme anomalous score (NaS) to identify majority class instances for removal, thereby rebalancing the dataset. It removes negative instances that lie on the boundary of an open ball centered at each instance, with a radius equal to its NaS value. This approach effectively eliminates majority class instances that lie close to minority class examples. Consequently, it expands the minority regions within the dataset. NEXUT offers three strategies for selecting which negative instances to remove: "All", "$Plus$", and "$PlusMinus$". The application of each selection approach defines a specific NEXUT variant: NEXUT-All, NEXUT-Plus, or NEXUT-PlusMinus. The NEXUT algorithm is presented in Algorithm 1.

Algorithm 1: Negative Extreme Anomalous Undersampling Technique (NEXUT) Algorithm

Input: Dataset D contains negative instances from a majority class, Neg, and positive instances from a minority class, Pos, and selection approach
Output: Undersampling dataset U

$DD = \varnothing$
n_diffNegPos $= |Neg| - |Pos|$
Compute the negative extreme anomalous score, $NaS(\boldsymbol{x})$, for each instance $\boldsymbol{x} \in D$
if selection approach == "*All*"
Sort the $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in D$ in descending order
for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** *n_diffNegPos* **do**
  Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in D$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
end
if selection approach == "*Plus*"
Sort the $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in Pos$ in ascending order
if $|Pos| >$ *n_diffNegPos* **then**
  for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** *n_diffNegPos* **do**
    Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
  end
else
  Collect all negative instances that lie on the boundary of an open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store them in DD
end
if selection approach == "*PlusMinus*"
Sort $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in Pos$ in ascending order
if $|Pos| >$ *n_diffNegPos* **then**
  for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** *n_diffNegPos* **do**
    Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
  end
else
  Collect all negative instances that lie on the boundary of an open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store them in DD
  Sort $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in Neg$ in ascending order
  for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** *n_diffNegPos* $- |DD|$ **do**
    Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in Neg$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
  end
end
return $U = D - DD$

Fig. 1 illustrates the flowchart of the proposed NEXUT technique. Given the training set D, the initial step is to compute the size difference between the majority and minority classes, defined as n_diffNegPos $= |Neg| - |Pos|$. Next, the negative extreme anomalous score (NaS) is computed for each instance in D. Then, a selection approach is applied to identify negative instances in the majority class D_{majority} that should be removed, resulting in a drop dataset DD.

Fig. 2a illustrates an example of a class-imbalanced dataset. Fig. 2b visualizes the corresponding NaS of the dataset. These illustrations highlight the problem setting that motivates NEXUT and provide an intuitive understanding of how NaS guides the removal of majority instances.

These three approaches lead to different algorithmic behaviors, depending on which classes are considered during the selection process. The "*All*" approach considers both negative and positive instances. The "*Plus*" approach considers only positive instances. The "*PlusMinus*" approach considers positive instances first, followed by negative instances. NEXUT then removes the selected negative instances in the set DD from the majority class D_{majority}. The resulting undersampled training set U is obtained by removing DD from the original dataset D.

A. NEXUT-All

The NEXUT-All technique employs the Negative Anomalous Score (NaS) to identify negative instances for removal while considering data from both classes. It begins by sorting all NaS values, from both the minority and majority classes, in descending order. Subsequently, it eliminates negative instances whose NaS rank is below the numerical difference between the minority and majority class sizes (see Fig. 2c).

To maintain a balance between the remaining negative and positive instances, the method also removes negative instances located near the boundaries of both classes. This removal process is biased toward instances closer to the minority class, since their NaS values generally tend to be higher than those of the majority class after sorting. The complete NEXUT-All algorithm, using the "*All*" selection approach, is detailed in Algorithm 1.

Sort the $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in D$ in descending order
for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** *n_diffNegPos* **do**
  Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in D$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
end

B. NEXUT-Plus

The NEXUT-Plus technique computes the Negative Anomalous Score (NaS) exclusively from positive instances. The process begins by sorting each NaS value (derived from positive instances) in ascending order. Negative instances whose NaS rank is less than the absolute numerical difference between the minority and majority class counts are then removed (Fig. 2d).

To ensure that the number of remaining negative instances does not drop below the number of positive instances, this method specifically targets negative instances located in close

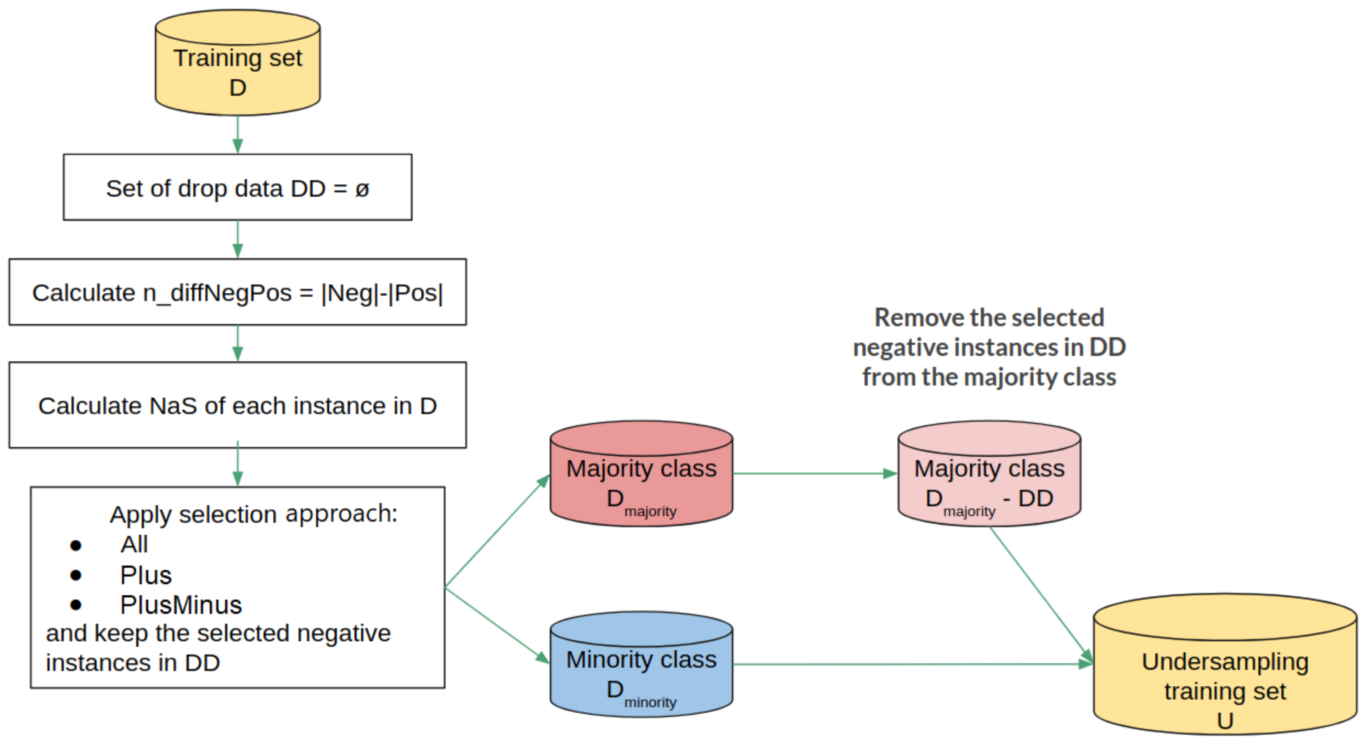


Fig. 1. The flowchart of NEXUT (Algorithm 1).

proximity to the minority class. The ascending sort order inherently prioritizes the removal of such negative instances.

It should be noted that when the numerical difference between the majority and minority classes equals or exceeds the number of positive instances, all negative instances associated with the minority class's NaS are removed. The complete algorithm for NEXUT-Plus, which applies the "$Plus$" selection approach, is detailed in Algorithm 1.

Sort the $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in Pos$ in ascending order
if $|Pos| > n\_diffNegPos$ **then**
  for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** $n\_diffNegPos$ **do**
    Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
  end
else
  Collect all negative instances that lie on the boundary of an open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store them in DD
end

C. NEXUT-PlusMinus

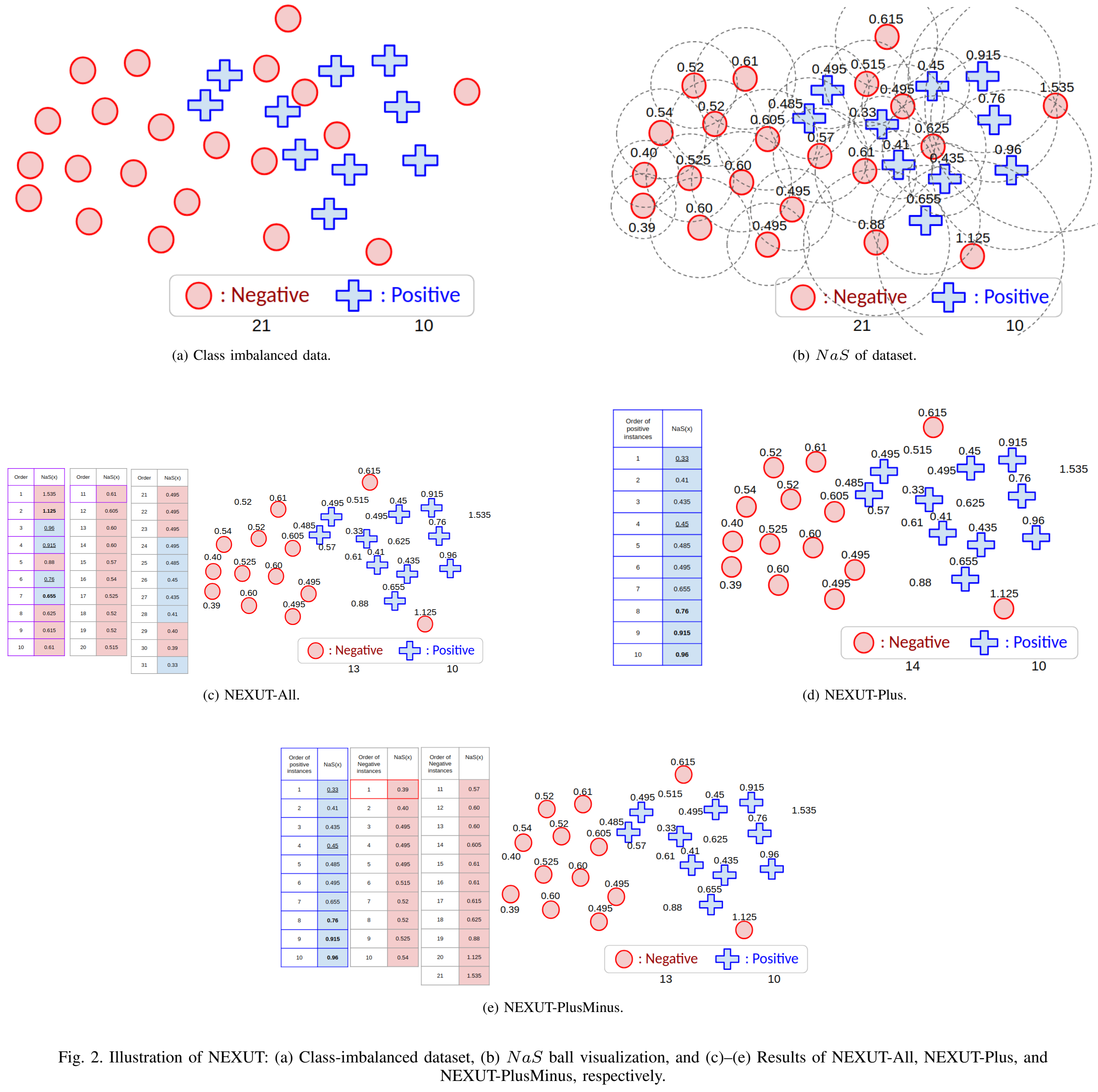
The NEXUT-PlusMinus technique extends the NEXUT-Plus method by employing a sequential two-stage removal strategy. In the first stage, it calculates the Negative Anomalous Score (NaS) exclusively from positive instances and sorts these values in ascending order. Negative instances whose NaS rank is less than the absolute difference between the minority and majority class counts are then removed.

In the second stage, NaS is computed for the remaining negative instances (majority class). These values are sorted in ascending order, and additional negative instances are removed until the total number of eliminated negatives across both stages equals the absolute class-size difference. This cumulative process, illustrated in Fig. 2e, ensures that the final number of remaining negative instances does not fall below the count of positive instances. The complete NEXUT-PlusMinus algorithm, which uses the "$PlusMinus$" selection approach, is detailed in Algorithm 1.

Sort $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in Pos$ in ascending order
if $|Pos| > n\_diffNegPos$ **then**
  for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** $n\_diffNegPos$ **do**
    Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
  end
else
  Collect all negative instances that lie on the boundary of an open ball centered at $\boldsymbol{x} \in Pos$ with a radius equal to $NaS(\boldsymbol{x})$, and store them in DD
  Sort $NaS(\boldsymbol{x})$ scores for all $\boldsymbol{x} \in Neg$ in ascending order
  for *ranking of* $NaS(\boldsymbol{x}) \leftarrow 1$ ***to*** $n\_diffNegPos - |DD|$ **do**
    Collect the negative instance that lies on the boundary of the open ball centered at $\boldsymbol{x} \in Neg$ with a radius equal to $NaS(\boldsymbol{x})$, and store it in DD
  end
end

IV. EXPERIMENT

This section presents the datasets used to assess the effectiveness of NEXUT compared with several established undersampling methods, namely AllKNN, NCL, OSS, and NB-Tomek. For baseline comparison, the original dataset without



(a) Class imbalanced data.

(b) NaS of dataset.

(c) NEXUT-All.

(d) NEXUT-Plus.

(e) NEXUT-PlusMinus.

Fig. 2. Illustration of NEXUT: (a) Class-imbalanced dataset, (b) NaS ball visualization, and (c)–(e) Results of NEXUT-All, NEXUT-Plus, and NEXUT-PlusMinus, respectively.

any resampling is also included. Further details regarding the experimental design and dataset properties are outlined below.

A. Experimental Setup

This study evaluates the effectiveness of the three proposed undersampling methods in comparison with well-established techniques, including AllKNN [24], NCL [25], OSS [23], and NB-Tomek [27]. A set of six classification algorithms is employed for the evaluation: the C4.5 decision tree [31], the k-nearest neighbors (k-NN) with $k = 3$ [32], the multilayer perceptron (MLP) [33], the naive Bayes [34], the random forest [35], and the logistic regression [36].

The experiments are conducted on a total of fourteen datasets: twelve real-world datasets obtained from the UCI Machine Learning Repository [37] and two synthetically generated datasets. Each experiment is performed with 30 replications of 10-fold cross-validation to ensure statistically robust and reliable results. Table I presents the parameter configurations used for each method. The experimental environment is based on Google Colaboratory (Colab), and all implementations are carried out using the Python programming language.

In the experiments, the selected undersampling techniques (AllKNN, NCL, OSS, and NB-Tomek) aim to remove negative instances based on their neighbors or to clean noisy data. However, these methods are not primarily designed to rebalance the dataset. Similar to these existing techniques, the proposed methods also focus on removing negative instances and reducing noise. Nevertheless, the proposed techniques are capable of eliminating a larger number of negative samples, which often leads to a more balanced dataset. In addition, they tend to preserve the underlying structure of the data distribution.

TABLE I. PARAMETER SETTINGS FOR THE PROPOSED TECHNIQUES AND COMPARISON TECHNIQUES

Techniques	Parameters setting
NEXUT-All	-
NEXUT-Plus	-
NEXUT-PlusMinus	-
AllKNN	n_neighbors = 3
NCL	n_neighbors = 3, threshold_cleaning = 0.5
OSS	n_seeds_S = 1
NB-Tomek	-

B. Datasets

To evaluate the performance of NEXUT-All, NEXUT-Plus, and NEXUT-PlusMinus against well-known undersampling methods such as AllKNN, NCL, OSS, and NB-Tomek, a total of fourteen datasets are used. These include twelve datasets from the UCI Machine Learning Repository [37] and two synthetically generated datasets. The twelve UCI datasets are Abalone, BreastCancer, Pima, BreastTissue, Haber, Glass, Vehicle, Ecoli, Yeast, Ozone8hr, Libras, and Ozone1hr. Table II presents descriptions of these datasets, arranged in ascending order based on their imbalance ratio (IR). The imbalance ratio is defined as $IR = \frac{|Neg|}{|Pos|}$, where $|Neg|$ and $|Pos|$ represent the numbers of negative and positive instances, respectively. A dataset is considered balanced if $IR = 1$, and imbalanced if $IR > 1$.

Synthetic datasets are used to evaluate whether the data distribution structure is preserved after applying each undersampling technique. These synthetic datasets are class-imbalanced and exhibit specific data distributions, namely the Moons dataset (Fig. 3a) and the Circles dataset (Fig. 3b) [38]. Each class-imbalanced Moons dataset and Circles dataset contains 200 negative instances and 20 positive instances, resulting in an imbalance ratio (IR) of 10.

C. Performance Evaluation

This research uses precision, recall, and F_1 score to evaluate the performance of the proposed techniques. These metrics are commonly applied in class-imbalanced research [39], [40]. Precision, recall, and F_1 score are computed using the following formulas:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

and

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where, TP denotes the number of positive instances accurately classified as positive, FP represents the number of negative instances misclassified as positive, and FN indicates the number of positive instances misclassified as negative.

D. Statistical Test

To compare the performance of each technique on each classifier, the Wilcoxon signed-rank test is employed. This test [41] is a non-parametric statistical method designed to evaluate paired performance outcomes between NEXUT and each of the alternative undersampling approaches. A two-tailed Wilcoxon signed-rank test with a 95% confidence level ($\alpha = 0.05$) is conducted, and the null and alternative hypotheses are formulated as follows:

H_0 : There is no significant difference in the median of the average F_1 scores between NEXUT and one of the other techniques.

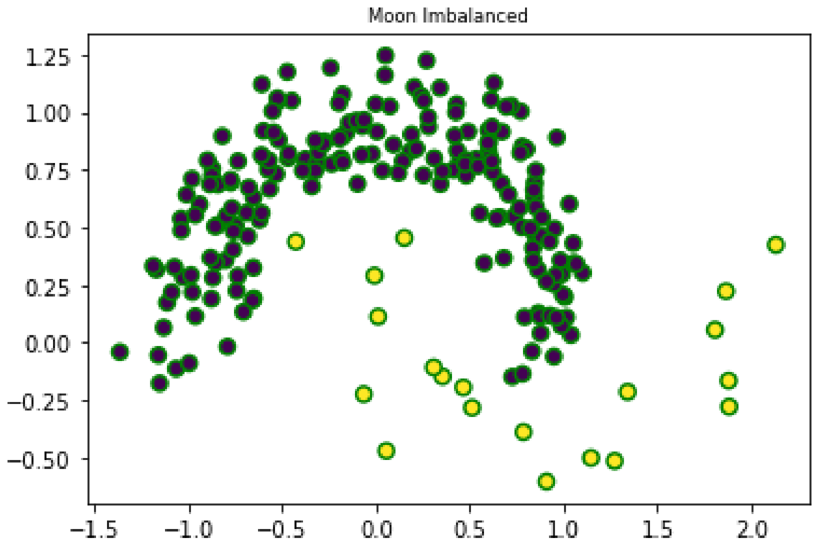
H_1 : There is a significant difference in the median of the average F_1 scores between NEXUT and one of the other techniques.

To perform the Wilcoxon signed-rank test, the differences in F_1 scores between NEXUT and one of the other techniques are calculated for each dataset. These differences are ranked after removing zero differences and ignoring the signs, with the remaining values sorted in ascending order. The sums of the ranks corresponding to the positive and negative differences are denoted as $Rk+$ and $Rk-$, respectively.

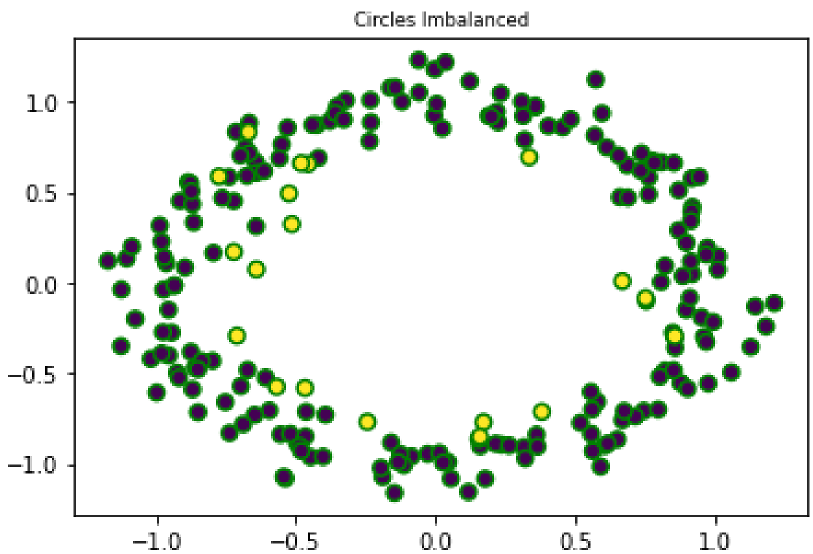
The test statistic (TS), used for comparison with the critical value, is defined as the smaller of $\{Rk+, Rk-\}$. If TS corresponds to $Rk-$, it suggests that NEXUT outperforms the compared method; otherwise, the alternative method performs better. When TS is less than the critical value, the null hypothesis is rejected, indicating a statistically significant difference in the median of average F_1 scores between NEXUT and the other technique.

E. Results

The average F_1 scores obtained from 30 replications of 10-fold cross-validation are reported for the original (unresampled) dataset and for seven resampling techniques: NEXUT-All, NEXUT-Plus, NEXUT-PlusMinus, AllKNN, NCL, OSS, and NB-Tomek. These results are generated using the six classifiers—namely, the decision tree (C4.5), the k-nearest neighbors ($k = 3$), the multilayer perceptron (MLP), the naive Bayes, the random forest, and the logistic regression—applied to twelve datasets from the UCI Machine Learning Repository. The results are summarized in Table III and Table IV. Table III presents the average F_1 scores for the decision tree (C4.5), the k-nearest neighbors ($k = 3$), and the multilayer perceptron (MLP). Table IV summarizes the results for the naive Bayes, the random forest, and the logistic regression. For each dataset, the techniques are ranked in descending order according to



(a) The Moons dataset.



(b) The Circles dataset.

Fig. 3. Synthetic datasets: (a) The Moons dataset and (b) The Circles dataset.

their average F_1 scores, with rank 1 assigned to the method that achieves the best performance.

As illustrated in Table III and Table IV, NEXUT-All consistently achieves the highest average rank when paired with the random forest classifier. NEXUT-Plus demonstrates the best performance with both the decision tree (C4.5) and the k-nearest neighbors (k-NN) classifiers. NEXUT-PlusMinus attains the top rank when evaluated with the logistic regression, while NCL performs best in combination with the multilayer perceptron (MLP). For the naive Bayes classifier, the highest average rank is shared by OSS and the original dataset without resampling.

Table V presents the statistical comparison between NEXUT and five other techniques—namely, the original dataset (without resampling), AllKNN, NCL, OSS, and NB-Tomek—across the six classifiers. The analysis is based on the Wilcoxon signed-rank test with a two-tailed evaluation at the 95% confidence level ($\alpha = 0.05$) using F_1 score. For twelve datasets ($n = 12$), the corresponding critical value is 13.

In Table V, the notation is interpreted as follows:

- A single plus sign (+) indicates that NEXUT outperforms the compared technique, but the result is not statistically significant ($TS = Rk-$, and $TS \geq 13$).
- A single minus sign (–) indicates that NEXUT underperforms, though again without statistical significance

TABLE II. THE DESCRIPTIONS OF UCI DATASETS

Datasets	#Inst	#Attr	Min. class	Maj. class	Pos. inst.	Neg. inst.	IR
Breast Cancer	683	9	'malignant'	'benign'	239 (34.99%)	444 (65.01%)	1.86
Pima	768	8	'1'	'0'	268 (34.90%)	500 (65.10%)	1.87
Breast Tissue	106	9	'CAR','FAD'	Others	36 (33.96%)	70 (66.04%)	1.94
Haber	306	3	'survived 5 years or longer'	'died within 5 years'	81 (26.47%)	225 (73.53%)	2.78
Glass	214	9	'5','6','7'	Others	51 (23.83%)	163 (76.17%)	3.20
Vehicle	846	18	'van'	Others	199 (23.52%)	647 (76.48%)	3.25
Ecoli	336	7	'im'	Others	77 (22.92%)	259 (77.08%)	3.36
Yeast	1484	8	'ME3','ME2','EXC','VAC','POX','ERL'	Others	304 (20.49%)	1180 (79.51%)	3.88
Ozone8hr	1848	72	'1'	'0'	128 (6.93%)	1720 (93.07%)	13.44
Libras	360	90	'1'	Others	24 (6.67%)	336 (93.33%)	14.00
Abalone	731	7	'18'	'9'	42 (5.75%)	689 (94.25%)	16.40
Ozone1hr	1848	72	'1'	'0'	57 (3.08%)	1791 (96.92%)	31.42

($TS = Rk+$, and $TS \geq 13$).

- A double plus sign (++) denotes that NEXUT's performance is significantly better at the 95% confidence level ($TS = Rk-$ and $TS < 13$).
- A double minus sign (–) signifies that NEXUT performs significantly worse ($TS = Rk+$ and $TS < 13$).

From the statistical results of F_1 score, NEXUT-All's F_1 score is better than all other techniques for the decision tree classifier, the multilayer perceptron, the random forest, and the logistic regression. Notably, for the logistic regression, NEXUT-All performs significantly better than all other techniques. NEXUT-All's F_1 score for the k-NN classifier is better than other techniques except OSS. Moreover, NEXUT-All's F_1 score is higher than those of NCL and NB-Tomek for all classifiers.

Regarding NEXUT-Plus, its F_1 score is better than all other techniques for the decision tree classifier, the k-NN classifier, the multilayer perceptron, and the random forest. Furthermore, NEXUT-Plus's F_1 score is higher than that of NB-Tomek for all classifiers.

For NEXUT-PlusMinus, its F_1 score is better than all other techniques for the random forest and the logistic regression. NEXUT-PlusMinus achieves better F_1 scores than other techniques for the decision tree classifier, except for NCL, where both methods perform equally. Additionally, NEXUT-PlusMinus's F_1 score is higher than that of NB-Tomek for all classifiers.

From the statistical results of F_1 score, NEXUT-All is suitable for use with the decision tree classifier, the multilayer perceptron, the random forest, and the logistic regression. NEXUT-Plus is suitable for use with the decision tree classifier, the k-NN classifier, the multilayer perceptron, and the random forest. NEXUT-PlusMinus is suitable for use with the random forest and the logistic regression.

For the synthetic datasets (the Moons dataset and the Circles dataset), the results after applying each undersampling technique, as well as the original datasets without applying any undersampling technique, are illustrated in Fig. 4 and Fig. 5, respectively. The proposed techniques successfully preserve the overall structure of the data distribution, even after a significant number of negative instances are removed.

Table VI and Table VII present the average F_1 scores obtained using the six classifiers: the decision tree classifier (C4.5) [31], the k-nearest neighbors ($k = 3$) [32], the multilayer perceptron [33], the naive Bayes [34], the random forest [35], and the logistic regression [36], evaluated on the Moons dataset and the Circles dataset, respectively. The techniques are ranked for each classifier in descending order of F_1 score, with Rank 1 representing the best-performing method.

For the Moons dataset, the results presented in Table VI indicate that NEXUT-PlusMinus achieves the highest F_1 score rank for the k-NN classifier, the multilayer perceptron, and the random forest. NEXUT-All obtains the highest rank for the logistic regression. For the decision tree classifier, NCL and NB-Tomek share the highest F_1 score rank. NCL also secures the highest rank for the naive Bayes classifier. Considering F_1 scores across all techniques, NEXUT-PlusMinus demonstrates the highest overall average rank.

For the Circles dataset, the results presented in Table VII indicate that NEXUT-All achieves the highest F_1 score rank for the decision tree classifier and the random forest. NCL secures the highest rank for the k-NN classifier. For the multilayer perceptron, the naive Bayes, and the logistic regression, several techniques demonstrate the highest F_1 score rank, notably NEXUT-All, NEXUT-Plus, NEXUT-PlusMinus, AllKNN, NCL, OSS, and the original dataset. It is important to note that, for these specific classifiers, all techniques except NB-Tomek exhibit comparable performance. Considering the overall F_1 scores for all techniques on the Circles dataset, NEXUT-All achieves the highest average rank.

F. Discussion

The experimental results demonstrate that the proposed NEXUT variants consistently improve classification performance across multiple classifiers and datasets. Specifically, NEXUT-All yields superior F_1 scores when applied to four classifiers: the decision tree, the multilayer perceptron (MLP), the random forest, and the logistic regression. These results indicate the consistent performance of NEXUT-All across different learning models. NEXUT-Plus performs best with the decision tree, the k-NN, the MLP, and the random forest classifiers, while NEXUT-PlusMinus excels with the random forest and the logistic regression classifiers. These findings highlight the effectiveness of the NEXUT approach in selectively removing extreme anomalous negative instances while preserving the overall class distribution.

When compared with established undersampling techniques such as AllKNN [24], NCL [25], OSS [23], and NB-Tomek [27], NEXUT achieves consistently higher or comparable performance across multiple classifiers. As shown in Table V, NEXUT-All significantly outperforms OSS and NB-Tomek on the logistic regression and the random forest classifiers at the 95% confidence level. This result aligns with previous findings showing that OSS [23] and NB-Tomek [27] sometimes removed informative instances, thereby reducing classifier performance. Similarly, NCL [25], which was designed to clean overlapping examples, was sometimes too aggressive in certain domains.

These findings are consistent with Batista et al. [42], who observed that Tomek-based approaches were effective at reducing class overlap. However, they sometimes removed informative instances, which degraded classifier performance. In addition, Drummond and Holte [43] reported that random or overly aggressive undersampling could eliminate useful majority instances. Chawla et al. [44] emphasized the importance of maintaining the class distribution to achieve balanced decision boundaries. More recently, Blagus and Lusa [45] noted that oversampling methods, such as SMOTE and its variants, could alter data variability and introduce correlations between samples, which reduced classifier robustness.

By contrast, NEXUT minimizes these drawbacks by targeting only extreme anomalous negatives. It also avoids excessive data removal and synthetic data generation, thereby preserving meaningful sample diversity while enhancing classification performance.

Despite these advantages, NEXUT has some limitations. It is primarily designed for binary class-imbalanced datasets, and its extension to multi-class scenarios requires further investigation. In cases of extreme data sparsity within the majority class, NEXUT's ability to retain representative instances is reduced. This limitation may affect classifiers that rely heavily on boundary information. While NEXUT is computationally efficient for moderate-sized datasets, further optimization will be necessary for large-scale applications.

In terms of practical applications, NEXUT is well-suited to domains where class imbalance is prevalent and accurate minority detection is critical. For example, in medical diagnosis, detecting rare diseases requires minimizing false negatives to avoid life-threatening errors. In fraud detection, reducing the influence of anomalous majority instances improves the identification of rare fraudulent transactions. In industrial anomaly detection, NEXUT enhances reliability by retaining representative operational data while isolating rare faults. Similarly, in cybersecurity, malicious activities often represent only a very small fraction of network traffic. In such cases, NEXUT helps to preserve decision boundaries that are critical for intrusion detection. By reducing the impact of extreme anomalous negatives, while maintaining meaningful class distributions, NEXUT improves classifier reliability in real-world scenarios where parameter tuning is challenging.

Overall, NEXUT demonstrates higher predictive performance compared to conventional undersampling methods while preserving the class distribution. Future work should explore integrating NEXUT with oversampling techniques to further enhance performance and address more challenging imbalanced datasets.

V. Conclusion

In this research, the Negative Extreme Anomalous Undersampling Technique (NEXUT) is proposed to address the imbalanced classification problem. Unlike other undersampling techniques from the imbalanced-learn Python package, which require user-defined parameters, NEXUT is parameter-free. It outperforms the existing undersampling techniques on continuous, binary, class-imbalanced datasets across the six classifiers. NEXUT applies negative extreme anomalous scores to identify negative instances for elimination. The negative instances selected by NEXUT may be noisy, affecting both the minority and majority classes, or they may simply be redundant. NEXUT effectively addresses overlapping instances between classes while maintaining the data distribution of each class. Moreover, it reduces the size of the majority class, resulting in a more balanced dataset. Thus, the proposed technique helps rebalance class instances.

For future work, NEXUT will be improved by integrating an oversampling technique to enhance performance efficiency. It is suggested to prioritize oversampling the minority class to balance the data, followed by applying undersampling to the majority class to remove noise and potentially overlapping instances.

Acknowledgment

We gratefully acknowledge the financial support provided by the Science Achievement Scholarship of Thailand (SAST) and the Applied Mathematics and Computational Science (AMCS) program. We also thank the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Thailand, for their support. Our sincere appreciation also goes to the anonymous reviewers for their constructive comments, which greatly enhanced this manuscript. The authors confirm that a large language model (LLM) was employed to improve the linguistic quality and readability of this manuscript. However, the integrity of the scientific content and the validity of the conclusions presented remain solely with the authors.

TABLE III. THE F_1 SCORE PERFORMANCE ON THE DECISION TREE CLASSIFIER (C4.5), THE K-NEAREST NEIGHBOR CLASSIFIER ($k = 3$), AND THE MULTILAYER PERCEPTRON FOR UCI DATASETS

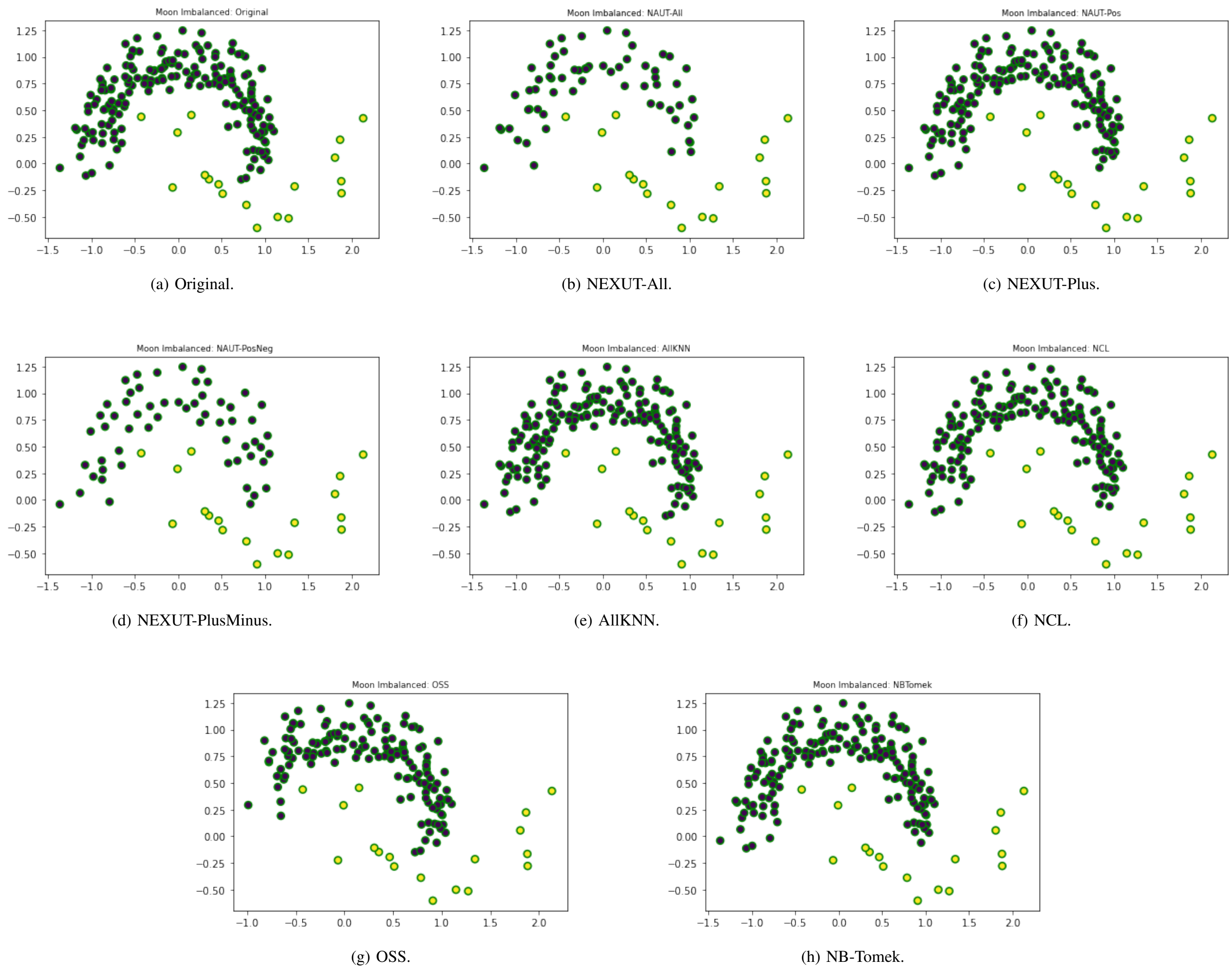
Dataset	**Original**		**NEXUT-All**		**NEXUT-Plus**		**NEXUT-PlusMinus**		**AllKNN**		**NCL**		**OSS**		**NB-Tomek**	
	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk
Decision tree classifier (C4.5)																
BreastCancer	0.9389	7	0.9536	4	0.9551	3	0.9552	2	0.9505	5	0.9472	6	0.9259	8	**0.9558**	**1**
Pima	0.6751	6	0.6927	3	**0.6994**	**1**	0.6933	2	0.6718	7	0.6754	5	0.6883	4	0.5319	8
BreastTissue	0.7643	3	**0.7774**	**1**	0.7544	4	0.7089	6	0.6540	8	0.7245	5	0.7656	2	0.6850	7
Haber	0.5447	6	**0.5831**	**1**	0.5642	2	0.5521	3	0.5306	7	0.5461	5	0.5490	4	0.4039	8
Glass	0.9072	7	0.9165	5	0.9289	3	0.9290	2	0.9094	6	0.9251	4	0.8953	8	**0.9382**	**1**
Vehicle	**0.9066**	**1**	0.8954	4	0.8950	5	0.8834	7	0.8916	6	0.8985	2	0.8957	3	0.8079	8
Ecoli	0.8286	7	0.8426	5	0.8472	4	0.8503	3	0.8393	6	0.8508	2	**0.8543**	**1**	0.8025	8
Yeast	0.7787	6	0.7894	3	**0.7962**	**1**	0.7608	7	0.7835	5	0.7895	2	0.7846	4	0.6875	8
Ozone8hr	0.6308	7	0.6310	6	0.6359	3	**0.6464**	**1**	0.6311	5	0.6404	2	0.6315	4	0.5943	8
Libras	0.7461	4	**0.7900**	**1**	0.7252	8	0.7815	2	0.7330	6	0.7257	7	0.7467	3	0.7354	5
Abalone	0.6377	3	0.6157	7	**0.6412**	**1**	0.6235	6	0.6375	4	0.6382	2	0.6262	5	0.6061	8
Ozone1hr	0.5909	7	0.5904	8	0.5977	3	0.5916	6	**0.6065**	**1**	0.5963	4	0.5956	5	0.5992	2
Average rank		5.33		4.00		**3.17**		3.92		5.50		3.83		4.25		6.00
K-nearest neighbor classifier ($K = 3$)																
BreastCancer	0.9664	8	0.9756	3	**0.9765**	**1.5**	**0.9765**	**1.5**	0.9739	5	0.9756	4	0.9681	7	0.9719	6
Pima	0.6586	4	0.6625	3	0.6636	2	0.6505	5	0.6442	6	0.6436	7	**0.6669**	**1**	0.4713	8
BreastTissue	**0.7300**	**1**	0.6692	5	0.6898	3	0.6695	4	0.5481	7	0.6408	6	0.7270	2	0.5249	8
Haber	0.5705	6	**0.6165**	**1**	0.6034	2	0.5884	4	0.5699	7	0.5972	3	0.5748	5	0.4027	8
Glass	0.9151	6	0.9146	7	0.9364	2	**0.9371**	**1**	0.9087	8	0.9358	3	0.9159	5	0.9232	4
Vehicle	**0.9266**	**1**	0.8788	6	0.8880	4	0.8657	7	0.8854	5	0.8954	3	0.9245	2	0.7983	8
Ecoli	0.8622	3	0.8564	6	**0.8700**	**1**	0.8592	5	0.8637	2	0.8534	7	0.8613	4	0.8448	8
Yeast	0.8038	4	**0.8126**	**1**	0.8124	2	0.7774	7	0.8019	6	0.8025	5	0.8086	3	0.6999	8
Ozone8hr	0.5706	7	0.5820	4	0.5792	5	0.5932	2	**0.5995**	**1**	0.5878	3	0.5771	6	0.5442	8
Libras	**0.8861**	**1**	0.8263	7	0.8221	8	0.8359	5	0.8574	3	0.8554	4	0.8830	2	0.8317	6
Abalone	0.6468	8	0.7029	3	0.6986	4	0.7077	2	0.6671	6	**0.7137**	**1**	0.6592	7	0.6901	5
Ozone1hr	0.5028	6	0.5134	3	0.5012	8	0.5160	2	0.5084	5	0.5111	4	0.5021	7	**0.5285**	**1**
Average rank		4.58		4.08		**3.54**		3.79		5.08		4.17		4.25		6.50
Multilayer perceptron																
BreastCancer	0.9616	7	0.9649	4	0.9648	5	0.9651	2	0.9647	6	**0.9659**	**1**	0.8171	8	0.9650	3
Pima	0.6465	4	0.6744	2	**0.6754**	**1**	0.6449	5	0.6163	7	0.6354	6	0.6499	3	0.3953	8
BreastTissue	0.5247	4	0.5258	3	**0.5442**	**1**	0.5164	6	0.4990	7	0.5236	5	0.5394	2	0.4773	8
Haber	0.5850	7	0.5934	5	**0.6383**	**1**	0.5887	6	0.6001	4	0.6154	2	0.6133	3	0.2305	8
Glass	0.6579	8	0.7057	3	0.6808	6	0.7139	2	0.6829	5	0.6875	4	**0.7188**	**1**	0.6737	7
Vehicle	0.8928	2	**0.8948**	**1**	0.8840	5	0.8711	7	0.8845	4	0.8849	3	0.8759	6	0.8423	8
Ecoli	0.8203	2	0.8111	7	0.8021	8	0.8131	5	**0.8284**	**1**	0.8154	4	0.8117	6	0.8187	3
Yeast	0.7829	7	**0.7972**	**1**	0.7933	4	0.7832	6	0.7965	2	0.7950	3	0.7903	5	0.7378	8
Ozone8hr	0.5187	5	0.5187	4	0.5138	7	0.5288	3	0.5170	6	0.5310	2	0.5053	8	**0.5565**	**1**
Libras	0.8542	6	0.7550	8	0.8584	4	0.7663	7	0.8601	3	0.8603	2	**0.8658**	**1**	0.8570	5
Abalone	0.5498	4	0.5831	2	0.5510	3	0.5427	7	0.5491	5	0.5473	6	0.5305	8	**0.6045**	**1**
Ozone1hr	0.4913	8	0.4919	5	0.4924	4	0.4945	2	0.4918	6	0.4925	3	0.4913	7	**0.5048**	**1**
Average rank		5.33		3.75		4.08		4.83		4.67		**3.42**		4.83		5.08

TABLE IV. THE F_1 SCORE PERFORMANCE ON THE NAIVE BAYES, THE RANDOM FOREST, AND THE LOGISTIC REGRESSION FOR UCI DATASETS

Dataset	Original		NEXUT-All		NEXUT-Plus		NEXUT-PlusMinus		AllKNN		NCL		OSS		NB-Tomek	
	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk
Naive Bayes																
BreastCancer	**0.9589**	**1**	0.9536	5	0.9536	4	0.9539	3	0.9498	7	0.9512	6	0.9560	2	0.9386	8
Pima	0.7237	2	0.7186	5	0.7229	3	**0.7266**	**1**	0.6810	7	0.6990	6	0.7187	4	0.5968	8
BreastTissue	0.6370	3	**0.6415**	**1**	0.6348	4	0.6335	5	0.6277	8	0.6322	6	0.6403	2	0.6279	7
Haber	0.5604	7	0.6372	2	0.6207	5	0.6224	4	**0.6523**	**1**	0.6350	3	0.5802	6	0.4018	8
Glass	0.8652	8	**0.8847**	**1**	0.8730	5	0.8766	2	0.8751	3	0.8728	6	0.8716	7	0.8738	4
Vehicle	0.6354	4	0.6321	8	0.6345	6	**0.6411**	**1**	0.6333	7	0.6347	5	0.6384	2	0.6366	3
Ecoli	0.6648	6	0.6409	7	0.6687	3	0.6263	8	**0.6704**	**1**	0.6698	2	0.6676	4	0.6676	5
Yeast	0.6604	4	0.4965	7	0.6662	3	**0.7242**	**1**	0.4968	5	0.4954	8	0.6802	2	0.4968	6
Ozone8hr	**0.5256**	**1**	0.5250	3	0.5214	4	0.5050	7	0.5161	6	0.5161	5	0.5252	2	0.4857	8
Libras	**0.8292**	**1**	0.8152	3	0.8137	4	0.8029	6	0.8178	2	0.8108	5	0.7926	8	0.7973	7
Abalone	0.5890	2	0.5326	8	0.5814	3	0.5775	5	0.5793	4	0.5724	6	**0.5927**	**1**	0.5375	7
Ozone1hr	0.4904	2	0.4864	4	0.4887	3	0.4772	7	0.4852	6	0.4857	5	**0.4909**	**1**	0.4651	8
Average rank		**3.42**		4.50		3.92		4.17		4.75		5.25		**3.42**		6.58
Random forest																
BreastCancer	0.9710	7	0.9742	2	**0.9745**	**1**	0.9742	3	0.9740	4	0.9739	5	0.9709	8	0.9730	6
Pima	0.7181	5	0.7338	2	**0.7341**	**1**	0.7259	4	0.6830	7	0.7005	6	0.7322	3	0.4857	8
BreastTissue	0.7758	3	0.7737	4	0.7931	2	0.7322	5	0.6403	8	0.7071	6	**0.8013**	**1**	0.6639	7
Haber	0.5583	7	**0.6097**	**1**	0.6044	2	0.5980	3	0.5743	5	0.5852	4	0.5727	6	0.3328	8
Glass	0.9326	5	0.9275	6	0.9394	2	0.9386	3	0.9175	8	**0.9400**	**1**	0.9266	7	0.9381	4
Vehicle	**0.9568**	**1**	0.9127	5	0.9225	4	0.9034	6	0.8988	7	0.9234	3	0.9397	2	0.8353	8
Ecoli	0.8502	7	0.8527	5	**0.8673**	**1**	0.8532	3	0.8531	4	0.8511	6	0.8571	2	0.8428	8
Yeast	0.8274	6	0.8297	5	**0.8367**	**1**	0.8260	7	0.8318	4	0.8331	3	0.8353	2	0.7453	8
Ozone8hr	0.5334	8	0.6475	2	0.5704	6	**0.6952**	**1**	0.6357	3	0.6276	5	0.5482	7	0.6314	4
Libras	0.6053	8	**0.7632**	**1**	0.6325	6	0.7469	2	0.6151	7	0.6333	5	0.6404	4	0.6494	3
Abalone	0.5662	8	**0.6563**	**1**	0.5843	6	0.6256	3	0.5941	5	0.6249	4	0.5717	7	0.6469	2
Ozone1hr	0.4921	4	0.5137	3	0.4921	6	0.5230	2	0.4920	8	0.4921	7	0.4921	5	**0.6448**	**1**
Average rank		5.75		**3.08**		3.17		3.50		5.83		4.58		4.50		5.58
Logistic regression																
BreastCancer	0.9638	8	0.9678	6	0.9682	3.5	0.9682	3.5	**0.9695**	**1**	0.9680	5	0.9672	7	0.9691	2
Pima	0.7334	3	**0.7420**	**1**	0.7369	2	0.7315	4	0.6925	7	0.7041	6	0.7282	5	0.5206	8
BreastTissue	0.7853	3	0.7821	5	0.7949	2	0.7823	4	0.6675	7	0.7721	6	**0.7984**	**1**	0.6499	8
Haber	0.5363	7	**0.6556**	**1**	0.6237	5	0.6299	4	0.6517	3	0.6527	2	0.5728	6	0.2636	8
Glass	0.8832	8	0.8892	4	0.8974	3	**0.9043**	**1**	0.8862	7	0.8999	2	0.8879	5	0.8875	6
Vehicle	0.9547	2	0.9433	5	0.9469	4	0.9327	7	0.9376	6	0.9522	3	**0.9560**	**1**	0.8852	8
Ecoli	0.7944	8	0.8276	3	0.8003	7	**0.8358**	**1**	0.8221	4	0.8168	5	0.8125	6	0.8311	2
Yeast	0.6226	8	0.7767	2	0.7162	6	**0.7848**	**1**	0.7492	5	0.7509	4	0.6602	7	0.7510	3
Ozone8hr	0.6195	7	0.6781	2	0.6528	5	**0.6936**	**1**	0.6708	4	0.6778	3	0.6333	6	0.6074	8
Libras	0.7045	8	**0.7766**	**1**	0.7139	7	0.7740	2	0.7144	6	0.7177	4	0.7171	5	0.7247	3
Abalone	0.5047	8	0.5733	2	0.5074	6	0.5223	3	0.5132	5	0.5198	4	0.5053	7	**0.5734**	**1**
Ozone1hr	0.5434	7	0.5710	3	0.5467	6	0.5825	2	0.5545	4	0.5485	5	0.5426	8	**0.6035**	**1**
Average rank		6.42		2.92		4.71		**2.79**		4.92		4.08		5.33		4.83

TABLE V. THE F_1 SCORE STATISTICAL RESULTS FOR UCI DATASETS

F_1 score of	Decision tree				K-NN classifier				MLP				Naive Bayes				Random forest				Logistic regression			
NEXUT-All vs	**Rk+**	**Rk-**	**TS**	**Sign**	**Rk+**	**Rk-**	**TS**	**Sign**	**Rk+**	**Rk-**	**T**	**Sign**	**Rk+**	**Rk-**	**TS**	**Sign**	**Rk+**	**Rk-**	**TS**	**Sign**	**Rk+**	**Rk-**	**TS**	**Sign**
Original	61	17	17	(+)	42	36	36	(+)	59	19	19	(+)	23	55	23	(-)	64	14	14	(+)	72	6	6	(++)
AllKNN	61	17	17	(+)	53	25	25	(+)	54	24	24	(+)	39	39	39	(+/-)	73	5	5	(++)	77	1	1	(++)
NCL	45	33	33	(+)	39	39	39	(+)	42	36	36	(+)	50	28	28	(+)	66	12	12	(++)	67	11	11	(++)
OSS	57	21	21	(+)	37	41	37	(-)	47	31	31	(+)	28	50	28	(-)	54	24	24	(+)	69	9	9	(++)
NB-Tomek	71	7	7	(++)	67	11	11	(++)	52	26	26	(+)	63	15	15	(+)	64	14	14	(+)	65	13	13	(++)
NEXUT-Plus vs																								
Original	59	19	19	(+)	46	32	32	(+)	59	19	19	(+)	35	43	35	(-)	67	11	11	(++)	72	6	6	(++)
AllKNN	66	12	12	(++)	56	22	22	(+)	42	36	36	(+)	56	22	22	(+)	63	15	15	(+)	34	44	34	(-)
NCL	61	17	17	(+)	43	35	35	(+)	39	39	39	(+)	64	14	14	(+)	46	32	32	(+)	20	58	20	(-)
OSS	57	21	21	(+)	43	35	35	(+)	56	22	22	(+)	33	45	33	(-)	56	22	22	(+)	55	23	23	(+)
NB-Tomek	68	10	10	(++)	69	9	9	(++)	53	25	25	(+)	74	4	4	(++)	54	24	24	(+)	49	29	29	(+)
NEXUT-PlusMinus vs																								
Original	47	31	31	(+)	37	41	37	(-)	33	45	33	(-)	33	45	33	(-)	60	18	18	(+)	69	9	9	(++)
AllKNN	57	21	21	(+)	48	30	30	(+)	38	40	38	(-)	36	42	36	(-)	74	4	4	(++)	69	9	9	(++)
NCL	39	39	39	(+/-)	33	45	33	(-)	19	59	19	(-)	37	41	37	(-)	63	15	15	(+)	63	15	15	(+)
OSS	44	34	34	(+)	31	47	31	(-)	31	47	31	(-)	39	39	39	(+/-)	51	27	27	(+)	69	9	9	(++)
NB-Tomek	72	6	6	(++)	75	3	3	(++)	50	28	28	(+)	69	9	9	(++)	64	14	14	(+)	65	13	13	(++)

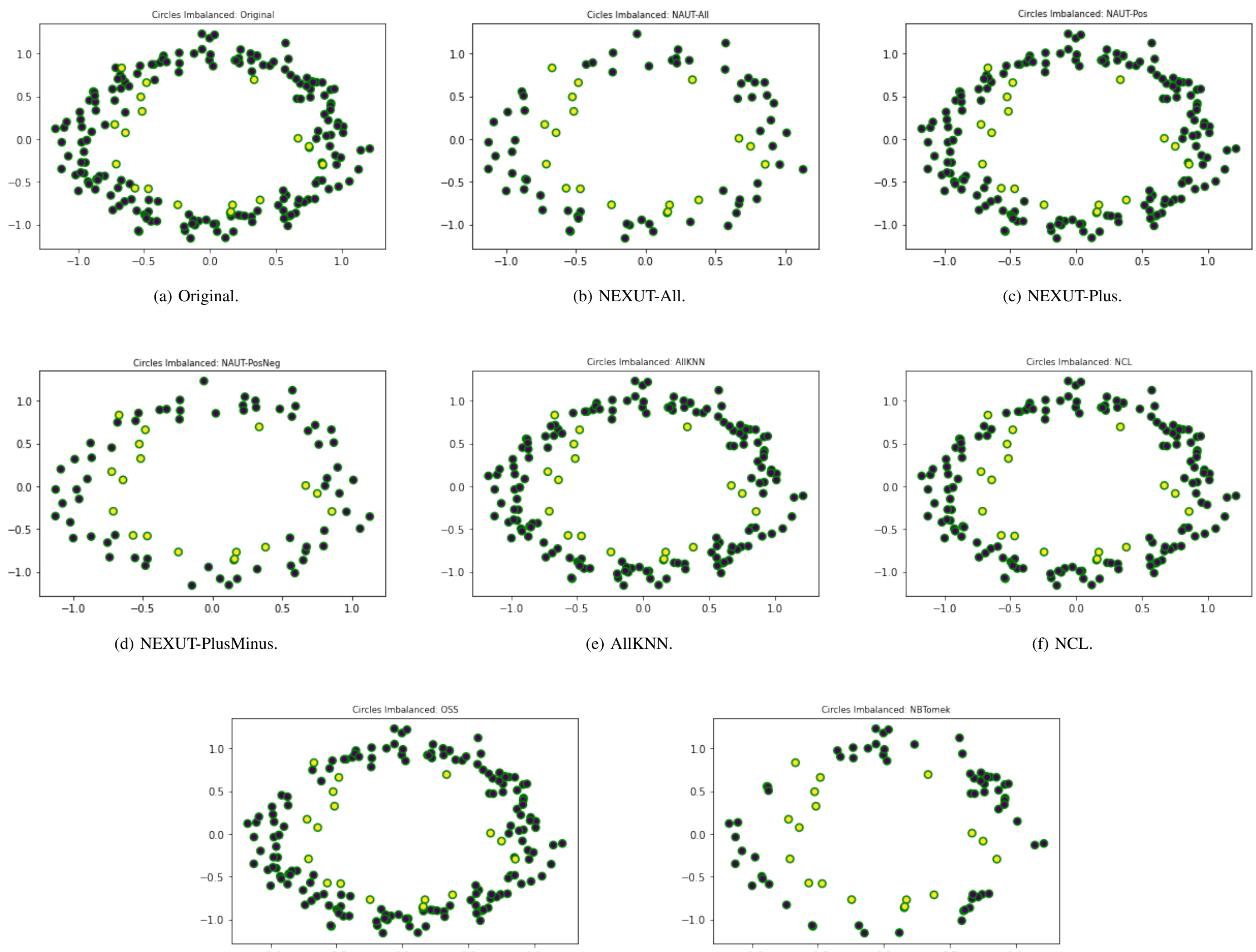


(a) Original. (b) NEXUT-All. (c) NEXUT-Plus.

(d) NEXUT-PlusMinus. (e) AllKNN. (f) NCL.

(g) OSS. (h) NB-Tomek.

Fig. 4. The Moons dataset after using each undersampling technique and without technique.



(a) Original.

(b) NEXUT-All.

(c) NEXUT-Plus.

(d) NEXUT-PlusMinus.

(e) AllKNN.

(f) NCL.

(g) OSS.

(h) NB-Tomek.

Fig. 5. The Circles dataset after using each undersampling technique and without technique.

TABLE VI. THE F_1 SCORE PERFORMANCES ON THE SIX CLASSIFIERS FOR THE MOONS DATASET

Classifiers	Original		NEXUT-All		NEXUT-Plus		NEXUT-PlusMinus		AllKNN		NCL		OSS		NB-Tomek	
	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk
Decision Tree	0.8366	8	0.8507	6	0.8688	3	0.8652	4	0.8383	7	**0.8692**	**1.5**	0.8643	5	**0.8692**	**1.5**
K-NN classifier	0.9442	8	0.9476	5	0.9605	2	**0.9644**	**1**	0.9460	6	0.9592	3	0.9453	7	0.9559	4
MLP	0.7013	7	0.7970	2	0.7210	4	**0.8001**	**1**	0.6978	8	0.7135	5	0.7278	3	0.7082	6
Naive Bayes	0.8295	5	0.7814	8	0.8369	3	0.7973	6	0.8299	4	**0.8420**	**1**	0.7940	7	0.8378	2
Random Forest	0.8684	8	0.8838	2	0.8822	3	**0.8927**	**1**	0.8691	7	0.8756	5	0.8723	6	0.8776	4
Logistic Regression	0.7030	7.5	**0.7993**	**1**	0.7443	3	0.7875	2	0.7030	7.5	0.7280	5	0.7333	4	0.7257	6
Average Rank		7.25		4.00		3.00		**2.50**		6.58		3.42		5.33		3.92

TABLE VII. THE F_1 SCORE PERFORMANCES ON THE SIX CLASSIFIERS FOR THE CIRCLES DATASET

Classifiers	Original		NEXUT-All		NEXUT-Plus		NEXUT-PlusMinus		AllKNN		NCL		OSS		NB-Tomek	
	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk	F_1	Rk
Decision Tree	0.6216	5	**0.6356**	**1**	0.6239	4	0.6240	3	0.6346	2	0.6208	6	0.6085	7	0.5544	8
K-NN classifier	0.5293	8	0.6332	2	0.6037	5	0.6154	4	0.6201	3	**0.6424**	**1**	0.5825	6	0.5434	7
MLP	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	0.4757	8
Naive Bayes	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	0.4684	8
Random Forest	0.5678	8	**0.6416**	**1**	0.6121	5	0.6403	2	0.6309	4	0.6309	3	0.5997	6	0.5944	7
Logistic Regression	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	**0.4762**	**4**	0.4738	8
Average Rank		5.50		**2.67**		4.33		3.50		3.50		3.67		5.17		7.67

REFERENCES

[1] M. Saini and S. Susan, "Deep transfer with minority data augmentation for imbalanced breast cancer dataset," *Applied Soft Computing*, vol. 97, p. 106759, 2020.

[2] F. Z. El Hlouli, J. Riffi, M. A. Mahraz, A. Yahyaouy, K. El Fazazy, and H. Tairi, "Credit card fraud detection: Addressing imbalanced datasets with a multi-phase approach," *SN Computer Science*, vol. 5, no. 1, p. 173, 2024.

[3] A. Naboureh, H. Ebrahimy, M. Azadbakht, J. Bian, and M. Amani, "Ruesvms: An ensemble method to handle the class imbalance problem in land cover mapping using google earth engine," *Remote Sensing*, vol. 12, no. 21, p. 3484, 2020.

[4] J. Kalajdjieski, E. Zdravevski, R. Corizzo, P. Lameski, S. Kalajdziski, I. M. Pires, N. M. Garcia, and V. Trajkovik, "Air pollution prediction with multi-modal data and deep neural networks," *Remote Sensing*, vol. 12, no. 24, p. 4142, 2020.

[5] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Computer Science*, vol. 8, p. e820, 2022.

[6] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artificial Intelligence Review*, vol. 57, no. 6, p. 137, 2024.

[7] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[8] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[9] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[10] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016.

[11] S. Kotsiantis and P. Pintelas, "Mixture of expert agents for handling imbalanced data sets," *Annals of Mathematics, Computing & Teleinformatics*, vol. 1, no. 1, pp. 46–55, 2003.

[12] I. Tomek, "Two modifications of cnn," *IEEE Trans. Systems, Man and Cybernetics*, vol. 6, pp. 769–772, 1976.

[13] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[14] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126, 2003.

[15] C. Chiamanusorn and K. Sinapiromsaran, "Extreme anomalous oversampling technique for class imbalance," in *Proceedings of the 2017 International Conference on Information Technology*. ACM, 2017, pp. 341–345.

[16] S. Rezvani and X. Wang, "A broad review on class imbalance learning techniques," *Applied Soft Computing*, vol. 143, p. 110415, 2023.

[17] S. S. Rawat and A. K. Mishra, "Review of methods for handling class imbalance in classification problems," in *International Conference on Data, Engineering and Applications*. Springer, 2022, pp. 3–14.

[18] N. U. Niaz, K. N. Shahariar, and M. J. Patwary, "Class imbalance problems in machine learning: A review of methods and future challenges," in *Proceedings of the 2nd International Conference on Computing Advancements*, 2022, pp. 485–490.

[19] S. Sharma and A. Gosain, "Addressing class imbalance in remote sensing using deep learning approaches: a systematic literature review," *Evolutionary Intelligence*, vol. 18, no. 1, p. 23, 2025.

[20] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 408–421, 1972.

[21] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE transactions on information theory*, vol. 14, no. 3, pp. 515–516, 1968.

[22] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409, pp. 17–26, 2017.

[23] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97. Nashville, USA, 1997, pp. 179–186.

[24] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Transactions on systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 448–452, 1976.

[25] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Conference on Artificial Intelligence in Medicine in Europe*. Springer, 2001, pp. 63–66.

[26] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Mute: Majority under-sampling technique," in *2011 8th international conference on information, communications & signal processing*. IEEE, 2011, pp. 1–4.

[27] P. Vuttipittayamongkol and E. Elyan, "Neighbourhood-based undersampling approach for handling imbalanced and overlapped data," *Information Sciences*, vol. 509, pp. 47–70, 2020.

[28] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe level graph for majority under-sampling techniques," *Chiang Mai Journal of Science*, vol. 41, no. 5.2, pp. 1419–1428, 2014.

[29] X. Liang, Y. Gao, and S. Xu, "Ase: Anomaly scoring based ensemble learning for highly imbalanced datasets," *Expert Systems with Applications*, vol. 238, p. 122049, 2024.

[30] C. Shi, X. Li, J. Lv, J. Yin, and I. Mumtaz, "Robust geodesic based outlier detection for class imbalance problem," *Pattern Recognition Letters*, vol. 131, pp. 428–434, 2020.

[31] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[32] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[33] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[34] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.

[35] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[36] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[37] M. Lichman *et al.*, "Uci machine learning repository," 2013.

[38] I. B. Khediri, C. Weihs, and M. Limam, "Kernel k-means clustering based local support vector domain description fault detection of multimodal processes," *Expert Systems with Applications*, vol. 39, no. 2, pp. 2166–2171, 2012.

[39] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation measures for models assessment over imbalanced data sets," *J Inf Eng Appl*, vol. 3, no. 10, 2013.

[40] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 2013, pp. 245–251.

[41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[42] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[43] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11, no. 1-8, 2003.

[44] 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.

[45] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," *BMC bioinformatics*, vol. 14, no. 1, p. 106, 2013.