IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 February 2023, accepted 6 March 2023, date of publication 10 March 2023, date of current version 15 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3255983

RESEARCH ARTICLE

A Comparative Analysis of Sampling Techniques for Click-Through Rate Prediction in Native Advertising

NADIR SAHLLAL AND EL MAMOUN SOUIDI
Laboratory of Mathematics, Computer Science, Applications and Information Security, Faculty of Sciences, Mohammed V University, Rabat 10000, Morocco
Corresponding author: Nadir Sahllal (Sahllal.Nadir@gmail.com)

ABSTRACT Native advertising is a popular form of online advertisements that has similar styles and functions with the native content displayed on online platforms, such as news, sports and social websites. It can better capture users' attention, and they have gained increasing popularity in many online platforms and among advertisers. In advertising, Click Trough Rate (CTR) prediction is essential but challenging due to data sparsity: the non-clicks constitute most of the data, whereas clicks form a significantly smaller portion. The performance of 19 class imbalance approaches is compared in this study with the use of four traditional classifiers, to determine the most effective imbalance methods for our native ads dataset. The data used is real traffic data from Finland over the course of seven days provided by the native advertising platform ReadPeak. The resampling methods used include seven undersampling techniques, four oversampling techniques, four hybrid sampling techniques, and four ensemble systems. The findings demonstrate that class imbalance learning can enhance the model's capacity for classification by as much as 20%. In general, oversampling is more stable comparatively. But, undersampling performed the best with Random Forest. Our study also demonstrates that the imbalance ratio plays an important role in the performance of the model and the features importance.

INDEX TERMS Class imbalance, data resampling, CTR prediction, native advertising, machine learning.

I. INTRODUCTION

In a class imbalanced dataset, there are significantly less samples in one of its classes than the other [3]. The difficulties of learning from such imbalanced data are inevitable. Standard learning classifiers are biased toward the majority class due to the skewed distribution of the training samples, making them unable to recognize unusual occurrences. It is possible to mistake noise for rare minority samples and vise-versa [24]. This kind of problem with uneven data is particularly prevalent in the advertising industry. The dataset contains a lot more non-clicked advertising than clicked advertisements, and the difference between the two is typically significant. To solve these issues, researchers have created numerous class imbalance techniques and performance evaluation criteria that play an important role in this paper. In this context, we implemented several class imbalance methods on a real-world class imbalanced dataset provided by the native advertising platform ReadPeak [1]. The dataset used contains information about the in-screen advertisements and whether they have been clicked or not. According to the dataset, there is an extreme imbalance between clicks and non-clicks, with 250 non-clicks for every 1 click.

Advertising is a key and crucial part of corporate operations. In 2021, advertisers are estimated to have spent 118.72 billion dollars on display advertising [52]. Demand-side platform (DSP)'s cost per click pricing model makes advertisers earnings directly correlated to the number of clicks. Predicting performance indicators such as the click-through rate (CTR) is crucial for DSP research [35].

The available literature presents a valid solution to the problem at hand, but addressing the unbalance of data specifically and the benefits we can get either in terms of

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

training speed and prediction accuracy has been neglected. In that spirit, the purpose of the current work is to explore the effect of sampling techniques on prediction accuracy with the combination of a number of classical machine learning models applied to the prediction of CTR in native advertisement.

The contributions of this paper are summarized as follows:

- The performance of 19 class imbalance methods with 4 classical classifiers is evaluated. The best performing methods are identified based on the *AUC* and *LogLoss* metrics.
- An analysis of the unbalanced data problem was conducted using real-world data from ReadPeak, including complete outline of the steps for feature engineering, selection, and data cleaning.
- Through experimentation, it was found that feature importance varies with different Imbalance Ratios. The Boruta algorithm was used to compute feature importance using three levels of sampling, highlighting the importance of balancing the data for accurate results.
- Using Random Forest and random undersampling, it is shown that as the balance between positive and negative samples is improved, the performance also improves. However, there is a threshold of imbalance ratio beyond which the performance improvement becomes marginal.
- We available the community access to a real world datasets that can help researchers study the phenomenon and find solutions to CTR Prediction in the native advertisement field.

The remainder of this paper is arranged in the following manner: in Section II, we give an overview of the sampling techniques. Starting by describing the data used, Section III is dedicated to the methodology. It expands on the steps and methods used to treat the data and explains how it is evaluated. Section IV provides a comparative study and analyzes the performance of different experiments. Further, the paper findings are discussed in Section V. To conclude, a summary of the work at hand in addition to potential future directions can be found in Section VI.

II. PRELIMINARIES AND RELATED WORK

In the following, we will be introducing some of literature related to CTR Prediction. We will also present the unbalanced methods that will be used in this paper and introduce some of the background information needed.

A. CLICK THROUGH RATE PREDICTION

Logistic Regression (LR) is the traditional model for CTR prediction [16], as it is quick, light, and clearly interpretable. Some enhanced models that would decrease the computing complexity and increase nonlinear expressiveness, such as factorization machines [47], field aware factorization machines [51], and gradient boosting Decision Tree. These models, have been proven to be effective in practice. Furthermore, in resent years, neural network techniques have been used to further enhance the current models. Instead of pooling interest vectors, approaches based on user behavior have made an effort to directly extract user interests from time-sequence data. One such method is the Recurrent Neural Network (RNN) model, which was used in GRU4Rec [8] to forecast preferences based on users' previous click-through patterns. Additionally, the Attentive Capsule Network (ACN) method was created by Li et al. [12] to reflect users' varied interests, where a transformer was used to separate feature interactions from users' varied interests. To boost performance, the targeting of segment groups for an advertisement and deliver customized advertisements to these segments can be very beneficial [13], [25], [50]. One of the most used segmentation methods is interest-based segmentation [32], which groups users with similar interests.

B. IMBALANCED DATA TECHNIQUES

Class imbalance is a well-known problem in machine learning [28]. When the classes in the data have an unbalanced distribution, machine learning model will favor samples from the majority class and will not pay enough attention to samples from the minority class. It will result in the model's output being biased in favor of the dominant class [27]. Due to the classifier's disregard for minority classes, its accuracy cannot be trusted. Many academics in the field of machine learning are now focusing on class unbalanced learning due to the impacts and potential that class imbalances can have on data and learning.

In advertising, class imbalance methods have already been used in many applications, such as click-fraud detection and CTR predictions. Class imbalanced data methods can be classified into three categories: (i) sampling techniques, (ii) algorithm level methods and (iii) ensemble methods [22].

1) SAMPLING TECHNIQUES

Data-level approaches entail steps taken in the training data to improve the class distribution by having less samples in the majority classes or more samples in the minority classes [48]. The data-level technique is primarily at the data pre-processing stage, and redistributing the training data of various classes in the data space through resampling [38]. To balance the unbalanced class, this kind of approach can alter the dataset structure as much as possible. Resampling the data to change the samples analog distribution has been demonstrated in some research to improve the model's performance to some extent [36].

Procedures for resampling can be divided further into (i) undersampling, (ii) oversampling and (iii) hybrid sampling. In the following, we provide a brief explanation of these techniques.

Undersampling techniques retain important data for learning while discarding samples from the majority class until the number of samples in each class is approximately equal [43]. However, it is unavoidable that certain samples that are significant to the training model may be overlooked

while undersampling the dataset [49]. After all, many undersampling techniques employ various filtering principles. Undersampling methods include:

1) *Random UnderSampling (RUS)*: *RUS* is the earliest undersampling method to have been invented, and it discards random samples from the majority class [23].
2) *Edited Nearest Neighbors (ENN)*: With the other samples in this procedure, each instance is checked using k-NN. The improperly identified samples will be discarded, and the updated dataset will be created from the remaining samples [6].
3) *Instance Hardness Threshold (IHT)*: This undersampling technique excludes examples that are difficult to classify or have a high likelihood of being misclassified after training a classifier to identify them [30].
4) *Near Miss*: By choosing majority samples whose average distances from the three nearest minority samples are the least, this method chooses majority samples that are close to some minority samples [19].
5) *Neighbourhood Cleaning Rule (NCR)*: The three closest neighbors of each instance in the dataset are taken into account by this strategy. A sample is eliminated from the dataset if it is misclassified by its three closest neighbors and is in the majority class. Additionally, the majority class samples in the vicinity of a sample that is a member of the minority class sample and is incorrectly classified by its three closest neighbors are eliminated [21].
6) *One-Sided Selection (OSS)*: First, 1-NN is used to choose minority class samples and majority samples that were incorrectly classified. Then the Tomek Links' vast majority of class examples are deleted [29].
7) Tomek Links (TL): Refer to a pair of instances, which belong to different classes and are each other's nearest neighbor. These links are considered to be noisy or boundary instances and are often removed from the majority class sample [28].

To provide a more equal class distribution of samples while preserving class borders, oversampling techniques create new samples based on data from the minority class. On the other hand, because it duplicates or synthesizes a small number of samples, oversampling can result in overfitting [54]. The length of training time rises with the quantity of samples. Oversampling methods include:

1) *Random OverSampling (ROS)*: *ROS* is the earliest oversampling approach to be created, and it replicates random minority class samples to achieve a more evenly distributed class [15].
2) *Adaptive Synthetic (ADASYN)*: Minority class samples are distributed in this manner based on their difficulty to learn. Minority samples that are harder to learn have more synthetic samples created for them. [18].
3) *Synthetic Minority OverSampling Technique (SMOTE)*: The k-Nearest Neighbors (kNN) interpolation method is used to generate synthetic samples from each minority sample. [34].
4) *Borderline SMOTE*: This technique utilizes borderline samples, which are frequently misclassified by their closest neighbors, to perform SMOTE [17].

The hybrid sampling approach combines under-sampling and over-sampling. Both approaches have their drawbacks: under-sampling can result in loss of important information, while over-sampling can lead to overfitting. To address these issues, methods that combine under-sampling and over-sampling have been proposed, such as SMOTEENN and SMOTETomek [15]. These methods use a combination of techniques such as SMOTE for over-sampling and ENN or Tomek links for under-sampling. These techniques aim to balance the training dataset, eliminate noisy points that are on the incorrect side of the decision boundary, discover better clusters, and build models with strong generalization capabilities.

2) ALGORITHM LEVEL METHODS

Algorithm-level methods are approaches in which the classifier is not affected by the skewed distribution, or where conventional machine learning classifiers are modified and linked to a weight or cost variable [10]. Many researchers have published relevant studies addressing the class imbalance issue at the algorithm level [4], [37].

3) ENSEMBLE METHODS

In ensemble systems, algorithmic and sampling techniques are used to handle data internally and modify the distribution of categories in the sample. They employ data-level approaches, and internally modify the learning process using specific algorithms [46]. This helps to prevent the model from excessively favoring the majority class during classification [28]. The standard ensemble techniques are listed below:

1) *Balanced Bagging*: This technique employs bagging and RUS to balance the dataset. It uses integrated estimators after resampling each subgroup of the data. The technique has an advantage over Sci-kit-Learn because it uses two additional parameters to control the behavior of the random sampler, namely sampling strategy and replacement [54].
2) *Balanced Random Forest*: This technique creates a balanced sample from which each tree is constructed. It starts by drawing bootstrap samples from the minority class, and then randomly selects an equal number of instances from the majority class with replacement. The final prediction is determined by a majority vote [9].
3) *Easy Ensemble*: This approach involves using AdaBoost to train classifiers on balanced subsets, and then combining the output of each classifier to produce an ensemble classifier [53].

4) *Random UnderSampling Boost (RUSBoost)*: This method combines sampling and boosting, and each round of boosting includes RUS [11].
5) *Balance Cascade*: This approach uses a double integration technique that combines bagging and boosting. It removes majority class samples that can be accurately identified during training using an iterative method and combines them with the minority class to create a base learner. This approach emphasizes samples that are more likely to be misclassified [53].

III. METHODOLOGY

This section aims to explain the approach taken to investigate the handling of class-imbalanced dataset in advertising data. The focus is on a dataset where the proportion of confirmed clicks is only 0.4%, which represents a significant class imbalance. To address this issue, various class imbalance techniques will be applied to predict clicks.

A detailed description of the specific methods and techniques utilized in this research will provide be provided in the following section, along with a justification of the choices made and the challenges encountered during the process of putting this paper together.

A. DATASET

The dataset used in this paper is provided by a real Demand Side Platform (DSP) called ReadPeak [1]. The data is collected from ReadPeak, which is a platform that enables advertisers to promote their content online by buying inventory from a publisher of their choice. The training data used in this study is one week of native advertisements that were shown to consumers all over Finland in the month of June 2022. The dataset comprises of a total of 17.925.833 lines of data, out of which 72.540 lines represent confirmed clicks. The testing set used in this study is one day's worth of data collected from the following week, comprising of 4.803.760 lines of data, out of which 17.179 lines represent confirmed clicks.

The dataset includes 13 features and a target feature named 'label.' Table 1 describes and details the features available. It is evident from the purpose of this study and the number of clicks observed in comparison to the total data, that the dataset used is highly unbalanced.

We would like to mention that in this work, our main focus is on addressing the data imbalance in the native advertising dataset. However, we also acknowledge and address some of the other challenges associated with this type of data, such as the overlap between clicks and non-clicks, the variability of the ads, and the limitation of available data. Although feature engineering was also considered in the data preparation in this study, our main emphasis is on exploring different sampling techniques and their effect on the performance of the models and CTR prediction.

B. PRE-PROCESSING

Data pre-processing includes addressing the issue of missing values and adjusting the features of the datasets. The part about scaling numerical data and label encoding of categories features will be discussed later.

TABLE 1. Features description.

Features	Format	Description
art	int	Article id identifies each unique advertisement
loc	int	The location id in the ReadPeak platform, one site represent a location.
tag	string	The placement identification. It identifies the advertisement placement within a location
dt	int	The device type
type	int	represents the type of advertisement it is; one of 2 types: native or banner
os	string	Operating system of the device e.g. Android
make	string	The maker brand of the device e.g. Samsung or Apple
client	int	Represents the client id within the ReadPeak platform
city	string	The city of where the advertisement will be shown
lang	string	The language of the navigator used to access the advertisement
cl	int	The number of clicks a certain advertisement accumulated in a certain location
ts	datetime	The time stamp of when the request to show the advertisement is received
label	boolean	Show either if the advertisement was clicked or not

1) MISSING VALUES

The initial dataset consists of almost 18 millions in-screen advertisements with 13 columns. Out of these columns, six were found to have missing values. The missing values in the 'lang' feature were replaced with the most common category, while the missing values in the 'tag', 'city', 'client', 'os', and 'make' columns were treated as a new category. This approach was chosen due to the nature of these features.

2) FEATURES CLEANING

The majority of the features present in our dataset underwent a number of operations to make them more manageable and to extract as much relevant information as possible. In this section, we will describe the process that each feature underwent.

- **tag**:For the majority of the variable, it was mainly an integer but there were instances that contained characters. We used Python's regular expression to clean the feature by removing the characters from the instances in question.
- **os**: We turned everything into lower case and categorized all the Operating Systems (OS) containing the word ''android'' into a category and did the same for ''ios''. From the data analysis, we curated a list of the most common OS. Any other values in the 'os' column not belonging to the list were put into their own category that we called 'unknown'.
- **make**: A similar process was done to the 'make' feature as was done to the 'os' feature. We categorized all Samsung and Apple devices in their own respective categories and limited the categories only to the most

common ones. Any other values not belonging to the list were put into a category called 'unknown'.

- **city**: We matched the cities that were entered in different manners and lowercased all entries. We only kept the top 25 cities obtained from the data analysis. Any other cities not belonging to the list were put into their own category called 'unknown'.
- **lang**: All languages that were repeated a very small number of times were removed. We matched all languages that were denoted in different ways to ensure consistency in the data.
- **ts**: From the 'timestamp' feature, we extracted the 'Day' and 'Hour' feature, which we used instead of the 'ts' feature. The 'Day' feature is represented with numbers from 0 to 6, with 0 being Sunday. The 'Hour' feature is a float, with the decimal values representing the minutes over 60.

3) NORMALIZATION

The purpose of normalization is to convert the values of numeric columns in the dataset to a common scale while preserving disparities in value ranges. This is only important when the ranges of the features differ. The benefit of data normalization is that many machine learning algorithms tend to perform better and it speeds up the training in some cases.

The data in this research is normalized to eliminate the influence of dimensionless disparities across features in the dataset. The goal is to set the data mean to 1 and the variance to 0. The following shows the calculation formula to normalize a certain feature X:

$$x' = \frac{X - X_{mean}}{X_{max} - X_{min}} \tag{1}$$

where X_{mean} is the mean value, X_{max} is the maximum value, and X_{min} is the minimum value.

4) CATEGORICAL ENCODING

Categorical encoding refers to the process of assigning numeric values to nominal (non-numeric) features in order to facilitate the processing task. Since we are using only neural networks in our model, it is necessary to convert the textual data of the dataset into numerical values. Generally speaking, there are two approaches to encode categorical variables:

1) One-hot (binary) encoding: A binary representation of nominal features where the categorical value is removed and a new binary variable is added for each unique nominal value.
2) Integer encoding: Refers to the process of coding a categorical variable using integers such as 1, 2, and 3.

The main difference between the two approaches is that One-Hot Encoding preserves the order relationship between the values of the nominal feature, while Label Encoding does not. Additionally, One-Hot Encoding requires a higher memory consumption than Label Encoding. In this research, We are using label encoding, also called integer encoding, because most of our categorical variables have a large number of categories. Using this approach allows for less memory consumption and makes training the data less computationally expensive.

C. FEATURE SELECTION

One of the key aspects of machine learning is feature selection, which aims to eliminate irrelevant and redundant information in the dataset, in order to boost model accuracy and improve computational efficiency. There are several popular feature selection techniques, such as filter, wrapper, and embedded methods [7]. In this study, we use the Boruta algorithm [26], an advanced feature selection method based on Random Forest, to select the most important features for our model. Boruta not only identifies the significance of each feature, but also evaluates whether it should be retained for further analysis. This approach is considered to be more efficient and accurate than other feature selection methods.

The Boruta algorithm's core process is as follows:

1) Creates a shadow feature by randomly sorting the original features to create a shadow feature matrix, then splicing the shadow feature matrix with the original feature matrix to create the new feature matrix.
2) For training, the new feature matrix is fed through a Random Forest classifier, which outputs the relevance of features v.
3) The original feature and shadow feature's z scores are computed, and the formula is as follows:

$$z_{score} = \frac{A_v}{S_v} \tag{2}$$

where the average value of feature importance is represented by A_v, while the standard deviation of feature importance is S_v.
4) The maximum z_{score}, indicated as Z_{max}, is searched in the shadow feature.
5) If the z_{score} of the original feature is larger than Z_{max}, the feature is classified as ''important'' if the original feature's z_{score} is less than Z_{max}, on the other hand, the feature will be classed as ''unimportant'' and removed.
6) Repeat steps 1–5 until all of the features have been noted.

D. DATA-SPLIT

We use Stratified KFold ($K = 5$) to split the dataset, retaining the sample category ratio while dividing the complete development set into five independent subgroups. For each split in this procedure, four-fifths of the dataset are used. The final fifth serves as the test set and the remaining as the training set. Each split can be regarded as the ith time ($i = 1, \ldots, 5$), and AUC is calculated on the ith test set [56]. It is important to note that the test set obtained each time is set aside and it was not be used in the scaling, recoding, or model-building processes. The test set needs to be separated from the training process, because the oversampling strategy will

clone or synthesize some minority samples, meaning the data obtained in this way cannot accurately represent the original dataset.

But to also put these sampling methods to the test in a real environment we re-train the model on all the available data and then test it on a one day data three days after the collected data.

E. IMBALANCED DATA STRATEGIES

The majority of the class-imbalanced learning strategies used in this study are data-level techniques and ensemble systems. Specifically, this research primarily investigates imbalance techniques in the Imblearn library [2]. Resampling techniques, including under-sampling, over-sampling, and hybrid sampling methods, are mostly used for data-level procedures.

Undersampling techniques include:

- Random UnderSampling (RUS).
- Edited Nearest Neighbors (ENN).
- Instance Hardness Threshold (IHT).
- Near Miss (NM).
- Neighbourhood Cleaning Rule (NCR).
- One-Sided Selection (OSS).
- Tomek Links(TL).

Given the size of the dataset used in this work, algorithms based on k-NN, such as All k-Nearest Neighbors (All k-NN), Cluster Centroids (CC), and Condense Nearest Neighbors (CNN), are computationally expensive and take a long time to run. Therefore, CC, All k-NN, and CNN are not included in this study.

Examples of oversaampling methods include:

- Synthetic Minority Oversampling Technique (SMOTE).
- Random Oversampling (ROS).
- Adaptive Synthetic (ADASYN).
- Borderline SMOTE.

Furthermore, methods for hybrid sampling include:

- SMOTE-ENN.
- SMOTE-Tomek.
- RUS&SMOTE
- RUS&ROS

To forecast advertisement clicks, these data level techniques are paired with classifiers. We also trained ensemble systems using the built-in classifiers, which are:

- Easy Ensemble.
- Random Undersampling Boost.
- Balanced Random Forest.
- Balanced Bagging (RUSBoost).

To note, this paper will not cover the Balance Cascade algorithm because it has been regularly modified by the Imblearn library in recent years and was finally dropped in version 0.6.0.

In the next section, we will discuss the effectiveness and safety of using oversampling techniques in our native advertising dataset.

F. OVERSAMPLING CONCERNS

Oversampling has been shown to improve model performance on imbalanced datasets, But, a recent review paper [55] has raised concerns about the risks associated with this approach. The paper argues that oversampling methods assume that all synthesized data belong to the minority class, without providing any guarantee that some of the generated examples may actually belong to the majority class.

Despite the concerns raised, we argue that oversampling techniques can still be a useful tool for CTR prediction on native advertising data. In fact, we found that oversampling techniques had a positive effect on the performance of CTR prediction models on our dataset. Additionally, the extreme imbalance and nature of the task at hand make it important to generate more positive samples, as they are so scarce.

It is worth noting that overlap in the native advertising data used can explain that some of the generated examples may belong to the majority class. This means that the risk of synthesizing minority class data that actually belongs to the majority class doesn't make the generated samples invalid. Additionally, we have explored hybrid methods that combine undersampling and oversampling techniques to create a balanced dataset. This approach can help alleviate some of the concerns of the mentioned paper.

Lastly, given the nature of the current study, it would be imprudent not to include oversampling in our systematic comparison. Oversampling is a commonly used techniques for addressing class imbalance in machine learning, and excluding it from our evaluation could limit the usefulness and relevance of our findings. By including oversampling in our comparison, we can provide a more comprehensive and nuanced analysis of the relative benefits and drawbacks of different sampling techniques for CTR prediction on native advertising data.

Overall, while it is important to be aware of the potential risks associated with oversampling, we believe that the benefits of generating more positive samples outweigh the risks in the current case. Therefore, we conclude that oversampling techniques can be a useful and tolerable approach for CTR prediction on native advertising data.

G. BASELINE MODEL SELECTION

The selection of algorithms for this study was based on their suitability for handling imbalanced datasets and their computational efficiency. The algorithms selected, namely Logistic Regression [41], Naive Bayes [40], Decision Tree [42], and Random Forest [39], are known for their simplicity, low computational cost, and ability to handle imbalanced datasets. These algorithms typically work well "out of the bag," meaning that they often provide a good baseline model without requiring extensive tuning.

While several other machine learning algorithms, including Support Vector Machines (SVM) and k-Nearest Neighbors (KNN), were initially considered, they were ultimately excluded from the study due to their high computational

cost and memory requirements. SVM is known for its high accuracy and ability to handle complex datasets, but it can be slow and require a large amount of memory. KNN, on the other hand, is a non-parametric algorithm that is easy to implement and interpret, but it can also be slow and require significant amounts of memory, especially when dealing with large datasets.

In addition, neural networks and factorization machines were not included in the selection of algorithms for this study. While neural networks have shown promising results in various domains, they can be computationally expensive and require significant amounts of data for training. Moreover, their suitability for this study was limited by the size of the dataset and the need to compare the performance of different sampling techniques. Similarly, factorization machines are also computationally expensive and can be challenging to implement, requiring a substantial amount of expertise and resources. Therefore, the selected algorithms were deemed sufficient to provide a useful baseline for comparison purposes and achieve the primary goal of identifying the most suitable sampling technique for predicting CTR in the context of native advertising.

H. EVALUATION

In order to determine the best class-imbalanced technique for the dataset based on the following standard, this study uses four traditional classifiers as the baseline model:

- Logistic Regression (LR).
- Random Forest (RF).
- Naive Bayes(NB).
- Decision Tree(DT).

We will compare the results based on the metrics mentioned in the section.

1) IMBALANCE RATIO

The imbalance ratio (IR) is an essential parameter in imbalanced learning. It measures the proportional relationship between the majority and minority classes in the experiment [31]. The formula is given by Eq. 3:

$$IR = Instances_{Minority}/Instances_{Majority} \tag{3}$$

Most of the data-level methods used in the research involve resampling the majority or minority class in the original dataset, which increases the minority class samples or decreases the majority class samples. This changes the imbalance ratio of the dataset. As the IR decreases, the difference in sample size between the majority and minority class becomes larger, indicating that the dataset is more imbalanced. On the other hand, when the IR value is closer to 1, the dataset tends to be more balanced. Therefore, this paper uses IR as a metric to evaluate sampling techniques.

2) MODELS EVALUATION

To assess the prediction outcomes, the output confusion matrix is used to calculate *Accuracy*, *Precision*, *Recall*, and *F1 Score*. Table 2 from [57] depicts the generated confusion matrix.

TABLE 2. Confusion matrix.

confusion matrix		Predicted	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	FN

True Positive (TP) denotes a situation where both the true and predicted values are positive, which means that the number of positive samples has been correctly predicted. False Positive (FP) denotes a situation where the true value is negative, but the predicted value is positive, meaning that the number of negative samples has been incorrectly predicted to be positive. True Negative (TN) denotes a situation where both the true and predicted values are negative, meaning that the number of negative samples has been correctly predicted. False Negative (FN) denotes a situation where the true value is positive, but the predicted value is negative, meaning that the number of positive samples has been incorrectly predicted to be negative.

The ratio of accurately predicted samples to the total number of samples is known as *Accuracy*, and the method for calculating it is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

However, most studies on imbalanced class data have found that accuracy may not be the best metric to evaluate the performance of a model on such datasets [33]. This is because in many real-world applications, the minority class is often more important and any errors in classifying it (i.e False Negatives or False Positives) may have a significant impact on the overall performance of the model. For example, in the case of click-through rate (CTR) prediction, the goal is to correctly identify the minority (positive) cases in order to find clicks, as they are usually the desired outcome of an online advertisement. Therefore, incorrectly assessing where to show the advertisements can lead to significant financial losses.

We describe an alternative performance evaluation metric, the area under the Receiver Operating Characteristic (ROC). ROC curve plots the True Positive Rate ($TPR = TP/(TP + FN)$) on the y-axis against the False Positive Rate ($FPR = FP/(TN + FP)$) on the x-axis at various threshold values [5]. The area under the ROC curve (AUC) identifies the classifier's ability to distinguish between classes and compares ROC curves [20]. As such, we use the following metrics to evaluate the compared methods:

AUC: A widely used metric for CTR prediction tasks. It indicates the Area Under the ROC Curve over the test set. It reflects the probability that a model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. Here, the larger the better. We will note that an improvement of 1% in *AUC* is usually regarded as

significant for the CTR prediction because it will bring a large increase in a company's revenue if the company has a very large user base.

LogLoss (cross entropy) a training goal function, and it may be computed as follows:

$$LogLoss = \frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) \quad (5)$$

where N is the total number of training samples, y_i is the true label of i^{th} instance and $\hat{y}_i$ is the prediction.

IV. EXPERIMENT AND ANALYSIS

This section presents the results of our experiments. All methodologies and models were tested on an Amazon Web Services (AWS) SageMaker instance with 32 vCPUs and 72GB of RAM. The implementation was done using two programming languages, Python 3.9.11 and R 4.2.0. The Python implementation used several packages such as Pandas, Numpy, and Sklearn. For feature importance, we primarily used the Boruta package in R.

Following, we will explore the effects of the IR on both the feature importance and the predictions result. Later also we list the imbalance ratio provided by the resampling technique and then show the prediction results of the imbalance technique model, which can help analyse the effect of the imbalance technique comprehensively. We have used the area under the curve (*AUC*) and *LogLoss* error for the evaluation of the proposed methods. The *AUC* and *LogLoss* provide the best indication of performance when the dataset is imbalanced [14], [27].

A. FEATURE SELECTION

The Boruta algorithm is used to select features for our dataset, and the results are illustrated in Figure 1. Blue boxplots indicate the minimal, average, and maximum Z score of a shadow attribute. As previously explained, these shadow attributes are generated by adding randomness to the dataset by creating shuffled copies of all features. As seen in the figure, the importance of shadow features should be minimal in comparison to the rest of the authentic features. The Boruta algorithm assigns one of three outcomes for each feature: accepted (green), inconclusive (yellow), and rejected (red) in the boxplot. The algorithm outputs can be observed in the figure 1.

Given that we are trying to figure out which sampling method work best with native advertising data, our first experiment pertained to how the feature importance of our variables change with the level of sampling. To do this, we used three levels of sampling:

- First level was with the full data with no sampling applied, where the clicks represent approximately 0.4% of the data. The results for the arrangement are depicted in the bottom of both Table 3c and Figure 1c.
- Represented in Figure 1a and Table 3a is level 2 where we applied a moderate random undersampling to make the clicks constitute 5%.
- The last level was where we made both clicks and non clicks completely balanced. The results for this level can be seen in Figure 1b and Table 3b.

Applying the Boruta algorithm on our data at different proportions shows differences in the importance of the features. Between the first and the second level, the city feature is rejected in both, while all the other features are accepted. Additionally, there is a change in the order of importance of the features between the first and the third level. The bottom three features remain consistent, and the city feature is rejected in both the full data and moderate undersampling, maintaining the lowest importance in the balanced layout. In general, it can be observed that the top, middle, and lower ranking features remain in the same splits, regardless of the different sampling proportions.

B. IMBALANCE RATIO EFFECT

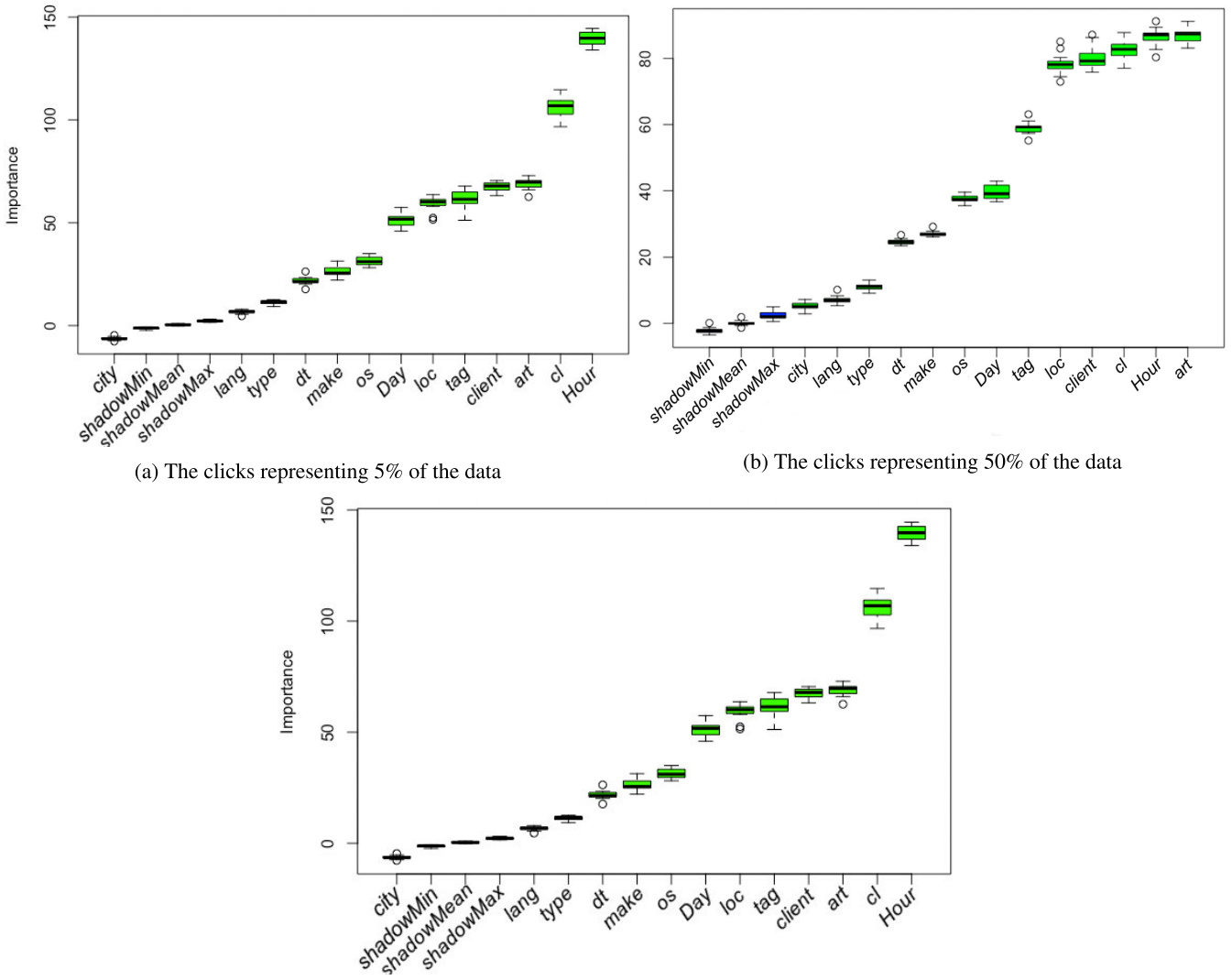
In this subsection, we discuss the imbalance ration effect. Given that the data studied in this work is very unbalanced, we ask the following questions: Can the unbalance itself be helpful in getting a better perdition? or, will balancing the data completely give the best results? To answer these question we run a Random Forest model on different imbalance ratios ranging from IR = [0.01 .... 1]. To measure the effect of the ratio imbalances we use the *AUC* metric. *AUC* has been shown through various research to be a great indicator of performance for imbalanced datasets.

In Figure 2 illustrates a gradual improvement of the *AUC* score from 0.547 to 0.663. This increase is a clear indication that balancing the data plays a significant role in the performance of our predictor. It is obvious that the substantial improvements in the *AUC* are mainly at the level where the IR is between [0.01...0.1]. While it is true that there is improvement as the IR increases to 1, the increase is quite marginal after IR = 0.25.

To answer the questions posed at the beginning of this subsection, it can be concluded that fully balancing the data yields the best results. It is also clear that the imbalance itself is not as helpful. With this in mind, another question arises: After the benefit of increasing the imbalance ratio using undersampling becomes marginal, would it make sense to combine other sampling techniques to further improve our predictions? This will be addressed by generating two hybrid sampling techniques.

C. SAMPLING TECHNIQUES EFFECT ON NATIVE ADVERTISEMENTS DATA

The class imbalanced dataset used has an imbalanced ratio of 0.004. Through resampling technology, the class proportion of the dataset has changed. Table 4 lists the class distribution in the training set after each sampling.



(a) The clicks representing 5% of the data

(b) The clicks representing 50% of the data

(c) Unsampled data

FIGURE 1. **Feature importance over different level of data sampling.**

The outcome shows that undersampling affects the majority of samples, oversampling only affects the minority of samples, and the hybrid method affects both categories.

All sampling techniques enhance the IR value, while oversampling and hybrid sampling have IR values that are almost equal to 1, ensuring that the dataset is as class-balanced as possible.

Applying various undersampling methods for the Read-Peak provided dataset, we show the resulting *AUC*s and *LogLoss*es for 4 different classifiers in Table 5. RUS showed the best performance with Random Forest and Decision Tree with an *AUC* of 0.666 and 0.59 respectively. While Naive Bayes's *AUC* of 0.607 IHT using Random Forest as a base learner.

For oversampling methods, ROS had the best performance for Logistic Regression, Naive Bayes. While for Random Forest and Decision Tree, SMOTE is the most suitable for performance. These are shown in Table 6. Naive Bayes had the highest mean *AUC* of 0.607.

On top of the hybrid methods provided by imlearn library [2] we manually manufacture two more hybrid methods based on RUS and SMOTE and ROS. These new sampling methods were inspired following the experiment in section IV-B.

For ensemble methods shown in Table 8, EasyEnsemble achieved the highest mean *AUC*, followed by RUSBoost.

We evaluate all the resampling methods using Random Forest. We chose Random Forest not only because it was the best-performing model with RUS, but also because it is the most sensitive to the applied sampling techniques. In Figure 4, blue represents the baseline, green represents the undersampling methods, red is for the oversampling methods, and yellow represents the hybrid methods. The baseline AUC value is 0.547; it can be seen that the lowest value that

TABLE 3. Feature importance over different level of data sampling.

	meanImp	medianImp	minImp	maxImp	normHits	decision
art	68.976	69.702	62.571	72.890	1	Confirmed
loc	59.131	60.156	51.456	63.700	1	Confirmed
tag	61.552	61.418	51.194	67.801	1	Confirmed
dt	21.815	21.489	17.663	26.275	1	Confirmed
type	11.294	11.665	9.237	12.583	1	Confirmed
os	31.516	31.067	28.121	35.012	1	Confirmed
make	26.479	25.562	22.102	31.356	1	Confirmed
client	67.433	67.900	63.204	70.505	1	Confirmed
city	-6.294	-6.492	-7.723	-4.631	0	Rejected
lang	6.604	6.770	4.575	7.996	1	Confirmed
cl	106.349	106.881	96.690	114.683	1	Confirmed
Day	51.238	51.739	45.907	57.475	1	Confirmed
Hour	139.534	139.746	133.984	144.501	1	Confirmed

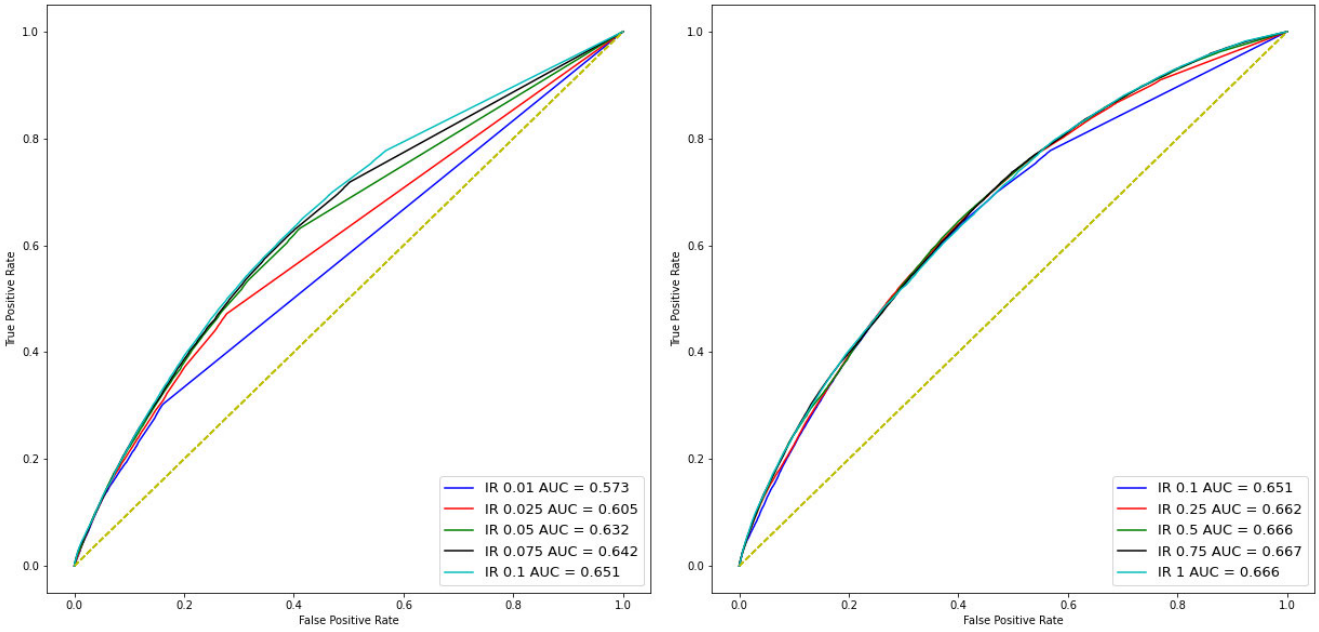
(a) The clicks representing 5% of the data

	meanImp	medianImp	minImp	maxImp	normHits	decision
art	86.913	87.357	83.130	91.198	1.00	Confirmed
loc	78.277	78.123	72.937	85.077	1.00	Confirmed
tag	59.011	59.256	55.166	63.089	1.00	Confirmed
dt	24.708	24.623	23.477	26.754	1.00	Confirmed
type	10.999	11.105	9.110	13.098	1.00	Confirmed
os	37.746	37.526	35.608	39.553	1.00	Confirmed
make	27.059	26.903	26.155	29.249	1.00	Confirmed
client	79.939	79.231	75.837	87.174	1.00	Confirmed
city	5.171	5.144	2.877	7.228	0.93	Confirmed
lang	7.166	7.006	5.357	10.162	1.00	Confirmed
cl	82.682	82.780	77.044	87.834	1.00	Confirmed
Day	39.619	39.144	36.800	42.877	1.00	Confirmed
Hour	86.407	87.109	80.334	91.250	1.00	Confirmed

(b) The clicks representing 50% of the data

	meanImp	medianImp	minImp	maxImp	normHits	decision
art	51.588	51.602	43.382	58.019	1	Confirmed
loc	40.607	38.776	35.797	51.969	1	Confirmed
tag	42.129	42.452	30.532	49.967	1	Confirmed
dt	16.664	16.125	12.952	20.040	1	Confirmed
type	12.390	12.627	10.760	13.262	1	Confirmed
os	23.885	23.651	21.371	26.104	1	Confirmed
make	16.942	17.920	13.704	19.956	1	Confirmed
client	48.559	50.380	37.632	56.698	1	Confirmed
city	-12.965	-12.903	-15.127	-11.643	0	Rejected
lang	5.742	5.637	3.740	7.426	1	Confirmed
cl	77.813	79.257	63.579	85.926	1	Confirmed
Day	51.454	51.083	38.766	63.577	1	Confirmed
Hour	94.360	90.421	84.359	104.858	1	Confirmed

(c) Unsampled data



(a) IR from 0.01 to 0.1

(b) IR from 0.1 to 1

FIGURE 2. Effects of ratio imbalance on training.

appears in Near Miss is 0.474, the highest value appears in ROS, and its AUC value is 0.666. Observing the bar chart shows that the AUC displayed by the undersampling method has more significant fluctuations than other methods. Also, through the LogLoss result from tables 7, 8, and 9, they show that oversampling is more stable than other imbalanced learning techniques, and undersampling is the most unstable.

D. GENERALIZATION TO FURTHER DATA

In this part we expose our last experiment. We take the best combinations between the used machine learning models and

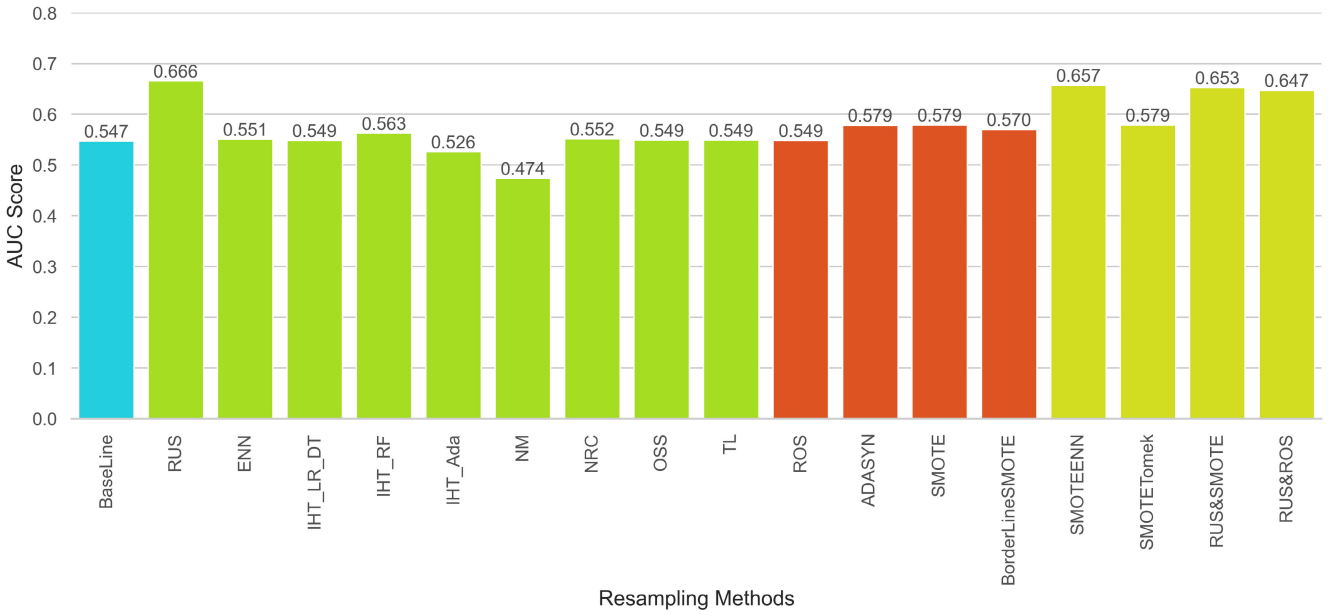


FIGURE 3. Comparison of sampling method on Random Forest.

TABLE 4. Class distribution for data-level methods.

Method	Imbalance Ratio	Majority Samples	Minority Samples
BaseLine	0.004	17853293	72540
RUS	1	72540	72540
ENN	0.004	17659810	72540
IHT_LR	0.004	17853293	72540
IHT_RF	0.005	15992013	72540
IHT_Ada	0.982	73887	72540
NM	1	72540	72540
NRC	0.004	17669435	72540
OSS	0.004	17843717	72540
TL	0.004	17843859	72540
ROS	1	17853293	17853293
ADASYN	0.99	17858792	17853293
SMOTE	1	17853293	17853293
SMOTEENN	0.964	17468821	16525698
SMOTETomek	1	17827350	17827350
BorderLineSMOTE	1	17853293	17853293
RUS&SMOTE	1	2176200	2176200
RUS&ROS	1	2176200	2176200

sampling technique, see if the results obtained trough cross validation can be generalized to future data. The best combinations found trough the previous experiments are: Random Forest with RUS, Decisition Tree with SMOTEENN, Naive Bayes and Logistic Regression with ROS.

From Figure 4 and Table 9, we can see that Random Forest is the best performing model when it is put in a real situation. To reiterate Random Forest generalize well for practical use. While we can also see that for ensemble methods apart from Balanced Random Forest which *AUC* slightly improved, all the others performed slightly worse. Making Random Forest based models the best performing for this type of data.

V. DISCUSSION

In this section, we will explore the implementation of class imbalance techniques in our study. First, we will examine the impact of data imbalance on machine learning algorithms and the significance of feature importance. Next, we will provide a more detailed analysis of the results obtained from the combinations of different data sampling techniques and baseline classifiers used. We will also address the challenges presented by the dataset and how these were reflected in the results. Finally, we will discuss the relevance and significance of our contribution.

A. DATA UNBALANCE EFFECT ON LEARNING

In this study, different data-level sampling techniques were applied to four baseline classifiers, including Random Forest, Logistic Regression, Decision Tree, and Naive Bayes. The techniques used were data-level oversampling, undersampling, and hybrid methods. The results showed that Random Forest achieved the highest AUC with the RUS technique compared to the other classifiers, and also showed the greatest improvement from the baseline, going from 0.547 to 0.666. Naive Bayes, on the other hand, was the least affected by sampling methods. These results suggest that the Random Forest classifier is well-suited for the imbalanced advertisement data used in this study.

Despite the initial low AUC values of Logistic Regression and Decision Trees, it is worth noting that the AUC values of most models were improved through the application of resampling approaches. This highlights the effectiveness of these techniques in improving a model's classification abilities. Additionally, our study found that most undersampling and hybrid sampling approaches had

TABLE 5. ***AUC*** **and** ***LogLoss*** **results for undersampling methods.**

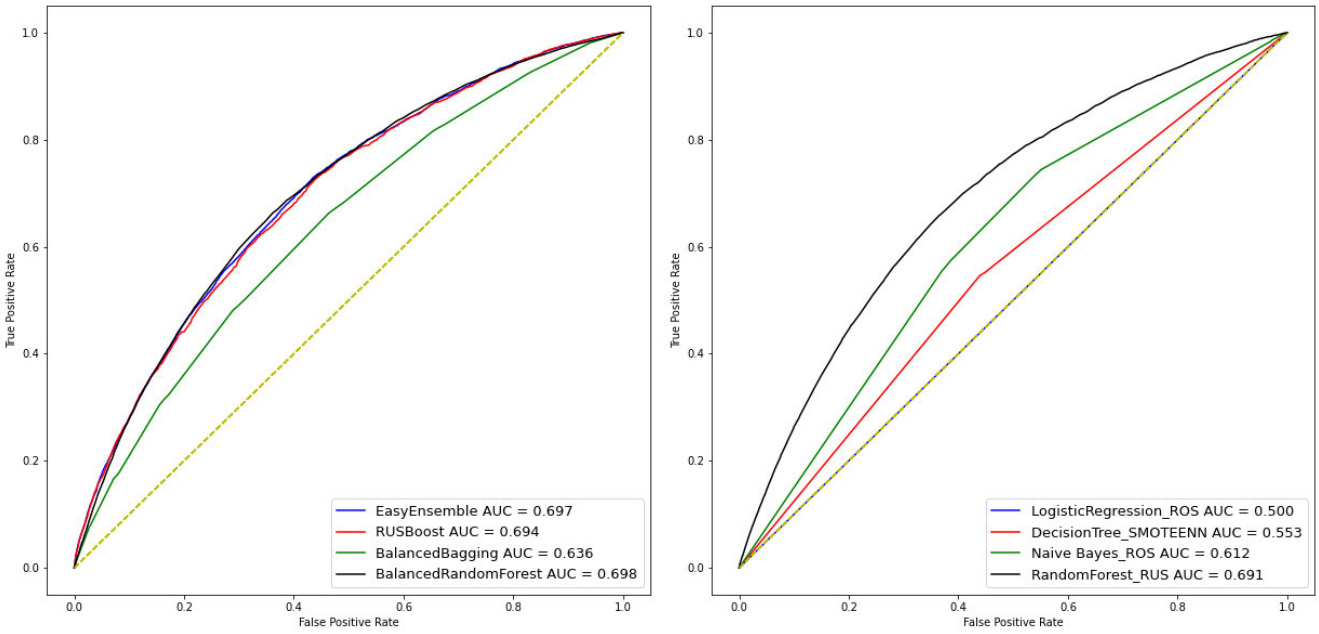
Method	Logistic Regression		Decision Tree		Naive Bayes		Random Forest	
	AUC	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*
BaseLine	0.500000	0.693147	0.526707	0.180339	0.603738	0.026056	0.547485	0.119183
RUS	0.5009305	0.69323475	**0.590349**	13.56434325	0.60582575	0.686635	**0.66606825**	1.03333775
ENN	0.5	0.693147	0.5294935	0.473606	0.6058525	0.026212	0.55096425	0.13126
IHT_LR_DT	0.5	0.693147	0.52787375	0.181053	0.6058525	0.02619775	0.548526	0.11978025
IHT_RF	0.5	0.693147	0.5329115	0.77200925	**0.606784**	0.026351	0.5629535	0.2122795
IHT_Ada	**0.51395**	0.694194	0.50601475	33.9179185	0.6045575	6.14427175	0.52633675	31.923336
NM	0.490552	0.73223225	0.49239475	34.15809675	0.39801575	1.4396265	0.47378175	32.5987575
NRC	0.5	0.693147	0.5288185	0.2365955	0.6058525	0.02621125	0.551633	0.12060425
OSS	0.5	0.693147	0.52808575	0.188301	0.6058425	0.02619875	0.5490755	0.11967825
TL	0.5	0.693147	0.5281335	0.1883825	0.6058425	0.02619875	0.54908475	0.11967675

TABLE 6. ***AUC*** **and** ***LogLoss*** **results for oversampling methods.**

Method	Logistic Regression		Decision Tree		Naive Bayes		Random Forest	
	AUC	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*
BaseLine	0.500000	0.693147	0.526707	0.180339	0.603738	0.026056	0.547485	0.119183
ROS	**0.569335**	0.693148	0.528662	0.297515	**0.607195**	0.686737	0.548611	0.257295
ADASYN	0.45561475	0.69314625	0.532529	0.443388	0.60546575	0.68677225	0.57850275	0.16850475
SMOTE	0.455315	0.69314625	**0.53267875**	0.440561	0.60572075	0.6867105	**0.57911525**	0.16841625
BorderLineSMOTE	0.5	0.693147	0.5305105	0.32716475	0.60438	0.68293625	0.5701645	0.155574

TABLE 7. ***AUC*** **and** ***LogLoss*** **results for hybrid-sampling methods.**

Method	Logistic Regression		Decision Tree		Naive Bayes		Random Forest	
	AUC	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*
BaseLine	0.500000	0.693147	0.526707	0.180339	0.603738	0.026056	0.547485	0.119183
SMOTEENN	0.500000	0.693147	**0.594370**	6.620541	0.605620	0.675026	**0.657399**	1.407990
SMOTETomek	0.45564425	0.69314625	0.5328025	0.44819025	0.60579925	0.6869205	0.57915525	0.17127225
RUS&SMOTE	**0.503822**	0.693147	0.569831	4.133750	0.605619	0.686698	0.652872	0.522852
RUS&ROS	0.482566	0.693147	0.556217	2.319643	**0.605901**	0.686387	0.647432	0.768557



(a) Ensembles Methods

(b) Base models with most suitable sampling method

FIGURE 4. **ROC curves for test data three days in the future from the training data.**

higher average AUC values than oversampling techniques. These findings suggests that undersampling and hybrid sampling methods are likely to provide the best performance for CTR prediction, as they effectively handle the class

TABLE 8. ***AUC*** **and** ***LogLoss*** **results for ensemble methods.**

Method	*AUC*	*LogLoss*
Balanced Bagging	0.671409	0.741580
Balanced Random Forest	0.695569	0.655088
EasyEnsemble	**0.710696**	0.690277
RUSBoost	0.709652	0.690229

TABLE 9. **AUC for data three days in the future from the training data.**

Method	*AUC*
EasyEnsemble	0.697
RUSBoost	0.694
Balanced Bagging	0.636
Balanced Random Forest	**0.698**

(a) Ensembles Methods

Method	*AUC*
Logistic Recression & ROS	0.500
Decision Tree & SMOTEENN	0.553
Naive Bayes & ROS	0.612
Random Forest & RUS	**0.693**

(b) Base models with suitable sampling method

imbalance in advertisement data while preserving the valuable information.

To further investigate the relationship between the sampling methods and the model's AUC, we used the imbalance ratio (IR) as a metric to measure the ability of resampling techniques to adjust the class distribution. The study's findings revealed that the oversampling and hybrid methods effectively achieved a near-equal distribution of classes in the dataset. However, the undersampling algorithms were not as successful in significantly reducing the majority class, suggesting that these methods may not be appropriate for a dataset with a high level of noise, like ours. It's worth noting that this conclusion is only based on the undersampling methods that were barely changed the data proportions, techniques such as RUS ans IHT_ada have proven to be more effective and suitable to be used on the same dataset.

In summary, the results of this study suggest that oversampling is generally more stable. However, for data that is as noisy and unbalanced as advertisement data, significant undersampling can outperform other approaches.

B. SAMPLING TECHNIQUES EFFECT ON THE ADVERTISEMENT DATA

The results of the study revealed that the feature importance of the native Advertisement dataset changed depending on the level of data sampling used. For example, when using the whole dataset, or when the IR was at 10%, it was observed that some features, such as city, were rejected by the Boruta algorithm. This suggests that the presence or absence of certain features can have a significant impact on the performance of the model when working with imbalanced datasets. It's also worth noting that this is just one example, and the feature importance and Boruta algorithm may have different results in different dataset and sampling ratios. It's important to keep in mind that while the results of this study were specific to the native ads dataset and specific sampling ratios, the takeaways and insights gained from this experimentation can be applied to other datasets and scenarios. By going through this process, we not only demonstrate good practice in handling imbalanced data, but we also see how the imbalance ratio can affect the feature selection process. Additionally, by testing the performance of different classifiers and sampling methods, we gain a deeper understanding of the strengths and weaknesses of each approach, which can inform future decisions when working with imbalanced data. Therefore, even though results may vary when applying these methods to other datasets, the experimentation and analysis conducted in this study can provide valuable guidance for practitioners.

After comparing the performance of different sampling methods, it can be seen that RUS has good performance when combined with Decision Tree and Random Forest. In fact, RUS had the best results of all combinations with Random Forest. In the case of oversampling techniques, ROC performs well with Logistic Regression and Naive Bayes, while SMOTE seems to be more suited for Random Forest. Additionally, for ensemble methods, EasyEnsemble and RUSBoost classifiers performed well. However, when evaluating the performance of these models on data from a few days in the future, it was found that the Balanced Random Forest classifier outperformed the others. This underlines the importance of considering real-world scenarios and testing models on future data when evaluating their performance.

From the baseline, it was observed that Random Forest had the most improvement with an increase of 0.119 in the *AUC* metric. After each classifier was processed by the sampling method in the table, the *AUC* of the model was increased, with the exception of Naive Bayes which showed little to no improvement. Additionally, among all the classifiers, Logistic Regression performed the worst, while all other classifiers combined with RUS performed well. As such, the Random Forest model using RUS was determined to be more suitable for processing imbalanced advertisement datasets and achieving the highest *AUC*. Another undersampling technique, Near Miss, was tested, but it showed lower results than the *AUC* of the corresponding baseline classifier. Therefore, it can be concluded that Near Miss is not suitable for datasets with an imbalance rate of about 0.004, as it was used in this study.

In conclusion, the results of this research suggest that the Random Forest model using RUS is a suitable approach for handling highly imbalanced advertisement datasets and can achieve the highest *AUC* even when generalized to future data. This highlights the effectiveness of the RUS technique in balancing the data and improving the performance of the Random Forest model.

C. ReadPeak's DATA

The dataset used in this study presents several challenges when it comes to predicting click-through rates. It is a

native advertising dataset that is unique in its nature, with limited access to user demographic information and a highly imbalanced distribution of positive and negative samples and by nature a significant percentage of overlap. These unique characteristics are suspected to have impacted the performance of the algorithms used in our analysis.

We uncovered challenges in using logistic regression with our dataset. While this algorithm has been shown to perform well on CTR prediction tasks in different datasets, our results showed that it's AUC Improved in combination with ROS, but in a more general scale it did not perform well. One possible explanation for this discrepancy is that Unlike most datasets used in the literature, our dataset has limited access to user demographics, which may have affected the accuracy of the logistic regression model. Additionally, the dataset exhibits overlap between positive and negative samples, violating logistic regression's assumption of independent and identically distributed samples.

The challenges presented by the used data suggest that additional care must be taken when selecting data sampling and machine learning techniques for an effective prediction. The performance of logistic regression in this context serves as an example. We aimed to provide a balanced study focusing on data sampling techniques that can serve as a guideline and a starting point for future works.

D. CONTRIBUTION

We have taken a systematic approach to feature engineering, selection, and data cleaning, outlining the complete steps taken in detail. Through experimentation using The Boruta algorithm to compute feature importance using three levels of sampling,it was found that the importance of features varies with different Imbalance Ratios.

Using Random Forest and RUS, it was shown that as the balance between positive and negative samples is improved, the performance of the algorithms also improves. However, there is a threshold of imbalance ratio beyond which the performance improvement becomes marginal. This highlights the importance of selecting the appropriate sampling technique for the dataset and algorithm in question.

The ultimate output of the sampling technique comparison resulted in RUS performing the best with Random Forest in terms of overall prediction AUC, it is important to note that this is not necessarily an obvious or expected result. While RUS has been successful in other studies, we noticed that there is a lack of rational on the selection of the algorithm and not enough evidence to support that it is the best technique for the problem at hand [44], [45].

In this work we provide valuable guidance for the development of effective click-through rate prediction models in native advertising by systematically comparing several practically used sampling techniques in combination with four different machine learning algorithms.

We would like to mention that the impact of these findings on machine learning in the field of native advertising. Advertisers and marketing professionals often deal with imbalanced datasets, where a small percentage of customers are responsible for the majority of conversions. In such scenarios, the ability to accurately identify and target potential customers becomes crucial for the success of an advertising campaign. The results of this study demonstrate that using the most appropriate sampling technique can help improve the performance of machine learning models on imbalanced native advertising datasets, leading to more effective targeting and better return on investment for advertisers. Additionally, by considering real-world scenarios and testing models on future data, advertisers can ensure that their models will continue to perform well over time. These insights can be used to inform the development of more effective and efficient advertising strategies.

VI. CONCLUSION

In this paper, we have investigated class imbalance techniques, including data-level and hybrid systems, to predict the click trough rate of advertisements. A dataset provided by ReadPeak's traffic in Finland for a week in June of 2022. The data had an imbalance ratio of 0.004. The imbalanced learning method was used to solve the problem of a skewed majority in prediction. This research discusses 19 imbalanced learning methods, including seven undersampling techniques, four oversampling techniques, four-hybrid resampling methods, and four ensemble systems. The class imbalance technology adjusts the majority or minority samples by discarding the majority samples, copying or synthesizing the minority samples to balance the categories in the dataset. In addition, foura classic classifiers (Logistic Regression, Random Forest, Decision Tree, Naive Bayes) combined with resampling techniques were used to train the dataset. The prediction results obtained using the classifier training pre-processing data (except for null values, etc.) were used as a baseline for comparison with models built using imbalance techniques. Additionally, the method used to evaluate the sampling technique was the imbalance ratio, and the index used to assess the classification ability of the model was *AUC*.

Further, using different levels of unbalance between the majority and minority samples we evaluated the features importance and the improvement of the *AUC*. For the current advertisement data when the undersampling was significant enough it was the more suitable approach. Interestingly, using RUS technology to process our data in the Random Forest model can achieve the highest *AUC* value.

This work's aim is to show how we can make use of sampling techniques in extremely skewed data such as native advertisement data. We would like to note that this methodology and results could be expanded to data related to cyber security; such as intrusion detection and click fraud, because of similarity in the data consistency of such use cases. In other words the results from the current work may apply to the mentioned cases.

We would like to point out that all the machine learning used in the current work have been used as is with no

fine tuning. For future work we can explore pushing these methods further by finding more suitable hyper parameters that can improve the result further. Another point that we would like to explore in the future is that advertisement data is extremely noisy and adding some more cleaning to it could improve the performance of the sampling method studied in this work even further. What is more explore the effect of other sampling methods that we could not include in the scope of the this work given the extremely high computational power needed.

DATA ACCESS STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] *Readpeak—The Nordic's Fastest-Growing Native Advertising Platform*, Helsinki, Finland, Jul. 2022.

[2] *Imbalanced-Learn Documentation!* Accessed: Oct. 14, 2022. [Online]. Available: https://imbalanced-learn.org

[3] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning From Imbalanced Data Sets*. Berlin, Germany: Springer, 2018.

[4] A. Majid, S. Ali, M. Iqbal, and N. Kausar, "Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines," *Comput. Methods Programs Biomed.*, vol. 113, no. 3, pp. 792–808, Mar. 2014.

[5] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[6] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972, doi: 10.1109/TSMC.1972.4309137.

[7] Q. Al-Tashi, R. H. Md, S. J. Abdulkadir, S. Mirjalili, and H. Alhussian, "A review of grey wolf optimizer-based feature selection methods for classification," in *Evolutionary Machine Learning Techniques* (Algorithms for Intelligent Systems). Singapore: Springer, 2019, pp. 273–286, doi: 10.1007/978-981-32-9990-0_13.

[8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.

[9] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. 666, 2004, vol. 110, nos. 1–12, p. 24.

[10] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 17, no. 1. Seattle, WA, USA: Lawrence Erlbaum Associates, 2001, pp. 973–978.

[11] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: Improving classification performance when training data is skewed," in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2008, pp. 1–4.

[12] D. Li, B. Hu, Q. Chen, X. Wang, Q. Qi, L. Wang, and H. Liu, "Attentive capsule network for click-through rate and conversion rate prediction in online advertising," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106522.

[13] D. Liu and V. Mookerjee, "Advertising competition on the internet: Operational and strategic considerations," *Prod. Oper. Manage.*, vol. 27, no. 5, pp. 884–901, May 2018.

[14] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci.*, vol. 513, pp. 429–441, Mar. 2020.

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

[16] H. B. McMahan, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, J. Kubica, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, and E. Davydov, "Ad click prediction: A view from the trenches," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Chicago, IL, USA, 2013, pp. 1222–1230.

[17] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 878–887.

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[19] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Workshop Learn. Imbalanced Datasets*, vol. 126, 2003, pp. 1–7.

[20] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, Mar. 2005.

[21] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Proc. Conf. Artif. Intell. Med. Eur.* Berlin, Germany: Springer, 2001, pp. 63–66.

[22] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[23] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano, "Using random undersampling to alleviate class imbalance on tweet sentiment data," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2015, pp. 197–202.

[24] JG. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[25] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piece-wise linear models from large scale data for ad click prediction," 2017, *arXiv:1704.05194*.

[26] M. B. Kursa and W. R. Rudnicki, "Feature selection with the Boruta package," *J. Stat. Softw.*, vol. 36, no. 11, pp. 1–13, 2010.

[27] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2017.

[28] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 42, no. 4, pp. 463–484, Jul. 2011.

[29] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. ICML*, vol. 97, 1997, pp. 179–186.

[30] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Mach. Learn.*, vol. 95, no. 2, pp. 225–256, May 2014.

[31] N. Noorhalim, A. Ali, and S. M. Shamsuddin, "Handling imbalanced ratio for class imbalance problem using SMOTE," in *Proc. 3rd Int. Conf. Comput., Math. Statist. (iCMS)*. Singapore: Springer, 2019, pp. 19–30.

[32] N. Schröder, A. Falke, H. Hruschka, and T. Reutterer, "Analyzing the browsing basket: A latent interests-based segmentation tool," *J. Interact. Marketing*, vol. 47, pp. 181–197, Aug. 2019.

[33] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2009, pp. 875–886.

[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jul. 2018.

[35] O. Chapelle, "Offline evaluation of response prediction in online advertising auctions," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015, pp. 919–922.

[36] P. Lee, "Resampling methods improve the predictive power of modeling in class-imbalanced datasets," *Int. J. Environ. Res. Public Health*, vol. 11, no. 9, pp. 9776–9789, Sep. 2014.

[37] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," 2020, *arXiv:2002.12478*.

[38] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," 2013, *arXiv:1305.1707*.

[39] A. Liaw and M. Wiener, "Classification and regression by random forest," *R Newslett.*, vol. 2, no. 3, pp. 18–22, 2002.

[40] H. Zhang, "The optimality of naive Bayes," in *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2004, pp. 562–567.

[41] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[42] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/BF00116251.

[43] S.-J. Yen and Y.-S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," in *Intelligent Control and Automation*. Berlin, Germany: Springer, 2006, pp. 731–740.
[44] K. Kim, E. Kwon, and J. Park, "Deep user segment interest network modeling for click-through rate prediction of online advertising," *IEEE Access*, vol. 9, pp. 9812–9821, 2021, doi: 10.1109/ACCESS.2021.3049827.
[45] M. Alali, M. AlQahtani, A. AlJuried, T. AlOnizan, D. Alboqaytah, N. Aslam, and I. Ullah Khan, "Click through rate effectiveness prediction on mobile ads using extreme gradient boosting," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 1681–1696, 2021, doi: 10.32604/cmc.2020.013466.
[46] S. M. Abd Elrahman and A. Abraham, "Class imbalance problem using a hybrid ensemble approach," *Int. J. Hybrid Intell. Syst.*, vol. 12, no. 4, pp. 219–227, Mar. 2016.
[47] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 995–1000.
[48] V. García, J. S. Sánchez, and R. A. Mollineda, "Exploring the performance of resampling strategies for the class imbalance problem," in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.* Berlin, Germany: Springer, 2010, pp. 541–549.
[49] W. Jindaluang, V. Chouvatut, and S. Kantabutra, "Under-sampling by algorithm with performance guaranteed for class-imbalance problem," in *Proc. Int. Comput. Sci. Eng. Conf. (ICSEC)*, Jul. 2014, pp. 215–221.
[50] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2016, pp. 45–57.
[51] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for CTR prediction," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 43–50.
[52] M. Yuen. (Jul. 7, 2022). *Native Advertising Industry 2022: Forecast, Trends, and the Rise of Video*. Insider Intelligence. Accessed: Sep. 14, 2022. [Online]. Available: https://www.insiderintelligence.com/insights/native-ad-spending
[53] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 39, no. 2, pp. 539–550, Apr. 2009.
[54] Z. Zheng, Y. Cai, and Y. Li, "Oversampling method for imbalanced classification," *Comput. Inform.*, vol. 34, no. 5, pp. 1017–1037, 2016.
[55] A. S. Tarawneh, A. B. Hassanat, G. A. Altarawneh, and A. Almuhaimeed, "Stop oversampling for class imbalance learning: A review," *IEEE Access*, vol. 10, pp. 47643–47660, 2022, doi: 10.1109/ACCESS.2022.3169512.
[56] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 36, no. 2, pp. 111–133, Jan. 1974. [Online]. Available: http://www.jstor.org/stable/2984809
[57] S. A. Alvarez, "An exact analytical relation among recall, precision, and classification accuracy in information retrieval," Boston College, Boston, MA, USA, Tech. Rep. BCCS-02-0, 2002.

NADIR SAHLLAL received the bachelor's degree in theoretical mathematics and the master's degree in code cryptography and information security from Mohammed V University, Morocco, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include algebra, information security, and machine learning applications.

EL MAMOUN SOUIDI received the Doctorat d'Etat degree in mathematics from Mohammed V University, Morocco, in 2002. He is currently a Professor of computer science with Mohammed V University. His expertise lies in the areas of algebra, information security, cryptography, and code-based steganography. His research interests include developing new methods and techniques for improving the security and privacy of digital communications.