How to Cite:
Gupta, A. K., & Garg, G. (2026). Audience expansion in the era of privacy regulations: Addressing shortened seed lists. *International Journal of Economic Perspectives*, *20*(1), 107–131. Retrieved from https://ijeponline.org/index.php/journal/article/view/1261

Audience expansion in the era of privacy regulations: Addressing shortened seed lists

Amit Kumar Gupta
Indian Institute of Management Lucknow, Lucknow - 226013, India
Email: efpm11010@iiml.ac.in
[0009-0004-8576-7266]

Gaurav Garg
Indian Institute of Management Lucknow, Lucknow - 226013, India
Email: ggarg@iiml.ac.in
[0000-0003-1962-7586]

***Abstract*---**Audience expansion enables businesses to acquire new customers by digitally targeting individuals who resemble their existing customer base, making it a critical lever for business growth. These models rely heavily on the diversity and quality of data available on audiences. However, emerging privacy regulations worldwide are limiting both the volume and variety of data that can be collected, which negatively impacts audience expansion models. Specifically, such restrictions reduce the size of the seed audience and weaken the signal in the feature space. A smaller seed list exacerbates class imbalance, which in turn degrades model performance. Synthetic oversampling techniques are commonly used to address class imbalance, but most overlook the challenges posed by high-dimensional binary covariate spaces. Existing methods that handle binary data often treat all features equally and do not selectively choose base samples for generating synthetic data—leading to the introduction of noise and borderline examples. We propose a novel oversampling algorithm, SMOTE-MSFB (SMOTE - Minority Focused Select Features for Binary data), that enhances synthetic sample quality by:

(a) Prioritizing minority samples near the decision boundary,
(b) Defining neighborhoods using a mutual information-weighted Jaccard distance to manage high dimensionality, and
(c) Improving signal strength through union-based voting across minority neighbors to counteract data sparsity.

Experiments on two publicly available audience expansion datasets demonstrate that SMOTE-MSFB outperforms existing resampling techniques for discrete features in a statistically significant result.

Corresponding author: *Gupta, A. K., Email: efpm11010@iiml.ac.in*
Submitted: 09 October 2025, Revised: 18 November 2025, Accepted: 27 December 2025

Also SMOTE-MSFB is at least ~70% more computationally efficient than the standard algorithm on the two datasets.

***Keywords*---**Audience Expansion, Look Alike models, Imbalanced dataset, Oversampling, Binary Sparse.

Introduction

Acquiring new customers is crucial for business growth. Customer prospecting helps businesses acquire new customers by targeting potential audiences who are likely to convert after exposure to ads [1]. Digital advertising plays a significant role in customer prospecting. Digital advertising is highly effective in this regard as internet users generate vast amounts of behavioral data through their interactions on websites and apps [2] [3]. Big data advancements allow advertisers and digital platforms to store and process granular consumer behavior data, enabling precise and effective targeting [1] [4].

Audience Expansion (AE), also known as Lookalike (LAL) models, is a crucial data modelling technique for optimizing digital advertising efforts by automating and enhancing the process of identifying new potential customers [5]. The primary idea behind AE is to expand a brand's existing customer base by targeting individuals who exhibit behaviors, interests, or demographics similar to the current customer base. This strategy assumes that if a person shares common characteristics with a brand's existing high-value customers, they are more likely to respond positively to advertising and make a purchase [6].

One of the key benefits of AE is its ability to scale advertising campaigns. AE enables the advertisers to automate the process of generation of target audiences by removing manual processing and subjective decision making [7] [8]. Instead of targeting a small, specific group of known customers, brands can now reach a broader audience that have a higher likelihood of converting into new customers in a very short span of time. AE helps brands allocate their marketing resources more effectively, ensuring ads are shown to individuals who are statistically more likely to become loyal customers [1].

Privacy related regulatory changes

Privacy laws are undergoing a significant change in the United States over the past few years. Enhanced privacy laws came into effect in 11 states in 2024 and laws from 8 states will go into effect during the year 2025. Fig 1 show the details. Although there are differences between the laws, broadly all the privacy laws coming into effect provide the following set of similar rights to the citizens of their state [9–22].

- To stop the sale of citizen personal data
- To stop the sharing of citizen personal data
- To access, correct and delete the personal data held by an organization
- Right to Opt-out of behavioral or targeted advertising
- Revoke consent of use of personal data
- Opt-out of profiling

Major privacy laws also mandate advertisers to implement an opt-in consent mechanism on their sites to block collection of personal data until a site visitor specifically opts-in. For example, when an internet user visits economist.com website via an internet browser, he/she will see a pop-up similar to one shown in Fig 2 specifically asking for visitor's opt-in consent for collecting data for this visit. Only if the visitor explicitly provides the consent for his/her data to be used, can the website collect the data on the visitor.

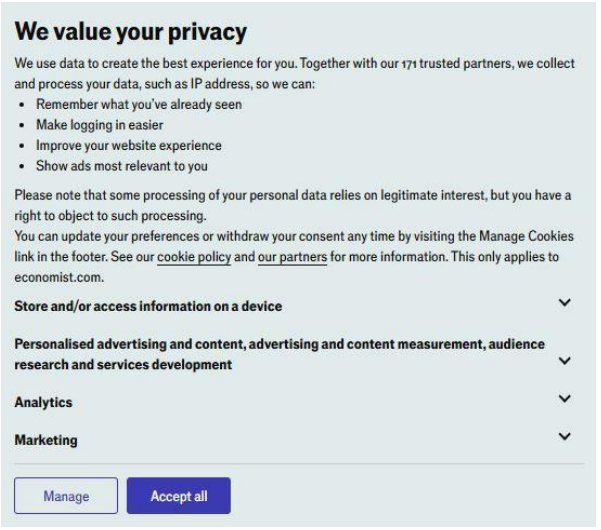

Fig. 1. – Opt-In Consent Pop up Implemented on economist.com

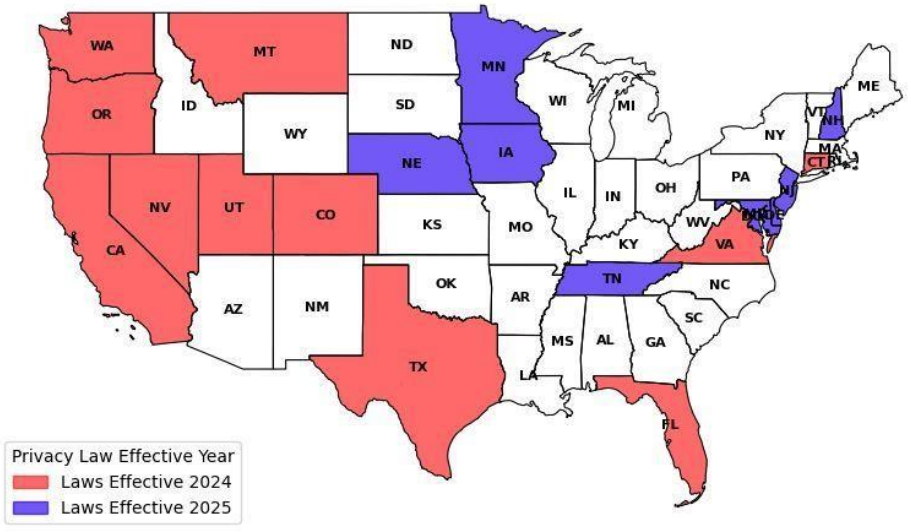

Fig. 2. Map of United States highlighting the states with recent changes in privacy laws

Impact of privacy regulations on Audience Expansion

From an advertising data perspective, these regulations result in increased sparseness in the data and loss of predictive power of the data specifically related to sensitive domains like healthcare, finance, age, gender, race etc. This way privacy laws fulfill their mandate by restricting the variety and volume of information that can be captured by advertisers on citizens, enabling the citizens to get their information deleted and prohibit the industry from using certain types of data for targeted advertising.

In Audience expansion, the seed list plays the most important role in determining the value an advertiser is able to drive from its advertising dollars. Seed lists are created using the first party data sources of the advertisers. The most typical case is using the recurring loyal customer base as the seed. However due to paucity of such customers, many a times website behavioral activity data of the audiences such as clicks and views are also considered for creating seed lists.

A large and representative seed list leads to good generalization capacity of the audience expansion model. This ensures that the model identifies the most suitable audiences who have a high probability of conversion when exposed to a relevant advertisement. Given the typical conversion rate in digital advertising ranges between 0.001 and 0.0000001 depending on the product and the targeting [1, 23], it has always been expensive to obtain an optimal data distribution. This the same situation as in many imbalanced classification problems where the cost of obtaining more minority class data samples is very high.

A direct impact of the privacy laws has been shortening of the seed lists. As the citizens become more aware and exercise their rights accorded by the privacy laws, the volume and variety of data available for targeting by digital advertising decreases. While the problem of shorter seed lists has always been present, it has become more frequent and acute with the implementation of the enhanced privacy laws. Another impact of the privacy laws is loss of predictive power of the data due to loss of features/characteristics of the audiences such as demographics due to privacy. This leads to presence of noisy features in the covariate space.

New Proposed Approach

From a modelling perspective, Audience expansion is an imbalanced binary classification problem with a high dimensional binary co-variate space and a noisy response variable. The seed list (S) represents the minority class and the large pool of audiences available for advertising (U) represent the majority class observations. The no. of audiences in seed is much less than the no. of audiences available for targeting ($S << U$) which leads to imbalance. The seed list provided by advertisers may contain samples which are distinctively very different from other samples in the seed [7, 24], this results in noisy minority class samples. The covariate space for LALs are usually constructed using the behavioral data (events, clicks and actions taken on the platform). Behavioral data is characterized by large no. of binary variables based on activities performed on the websites or apps.

In this paper, we present a novel approach to generate expanded audience for targeted advertisement based on short seed lists and behavioral data. The main contribution of this paper is presenting a novel approach for handling shortened seed list for audience expansion based on imbalanced classification in high dimensional sparse binary co-variate space. We create a modification of much renowned approach Synthetic Minority Oversampling Technique (SMOTE) for large/high dimensional binary feature vector space.

The rest of the research paper is organized as follows. In section 2, we review and summarize the existing literature on Audience expansion. The proposed approach/algorithm developed to solve for shortened seed list problem is presented in section 3. Experimental results using real world dataset are given in section 4. Section 5 provides the details on future potential research areas.

Literature review

We briefly review the relevant literature for Audience Expansion and techniques for oversampling data in imbalanced classification.

Audience Expansion – Literature review

In audience targeted advertising, Look alike modelling techniques can be classified into following three broad categories.

Rule based segment level approaches. Initial efforts in Lookalike Audience (LAL) modeling used data to automate segment determination. Rule-based segment level approaches focus on refining audience targeting by using predefined segment criteria such as age, gender, geography, and interests for identifying relevant audiences. [23] Introduced an associative classifier to identify key rules based on feature pairs, with rules sorted by a frequency-weighted log-likelihood ratio (FLLR) metric. [25] Propose a weighted audience extension approach using three criteria: interest, novelty, and quality, allowing advertisers to adjust weights to generate expanded audience sets. [26] Addressed data transfer latency by proposing an in-database Kmeans clustering method based on columnar databases. [27] Expanded audiences by applying the Generalization of User Segments (GUS) approach, leveraging the IAB taxonomy to extend audience segments to higher-level "head subjects."

Rules based approaches only use high level users features but are not able to capture sophisticated user behavior for targeting [4]. Simplicity and ease of explain ability of selection criteria to business stakeholders is the main advantage.

Locality Sensitive Hashing (LSH) based approaches. LSH-based approaches are appropriate for large-scale Lookalike (LAL) modeling due to the high number of features and large data volumes. [28] proposed LSH to transform the feature space into smaller similarity-preserving signatures. This method reduces the need for expensive pairwise comparisons, enhancing efficiency. [5] applied MinHash to preserve Jaccard similarity on binary features, selecting all audiences in a cluster containing at least one seed audience. In a refined approach, [29] introduced feature weighting based on Information Value (IV) to adjust for varying importance of features, improving the model's performance.

Very low latency in expanding seed list audience post generation of the hashing tables is the main benefit of LSH based approaches. However LSH based approaches do not take into consideration the interaction of features indicating user behavior [4].

Regression based approaches. Regression-based classification approaches treat Lookalike Audience (LAL) modeling as a binary classification problem with a large binary feature space. The seed list audiences are considered the minority class, while usually a random sample from the remaining audiences is the majority class. Methods ranging from Linear models [1, 5], Tree based models [6, 30], SVMs [30, 31], Neural networks [32, 33] and Semi-Supervised learning methods [32] are proposed in the literature.
These methods leverage different techniques to handle large-scale, imbalanced datasets in LAL modeling, with each offering strengths in different aspects of model accuracy and efficiency.
Apart from the above methods graph based approaches and hybrid approaches combining the above three methods have also been suggested in literature as well.

Oversampling in Imbalanced classification
Random oversampling (ROS) is the simplest of the oversampling techniques used for Imbalanced classification problems. ROS randomly replicates the minority class observations until a desired level of class balance is achieved in the resampled data [34]. A key impact of replication is decrease in the variance in the data which causes overfitting issues [34–39]. To overcome this challenge, new oversampling methods were developed which generate synthetic samples.
One of the most renowned oversampling developed to address the above issues was Synthetic Minority Oversampling Technique (SMOTE) [36]. In SMOTE, new observations are added to the sample which are not replication of the existing samples rather are synthetic observations generated by linearly interpolating in the covariate space between minority class samples. These synthetic observations leads to broadening of the decision area and better generalization by the classification model.
However, SMOTE introduces noise and borderline samples in the data [40]. To tackle these challenges, several advanced oversampling techniques have been introduced, such as Borderline SMOTE [41], Advanced SMOTE (A-SMOTE) [40], SafeLevel SMOTE [35], the Adaptive Synthetic Sampling Technique for Imbalanced
Learning (ADASYN) [42], Attribute Weighted and kNNHub on SMOTE (AWHSMOTE) [37] and Constrained Oversampling [34]. These methods address previous limitations by generating synthetic samples based on selected minority class sample, typically chosen according to the number of majority class neighbors identified using k-nearest neighbors (kNN).

Oversampling for categorical covariate space. These diverse oversampling techniques have demonstrated their effectiveness in improving prediction model performance. However, they primarily target continuous features [43]. Real-world datasets often only have binary features especially in domain like digital advertising, natural language processing, image processing and applications using internet of things based sensors and using the standard methods on such datasets can result in suboptimal outcomes.

SMOTE-N is an oversampling method specifically designed to handle all discrete features. Like the original SMOTE approach, it generates synthetic samples from existing minority instances. However, in SMOTE-N, the feature values of the synthetic samples are determined through majority voting among the features of the minority class and its k-nearest neighbors (kNN) [36], with distances calculated using the Value Difference Metric (VDM) [44]. Despite its adaptations, SMOTE-N shares the common limitation with SMOTE, as it can potentially introduce noise in the resampled training dataset.
Another oversampling method developed to handle discrete features is SMOTEENC, introduced by Mukherjee and Khushi [43]. In this approach, discrete features are first numerically encoded before generating synthetic samples, where higher numerical values indicate a stronger correlation with the minority class. This encoding process is based on Pearson's chi-squared test.
Both SMOTE-N and SMOTE-ENC create synthetic samples from each positive instance using the k-nearest neighbors (kNN) method, which may result in the introduction of noise and borderline cases. Moreover, treating all features with equal importance during the neighbor selection process can lead to the inclusion of unrepresentative neighbors, potentially affecting the quality of the generated samples and negatively impacting accuracy of the model.

Oversampling in high dimensional datasets. The class imbalance problem is frequently accompanied by high-dimensionality, making the identification of relevant features crucial for minimizing class overlap [45]. To address this, both resampling and cost-sensitive approaches have been combined with feature selection techniques in high-dimensional, imbalanced datasets [46–49]. Alternatively, the impact of highdimensionality can be reduced through feature extraction and manifold learning methods**.**
SMOTE has also been used in high dimensional setting in the literature. In the work by Deepa and Punithavalli [50], the authors introduce E-SMOTE, which combines genetic algorithms for feature selection with SMOTE-based oversampling applied to the selected subset of relevant features. Similarly, Qazi and Raza [51] investigate the impact of two feature selection methods—one based on redundancy and the other on information gain—alongside two resampling strategies, namely random under sampling and SMOTE, on a network intrusion detection dataset.
Blagus and Lusa [48] highlighted that SMOTE may be ineffective in highdimensional settings, often failing to reduce bias toward the majority class. Comparing SMOTE with threshold adjustment across various classifiers, they found that SMOTE offers no performance gains—and can even degrade results—when applied to high-dimensional datasets with limited sample sizes.
To best of our knowledge, No previous research has studied the use of SMOTE on high dimensional sparse binary covariate space which is one of the key novelty contributions of this study.

Proposed algorithm for shortened seed lists
The main objective of this study is to generate look alike audiences based on short seed lists to enable advertises to attract new customers in a privacy compliant way. Short seed lists lead to a significant imbalance in the data. This imbalance negatively impacted the performance of the model especially for the minority class which is of main interest for the advertisers. Additionally, Look Alike models are

built on behavioral data of platforms which leads to a large sparse binary covariate space running into thousands/millions of features at times [1, 7, 23]. As a results, there is a need to develop way to mitigate the impact of the class imbalance in high dimensional binary covariate space.
To address the aforementioned problem, synthetic data oversampling for large binary covariate space was used in this investigation.
SMOTE forms a neighborhood around minority class instances using KNN algorithm and creates synthetic samples by selecting points from within this neighborhood. SMOTE-N algorithm [36] use value difference matrix [44] based distance for nominal variables. High dimensional data usually have a high percentage of redundant and/or irrelevant variables that introduce noise in the algorithm. KNN classifier's accuracy is negatively impact in high dimensional data settings with large no. of redundant and irrelevant variables [52]. Also, SMOTE-N randomly samples minority class samples without considering the difficulty of classification of the minority sample. It is possible for SMOTEN to generate synthetic observations based on minority samples which can be correctly without resampling as well.
Specifically in the context of behavioral data, each binary feature represent an action taken by the audience on the advertiser platform or represent some attribute of the audience. The covariate space remains the same across advertisers from different domains. Hence only a very small no. of features are relevant for a particular advertiser.
Our approach modifies the standard SMOTE-N algorithm to take care of the above deficiencies. The proposed oversampling methodology is visually represented in Fig 3. Our approach consists of following three phases:

Defining minority class neighborhood
First, we use mutual information based criteria for selecting a smaller set of features out of the complete feature space which contribute to user specified proportion of (suggested value = 95%) of the total mutual information. The MI based feature selection leads to removal of a large no. of irrelevant features from the dataset. The neighborhood for each minority sample is created using KNN on this reduced feature space using weighted Jaccard distance metric.[29] had used the same distance metric for improving on the base LSH approach for audience expansion. The weights being the corresponding MI score for the feature.

Reasons for using weighted Jaccard distance. Since Jaccard distance ignores features that are absent (0) in both samples and focusses only on features that are present (1) in at least one of the samples, it is more relevant to use in sparse binary feature space. Using the weights from MI scores keeps the samples which are similar in important features to be closer together while defining the neighbourhood.

Identifying minority samples for resampling using base learner
In order for the resampling to be effective, it must be targeted at the correct minority class samples. Creating synthetic samples from minority samples far away from the decision boundary on either side does not improve the accuracy. Minority samples deep inside the minority space do not need resampling as those will be correctly classified without resampling. Minority samples deep inside the majority space should not be used for resampling as it will only be add noise to

the dataset. This is the same idea behind borderline SMOTE for continuous feature space [41].
We use a logistic regression as base learner for identifying minority samples to be resampled. We remove any minority samples which satisfy either of the two conditions: a) Correctly classified by base learner OR b) 2) Probability from base learner is < 0.1.
First condition is targeted towards removing minority samples which are deep inside the minority space while the second condition removes minority samples deep inside the majority space. For creating synthetic observations, minority samples are selected randomly from all the remaining minority samples.

Generating Synthetic minority class samples
The synthetic minority class samples are generated using the neighborhood created in the step 1 and only for the minority samples identified in step 2. However, the covariate space of the synthetic sample is created using a union voting scheme across the complete feature space of the neighbors. Thus the synthetic samples generated are of the same feature length as the original samples.

Union voting scheme for generating synthetic observations from neighbors. In Behavioral data context, the no. of features observed/present per unit audience is very small as compared to the total feature set. And a large majority of features are unobserved (tagged as 0) for most of the audiences. This sparseness in the data is very difficult for classification algorithms to handle. Hence In order to increase the signal present in the data, the new synthetic observation is created using the union across the neighbor. That is the synthetic observation has a feature set to 1 if the feature is 1 in any of the neighbor minority observation. In business terms, it implies that if one seed audience with age = [15-20 years] and its neighboring seed audience with income = [> $50K] are in the seed list, then the synthetic audiences with age = [15-20 years] and income = [> $50K] should also be in the seed list.

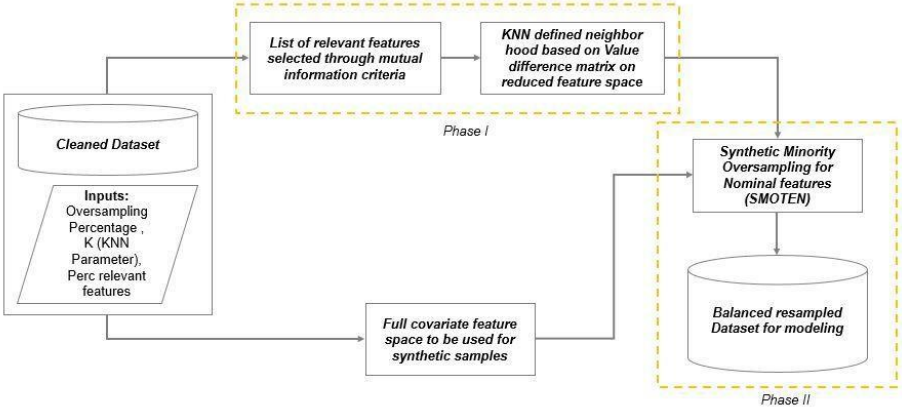

Fig. 3. High Level Flow Diagram of Resampling Approach

A key feature of our approach is the feature selection for classification is performed after the resampling process. As the features for SMOTE oversampling may not be the same for the classification task, this is the best way to model the

data. We name our algorithm SMOTEN-Minority focused Select Features for Binary data (SMOTEMSFB). The pseudo code for our methodology is given in Figure 4.

Algorithm 1 Feature-Selective KNN-based SMOTEN for High-Dimensional Binary Covariates

Input: Imbalanced dataset $D = \{X, Y\}$ with binary features; K: number of nearest neighbors; $PercFeatures$: percentage of top features for neighbourhood creation
Output: Balanced dataset D_{bal} with equal number of minority and majority class samples
Steps:
$FeatureImportanceScoreList \leftarrow$ `mutual_info_classif`(X, Y)
$TopFeatureList \leftarrow$ Select top $PercFeatures\%$ features from $FeatureImportanceScoreList$
$X' \leftarrow X$[TopFeatureList_Index]
$X'_{\text{minority}} \leftarrow X'[Y == 1]$
$Y'_{\text{minority}} \leftarrow Y'[Y == 1]$
$VDM \leftarrow$ Value Difference Metric computed on $X'_{\text{minority}}$and Y'_{minority}
$KNN \leftarrow$ Nearest Neighbors of X'_{minority} using VDM and K
$n_{\text{samples}} \leftarrow$ Count$(Y == 0)$ – Count$(Y == 1)$
$X_{\text{synthetic}} \leftarrow []$
for $i = 1$ to n_{samples} **do**
    Randomly select a minority instance C from X'_{minority}
    $neighbors \leftarrow KNN(C)$
    $C_{\text{synthetic}} \leftarrow$ Majority vote on binary features of $neighbors$
    Append $C_{\text{synthetic}}$ to $X_{\text{synthetic}}$
end for
$Y_{\text{synthetic}} \leftarrow$ Vector of 1s with length equal to n_{samples}
Append $X_{\text{synthetic}}$ to X and $Y_{\text{synthetic}}$ to Y
return $D_{\text{bal}} = \{X, Y\}$

Fig. 4. Pseudo Code for SMOTE-MSFB

Visual representation of the SMOTE-MSFB sampling using t-SNE. In order to better understand the resampling mechanism, we generated a simulated dataset with 50 samples having a 1:3 imbalance between the minority and majority class and 100 binary features covariate space. The dataset was resampled using random oversampling and SMOTE-MSFB to oversample the minority class and bring it at par with no. of majority class samples.

The resampled data was transformed to a two-dimensional map using t-distributed stochastic neighbor embedding (t-SNE) [53]. t-SNE preserves the neighbourhood structure of the data [54]. This property helps us in understanding the resampling process. The resampled data visualized using t-SNE is given in figure 5.

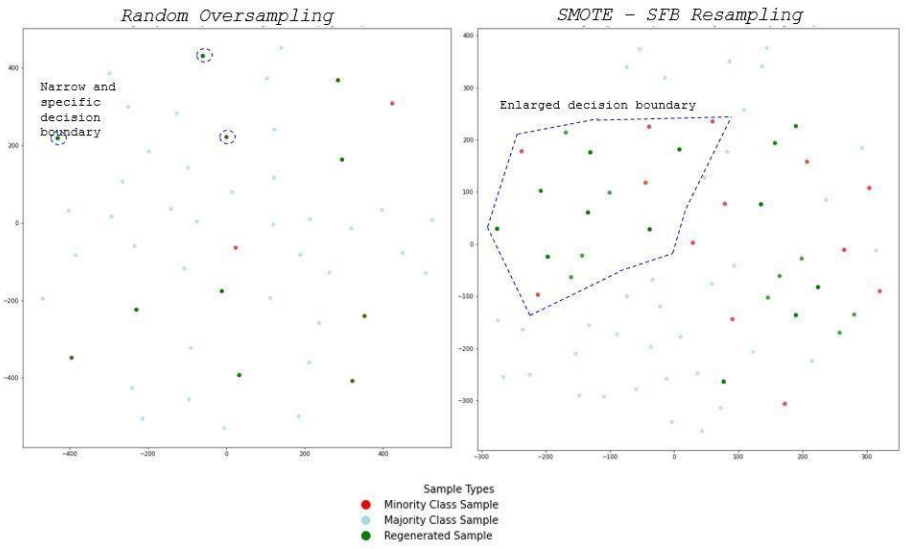

Fig. 5. – t-SNE visualization for resampled data. Left panel: Random oversampling based resampled data. Right panel: SMOTE-MSFB based resampled data.

As observed in the visualization, Random oversampling replicates the existing minority class observations. This leads the classification model to identify similar but more specific regions in the feature space as the decision region for the minority class [36]. With SMOTE-MSFB, the synthetic observations are generated in regions such that the decision region is enlarged to include spaces between the minority class samples.

Empirical evaluation & results

In order to establish the efficacy of our approach, we conducted experiments on two publicly available datasets using the Movie Lens dataset and the Tencent Ads competition dataset 2018.

Movie Lens Dataset

The public dataset [55] contains 6,040 users; each of them consists of user ID, gender, age, occupation, and zip code, and holds ~3,900 movies, each movie including movie ID, title, and genres. And it was rated by user with a score that among of 5 scale, and recorded timestamp for the rating behavior.

Generating small seed list datasets. Our problem statement is targeted towards small seed list situations. In order to generate the data suitable for evaluating our approach, we firstly clustered the all the movie groups into 100 clusters based movie genres and the normalized years which was extracted from the movie title using the kmeans method. Each cluster was regarded as an ad group and target audiences were found for each ad group. We selected all the clusters where the ratio of minority class is less than 25% of the dataset and the size of the minority class is less than 1000. There were 48 such clusters. The rating data was used for creating the binary classification dataset. All users who rated the movie 5 were tagged as seed users (labeled as 1) and rest were tagged as non-seed users (labeled as 0). This is the same approach as used by [56] except they use rating 5

and 4 for generating seed user where as we use only rating 5. The details of size and balance of each of the cluster is provided in the appendix.

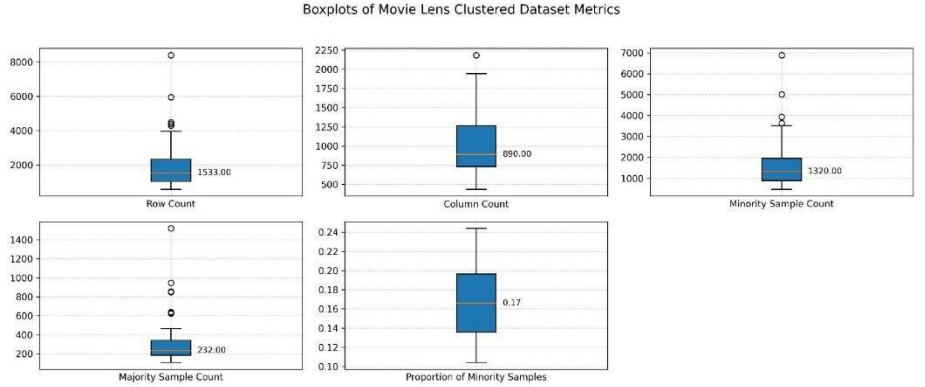

Fig. 6. Summary statistics of the clustered datasets generated from Movie Lens Dataset

Tencent Ads dataset 2018

It is a public dataset released for Tencent Ads Competitions in 2018. The dataset contains the data for 173 advertiser IDs including the seed and non-seed audiences. It contains features related to creatives, product, campaign and audience attributes. All the features were converted to binary attributes denoting the presence or absence of a feature. Many of the attributes are multi valued attributes like interest1 which are first parsed to create a list of unique values and then one hot encoded to create the final feature list. This leads to an average dimensionality of 18078 binary covariate variables across the 173 datasets with a max and min covariate length of 23004 and 13284 respectively.

As we are interested in shorter seed list situation, for each advertiser ID we randomly sampled 400 seed audiences as minority class observations and 2000 non-seed audiences as majority class observation keeping the imbalance at 25%.

Comparison Methodology

We used SMOTEN as the benchmarking resampling algorithm for this study. 5-fold cross validation procedure was used for benchmarking the resampling algorithms. In order to ensure strict comparison, the same cross validation folds were applied on the dataset for both SMOTEN and SMOTE-MSFB and subsequent classification algorithms. The resampling was only performed on the 4 fold training data of the folds ensuring no data leakage. The accuracy metrics were calculated on the remaining fold.

Logistic regression, Naïve Bayes and SVC with linear kernel were used as the classification algorithms post resampling for the benchmarking metrics. Logistic regression and Naïve Bayes does not required any parameter setting in their formulation. For SVC, we used the value of C=1 as it is suggested as good default values in the literature [57]. The sampling strategy used for SMOTE-MSFB and SMOTEN is to balance the majority and minority samples in the data post resampling for each cluster.

Test ROC AUC metric is used for comparing the performance of two resampling algorithms. While overall accuracy is a commonly used performance metric for predictive models, it can be misleading in the context of class-imbalanced

datasets. In such scenarios, a model may achieve high accuracy despite misclassifying a large number of minority class instances. To address this limitation, alternative evaluation metrics such as recall, precision, F-measure, and the Area under the Receiver Operating Characteristic Curve (AUC) are commonly employed. A key advantage of the ROC curve is its robustness to class imbalance, making it a valuable tool for evaluating model performance on imbalanced datasets. The Area Under the Curve (AUC) serves as a quantitative measure of the ROC curve's performance [36, 49, 52, 58].
Statistical tests were conducted on the results of the experiments to establish better performance. We use paired t-test for comparing the AUC across multiple data clusters generated from Movie Lens and Tencent data. The NULL hypothesis posits that SMOTEN is better than SMOTE-MSFB while the alternative hypothesis denoting otherwise.

Results

Movie Lens Dataset - Out of the 48 clusters with imbalanced response, SMOTEMSFB performs better than SMOTEN resampling algorithm for 41 clusters. Table 1 provide the details on the results. The bolded values in the table is the higher of the AUC value.

Table 1. ROC AUC Results comparing the SMOTEN Vs SMOTE-MSFB for 48 success clusters of Movie Lens Dataset

	SMOTE-MSFB				**SMOTEN**			
ID	**LR**	**Naïve Bayes**	**SVC**	**Mean AUC**	**LR**	**Naïve Bayes**	**SVC**	**Mean AUC**
1	0.5017	0.5081	0.5184	**0.5094**	0.4670	0.4269	0.5218	0.4719
2	0.4812	0.5223	0.4943	0.4993	0.5038	0.4872	0.5083	**0.4998**
3	0.4832	0.5115	0.5015	0.4987	0.5188	0.5329	0.4934	**0.5150**
4	0.5511	0.5429	0.5418	**0.5453**	0.5148	0.4798	0.4846	0.4931
5	0.5313	0.5359	0.5351	**0.5341**	0.5152	0.5145	0.5412	0.5236
6	0.5613	0.5437	0.5617	**0.5556**	0.5316	0.4552	0.5313	0.5060
7	0.5475	0.5543	0.5546	**0.5521**	0.5254	0.4793	0.5304	0.5117
8	0.5556	0.5390	0.5475	**0.5473**	0.5326	0.5096	0.5220	0.5214
9	0.5333	0.5227	0.5288	0.5283	0.5393	0.5441	0.5418	**0.5417**
10	0.5547	0.5178	0.5369	**0.5365**	0.5408	0.5408	0.5211	0.5342
11	0.5206	0.5248	0.5676	**0.5376**	0.5236	0.5208	0.5631	0.5358
12	0.5304	0.5328	0.5525	0.5386	0.5354	0.5235	0.5649	**0.5413**
13	0.5543	0.5676	0.5499	**0.5573**	0.5219	0.5376	0.5512	0.5369
14	0.5586	0.5595	0.5540	**0.5574**	0.5500	0.5429	0.5559	0.5496
15	0.5490	0.5587	0.5847	**0.5642**	0.5253	0.5372	0.5760	0.5462
16	0.5603	0.5504	0.5571	0.5560	0.5575	0.5541	0.5652	**0.5589**
17	0.5771	0.5700	0.5684	**0.5719**	0.5621	0.5457	0.5612	0.5563
18	0.5762	0.5788	0.5889	**0.5813**	0.5616	0.5433	0.5654	0.5568
19	0.5688	0.5985	0.5987	**0.5887**	0.5494	0.5460	0.5554	0.5502
20	0.5881	0.5752	0.5836	0.5823	0.5886	0.5977	0.5960	**0.5941**

21	0.6020	0.5925	0.5955	**0.5967**	0.5877	0.5575	0.5982	0.5811
22	0.5823	0.5856	0.6062	**0.5914**	0.5835	0.6000	0.5858	0.5898
23	0.6144	0.6030	0.5929	**0.6034**	0.5953	0.5540	0.5930	0.5808
24	0.6093	0.6016	0.5984	**0.6031**	0.6058	0.5867	0.5864	0.5930
25	0.6343	0.6254	0.6030	**0.6209**	0.5889	0.5783	0.5711	0.5795
26	0.6316	0.6244	0.6188	**0.6249**	0.6051	0.5980	0.5502	0.5845
27	0.6143	0.6174	0.6223	**0.6180**	0.5986	0.5867	0.6134	0.5996
28	0.6226	0.6061	0.6208	**0.6165**	0.6069	0.5892	0.6114	0.6025
29	0.6158	0.6193	0.6147	**0.6166**	0.6194	0.5872	0.6184	0.6083
30	0.6264	0.6184	0.6251	**0.6233**	0.6121	0.5996	0.6121	0.6079
31	0.6357	0.6002	0.6392	**0.6250**	0.6164	0.5848	0.6291	0.6101
32	0.6280	0.6266	0.6145	**0.6230**	0.6266	0.5999	0.6112	0.6126
33	0.6251	0.6055	0.6192	0.6166	0.6306	0.6209	0.6214	**0.6243**
34	0.6327	0.5888	0.6467	**0.6228**	0.6244	0.6123	0.6228	0.6199
35	0.6426	0.6173	0.6454	**0.6351**	0.6291	0.6189	0.6156	0.6212
36	0.6512	0.6298	0.6508	**0.6439**	0.6321	0.6306	0.6187	0.6271
37	0.6622	0.6854	0.6372	**0.6616**	0.6244	0.6531	0.5864	0.6213
38	0.6742	0.6627	0.6588	**0.6652**	0.6468	0.6133	0.6278	0.6293
39	0.6678	0.6511	0.6614	**0.6601**	0.6444	0.6244	0.6523	0.6404
40	0.6718	0.6390	0.6550	**0.6552**	0.6652	0.6287	0.6446	0.6462
41	0.6778	0.6535	0.6702	**0.6672**	0.6762	0.6557	0.6405	0.6575
42	0.6929	0.6632	0.7025	**0.6862**	0.6502	0.6266	0.6785	0.6518
43	0.7145	0.6396	0.7188	**0.6910**	0.6822	0.6344	0.6814	0.6660
44	0.6990	0.6713	0.6819	**0.6840**	0.6927	0.6788	0.6720	0.6812
45	0.7007	0.7151	0.6656	**0.6938**	0.6827	0.6990	0.6374	0.6731
46	0.7475	0.7387	0.7495	**0.7452**	0.7321	0.7242	0.7309	0.7291
47	0.7678	0.7411	0.7649	**0.7579**	0.7440	0.7339	0.7368	0.7382
48	0.7745	0.7468	0.7516	**0.7576**	0.7659	0.7211	0.7297	0.7389

Statistical Significance results: Using cross validated AUC metric on the 48 distinct clusters, Paired t-test was executed to compare the statistical performance. Table 2 below elaborates on the results. SMOTE-MSFB is performs better than SMOTEN and results are statistically significant.

Table 2. Paired t-test results on Movie Lens Dataset

Resampling Method	**Mean AUC**	**Standard Deviation**
SMOTEN	0.5908	0.0627
SMOTE-MSFB	0.6072	0.0637
T-Statistics	7	
P-value	8.2830e-09	

Runtime comparison. Fig 7 provides the comparison of standard SMOTEN algorithm to SMOTE-MSFB in terms of run time. The analysis was performed on an Ubuntu Linux machine with 500GB RAM, Intel Xeon Platinum CPU @ 2.50 GHz processor. SMOTE-MSFB is ~70% efficient in terms of run time as compared

to the standard SMOTEN algorithm. Across the 48 clusters, SMOTE-MSFB takes ~3.74 seconds to resample the data as compared to ~13.62 seconds taken by SMOTEN.

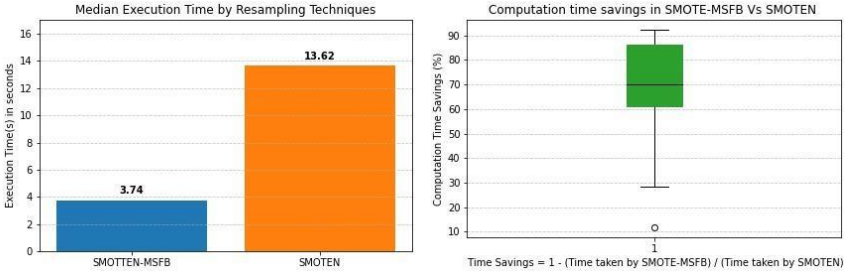

Fig. 7. Comparison of the run time of resampling algorithms 1) SMOTE-MSFB Vs 2) SMOTEN. SMOTE-MSFB is ~70% more efficient than the standard SMOTEN algorithm

Tencent Ads Dataset. Out of the 173 distinct aids present in the data, SMOTEMSFB performs better than SMOTEN resampling algorithm for 152 instances. Table 2 provide the details on the results. The bolded values in the table is the higher of the mean AUC value.

Table 3. ROC AUC Results comparing the SMOTEN Vs SMOTE-MSFB for 173 AIDs clusters of Tencent Dataset

	SMOTE - MSFB				**SMOTEN**			
ID	**LR**	**Naïve Bayes**	**SVC**	**Mean AUC**	**LR**	**Naïve Bayes**	**SVC**	**Mean AUC**
6	0.6107	0.5355	0.5978	**0.5814**	0.5939	0.5103	0.5854	0.5632
7	0.7596	0.6195	0.7532	**0.7108**	0.7510	0.5773	0.7497	0.6927
12	0.7800	0.6790	0.7289	**0.7293**	0.7501	0.5991	0.7067	0.6853
18	0.5234	0.5597	0.5147	**0.5326**	0.5028	0.4521	0.5053	0.4867
70	0.8271	0.7524	0.7858	**0.7884**	0.8015	0.6802	0.7693	0.7503
74	0.6621	0.5936	0.6472	**0.6343**	0.6541	0.5005	0.6400	0.5982
86	0.5532	0.5986	0.5575	**0.5697**	0.5446	0.4530	0.5490	0.5155
98	0.7602	0.6473	0.7367	**0.7147**	0.7491	0.5489	0.7345	0.6775
113	0.5596	0.5495	0.5474	**0.5522**	0.5661	0.4693	0.5578	0.5310
117	0.6156	0.5522	0.6024	**0.5901**	0.6149	0.5499	0.5996	0.5881
121	0.8131	0.7302	0.7824	**0.7752**	0.7931	0.6584	0.7726	0.7414
136	0.5959	0.5888	0.5766	**0.5871**	0.5713	0.4783	0.5670	0.5388
145	0.7436	0.6260	0.7310	**0.7002**	0.7297	0.6000	0.7238	0.6845
164	0.8619	0.8240	0.8220	**0.8360**	0.8345	0.6688	0.7975	0.7670
173	0.7125	0.6411	0.6835	**0.6790**	0.6932	0.4978	0.6696	0.6202
174	0.6724	0.5118	0.6538	0.6127	0.6682	0.5640	0.6508	**0.6277**
177	0.5425	0.5714	0.5388	**0.5509**	0.5471	0.4670	0.5419	0.5186
191	0.6929	0.6243	0.6604	**0.6592**	0.6747	0.6156	0.6533	0.6479
205	0.9305	0.9115	0.9142	**0.9188**	0.9098	0.7937	0.8982	0.8672

206 0.5091 0.5518 0.5097 **0.5235** 0.4979 0.4545 0.5034 0.4853
212 0.5671 0.5512 0.5497 **0.5560** 0.5682 0.5156 0.5503 0.5447
231 0.6108 0.5597 0.5855 **0.5854** 0.6117 0.5287 0.5836 0.5747
272 0.8972 0.7932 0.8903 **0.8602** 0.8833 0.7072 0.8812 0.8239
286 0.7131 0.6273 0.6926 **0.6777** 0.6922 0.6492 0.6761 0.6725
302 0.6437 0.5407 0.6216 **0.6020** 0.6316 0.5568 0.6173 0.6019
311 0.7066 0.7202 0.6666 **0.6978** 0.6677 0.6815 0.6426 0.6640
313 0.9272 0.9075 0.9128 **0.9158** 0.9116 0.7782 0.9034 0.8644
336 0.9244 0.8656 0.9212 **0.9038** 0.9080 0.7370 0.9014 0.8488
369 0.5736 0.5325 0.5705 **0.5588** 0.5659 0.4819 0.5664 0.5380
389 0.6800 0.6248 0.6546 **0.6531** 0.6701 0.6137 0.6501 0.6446
404 0.5862 0.6171 0.5824 **0.5952** 0.5619 0.4012 0.5736 0.5122
411 0.6488 0.5995 0.6071 **0.6185** 0.6343 0.5573 0.6035 0.5984
420 0.6018 0.5836 0.5812 **0.5889** 0.5890 0.4636 0.5739 0.5422
432 0.6326 0.5357 0.6052 0.5912 0.6358 0.5527 0.6068 **0.5985**
436 0.5845 0.4965 0.5553 0.5454 0.5856 0.5604 0.5591 **0.5683**
450 0.9564 0.8739 0.9534 **0.9279** 0.9394 0.7594 0.9360 0.8783
454 0.6583 0.5257 0.6461 0.6100 0.6513 0.5652 0.6319 **0.6162**
471 0.6189 0.4819 0.6214 0.5741 0.6095 0.5359 0.6141 **0.5865**
516 0.8363 0.7622 0.8297 **0.8094** 0.8223 0.6213 0.8223 0.7553
519 0.9117 0.8441 0.8889 **0.8816** 0.8925 0.6844 0.8676 0.8149
529 0.5466 0.5560 0.5357 **0.5461** 0.5388 0.5644 0.5333 0.5455
543 0.5388 0.5673 0.5385 **0.5482** 0.5403 0.4585 0.5418 0.5135
561 0.6316 0.5310 0.6044 0.5890 0.6278 0.5500 0.6005 **0.5928**
562 0.6129 0.5517 0.5987 **0.5878** 0.5958 0.5247 0.5875 0.5693
613 0.9175 0.8715 0.9044 **0.8978** 0.9081 0.7993 0.8903 0.8659
624 0.5322 0.5923 0.5168 **0.5471** 0.5038 0.4203 0.5011 0.4751
647 0.5386 0.5715 0.5277 **0.5460** 0.5323 0.4568 0.5212 0.5034
660 0.5686 0.5288 0.5550 **0.5508** 0.5620 0.4973 0.5478 0.5357
671 0.8790 0.7429 0.8541 **0.8253** 0.8626 0.6612 0.8400 0.7879
681 0.7324 0.6392 0.7125 **0.6947** 0.7243 0.5821 0.7117 0.6727
686 0.7781 0.6391 0.7658 **0.7277** 0.7496 0.5265 0.7434 0.6732
688 0.6865 0.6447 0.6511 **0.6608** 0.6805 0.6041 0.6486 0.6444
692 0.5708 0.5568 0.5598 **0.5624** 0.5487 0.4936 0.5415 0.5279
699 0.5490 0.5645 0.5491 **0.5542** 0.5510 0.4799 0.55330.5281
725 0.8633 0.8074 0.8337 **0.8348** 0.8375 0.6384 0.80720.7611
727 0.7506 0.6526 0.7245 **0.7092** 0.7349 0.6244 0.70840.6893
748 0.4974 0.5380 0.4916 **0.5090** 0.4847 0.4695 0.48260.4790
765 0.8221 0.7434 0.7892 **0.7849** 0.7964 0.6362 0.7654 0.7327
792 0.6975 0.5726 0.6838 **0.6513** 0.6915 0.5343 0.6813 0.6357
838 0.9047 0.8573 0.8890 **0.8836** 0.8903 0.7121 0.8782 0.8269
846 0.8192 0.7265 0.7916 **0.7791** 0.7979 0.6783 0.7748 0.7503
853 0.5931 0.5603 0.5921 **0.5818** 0.5878 0.5078 0.5892 0.5616
875 0.8240 0.7488 0.8210 **0.7979** 0.8114 0.6549 0.8154 0.7606
886 0.7153 0.6293 0.6977 **0.6808** 0.7115 0.6054 0.6954 0.6708
894 0.6385 0.5083 0.6119 0.5862 0.6436 0.5701 0.6177 **0.6105**
903 0.7951 0.7393 0.7578 **0.7641** 0.7851 0.6930 0.7529 0.7437
914 0.5789 0.5838 0.5696 **0.5775** 0.5663 0.4540 0.5562 0.5255
916 0.8466 0.7425 0.8173 **0.8021** 0.8271 0.7163 0.8006 0.7813
927 0.9042 0.7857 0.8966 **0.8621** 0.8887 0.6573 0.8800 0.8087

932	0.7854	0.6714	0.7647	**0.7405**	0.7667	0.5697	0.7507	0.6957
939	0.5681	0.5617	0.5594	**0.5631**	0.5548	0.4670	0.5512	0.5243
951	0.8132	0.7437	0.8077	**0.7882**	0.8125	0.6441	0.8126	0.7564
960	0.5109	0.5439	0.5285	**0.5278**	0.5049	0.4576	0.5264	0.4963
966	0.6556	0.5874	0.6407	**0.6279**	0.6463	0.5763	0.6448	0.6225
975	0.8256	0.7350	0.8186	**0.7931**	0.8245	0.6564	0.8171	0.7660
977	0.5353	0.5025	0.5325	**0.5234**	0.5194	0.5182	0.5171	0.5182
1017	0.6370	0.5665	0.6261	**0.6099**	0.6201	0.5413	0.6111	0.5908
1021	0.8494	0.7498	0.8148	**0.8047**	0.8331	0.6122	0.8030	0.7494
1023	0.6876	0.5961	0.6639	**0.6492**	0.6703	0.5610	0.6488	0.6267
1027	0.8148	0.6903	0.8108	**0.7720**	0.8035	0.6715	0.7995	0.7582
1044	0.6065	0.5420	0.5793	**0.5759**	0.5991	0.5394	0.5753	0.5713
1057	0.9433	0.8832	0.9338	**0.9201**	0.9395	0.7009	0.9287	0.8564
1085	0.8131	0.7496	0.8021	**0.7883**	0.8160	0.6402	0.8100	0.7554
1107	0.7911	0.8118	0.7402	**0.7810**	0.7790	0.7435	0.7352	0.7525
1119	0.5577	0.5478	0.5411	**0.5488**	0.5519	0.5123	0.5378	0.5340
1140	0.6958	0.5989	0.6689	**0.6545**	0.6870	0.5631	0.6667	0.6389
1171	0.8713	0.8387	0.8267	**0.8456**	0.8497	0.6753	0.8115	0.7789
1182	0.6613	0.5974	0.6419	**0.6335**	0.6556	0.5822	0.6411	0.6263
1201	0.7755	0.6412	0.7422	**0.7196**	0.7477	0.6052	0.7251	0.6926
1202	0.5912	0.5890	0.5899	**0.5900**	0.5887	0.4541	0.5896	0.5441
1215	0.5321	0.5666	0.5193	**0.5393**	0.5119	0.4618	0.5081	0.4939
1230	0.6003	0.5154	0.5833	**0.5663**	0.5940	0.5183	0.5831	0.5651
1242	0.8865	0.8510	0.8775	**0.8717**	0.8770	0.6652	0.8688	0.8037
1254	0.7639	0.6287	0.7459	**0.7129**	0.7453	0.5744	0.7333	0.6843
1277	0.6879	0.5791	0.6648	**0.6440**	0.6590	0.5295	0.6472	0.6119
1284	0.8004	0.6586	0.7852	**0.7481**	0.7724	0.6124	0.7615	0.7155
1291	0.7252	0.6147	0.6955	**0.6785**	0.7109	0.5736	0.6907	0.6584
1317	0.6195	0.5490	0.6051	0.5912	0.6340	0.5422	0.6173	**0.5978**
1335	0.8173	0.7616	0.8046	**0.7945**	0.8168	0.6450	0.8106	0.7575
1338	0.6926	0.6230	0.6706	**0.6621**	0.6864	0.5472	0.6722	0.6353
1350	0.5320	0.4951	0.5271	0.5181	0.5320	0.5303	0.5290	**0.5304**
1351	0.8749	0.7999	0.8717	**0.8488**	0.8513	0.5738	0.8505	0.7585
1375	0.8829	0.8140	0.8508	**0.8492**	0.8556	0.6124	0.8265	0.7648
1377	0.8250	0.8022	0.7798	**0.8023**	0.7861	0.6812	0.7518	0.7397
1379	0.7572	0.6579	0.7255	**0.7135**	0.7379	0.6348	0.7168	0.6965
1407	0.5276	0.5492	0.5211	**0.5326**	0.5277	0.4615	0.5228	0.5040
1415	0.5969	0.4938	0.5955	0.5621	0.5961	0.5275	0.5918	**0.5718**
1429	0.9194	0.8444	0.9167	**0.8935**	0.9004	0.7594	0.9041	0.8546
1449	0.8281	0.7287	0.8077	**0.7881**	0.8048	0.5358	0.7873	0.7093
1468	0.6602	0.5137	0.6417	0.6052	0.6547	0.6133	0.6464	**0.6381**
1483	0.7760	0.7030	0.7428	**0.7406**	0.7469	0.6628	0.7234	0.7110
1496	0.5684	0.5678	0.5586	**0.5650**	0.5508	0.4539	0.5451	0.5166
1503	0.7417	0.6810	0.7224	**0.7150**	0.7104	0.4346	0.7067	0.6172
1507	0.9183	0.8146	0.9071	**0.8800**	0.9046	0.6581	0.8942	0.8190
1508	0.7463	0.6674	0.7153	**0.7097**	0.7403	0.6394	0.7169	0.6989
1512	0.8210	0.7536	0.7944	**0.7897**	0.8047	0.6895	0.7851	0.7598
1530	0.6183	0.5282	0.5984	**0.5816**	0.6157	0.5216	0.6022	0.5798
1566	0.5523	0.5619	0.5455	**0.5532**	0.5420	0.4959	0.5338	0.5239
1580	0.5605	0.5879	0.5490	**0.5658**	0.5560	0.4643	0.5449	0.5217

15960.7689 0.7439 0.7329 **0.7486** 0.7523 0.6677 0.7198 0.7132
15990.5969 0.5983 0.5910 **0.5954** 0.5734 0.4413 0.5779 0.5309
16050.6587 0.5474 0.6168 0.6076 0.6539 0.5614 0.6135 **0.6096**
1621 0.79150.7395 0.7562 **0.7624** 0.7752 0.6466 0.7470 0.7229
1622 0.81800.7585 0.7784 **0.7850** 0.7939 0.6895 0.7609 0.7481
1635 0.63460.6079 0.6130 **0.6185** 0.6064 0.5341 0.5965 0.5790
1671 0.6346 0.6125 0.6047 **0.6172** 0.6270 0.5179 0.6031 0.5827
16720.9176 0.8649 0.8960**0.8928** 0.9087 0.7360 0.88920.8446
17140.5434 0.5626 0.5275 **0.5445** 0.5377 0.4578 0.52580.5071
17160.7045 0.6288 0.6676 **0.6670** 0.6863 0.6124 0.65770.6521
1728 0.5538 0.5467 0.5587 **0.5531** 0.5442 0.4557 0.5507 0.5169
1746 0.8531 0.7348 0.8511 **0.8130** 0.8448 0.6193 0.8440 0.7694
1749 0.6276 0.5446 0.6117 0.5946 0.6120 0.5817 0.5999 **0.5979**
1781 0.5739 0.5085 0.5575 0.5466 0.5709 0.5201 0.5546 **0.5486**
1790 0.7604 0.6760 0.7250 **0.7204** 0.7443 0.4698 0.7120 0.6420
1819 0.7718 0.6713 0.7444 **0.7292** 0.7668 0.6539 0.7474 0.7227
1827 0.7256 0.6872 0.6938 **0.7022** 0.7076 0.6342 0.6816 0.6745
1841 0.6831 0.6064 0.6649 **0.6515** 0.6662 0.5497 0.6549 0.6236
1842 0.8024 0.7546 0.7570 **0.7713** 0.7756 0.6821 0.7381 0.7319
1847 0.7861 0.7188 0.7535 **0.7528** 0.7716 0.6361 0.7443 0.7173
1871 0.6235 0.6002 0.5883 **0.6040** 0.6208 0.5167 0.5904 0.5760
1894 0.8027 0.7326 0.7661 **0.7671** 0.7887 0.6584 0.7551 0.7341
1904 0.5463 0.5246 0.5321 0.5344 0.5487 0.5447 0.5348 **0.5427**
1910 0.6826 0.6247 0.6591 **0.6555** 0.6640 0.4650 0.6480 0.5923
1918 0.6033 0.4707 0.5841 0.5527 0.6094 0.5621 0.5900 **0.5872**
1925 0.8128 0.7232 0.8137 **0.7832** 0.7923 0.4958 0.7912 0.6931
1930 0.8674 0.8133 0.8216 **0.8341** 0.8374 0.5801 0.7957 0.7377
1931 0.5486 0.5004 0.5350 0.5280 0.5474 0.5320 0.5336 **0.5376**
1940 0.8574 0.8658 0.8160 **0.8464** 0.8232 0.7123 0.7835 0.7730
1950 0.7697 0.6752 0.7467 **0.7305** 0.7400 0.5145 0.7240 0.6595
1957 0.8905 0.8421 0.8541 **0.8622** 0.8490 0.5302 0.8000 0.7264
1962 0.5548 0.5926 0.5507 **0.5660** 0.5440 0.4630 0.5398 0.5156
1966 0.8193 0.7439 0.7826 **0.7819** 0.7982 0.6565 0.7633 0.7393
1970 0.7678 0.7092 0.7259 **0.7343** 0.7376 0.6401 0.7037 0.6938
1991 0.6588 0.5803 0.6316 **0.6236** 0.6387 0.6088 0.6197 0.6224
1998 0.6089 0.5738 0.5951 **0.5926** 0.5932 0.5105 0.5860 0.5632
2013 0.6972 0.6373 0.6685 **0.6677** 0.6760 0.5927 0.6488 0.6392
2031 0.7808 0.6568 0.7512 **0.7296** 0.7563 0.6215 0.7347 0.7042
2044 0.6354 0.6000 0.6158 **0.6171** 0.6312 0.4972 0.6106 0.5796
2047 0.55020.5592 0.5498 **0.5531** 0.5470 0.4686 0.5500 0.5219
2048 0.54790.5742 0.5401 **0.5541** 0.5255 0.4535 0.5297 0.5029
20500.6903 0.6343 0.6638 **0.6628** 0.6750 0.5575 0.6506 0.6277
20540.9323 0.7177 0.9205 0.8568 0.9136 0.7790 0.8980 **0.8635**
20660.5997 0.5368 0.5815 0.5727 0.5967 0.5463 0.5812 **0.5748**
20680.8141 0.7603 0.7893 **0.7879** 0.8040 0.4540 0.7885 0.6822
21120.6519 0.5260 0.6232 0.6004 0.6541 0.6191 0.6276 **0.6336**
2118 0.6414 0.5356 0.6127 0.5966 0.6373 0.5717 0.6144 **0.6078**
2154 0.8005 0.6942 0.7966 **0.7638** 0.7837 0.6206 0.7862 0.7302
2169 0.6515 0.5733 0.6224 **0.6157** 0.6335 0.5630 0.6112 0.6026
2196 0.8301 0.7248 0.8230 **0.7926** 0.8252 0.6119 0.8203 0.7524

2197	0.6195	0.6487	0.5861	**0.6181**	0.5858	0.5437	0.5564	0.5619
2201	0.8143	0.7054	0.7983	**0.7727**	0.7994	0.5806	0.7926	0.7242
2205	0.8783	0.7665	0.8607	**0.8352**	0.8633	0.6578	0.8509	0.7907
2216	0.5360	0.5076	0.5221	**0.5219**	0.5274	0.5118	0.5163	0.5185

Statistical Significance results. Using cross validated AUC metric on the 173 AID dataset, Paired t-test was executed to compare the statistical performance. Table 3 below elaborates on the results. SMOTE-MSFB is performs better than SMOTEN and results are statistically significant.

Table 4. Paired t-test results on Tencent Ads Dataset

Resampling Method	Mean AUC	Standard Deviation
SMOTEN	0.6499	0.1036
SMOTE-MSFB	0.6815	0.1147
T-Statistics	15.01	
P-value	4.1524e-33	

Runtime comparison. Fig 8 provides the comparison of standard SMOTEN algorithm to SMOTE-MSFB in terms of run time. The analysis was performed on an Ubuntu Linux machine with 500GB RAM, Intel Xeon Platinum CPU @ 2.50 GHz processor. SMOTE-MSFB is ~84% efficient in terms of run time as compared to the standard SMOTEN algorithm. Across the 173 AIDs, SMOTE-MSFB takes ~69 seconds to resample the data as compared to ~425 seconds taken by SMOTEN

Fig. 8. Comparison of the run time of resampling algorithms 1) SMOTE-MSFB Vs 2) SMOTEN. SMOTE-MSFB is ~80% more efficient than the standard SMOTEN algorithm

Conclusion

In this work, we present a novel way of handling shorter seed lists for generating expanded audiences for advertisers. As new privacy laws are incrementally applied across geographies, they reduce the predictive power of the data for targeting the right audiences. Our approach helps the advertisers mitigate the impact of the privacy laws.

We present a novel extension of SMOTE called SMOTE-MSFB, oversampling approach specifically designed for large/high dimensional sparse binary data of digital advertising. The technique was carried out in stages. First, identification of misclassified seed observations to focus the resampling efforts on hard to classify minority samples. Secondly, using mutual information based feature selection process to remove irrelevant features enabling a more apt the KNN based neighbourhood definition. Lastly, given the sparse nature of the binary covariate space, union voting to generate synthetic observations from the defined neighbourhood. The synthetic data generated through SMOTE-MSFB clearly expands the decision boundary to include sub-spaces surrounded by minority samples. Thus generating superior performance as compared to standard SMOTEN algorithm on the problem dataset. Also our algorithm takes less than half the computation time as compared to SMOTEN.

Appendix

Details of the clusters generated using the Movie Lens Dataset.

Cluster ID	Rows	Cols	Majority Samples	Minority Samples	Minority Class Proportion
1	1022	696	892	130	13%
2	781	603	670	111	14%
3	604	448	495	109	18%
4	1018	724	885	133	13%
5	2170	1149	1828	342	16%
6	973	829	783	190	20%
7	1769	1262	1450	319	18%
8	1799	1436	1446	353	20%
9	725	508	550	175	24%
10	704	569	536	168	24%
11	1533	1159	1320	213	14%
12	1011	657	889	122	12%
13	1475	856	1137	338	23%
14	3449	1304	2812	637	18%
15	2614	1521	1990	624	24%
16	1389	846	1060	329	24%
17	3969	1437	3506	463	12%
18	1567	879	1333	234	15%
19	1297	1000	1146	151	12%
20	951	651	737	214	23%
21	634	434	494	140	22%
22	1046	657	892	154	15%
23	1900	1199	1645	255	13%
24	2169	1334	1814	355	16%
25	4484	1411	3636	848	19%

Cluster ID	Rows	Cols	Majority Samples	Minority Samples	Minority Class Proportion
26	1588	837	1354	234	15%
27	1636	997	1462	174	11%
28	1292	928	1051	241	19%
29	1224	705	1025	199	16%
30	3858	1542	3230	628	16%
31	2640	1162	2343	297	11%
32	1548	1004	1328	220	14%
33	1215	812	1050	165	14%
34	1385	736	1094	291	21%
35	1966	1016	1640	326	17%
36	984	733	753	231	23%
37	578	497	471	107	19%
38	1631	810	1455	176	11%
39	1056	682	825	231	22%
40	1254	860	1022	232	19%
41	1618	1002	1416	202	12%
42	1601	890	1391	210	13%
43	2840	1309	2508	332	12%
44	1234	814	1022	212	17%
45	754	665	621	133	18%
46	4400	1616	3942	458	10%
47	2329	1000	1961	368	16%
48	1710	856	1507	203	12%

References

1. Perlich, C., Dalessandro, B., Raeder, T., Stitelman, O., Provost, F.: Machine learning for targeted display advertising: transfer learning in action. Mach Learn. 95, 103–127 (2014). https://doi.org/10.1007/s10994-013-5375-2.
2. Liu, H., Pardoe, D., Liu, K., Thakur, M., Cao, F., Li, C.: Audience Expansion for Online Social Network Advertising. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 165–174. ACM, San Francisco California USA (2016). https://doi.org/10.1145/2939672.2939680.
3. Yan Qu, Jing Wang: System and methods for generating expanded user segments.
4. Jiang, J., Lin, X., Yao, J., Lu, H.: Comprehensive Audience Expansion based on End-to-End Neural Prediction. (2019).
5. Ma, Q., Wen, M., Xia, Z., Chen, D.: A Sub-linear, Massive-scale Look-alike Audience Extension System. (2016).
6. Carvalhaes, C.: Reframing Audience Expansion through the Lens of Probability Density Estimation, http://arxiv.org/abs/2311.05853, (2023).

7. Popov, A., Iakovleva, D.: Adaptive look-alike targeting in social networks advertising. Procedia Computer Science. 136, 255–264 (2018). https://doi.org/10.1016/j.procs.2018.08.264.
8. Tziortziotis, N., Qiu, Y., Hue, M., Vazirgiannis, M.: Audience expansion based on user browsing history. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE, Shenzhen, China (2021). https://doi.org/10.1109/IJCNN52387.2021.9533392.
9. The Connecticut Data Privacy Act, https://portal.ct.gov/ag/sections/privacy/theconnecticut-data-privacy-act, last accessed 2025/06/05.
10. California Consumer Privacy Act (CCPA), https://oag.ca.gov/privacy/ccpa, last accessed 2025/06/05.
11. New York privacy act, https://nyassembly.gov/leg/?default_fld=&leg_video=&bn=S00365&term=2023&Su mmary=Y&Actions=Y&Text=Y, last accessed 2025/06/05.
12. Colorado Privacy Act (CPA), https://coag.gov/resources/colorado-privacyact/, last accessed 2025/06/05.
13. Virginia - Consumer Data Protection Act, https://law.lis.virginia.gov/vacodefull/title59.1/chapter53/, last accessed 2025/06/05.
14. Utah Consumer Privacy Act, https://dcp.utah.gov/ucpa/, last accessed 2025/06/05.
15. Washington My Health My Data Act, https://www.atg.wa.gov/protectingwashingtonians-personal-health-data-and-privacy, last accessed 2025/06/05.
16. Nevada consumer health data privacy law, https://www.leg.state.nv.us/App/NELIS/REL/82nd2023/Bill/10323/Overview, last accessed 2025/06/05.
17. Texas Data Privacy And Security Act, https://www.texasattorneygeneral.gov/consumer-protection/file-consumercomplaint/consumer-privacy-rights/texas-data-privacy-and-security-act, last accessed 2025/06/05.
18. Florida digital bill of rights, https://www.flsenate.gov/Session/Bill/2023/262/BillText/er/HTML, last accessed 2025/06/05.
19. Delaware Personal Data Privacy Act, https://legis.delaware.gov/json/BillDetail/GenerateHtmlDocument?legislationId=1403 88&legislationTypeId=1&docTypeId=2&legislationName=HB154, last accessed 2025/06/05.
20. Nebraska data privacy act, https://protectthegoodlife.nebraska.gov/dataprivacy-homepage, last accessed 2025/06/05.
21. New Hampshire Data Privacy Act, https://www.doj.nh.gov/data-privacyenforcement, last accessed 2025/06/05.
22. New Jersey Data Privacy Law, https://www.njconsumeraffairs.gov/ocp/Pages/NJ-Data-Privacy-Law-FAQ.aspx, last accessed 2025/06/05.
23. Mangalampalli, A., Ratnaparkhi, A., Hatch, A.O., Bagherjeiran, A., Parekh, R., Pudi, V.: A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns. In:

Proceedings of the 20th international conference companion on World wide web. pp. 85–86. ACM, Hyderabad India (2011). https://doi.org/10.1145/1963192.1963236.
24. Zhuzhel, V., Grabar, V., Kaploukhaya, N., Rivera-Castro, R., Mironova, L., Zaytsev, A., Burnaev, E.: No Two Users Are Alike: Generating Audiences with Neural Clustering for Temporal Point Processes. Dokl. Math. 108, S511–S528 (2023). https://doi.org/10.1134/S1064562423701661.
25. Shen, J., Geyik, S.C., Dasdan, A.: Effective Audience Extension in Online Advertising. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 2099–2108. ACM, Sydney NSW Australia (2015). https://doi.org/10.1145/2783258.2788603.
26. Ramesh, A., Teredesai, A., Bindra, A., Pokuri, S., Uppala, K.: Audience segment expansion using distributed in-database k-means clustering. In: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising. pp. 1– 9. ACM, Chicago Illinois (2013). https://doi.org/10.1145/2501040.2501982.
27. Frolov, D., Taran, Z., Mirkin, B.: A Method for Audience Extending in Programmatic Advertising by Using Parsimonious Generalization of User Segments. In: Ahram, T., Taiar, R., Colson, S., and Choplin, A. (eds.) Human Interaction and Emerging Technologies. pp. 837–841. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-25629-6_131.
28. Rajaraman, A., Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. (2012).
29. Ma, Q., Wagh, E., Wen, J., Xia, Z., Ormandi, R., Chen, D.: Score Look-Alike Audiences. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). pp. 647–654. IEEE, Barcelona, Spain (2016). https://doi.org/10.1109/ICDMW.2016.0097.
30. Doan, K.D., Yadav, P., Reddy, C.K.: Adversarial Factorization Autoencoder for Look-alike Modeling. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2803–2812. ACM, Beijing China (2019). https://doi.org/10.1145/3357384.3357807.
31. Liu, Y., Ge, K., Zhang, X., Lin, L.: Real-time Attention Based Look-alike Model for Recommender System. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2765–2773 (2019). https://doi.org/10.1145/3292500.3330707.
32. Liu, Z., Niu, X.-F., Zhuang, C., Tan, Y., Mu, Y., Gu, J., Zhang, G.: TwoStage Audience Expansion for Financial Targeting in Marketing. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 2629–2636. ACM, Virtual Event Ireland (2020). https://doi.org/10.1145/3340531.3412748.
33. Zhuang, C., Liu, Z., Zhang, Z., Tan, Y., Wu, Z., Liu, Z., Wei, J., Gu, J., Zhang, G., Zhou, J., Qi, Y.: Hubble: An Industrial System for Audience Expansion in Mobile Marketing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2455–2463. ACM, Virtual Event CA USA (2020). https://doi.org/10.1145/3394486.3403295.
34. Liu, C., Jin, S., Wang, D., Luo, Z., Yu, J., Zhou, B., Yang, C.: Constrained Oversampling: An Oversampling Approach to Reduce Noise Generation in Imbalanced Datasets With Class Overlapping. IEEE Access. 10, 91452–91465 (2022). https://doi.org/10.1109/ACCESS.2020.3018911.

35. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-LevelSMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., and Ho, T.-B. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 475–482. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3642-01307-2_43.
36. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. jair. 16, 321–357 (2002). https://doi.org/10.1613/jair.953.
37. Fahrudin, T., Buliali, J.L., Fatichah, C.: enhancing the performance of smote algorithm by using attribute weighting scheme and new selective sampling method for imbalanced data set.
38. Kirshners, A., Parshutin, S., Gorskis, H.: Entropy-Based Classifier Enhancement to Handle Imbalanced Class Problem. Procedia Computer Science. 104, 586– 591 (2017). https://doi.org/10.1016/j.procs.2017.01.176.
39. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson, Harlow (2014).
40. Saad Hussein, A., Li, T., Yohannese, C.W., Bashir, K.: A-SMOTE: A New Preprocessing Approach for Highly Imbalanced Datasets by Improving SMOTE: IJCIS. 12, 1412 (2019). https://doi.org/10.2991/ijcis.d.191114.002.
41. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A New OverSampling Method in Imbalanced Data Sets Learning. In: Huang, D.-S., Zhang, X.-P., and Huang, G.-B. (eds.) Advances in Intelligent Computing. pp. 878–887. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/11538059_91.
42. Haibo He, Yang Bai, Garcia, E.A., Shutao Li: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). pp. 1322–1328. IEEE, Hong Kong, China (2008). https://doi.org/10.1109/IJCNN.2008.4633969.
43. Mukherjee, M., Khushi, M.: SMOTE-ENC: A Novel SMOTE-Based Method to Generate Synthetic Data for Nominal and Continuous Features. ASI. 4, 18 (2021). https://doi.org/10.3390/asi4010018.
44. Stanfill, C., Waltz, D.: Toward memory-based reasoning. Commun. ACM. 29, 1213–1228 (1986). https://doi.org/10.1145/7902.7906.
45. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor. Newsl. 6, 1–6 (2004). https://doi.org/10.1145/1007730.1007733.
46. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. SIGKDD Explor. Newsl. 6, 80–89 (2004). https://doi.org/10.1145/1007730.1007741.
47. Shanab, A.A., Khoshgoftaar, T.M., Wald, R., Van Hulse, J.: Comparison of approaches to alleviate problems with high-dimensional and class-imbalanced data. In: 2011 IEEE International Conference on Information Reuse & Integration. pp. 234–239. IEEE, Las Vegas, NV, USA (2011). https://doi.org/10.1109/IRI.2011.6009552.
48. Blagus, R., Lusa, L.: Class prediction for high-dimensional class-imbalanced data. BMC Bioinformatics. 11, 523 (2010). https://doi.org/10.1186/1471-2105-11523.

49. Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A., Wald, R.: Feature Selection with High-Dimensional Imbalanced Data. In: 2009 IEEE International Conference on Data Mining Workshops. pp. 507–514. IEEE, Miami, FL (2009). https://doi.org/10.1109/ICDMW.2009.35.
50. Deepa, T., Punithavalli, M.: An E-SMOTE technique for feature selection in High-Dimensional Imbalanced Dataset. In: 2011 3rd International Conference on Electronics Computer Technology. pp. 322–324. IEEE, Kanyakumari, India (2011). https://doi.org/10.1109/ICECTECH.2011.5941710.
51. Qazi, N., Raza, K.: Effect of Feature Selection, SMOTE and under Sampling on Class Imbalance Classification. In: 2012 UKSim 14th International Conference on Computer Modelling and Simulation. pp. 145–150. IEEE, Cambridge, United Kingdom (2012). https://doi.org/10.1109/UKSim.2012.116.
52. Maldonado, S., López, J., Vairetti, C.: An alternative SMOTE oversampling strategy for high-dimensional datasets. Applied Soft Computing. 76, 380–389 (2019). https://doi.org/10.1016/j.asoc.2018.12.024.
53. Van der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research. (2008).
54. Schubert, E., Gertz, M.: Intrinsic t-Stochastic Neighbor Embedding for Visualization and Outlier Detection. In: Beecks, C., Borutta, F., Kröger, P., and Seidl, T. (eds.) Similarity Search and Applications. pp. 188–203. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-68474-1_13.
55. Harper, F.M., Konstan, J.A.: The MovieLens Datasets: History and Context. ACM Trans. Interact. Intell. Syst. 5, 1–19 (2016). https://doi.org/10.1145/2827872.
56. Ni, H., Wang, Z.: Feature Dual Supervision Model for the Searches of Online Advertising Audiences. Scientific Programming. 2023, 1–14 (2023). https://doi.org/10.1155/2023/1217898.
57. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 1–27 (2011). https://doi.org/10.1145/1961189.1961199.
58. Chen, X., Wasikowski, M.: FAST: a roc-based feature selection metric for small samples and imbalanced data classification problems. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 124–132. ACM, Las Vegas Nevada USA (2008). https://doi.org/10.1145/1401890.1401910.