IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 14, 2016, accepted October 1, 2016, date of publication October 26, 2016,
date of current version November 28, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2619719

Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study

ADNAN AMIN[1], SAJID ANWAR[1], AWAIS ADNAN[1], MUHAMMAD NAWAZ[1], NEWTON HOWARD[2], JUNAID QADIR[3], (Senior Member, IEEE), AHMAD HAWALAH[4], AND AMIR HUSSAIN[5], (Senior Member, IEEE)

[1]Center for Excellence in Information Technology, Institute of Management Sciences, Peshawar 25000, Pakistan
[2]Nuffield Department of Surgical Sciences, University of Oxford, Oxford, OX3 9DU, U.K.
[3]Information Technology University, Arfa Software Technology Park, Lahore 54000, Pakistan
[4]College of Computer Science and Engineering, Taibah University, Medina 344, Saudi Arabia
[5]Division of Computing Science and Maths, University of Stirling, Stirling, FK9 4LA, U.K.

Corresponding author: A. Amin (adnan.amin@live.co.uk)

The work of A. Hussain was supported by the U.K. Engineering and Physical Sciences Research Council under Grant EP/M026981/1.

ABSTRACT Customer retention is a major issue for various service-based organizations particularly telecom industry, wherein predictive models for observing the behavior of customers are one of the great instruments in customer retention process and inferring the future behavior of the customers. However, the performances of predictive models are greatly affected when the real-world data set is highly imbalanced. A data set is called imbalanced if the samples size from one class is very much smaller or larger than the other classes. The most commonly used technique is over/under sampling for handling the class-imbalance problem (CIP) in various domains. In this paper, we survey six well-known sampling techniques and compare the performances of these key techniques, i.e., mega-trend diffusion function (MTDF), synthetic minority oversampling technique, adaptive synthetic sampling approach, couples top-N reverse k-nearest neighbor, majority weighted minority oversampling technique, and immune centroids oversampling technique. Moreover, this paper also reveals the evaluation of four rules-generation algorithms (the learning from example module, version 2 (LEM2), covering, exhaustive, and genetic algorithms) using publicly available data sets. The empirical results demonstrate that the overall predictive performance of MTDF and rules-generation based on genetic algorithms performed the best as compared with the rest of the evaluated oversampling methods and rule-generation algorithms.

INDEX TERMS SMOTE, ADASYN, mega trend diffusion function, class imbalance, rough set, customer churn, mRMR. ICOTE, MWMOTE, TRkNN.

I. INTRODUCTION

In many subscription-based service industries such as telecommunications companies, are constantly striving to recognize customers that are looking to switch providers (i.e., Customer churn). Reducing churn is extremely important in competitive markets since acquiring new customers in such markets is very difficult (attracting non-subscribers can cost up to six times more than what it costs to retain the current customers by taking active steps to discourage churn behavior [1]). Churn-prone industries such as the telecommunication industry typically maintain customer relationship management (CRM) databases that are rich in unseen knowledge and certain patterns that may be exploited for acquiring customer information on time for intelligent decision-making practice of an industry [2].

However, knowledge discovery in such rich CRM databases, which typically contains thousands or millions of customers' information, is a challenging and difficult task. Therefore, many industries have to inescapably depend on prediction models for customer churn if they want to remain in the competitive market [1]. As a consequence, several competitive industries have implemented a wide range of statistical and intelligent machine learning (ML) techniques to develop predictive models that deal with customer churn [2].

Unfortunately, the performance of ML techniques is considerably affected by the CIP. The problem of imbalanced dataset appears when the proportion of majority class has a higher ratio than minority class [3], [4]. The skewed distribution (imbalanced) of data in the dataset poses challenges for machine learning and data mining algorithms [3], [5]. This is an area of research focusing on skewed class distribution where minority class is targeted for classification [6]. Consider a dataset where the imbalance ratio is 1:99 (i.e., where 99% of the instances belong to the majority class, and 1% belongs to the minority class). A classifier may achieve the accuracy up to 99% just by ignoring that 1% of minority class instances—however, adopting such an approach will result in the failure to correctly classify any instances of the class of interest (often the minority class).

It is well known that churn is a rare object in service-based industries, and that misclassification is more costly for rare objects or events in the case of imbalance datasets [7]. Traditional approaches, therefore, can provide misleading results on the class-imbalanced dataset—a situation that is highly significant and one that occurs in many domains [3], [6]. For instance, there are a significant number of real-world applications that are suffering from the class imbalance problem (e.g., medical and fault diagnosis, anomaly detection, face recognition, telecommunication, the web & email classification, ecology, biology and financial services [3], [4], [6], [85], [86]).

Sampling is a commonly used technique for handling the CIP in various domains. Broadly speaking, sampling can be categorized into oversampling and undersampling. A large number of studies focused on handling the class imbalance problem have been reported in literature [2], [4]. These studies can be grouped into the following approaches based on their dealing with class imbalance issue: (i) the internal-level: construct or update the existing methods to emphasize the significance of the minority class and (ii) the external level: adding data in preprocessing stage where the distribution of class is resampled in order to reduce the influence of imbalanced distribution of class in the classification process. The internal level approach is further divided into two groups [5]. Firstly, the cost-sensitive approach, which falls between internal and external level approaches, is based on reducing incorrect classification costs for minority class leading to reduction of the overall cost for both internal and external level approaches. Secondly, the ensemble/boosting approach, which adopts the use of multiple classifiers to follow the similar idea adopted by the internal approach. In this study, we have used six well-known advanced oversampling techniques—namely, Mega-trend Diffusion Function (MTDF), Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling approach (ADASYN), Majority Weighted Minority Oversampling Technique (MWMOTE), Immune centroids oversampling technique (ICOTE) and Couples Top-N Reverse k-Nearest Neighbor (TRkNN).

SMOTE [6] is commonly used as a benchmark for oversampling algorithm [7], [8]. ADASYN is also an important oversampling technique which improves the learning about the samples distribution in an efficient way [9]. MTDF was first proposed by Li *et al.* [10] and reported improved performance of classifying imbalanced medical datasets [11]. CUBE is also another advanced oversampling technique [12] but we have not considered this approach since as noted by Japkowicz [13], CUBE oversampling technique does not increase the predictive performance of the classifier. The above-mentioned oversampling approaches are used to handle the imbalanced dataset and improve the performance of predictive models for customer churn, particularly in the telecommunication sector. Rough Set Theory (RST) is applied to the four different rule-generation algorithms—(i) Learning from Example Module, version 2 (LEM2), (ii) Genetic (Gen), (iii) Covering (Cov) and (iv) Exhaustive (Exh) algorithms—in this study to observe the behavior of customer churn and all experiments are applied on the publicly available dataset. It is specified here that this paper is an extended version of our previous work [14], and makes the following contributions: (i) more datasets are employed to obtained more generalized results for the selected oversampling techniques and the rules-generation algorithms, (ii) another well-known oversampling technique—namely, ADASYN—is also used (iii) detailed analysis and discussion on the performance of targeted oversampling techniques (namely, ADASYN, MTDF, SMOTE, MWMOTE, ICOTE and TRkNN) followed by the rules-generation algorithms—namely, Gen, Cov, LEM2 and Exh, and (iv) detailed performance evaluation—in terms of the balance accuracy, the imbalance ratio, the area under the curve (AUC) and the McNemar's statistical test—is performed to validate the results and avoid any biases. Many comparative studies [2], [7], [11], [15], [16] have already been carried out on the comparison of oversampling and undersampling methods for handling the CIP; however, the proposed study differs from the previous studies in that in addition to evaluating six oversampling techniques (SMOTE, ADASYN, MTDF, ICOTE, MWMOTE and TRkNN), we also compare the performance of four rules-generation algorithms (Exh, Gen, Cov and LEM2). The proposed study is also focused on considering the following research questions (RQ):

- RQ1: What is the list of attributes that is highly symptomatic in the targeted data set for prediction of customer churn?
- RQ2: Which of the oversampling technique (e.g. SMOTE, MTDF, ADASYN, ICOTE, MWMOTE and TRkNN) is more suitable for creating synthetically samples that not only handle the CIP in a dataset of the telecommunication sector but also improves the classification performance.
- RQ3: Which of the rule-generation algorithm (Exh, Gen, Cov & LEM2) is more suitable using RST based classification for customer churn prediction in the imbalanced dataset.

Remaining paper is organized as follows: the next section presents the existing work of class imbalance and approaches to handle the CIP. The background study and evaluation measures are explored in section III. The experiments are detailed in section IV. The section V explains the results followed by section VI that concludes the paper.

II. HANDLING THE CLASS-IMBALANCE PROBLEM

A. CLASS IMBALANCE PROBLEM (CIP)

In this section, firstly it is explained that the CIP in the context of classification followed by techniques used in handling the CIP in a dataset and its relationship to potential domains. This section also contains a brief literature review on class imbalance/skewed distribution of samples in datasets.

Prior to the overview of handling CIP, first, there is a need to address the notion of classification. The aim of classification is to train the classifier on some dataset, making it capable to correctly classify the unknown classes of unseen objects [4], [17]. If the samples in the dataset are not balanced, there is a great chance that the classification task will result in misleading results.

CIP exists in many real-world classifications including Social Network Services [18]–[22], Banks & Financial Services [16], [23]–[26], Credit Card Account Services [27], [28], Online Gaming Services [29], [30], Human Resource Management [31]–[33], Discussion & Answer forums [34], Fault Prediction & Diagnosis [11], [35], User's profile personalization [36], Wireless Networks [37], [38], 5G future network [39] and Insurance & Subscription Services [40]–[42]. Considering the scenario of class imbalance in any application domain, almost all the objects belong to specific class (majority class) and far less number of objects are assigned to other class (minority class) [26]. The classification problems observation shows that training the classifier using conventional classification techniques results on higher performance, but it tends to classify all the samples data into the majority class; usually, which is often not the desired goal of the classification study [26]. In contrast, research studies [43], [44] show that latest machine learning techniques result in low performance due to dealing with large imbalanced datasets. Also, the class imbalance may cause classification approaches to pass from difficulties in learning, which eventually results in poor classification performance [15]. Therefore, learning from imbalanced class data has received a tremendous amount of attention from the machine learning and data mining research community [45]. The following figures illustrate difficulties in imbalanced datasets such as figure 1 (a) describes the overlapping problem & small disjoints in figure 1(b).

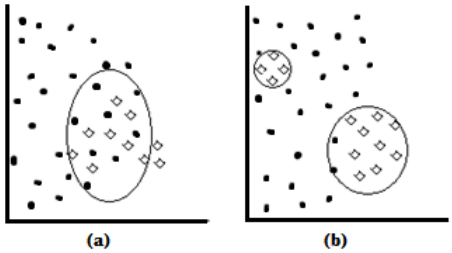

FIGURE 1. When overlapping occurs then hard to induce the discriminative rules [see figure 1 (a)] while the figure 1 (b) reflects the existence of small disjoints that increases the complexity due to unbalanced instances.

1) APPROACHES FOR HANDLING THE CIP

Handling the CIP is one of the most important approaches for dealing with rarity in sampling. The basic idea of handling CIP is to reduce the majority class or increase the minority class samples by altering the samples in the dataset distribution. Following are the three major approaches used for handling CIP:

- **The basic sampling method for under/over-sampling:** The basic sampling method for under/over-sampling is usually referred to a manual approach which is the simplest form to deal with the undersampling or oversampling in datasets [2]. In this method, some of the samples from the majority class are either eliminated or the samples of minority class are duplicated to balance the data distribution in the dataset. However, this method for undersampling has a drawback [46], i.e., it discards potentially important samples (lack of data) in majority class and thus can receive low classifier's performance. On the other hand, oversampling replicates data in the dataset. This does not degrade the classifier's performance but can usually take more time to train a classifier.
- **The advanced sampling method for under/over-sampling:** The advanced sampling methods may involve some data mining or statistical approach to cut the samples or combine the undersampling and oversampling techniques [2]. There are many intelligence techniques which are used for handling CIP such as SMOTE [6], ADASYN [9], MTDF [10], etc.
- **Random undersampling/oversampling technique:** In this technique, randomly removed samples from the majority class are combined with the minority class samples. While in random oversampling technique, the samples are replicated for random times and combined them with the samples population of the majority class [26]. Classification techniques usually produce a better performance when the samples of both classes are nearly equally distributed in the dataset.

Figure 2(a) depicts the examples of random ignorance of the majority class sample while figure 2(b) replicate the minority class samples.

B. METHODS FOR DEALING WITH CIP

The methods for dealing with the CIP, as discussed in the introduction section, can be classified into two groups depending on how these methods are used, for either undersampling or oversampling, are the following:

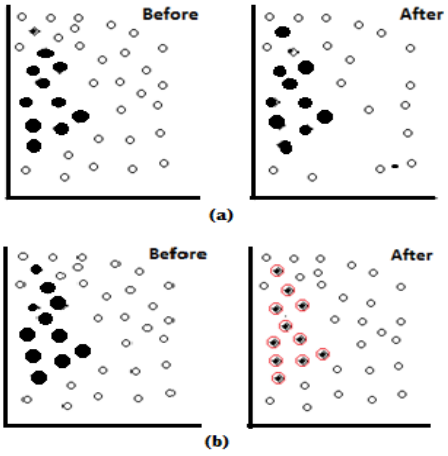

FIGURE 2. (a) Reflects the randomly removes the instances from majority class. Figure 2(b) illustrate oversampling method by replicating the instances of minority class to balance the dataset.

- **Internal (algorithm level) approaches:** In this approach, the concept is to resample the data through existing classification algorithms in order to rebalance the class distribution [4], [5]. Where the oversampling approach increases the minority class by bias data, while eliminating examples from the majority class is done through undersampling approach. The algorithm level methods are more dependent on problem & require special knowledge of targeted classifier(s) along with domain knowledge [4], [47]. This method is further divided into two groups [5]. Firstly, cost-sensitive learning, which is based on reducing the misclassification costs for minority class, and on reducing the total cost errors of both classes. Secondly, ensemble or boosting method is based on multiple sets of classifiers (ensemble technique) to handle the CIP [4].
- **External (data level) approach:** In this approach, a preprocessing step is involved in order to reduce the effect of imbalanced class distribution in the classification process. There is no need to have special knowledge of classifiers and domain. Therefore, data level approach is usually more versatile. The objective of this approach is to rebalance the skewed distribution of samples by resampling the data space [48]. Resampling method is used to rebalance the distribution of the class data [4], [5], [47], and therefore avoids the modification of the learning algorithm.

The data level approaches are usually considered to preprocess the data before training the classifier and then include into learning algorithms [4], [49]. On the other hand, cost-sensitive and internal level methods are depended on the problem and require special knowledge of targeted classifier along with the domain. Similarly, the cost-sensitive technique has a major problem of defining misclassification costs, which are not usually known/available in the data space [50]. The comparison between oversampling and undersampling has already been performed [2] with the conclusion that oversampling performs best as compared to undersampling for handling CIP. It is also reported that undersampling technique has a major problem of losing classifier's performance when some potential samples are discarded from the majority class [46]. Due to these motives, our objective is to deeply review the state-of-the-art data level methods to address binary CIP. The data level method have following advantages: (i) it is independent of the obligation to train the classifier; (ii) it is usually used in preprocessing data stage of other methods (i.e. ensemble-based approach, cost-sensitive level); and (iii) it can be easily incorporated into other methods (i.e. internal methods) [51].

C. TECHNIQUES FOR HANDLING CIP

Kubat *et al.* [52] have addressed the CIP by applying undersampling technique for the majority class and kept the original instances of the minority class. They applied geometric mean (related to ROC Curve) for performance evaluation of classifiers. Burez and Van den Poel [2] reported that random undersampling can improve the prediction accuracy as compared to the boosting techniques but did not help them in their experiments. However, by randomly oversampling or resampling the minority class may result in over-fitting. Chawla *et al.* [6] introduced a novel SMOTE considered as widespread technique for oversampling. SMOTE produces new minority observations based on weighted mean/average of the k-nearest neighbor giving positive observations. It reduces the samples inconsistency and creates a correlation between objects of the minority class. The SMOTE oversampling technique is experimentally evaluated on a variety of datasets with various levels of imbalance and different sizes of data. SMOTE with C4.5 and Ripper algorithms, outperformed as compared to Ripper's Loss Ratio and Naïve Bayes [6], [45]. Verbeke *et al.* [15] illustrated an oversampling technique by simply copying the minority class data and adding it to the training set. They reported that just oversampling the minority class by same data (i.e. copied samples) did not show significant improvement in performance of the classifier. Therefore, they have suggested using more appropriate oversampling methods (i.e. SMOTE). On the other hand, Jo and Japkowicz [3] showed that the decision tree C4.5 and Backpropagation Neural Network (BNN) algorithms both degrade the performance of classifiers on small and complex dataset due to class imbalance. They have proposed to use the cluster-based oversampling for a small and complex dataset with class imbalance dataset. Ling and Li [53] have combined oversampling with undersampling of minority and majority classes respectively. They performed different experiments such as undersampling the data in majority class followed by oversampling the data in minority class and finally combining the oversampled with under-sampled data. The conclusive results did not show significant improvement. Tang *et al.* [54], used support vector machine (SVM) and granular computing for handling CIP through undersampling technique. They have removed the

noisy data (e.g., redundant data or irrelevant data) from the dataset while keeping only those samples that have maximum relevant information. They investigated that undersampling can significantly increase the classification performance, but it was also observed that random undersampling might not provide highly accurate classification. On the other hand, Wu and Chang [55] performed an experiment showing results weak performance of SVM on the dataset that suffered from CIP. Foster Probost [56] empirically observed that during classification process with imbalanced data the number of instances of minority class are usually very less. Therefore, the trained classifiers can accurately recognize the objects of majority class instead of minority class. The reason is that the minority class cannot contribute more as compared to majority class. Due to this reason, the misclassification of instances that belong to minority class cannot be reduced in CIP. Batista *et al.* [48] introduced a comparative investigation of various sampling schema (i.e., Edited Nearest Neighbor rule or ENN and SMOTE) to balance the training-set. They removed the redundant or irrelevant data from the training process, which improved the mean number of induced rules and increased the performance of SMOTE+ENN. In connection to this work, another sampling technique was proposed by Guo and Viktor [57] for handling CIP. He modified the existing procedure of DataBoost which performed much better than SMOTEBoost [49].

He *et al.* [9] introduced ADASYN oversampling algorithm which was an extension of the SMOTE algorithm. They reported that ADASYN algorithm can self-decide the number of artificial data samples that are required to be produced for minority class. They also investigated that ADASYN not only provided a balanced data distribution but also forced the learning algorithm to focus on complex samples in the dataset. On the other hand, SMOTE algorithm [6], generated alike numbers of artificial data for minority class while DataBoost-IM [57] algorithm has generated various weightage for changed minority class samples to compensate for the distribution of skewed data. However, in their study, they have shown that ADASYN has produced more efficient results than SMOTE and DataBoost-IM.

III. BACKGROUND: OVERSAMPLING TECHNIQUES AND EVALUATION METRICS

This section presents a study of six well-known oversampling techniques (i.e., MTDF, SMOTE, ADASYN, MWMOTE, TRkNN and ICOTE), feature selection algorithm (i.e., mRMR) and Rough Set Theory (RST).

A. MEGA-TREND-DIFFUSION FUNCTION (MTDF)

Li *et al.* [10] introduced the MTDF, procedure to facilitate the estimation of domain samples systematically. It generates artificial data to balance the dataset. In their work, MTDF is applied for oversampling in order to address the CIP for customer churn prediction in the targeted domain. Due to insufficient data, small samples size or imbalance distribution of class samples provides imprecise information [10]. Therefore, MTDF apply a mutual diffusion function to diffuse the data. Let h_{set}, is diffusion coefficient set, i.e.

$$h_{set} = \hat{S}_x^2/n \tag{1}$$

$$\hat{S}_x^2 = \sum_{i=1}^{n}(X_i - \overline{X})^2/n - 1 \tag{2}$$

Where equation **(2)** is sample set variance and **n** is representing the size of sample. If the set of the sample is variant, then, i.e., $u_{set} = \frac{min+max}{2}$, The lower and upper boundaries of diffused sample are;

$$a = u_{set} - \sqrt{-2 \times \hat{S}_x^2/N_L \times ln(\varphi(a))} \tag{3}$$

$$b = u_{set} + \sqrt{-2 \times \hat{S}_x^2/N_U \times ln(\varphi(b))} \tag{4}$$

Let $Skew_L = \frac{N_L}{(N_L+N_U)}$ and $Skew_U = \frac{N_U}{(N_L+N_U)}$ are the left and right skewness magnitudes of $\sqrt{-2 \times h \times ln(\varphi(a))}$ in order to characterize the asymmetric diffusion. In propose study the diffusion function is used and accordingly revised as:

$$a = u_{set} - Skew_L \times \sqrt{-2 \times \hat{S}_x^2/N_L \times ln(\varphi(a)} \tag{5}$$

$$b = u_{set} - Skew_U \times \sqrt{-2 \times \hat{S}_x^2/N_U \times ln(\varphi(b)} \tag{6}$$

Besides these it is important to note that if the variance of any features for a target class is zero, then the value of N_U and $N_L$will be zero. Under such conditions equations (5) and (6) are not used. However, they propose that either equation (7) or (8) can be used.

$$a = \frac{min}{5} \tag{7}$$

$$b = max \times 5. \tag{8}$$

B. SMOTE

To overcome the issue of over-fitting and extend the decision area of the minority class samples, a novel technique SMOTE ''Synthetic Minority Oversampling TEchnique'' was introduced by Chawla [45], This technique produces artificial samples by using the feature space rather than data space. It is used for oversampling of minority class by creating the artificial data instead of using replacement or randomized sampling techniques. It was the first technique which introduced new samples in the learning dataset to enhance the data space and counter the scarcity in the distribution of samples [45]. The oversampling technique is a standard procedure in the classification of imbalance data (e.g., minority class) [7]. It has received incredible effort from the researcher of machine learning domain in a recent decade. The pseudo code of the SMOTE algorithm and detail can be found in [45].

C. mRMR

Peng *et al.* [58], introduced Minimal Redundancy Maximal Relevance (mRMR) technique for attributes selection as per the procedure of the maximal statistical dependency criterion.

mRMR algorithm not only minimizes the features space by increasing the mutual dissimilarity with the class but also extracts more appropriate features subset. The set S in term of relevance of attributes for class c is expressed by the average value of MI between each attribute f_i and c as shown in equation (9):

$$D(S, c) = \frac{1}{|s|} \sum_{fi \in S} I(fi; c) \tag{9}$$

The redundancy of all attributes in the set=S is the mean value of mutual information between the attributes (features) f_i and f_j.

$$R(S) = \frac{1}{|S|^2} \sum_{fi \in S} I(f_i : f_j) \tag{10}$$

The mRMR criterion is a group of two measures given in equations (9) and (10), which can be defined as follows (i.e., equation (11)):

$$mRMR = \max_{S} \left[\frac{1}{|S|} \sum_{fi \in S} I(fi, c) - \frac{1}{|s|^2} \sum_{fi, fj \in S} I(fi, fj) \right] \tag{11}$$

The equation above may then be expressed as an optimization problem:

$$mMRM = \max_{x \in \{0,1\}^n} \left[\frac{\sum_{i=1}^{n} c_i x_i}{\sum_{i-1}^{n} x_i} - \frac{\sum_{j=1}^{n} a_{ij} x_i x_j}{(\sum_{i=1}^{n} x_i)^2} \right]. \tag{12}$$

D. MWMOTE

Barua *et al.* [59] have presented a new method for efficiently handling the CIP, called MWMOTE. MWMOTE first identifies the difficult to learn important minority class samples and assigns them weightage based on Euclidean distance from the nearest larger class samples. For in-depth study, read the original work of Barua *et al.* [59].

E. TRkNN

TRkNN originally proposed by Tsai and Yu [60] in 2016 to overcome the CIP and improve the accuracy rate in predicting the samples of majority and minority classes. TRkNN algorithm also solves the issue of noisy and borderline sample in CIP. The advantage of TRkNN is to avoid the production of unnecessary minority examples. For in-depth study, read the original work of Tsai and Yu [60].

F. ICOTE

Ai *et al.* [61] introduced another oversampling technique "ICOTE" in 2015 to improve the performance of classification in CIP. ICOTE is based on an immune network and it produces a set of immune centroids to broaden the decision space of the minority class. In this algorithm, the immune network is used to produce artificial samples on clusters with the high data densities and these immune centroids are considered synthetic examples in order to resolve the CIP. For in-depth study, read the original work of Ai *et al.* [61].

G. ROUGH SET THEORY (RST)

RST [62] was initially proposed by Pawlak in 1982, which is used as a mathematical tool in order to address ambiguity. RST philosophy is centered on the assumption that there is information (knowledge, data) associated with all instances in the universe of discourse. RST has a precise idea of rough set approximation (i.e., *LB*=lower bound and *UB*=upper bound), and the boundary region (BR). The *BR* separates the *LB* from *UB* (i.e. boundary-line). For example, those samples that may not be classified with certainty are members of either the *LB* or *UB*. It is difficult to characterize the borderline samples due to unavailability of clear knowledge about these elements. Therefore, any rough concept is replaced by either *LB* or *UB* approximation of vague concept [58]–[60]. Mathematically, the concepts of *LB*, *UB* and *BR* have been defined as; let $X \subseteq U$ and B is an equivalence relation (i.e. the partition of the universe set U to create new subset of interest from U which has the same value of outcome attribute) in information system or $IS = (U, B)$ of non-empty finite set U and B, where U is the universe of objects and B is a set which contains features. Then $LB = \bigcup Y \in U/B : Y \subseteq X$ is a *LB* approximation and an exact member of X while $UB = \bigcup Y \in U/B : Y \bigcap X \neq \phi$ is *UB* approximation that can be an element of X. $BR = UB - LB$ in the boundary region. The detail study can be found at [62]–[65].

H. RULES GENERATION

Decision rules are often denoted as "IF C then D" where D represents the decision feature and C is the set of conditional attributes in the decision table [63]. Given two unary predicate formulae are $\alpha(\chi)$ and $\beta(\chi)$, where χ executes over a finite set U. Łukasiewicz defined this in 1913 as: i.e. $\frac{card\|\alpha(\chi)\|}{card(U)}$, assign to $\alpha(\chi)$ where $\|\alpha(\chi)\| = \{\chi \epsilon U : \chi\}$satisfies α while the fractional value is assigned to implication $\alpha(x) => \beta(x)$ is then $\frac{card(|\alpha(\chi)\beta(\chi)|)}{card(|\alpha(\chi)|)}$with assumption that $||\alpha(x)|| \neq \phi$. The decision rules can easily be built by overlaying the reduct sets over the IS. Mathematically, it can be represented as: $(a_{i1} = v_1)^\wedge \ldots\ldots\ldots\ldots^\wedge (a_{ik} = v_k) => d = v_d$, where $1 \leq i_j \prec \ldots\ldots\ldots \prec i_k \leq m, v_i \in V_{ai}$; for simplicity it can be represented in IF-ELSE statement as "IF C then D" where C is set of conditions and D is decision part. To retrieve the decision rules, the following well-known rule-generation algorithms are used [65]:

- **Exhaustive Algorithm (Exh):** It takes subsets of attributes incrementally and then returns reduced set and minimal decision rules. The generated decision rules are those rules, which have minimal descriptors in the conditional attributes. It requires more focus due to the extensive computations needed in the case of large and complex Boolean reasoning method [66].
- **Genetic Algorithm (Gen):** This method depends on order-based genetic algorithm combined with a heuristic. It is applied to minimize the computational cost in complex and large IS [64], [67].

- **Covering Algorithm (Cov):** It is the modified implementation of the Learning from Example Module, version 1 (LEM1) algorithm and deployed in the Rough Set Exploration System (RSES) as a rule-generation method. It was presented by Jerzy Grzymala [68].
- **RSES LEM2 Algorithm (LEM2):** It is a divide-and-conquer based method which coupled with the RST approximation and it depends on the local covering determination of every instance from the decision attribute [68], [69].

I. EVALUATION MEASURES

It may not be possible to construct classifier that could perfectly classify all the objects of the validation set [2], [42]. To evaluate the classification performance, we calculate the count of *TP* (e.g., True Positive), *FN* (e.g., False Negative), *FP*(e.g., False Positive) and *TN* (e.g., True Negative). The *FP* value is part of *N* or negative but incorrectly classified as *P* or positive. The *FN* result actually belongs to *P*. it can be formulate as $P = FN + TP$ but when incorrectly classified the instances will then belongs to *N*. Mathematically it can be expressed as: $N = FP + TN$. The following evaluations measures are used for performance validation of the proposed approach.

- **Regular Accuracy (RA):** It is a measure that calculates the classifier's overall accuracy. It is formulated as:

$$RA = \frac{TN + TP}{N + P} \tag{13}$$

- **Sensitivity (Recall):** It is the proportion of those cases which are correctly classified as true positive, and calculated as:

$$Recall = \frac{TP}{P} \tag{14}$$

- **Precision:** It is fraction of the predicted positive instances that characterized as correctly churned. Formally, it can be expressed as;

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

- **F-Measure**: It is based on the harmonic mean between both the precision and recall. The high F-measure value represents that both precision and recall are reasonably high. It can also be considered as the weighted-average of recall and precision.

$$F.Measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{16}$$

- **Coverage**: The ratio of classified objects that are recognized by a classifier from the class to the total number of instances in the class. Where C is a classifier, A is a decision table, Match A (C) is a subset of objects in A that is classified by classifier C.

$$Coverage\ A(C) = \frac{|Match\ A(C)|}{||A||} \tag{17}$$

- **Mutual Information (MI):** It measures the information regarding how much the attribute's value pays to creating the right decision in classification. It is computing the MI between the predicted churn label (i.e., $\hat{y}$) and actual churn (i.e., y). The MI between predicted and actual churn label is as following:

$$I\left(\hat{Y}, Y\right) = \sum_{\hat{y}=0}^{1} \sum_{y=0}^{1} p\left(\hat{y}\right)(y)\, log \frac{p(\hat{y}, y)}{p(\hat{y})p(y)} \tag{18}$$

- **Imbalanced Ratio (IR):** The imbalance ratio can be expressed as the fraction of a number of samples in the majority class to the number of samples in the minority class [70]. It can be formulated as;

$$IR = \frac{Number\ of\ negative\ class\ instances}{Number\ of\ positive\ class\ instances} \tag{19}$$

- **Area Under Curve (AUC):** It is the fraction of total area that lies under the ROC graph. The ROC is used to compute an overall measure of quality while AUC provides a single value for performance evaluation of classifier. It can also be used as evaluation measure for the imbalanced datasets. The AUC measure is computed as [5], [51]:

$$AUC = \frac{TP_{rate} + TN_{rate}}{2} \tag{20}$$

- **Balanced Accuracy (CBA):** The use of Balanced Accuracy is not widespread in the class imbalance literature, likely because of the aforementioned shortcoming, This study would use balanced accuracy for 2×2 confusion table. i,j ∈ G, where G denotes the set of all possible class labels.

$$CBA = \frac{\sum_{i}^{k} \frac{c_{ij}}{max(c_i, c_j)}}{k}. \tag{21}$$

IV. EVALUATION SETUP

In this section, an experimental environment is established for the propose approach. mRMR algorithm is used for selecting more suitable attributes subset. six well-known oversampling techniques—SMOTE, ADASYN, MTDF, ICOTE, MWMOTE and TRkNN—have been used to address the CIP using publicly available datasets. Finally, a more appropriate approach for customer churn prediction is proposed. Classification tasks, and trade-off among the four rules-generation algorithms, are performed using RST.

A. DATASET

In the proposed study, we have used publicly available dataset related to telecom-sector. These datasets can be obtained from URLs (i.e.,given in table 1). Figures 3 present the original distribution of class samples & sample sizes in datasets. The vertical bar represent the instances of non-churn class and horizontal bars used for the instances of churn class in four datasets. It is clear from the figure 3 that CIP exists in the available dataset.

Table 1 illustrates the description of the imbalanced datasets used in this empirical study. For each dataset,

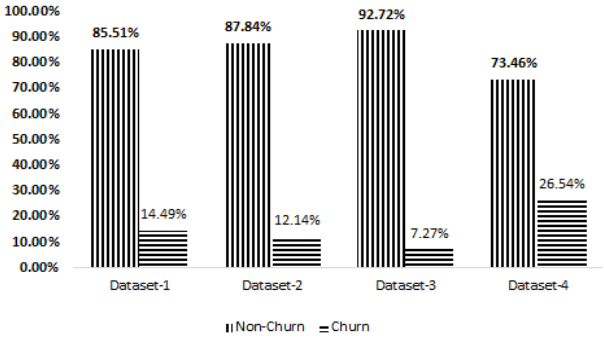

FIGURE 3. Original class sample distribution, where the vertically striped-bars represent the samples of the Non-Churn (NC) class while horizontally striped-bars represent the samples of the churn (C) class.

TABLE 1. Summary description of imbalanced datasets.

Source	#Ins	#Att	Class {Pos, Neg}	{%Pos, %Neg}	IR
Dataset 1 [71]	3333	21	{Churn, Non-Churn}	14.49%, 85.51%	5.94
Dataset 2 [72]	5782	232	{-1, 1}	87.84%, 12.14%	7.24
Dataset 3 [73]	38162	251	{3G, 2G}	92.72%, 7.27%	12.75
Dataset 4 [74]	7043	20	{Yes, No}	73.46%, 26.54%	2.77

dataset source, the number of instances (#Ins), the number of attributes (#Att), class name of each class (pos and neg where "pos" represent the minority class while "neg" represent the majority class), the percentage of instances in each class (%Pos, %Neg) and imbalance ratio (IR) is shown.

Table 1, also shows the obtained IRs for different datasets: from highly imbalanced ratio value to low imbalanced dataset. The IR 2.77 is the lowest and 12.75 is the highest value, which shows that dataset 4 is low imbalanced dataset while dataset 3 is a highly imbalanced dataset.

B. FEATURES/ATTRIBUTES SELECTION & DATASET PREPARATION

For attributes selection, we used mRMR method [58]. The following attributes were selected for preparation of final datasets as shown in Table 2, also reports to RQ1.

To focus on the CIP using available datasets, SMOTE, MTDF, ADASYN, MWMOTE, ICOTE and TRkNN methods are used to generate artificial samples in minority class. SMOTE algorithm is used using the Weka toolkit [75] and MTDF is used manually while ADASYN technique [9], TRkNN [60], ICOTE [61]and MWMOTE [59] are applied using MATLAB Code. Table 3 reflects the ranges of each feature based on obtained values for "a" and "b" after applying MTDF. The artificial samples are produced for "churn" based on ranges of "a" and "b" while the churn and non-churn class size was equal to each other. It was investigated that MTDF method produces values between 0 and 5 range for both features "VMail_Plan" and "Intl_Plan", but it is observed that the required values for these features were in the range between 0 and 1. Therefore, normalization process was applied by using a standardized function to obtained the required values [76].

Once the range of individual attributes (i.e., a and b) is defined by using MTDF, the next step is to generate the artificial samples using the above mentioned oversampling technique. Then prepare the decision tables to apply the base-classifier (i.e., rough set theory). Tables 4, 5, 6 and 7 represent the prepared decision tables for dataset 1, 2, 3 and 4 respectively.

These tables contain objects, attributes with condition that needs to be fulfilled and the decision attributes. For each experiment (i.e. original population extended by MTDF, SMOTE, ADASYN, MWMOTE, ICOTE and TRkNN oversampling techniques) the same structure of decision tables with four different datasets were used.

The cut and discretization process is an important approach to reduce the dataset horizontally in order to handle the large data efficiently. It is a common approach used in the RST where the attributes that contain continuous values are split

TABLE 2. Feature subset extracted through mRMR.

Features	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
Sequence	Features List	Score	Features List	Score	Features List	Score	Features List	Score
F1	*Intl_Plan*	0.041	*Var_ar_flag*	0.053	*Var126*	0.016	*Contract*	0.094
F2	*Day_Charges*	0.040	*Avg_call_intran*	0.035	*Var72*	0.009	*PaymentMethod*	0.018
F3	*Day_Mins*	0.039	*Tot_usage_days*	0.034	*Var7*	0.008	*Tenure*	0.009
F4	*CustServ_calls*	0.013	*Avg_usage_days*	0.034	*Var189*	0.007	*DeviceProtection*	0.005
F5	*Intl_Charges*	0.010	*Avg_call*	0.033	*Var65*	0.006	*Dependents*	0.002
F6	*Intl_Mins*	0.010	*Highend_prog_flag*	0.033	*Var113*	0.005	*Gender*	-0.001
F7	*Eve_Charges*	0.007	*Avg_call_local*	0.032	*Var133*	0.005	*PaperlessBilling*	-0.002
F8	*Eve_Mins*	0.007	*Avg_call_ob*	0.031	*Var16*	0.003	*SeniorCitizen*	-0.005
F9	*Account_Length*	0.006	*Std_vas_arc*	0.030	*Var153*	0.002	*Partner*	-0.021
F10	*VMail_Plan*	0.003	*Avg_mins*	0.029	*Var73*	0.002	*PhoneService*	-0.048
F11	*Class label attribute*		*Class label attribute*		*Class label attribute*		*Class label attribute*	

TABLE 3. The a and b values of MTDF function for each attribute.

Features	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
Sequence	MTDF Ranges		MTDF Ranges		MTDF Ranges		MTDF Ranges	
	a	b	a	b	a	b	a	b
F1	28.4636	44.6562	0.0000	1.0000	-30	56.88	0.873547	6.655153
F2	0.0000	1.0000	380.069	393.496	3.45	15.58	2.412609	8.02709
F3	173.517	210.653	91.9169	127.634	10.43	20.17	0.361109	4.892115
F4	0.0000	1.0000	15.2519	29.8818	308.83	426.43	0.881878	6.282582
F5	111.32	138.139	2409.06	2443.80	22.93	36.07	35.89648	63.77696
F6	207.681	239.084	0.0000	1.0000	3552064	3263292	0.348405	5.293856
F7	2.72140	9.83579	2403.23	2437.72	5444049	5451211	0.350999	5.211826
F8	2.0746	5.80143	1292.73	1318.407	37.58	576.24	0.388244	4.034051
F9	9.2768	16.4491	98.0268	103.876	6155159	6168114	0.348485	5.291311
F10	15.665	24.8207	7646.24	7687.29	105.22	136.59	0.410333	3.335514

TABLE 4. Attributes for the decision table using dataset 1.

Sets	Desciption
Number of Objects	{5700 distinct objects}
Conditional Attributes	{*Intl_Plan, Day_charges, Day_mins, Custserv_calls, Intl_charges, Eve_mins, Account_length, Vmail_plan*}
Decision Attribute	{Churn}

TABLE 5. Attributes for the decision table using dataset 2.

Sets	Desciption
Number of Objects	{38162 distinct objects}
Conditional Attributes	{*Var ar flag, Avg call intran, Tot usage days, Avg usage days, Avg call, Highend program flag, Avg call local, Avg call ob, Std vas,arc, Avg mins*}
Decision Attribute	{Churn}

TABLE 6. Attributes for the decision table using dataset 3.

Sets	Desciption
Number of Objects	{38162 distinct objects}
Conditional Attributes	{*Var126, Var72, Var7, Var189, Var65, Var113, Var133, Var16, Var153, Var73*}
Decision Attribute	{Churn}

TABLE 7. Attributes for the decision table using dataset 4.

Sets	Desciption
Number of Objects	{7043 distinct objects}
Conditional Attributes	{*Contacts, PaymentMethod, Dependents, DeviceProtection, Tenure, Gender, PaperlessBilling, SeniorCitizen, Partner, PhoneService*}
Decision Attribute	{Churn}

TABLE 8. Cuts distribution of attribute Day_Min.

Group#	Intervals	Count	Percentage
1	{263.55, 281.05}	115	3.45%
2	{151.05, 163.45}	295	8.85%
3	{237.85, 251.85}	158	4.74%
4	{281, **}	103	3.09%
5	{163.45, 178.15}	339	10.17%
6	{217.65, 178.15}	337	10.11%
7	{178.15, 189.25}	235	7.05%
8	{251.85, 263.55}	95	2.85%
9	{108.8, 151.05}	687	20.61%
10	{195.45, 208.85}	302	9.06%
11	{189.25, 195.45}	168	5.04%
12	{*, 78.65}	109	3.27%
13	{78.65, 108.8}	202	6.06%
14	{208.85, 217.65}	188	5.64%

**=Maximum number, *=Minimum number

into a finite number of intervals [47]. The cut and discretization processes were carefully accomplished on the prepared decision table. These cuts on the decision table were made at every iteration to minimize the number of cuts to the data. This was done in light of the recommendation in study [65]. For example, the cuts of attribute ''Day_Min'' in dataset 1 were grouped after discretization process as listed in Table 8. The first field denotes the groups that are represented by numeric numbers for simplicity purpose and listed in ascending order. The second column represents the intervals that are obtained after the discretization process. The third column is the count of the attribute's values that fall into certain groups while the last column is the percentage of variable's value in each interval. It is clear from table 8 that the range of Day_Min has been changed from the continuous nature into 14 different intervals or groups after cut and discretization process.

C. TRAINING AND VALIDATION SETS

In data mining, training and validation is an extremely important step. Once the training process is finished, then the

validation step is to be performed to confirm the performance of the predictive models on known and unknown objects [77]. To perform the validation process, the following procedure is applied: (i) some of the data is excluded from the training set that is used for the learning process of the classifier, (ii) when the training process is finished, and the classifier is to be considered as trained, the excluded samples can be included to validate the results of the trained classifier on unseen data samples. This procedure is known as the cross-validation process. The K-fold cross-validation is applied in the proposed study to avoid bias during the validation of methods.

V. RESULTS AND DISCUSSION

In this section, the four rules-generation algorithms using RST on dataset 1, 2, 3, and 4 that were expanded using MTDF, SMOTE, ADASYN, ICOTE, MWMOTE and TRkNN to handle the CIP were considered. Tables 9, 10, 11 and 12 reflect the performance of classifiers through evaluation measures. The following tables 9, 10, 11 and 12 clearly shows the performance of various oversampling techniques (i.e., MTDF, SMOTE, ADASYN, MWMOTE, ICOTE and TRkNN) on publically available telecommunication's dataset by applying classification process based on rough set theory and four rules generation algorithms (i.e., Gen, Exh, Cov and LEM2). It can be observed (i.e., as given tables 9-12) that the Cov and LEM2 rules generation algorithms have achieved maximum performances as compared to Gen and Exh algorithms but these two (i.e. Cov & LEM2) have not been covered all the instances available in the dataset. Therefore, all the underlined values corresponding to targeted oversampling techniques are ignored from further analysis in this study while the bold values reflect the best performed techniques in our empirical environment.

To report RQ2 and RQ3, the results reflect in Tables 9, 10, 11, and 12 were thoroughly analyzed to cover each relevant detail. It is observed that both algorithms (e.g., Cov & LEM2) shown higher accuracy, but do not deliver full coverage (e.g., the underlined values does not seem to follow from full coverage). The coverage of a learning algorithm is the number of samples or instances that can be learned by that algorithm from samples of a given size for given accuracy [78].

On the other hand, both algorithms (i.e. Exh and Gen) have covered all instances in given datasets (i.e. Dataset 1, 2, 3 & 4). It can be observed in Figure 4 that both the Cov and LEM2 algorithms achieved higher accuracy as compared to Exh and Gen algorithms but these results seem not to follow the full objects coverage. So these results (i.e. not fully covering the dataset objects) were ignored and not shown in Figures 5 (a), (b), (c) and (d) for datasets 1, 2, 3 and 4 respectively. Figure 5 describes the performance of oversampling algorithms (e.g., MTDF, SMOTE, ADASYN, ICOTE,

TABLE 9. Performance evaluation of classifier for dataset 1.

Techniques & Algorithm		Accuracy	Recall	Precision	F-Measure	Coverage	MI
MTDF	Exh_M	0.935	0.960	0.911	0.933	1.0	0.66
	Gen_M	0.936	**0.964**	0.913	0.934	1.0	0.67
	Cov_M	0.914	0.919	0.913	0.912	0.528	0.58
	LEM2_M	0.964	0.979	0.944	0.967	0.799	0.78
SMOTE	Exh_M	0.924	0.930	0.917	0.923	1.0	0.61
	Gen_M	0.925	0.933	0.916	0.924	1.0	0.62
	Cov_M	0.953	0.969	0.973	0.864	0.226	0.41
	LEM2_M	0.941	0.944	0.944	0.938	0.752	0.68
ADASYN	Exh_M	0.925	0.931	0.920	0.925	1.0	0.62
	Gen_M	0.924	0.933	0.917	0.924	1.0	0.62
	Cov_M	0.944	0.919	0.913	0.912	0.216	0.08
	LEM2_M	0.950	0.951	0.953	0.947	0.75	0.71
MWMOTE	Exh_M	0.952	0.942	0.960	0.953	1.0	0.725
	Gen_M	**0.969**	0.961	**0.976**	**0.969**	1.0	**0.803**
	Cov_M	0.992	0.962	0.962	0.996	0.41	0.40
	LEM2_M	0.985	0.96	0.888	0.992	0.42	0.35
ICOTE	Exh_M	0.926	0.951	0.906	0.924	1.0	0.63
	Gen_M	0.927	0.958	0.904	0.926	1.0	0.64
	Cov_M	0.927	0.917	0.905	0.938	0.629	0.60
	LEM2_M	0.959	0.970	0.940	0.962	0.80	0.75
TRkNN	Exh_M	0.88	0.866	0.892	0.882	1.0	0.47
	Gen_M	0.89	0.879	0.898	0.891	1.0	0.50
	Cov_M	0.861	0.799	0.822	0.892	0.347	0.38
	LEM2_M	0.874	0.843	0.857	0.893	0.638	0.44

TABLE 10. Performance evaluation of classifier for dataset 2.

Techniques & Algorithm		Accuracy	Recall	Precision	F-Measure	Coverage	MI
MTDF	Exh_M	0.940	**0.955**	0.927	0.940	1.0	0.68
	Gen_M	**0.946**	0.930	**0.964**	**0.945**	1.0	0.70
	Cov_M	0.789	0.859	0.795	0.732	0.38	0.237
	LEM2_M	0.834	0.861	0.872	0.780	0.579	0.317
SMOTE	Exh_M	0.847	0.851	0.880	0.824	1.0	0.376
	Gen_M	0.849	0.841	0.812	0.867	1.0	0.379
	Cov_M	0.999	1.0	0.666	0.999	0.199	0.0099
	LEM2_M	0.898	0.885	0.873	0.912	0.67	0.5096
ADASYN	Exh_M	0.85	0.85	0.88	0.82	1.0	0.38
	Gen_M	0.854	0.867	0.85	0.84	1.0	0.38
	Cov_M	0.98	0.66	0.97	0.99	0.19	0.11
	LEM2_M	0.90	0.89	0.87	0.91	0.67	0.52
MWMOTE	Exh_M	0.743	0.756	0.741	0.739	1.0	0.18
	Gen_M	0.746	0.756	0.741	0.743	1.0	0.18
	Cov_M	0.689	0.66	0.589	0.736	0.25	0.10
	LEM2_M	0.780	0.821	0.814	0.722	0.34	0.22
ICOTE	Exh_M	0.860	0.839	0.876	0.863	1.0	0.418
	Gen_M	0.862	0.847	0.873	0.864	1.0	0.422
	Cov_M	0.981	0.979	0.894	0.988	0.547	0.48
	LEM2_M	0.973	1.00	0.939	0.975	0.843	0.027
TRkNN	Exh_M	0.802	0.80	0.803	0.802	1.0	0.28
	Gen_M	0.812	0.794	0.823	0.815	1.0	0.30
	Cov_M	0.770	0.736	0.68	0.786	0.362	0.22
	LEM2_M	0.831	0.813	0.813	0.843	0.658	0.34

MWMOTE and TRkNN) on the datasets that were prepared and balanced by using the same algorithms. It is observed from the results that Gen and Exh algorithms have performed better as compared to the rest of the other two. Although, Gen and Exh algorithms have achieved alike performance; however, the Gen algorithm has acquired more accuracies 0.929, 0.862, 0.982 and 0.813 with 100% objects coverage on dataset 1, 2, 3, and 4 respectively.

The following acronyms have been used in Figures 4- 6:

Gen_M: Genertic Algorithm & MTDF (dataset oversampled with MTDF).
Exh_S: Exhaustive Algorithm & SMOTE (dataset oversampled with SMOTE).
Gen_S: Genertic Algorithm & SMOTE (dataset oversampled with SMOTE).
Exh_A: Exhaustive Algorithm & ADASYN (dataset oversampled with ADASYN).
Gen_A: Genetic Algorithm & ADASYN (dataset oversampled with ADASYN).
Exh_M: Exhaustive Algorithm & MTDF (dataset oversampled with MTDF)
Gen_I: Genetic Algorithm & ICOTE (dataset oversampled with ICOTE)
Exh_I: Exhaustive Algorithm & ICOTE (dataset oversampled with ICOTE)
Gen_W: Genetic Algorithms & MWMOTE (dataset oversampled with MWMOTE)
Exh_W: Exhaustive Algorithm & MWMOTE (dataset oversampled with MWMOTE)
Gen_T: Genetic Algorithms & TRkNN (dataset oversampled with TRkNN)
Exh_T: Exhaustive Algorithm & TRkNN (dataset oversampled with TRkNN)

The F-Measure and MI can give the best result. The MI value of Gen_M is larger than other targeted algorithms. It describes the correct ranking through MI measure between the projected and positive churn in all datasets (e.g., datasets 1, 2, 3 & 4).

A. COMPARISON BETWEEN BINARY CLASS IMBALANCE ACCURACY AND CLASS BALANCE ACCURACY (CBA)

The balanced accuracy, which accounts for the recall over each class, has the weakest ability to discriminate between confusion matrix inputs. It is the ability of CBA measure to account for rows and column sum simultaneously that allows it to discriminate between all other measures (i.e. Regular Accuracy, F-Measure, Mutual Information, Precision, and Recall). CBA's ability to detect changes in the false negative and false positive counts separates it from overall regular accuracy. Therefore, table 13 shows the overall regular accuracy followed by the AUC results (+ for standard deviation). The comparison of the CBA and regular accuracy for the binary class imbalance is shown in Table 13, where each table represent the results of each dataset

TABLE 11. Performance evaluation of classifier for dataset 3.

Techniques & Algorithm		Accuracy	Recall	Precision	F-Measure	Coverage	MI
MTDF	Exh_M	0.970	**0.97**	0.98	0.97	1.0	0.85
	Gen_M	**0.980**	0.96	**0.99**	**0.981**	1.0	0.88
	Cov_M	0.990	0.99	0.98	0.980	0.70	0.85
	LEM2_M	0.980	0.96	0.99	0.981	0.97	0.87
SMOTE	Exh_M	0.920	0.91	0.92	0.92	1.0	0.61
	Gen_M	0.921	0.91	0.93	0.92	1.0	0.60
	Cov_M	0.920	0.91	0.93	0.921	0.46	0.59
	LEM2_M	0.950	0.96	0.93	0.96	0.76	0.73
ADASYN	Exh_M	0.928	0.92	0.93	0.92	1.0	0.62
	Gen_M	0.930	0.917	0.94	0.93	1.0	0.63
	Cov_M	0.916	0.909	0.931	0.91	0.45	0.58
	LEM2_M	0.960	0.97	0.94	0.97	0.76	0.77
MWMOTE	Exh_M	0.745	0.756	0.741	0.742	1.0	0.182
	Gen_M	0.746	0.756	0.741	0.743	1.0	0.18
	Cov_M	0.692	0.66	0.589	0.736	0.387	0.096
	LEM2_M	0.779	0.821	0.814	0.722	0.682	0.217
ICOTE	Exh_M	0.907	0.926	0.892	0.905	1.0	0.555
	Gen_M	0.915	0.94	0.895	0.912	1.0	0.584
	Cov_M	0.895	0.914	0.803	0.918	0.70	0.480
	LEM2_M	0.936	0.970	0.886	0.943	0.816	0.663
TRkNN	Exh_M	0.692	0.712	0.684	0.685	1.0	0.109
	Gen_M	0.706	0.722	0.699	0.701	1.0	0.126
	Cov_M	0.671	0.665	0.658	0.692	0.422	0.092
	LEM2_M	0.710	0.681	0.625	0.748	0.555	0.119

(i.e. datasets 1, 2, 3 & 4) respectively. The bold values in each table represent the best-performing method. it can be observed in the tables 13 that both Gen_M (genetic algorithm applied on oversampled dataset using MTDF) and Exh_M (exhaustive algorithm applied on oversampled dataset using MTDF) performed better as compared to the rest of the techniques. It is also investigated that out of these two techniques (i.e. Gen_M and Exh_M) the Gen_M have achieved overall much better results.

The AUC can be used as evaluation measure to evaluate the performance of classifiers. It is observed that the best samples distribution for training set tends to be very balanced class samples distribution [48]. The CIP significantly decreases the performance of imbalanced datasets. It might be projected that the AUC decreases for the very imbalanced dataset. The results obtained for AUC from experiments are shown in figure 6. As stated earlier, these results were calculated through 10-fold cross validation and AUCs were obtained from the over-sampled dataset by applying target algorithms. However, here only two algorithms (i.e. the Genetic and Exhaustive) have been shown because these two algorithms provided full coverage during classification of the validation set instances (e.g. shown in figure 4).

B. SIMULATION RESULTS

Tables 9, 10, 11, 12 and 13 summarize the results of SMOTE, ADASYN, MTDF, MWMOTE, ICOTE and TRkNN algorithms on four publically available telecommunication related datasets. Each result is the average of the 10 independent executions of classifiers and selected the best result of each targeted algorithms for this study. In tables 9—12 precision, recall also included as evaluation measures of different techniques but these evaluations are also used as input for other evaluation measures (i.e., F-measure). The bold-face values in the recall column of each technique in all datasets (e.g., Tables 9—12) have misclassify many majority class samples as minority. Hence, in spite of the higher recall, these targeted techniques reduce the F-measure and AUC because the erroneous generation of the synthetic minority class samples falls inside the majority class region. The probability of wrongly classified minority class samples thus providing improve precision and decrease recall. Unlike recall and precision, the overall simulation results are based on F-measure, AUC and CBA. The AUC values are shown in figure 6 which illustrate a comparison of selected algorithms. It is observed that the performances of SMOTE, ADASYN and ICOTE algorithms are closed to each other and these algorithms have achieved a performance higher than TRkNN algorithm and lower than the performance of MTDF. It is also investigated that the performance of SMOTE, ADASYN and ICOTE have much better results than MWMOTE where the dataset have less categorical values attributes but low performance obtained where the data have too many categorical values in attributes as compare to MWMOTE. It is also investigated that MWMOTE can

TABLE 12. Performance evaluation of classifier for dataset 4.

Techniques & Algorithm		Accuracy	Recall	Precision	F-Measure	Coverage	MI
MTDF	Exh_M	0.812	0.73	0.871	0.825	1.0	0.314
	Gen_M	**0.848**	0.794	**0.881**	**0.858**	1.0	0.389
	Cov_M	0.999	0.5	0.5	0.999	0.213	6.904
	LEM2_M	0.857	0.830	0.830	0.871	0.649	0.402
SMOTE	Exh_M	0.808	0.729	0.866	0.822	1.0	0.308
	Gen_M	0.811	**0.808**	0.813	0.811	1.0	0.301
	Cov_M	0.825	0.796	0.904	0.797	0.329	0.335
	LEM2_M	0.854	0.772	0.860	0.881	0.681	0.379
ADASYN	Exh_M	0.772	0.697	0.873	0.769	1.0	0.251
	Gen_M	0.806	0.731	0.860	0.821	1.0	0.302
	Cov_M	0.823	0.813	0.806	0.808	0.315	0.296
	LEM2_M	0.849	0.814	0.809	0.810	0.650	0.301
MWMOTE	Exh_M	0.693	0.664	0.704	0.701	1.0	0.110
	Gen_M	0.673	0.642	0.684	0.682	1.0	0.088
	Cov_M	0.647	0.623	0.733	0.615	0.4	0.066
	LEM2_M	0.703	0.703	0.679	0.715	0.486	0.122
ICOTE	Exh_M	0.802	0.794	0.806	0.803	1.0	0.282
	Gen_M	0.793	0.788	0.795	0.794	1.0	0.264
	Cov_M	0.805	0.832	0.736	0.823	0.384	0.288
	LEM2_M	0.836	0.841	0.868	0.812	0.672	0.348
TRkNN	Exh_M	0.693	0.724	0.681	0.683	1.0	0.110
	Gen_M	0.691	0.734	0.675	0.677	1.0	0.108
	Cov_M	0.674	0.664	0.614	0.705	0.373	0.088
	LEM2_M	0.699	0.736	0.703	0.675	0.492	0.116

outperform on such dataset which has much more categorical values attributes in a dataset. on the other hand, SMOTE have not performed better (e.g., obtained AUC values 0.92, 0.46, 0.69 and 0.57 on dataset 1, 2, 3 and 4 respectively) due to over-generalization and variance because SMOTE assign the same number of synthetic samples for every genuine minority samples without considering to the samples in neighbored while over generalization is major drawback of the SMOTE algorithm [79]. Similarly, ICOTE also uses the generalization procedure and create a larger cluster during the learning process of a classification [61]. To get this over generalization issue, ADASYN algorithm, which is based on adaptive sampling method [8], [9] was applied. ADASYN uses a weighted distribution for minority class samples and reducing the bias in generated synthetic samples [79]. ADASYN algorithm also has not been sufficiently achieved higher performance (e.g., average performance based on obtained AUC values 0.92, 0.85, 0.93 and 0.80 on dataset 1, 2, 3 and 4 respectively), due to interpolation between minority class samples and their random same class neighbors for producing the artificial samples. According to Dhurjad and Banait [79] sometime both algorithms (i.e., ADASYN and SMOTE) become inappropriate and cannot provide the required amount of useful synthetic minority class samples. Finally, it is summarized from all the simulation results given in tables 9—13 based on various evaluation measures (detailed given in section III) that MTDF have better performed as a whole as compared to

TABLE 13. Comparison between regular accuracy & class balance accuracy.

Classifiers	Dataset 1	Dataset 2	Dataset 3	Dataset 4
	RA+CBA	RA+CBA	RA+ CBA	RA+CBA
Gen_I	0.929±0.978	0.862±0.923	0.915±0.959	0.802±0.885
Exh_I	0.927±0.974	0.860±0.919	0.907±0.952	0.793±0.882
Gen_S	0.925±0.965	0.849±0.921	0.923±0.965	0.811±0.890
Exh_S	0.924±0.963	0.849±0.920	0.923±0.964	0.809±0.901
Gen_A	0.927±0.964	0.852±0.919	0.930±0.969	0.807±0.864
Exh_A	0.927±0.963	0.847±0.920	0.929±0.964	0.773±0.847
Gen_W	**0.969±0.979**	0.746±0.865	0.746±0.865	0.673±0.807
Exh_W	0.879±0.930	0.745±0.854	0.744±0.862	0.693±0.819
Gen_M	0.935±0.980	**0.946±0.971**	**0.993±0.994**	**0.848±0.896**
Exh_M	0.936±0.980	0.941±0.967	0.975±0.991	0.811±0.864
Gen_T	0.890±0.938	0.812±0.885	0.706±0.847	0.691±0.852
Exh_T	0.881±0.931	0.802±0.888	0.692±0.842	0.693±0.847

the rest of applied oversampling techniques (e.g., SMOTE, ADASYN, MWMOTE, ICOTE and TRkNN) on the targeted domain datasets (i.e., Tables 1 and 2). It is also observed about MTDF that it can solve the issue of CIP of small dataset very efficiently. However, the other oversampling techniques needed a considerable number of samples during the learning process to approximate the real function. Consequently, the

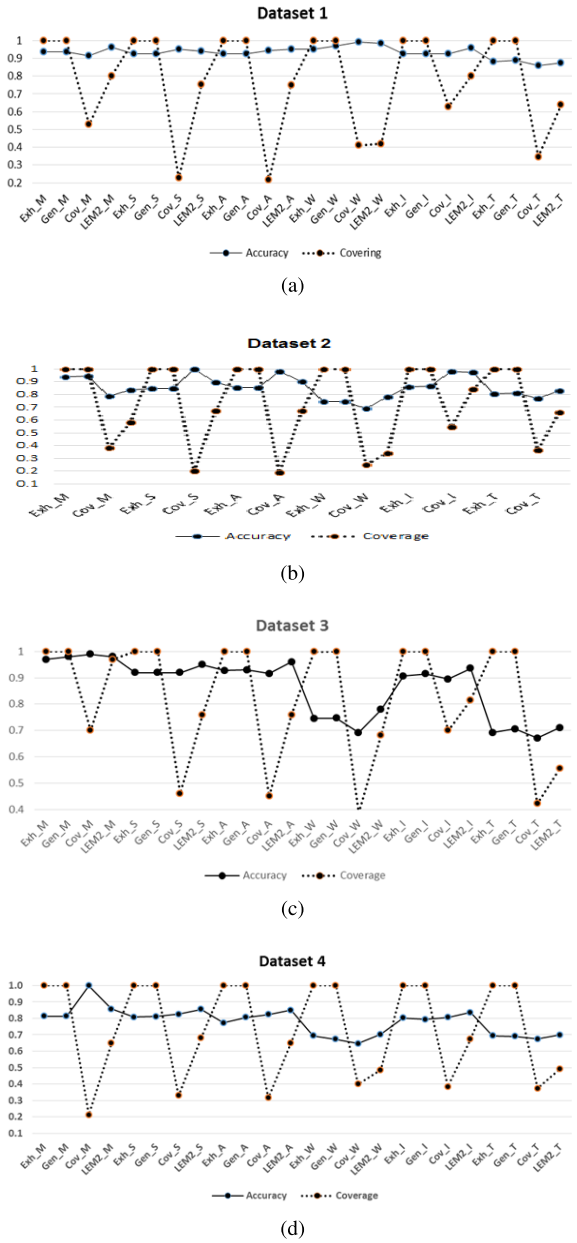

FIGURE 4. Coverage of objects and Accuracy of techniques on all datasets (1-4), where (a), (b), (c) and (d) represents the accuracy and coverage of selected algorithms on dataset 1,2, 3 and 4 respectively. List of algorithms is given on x-axis while y-axis reflects the number of samples (instances).

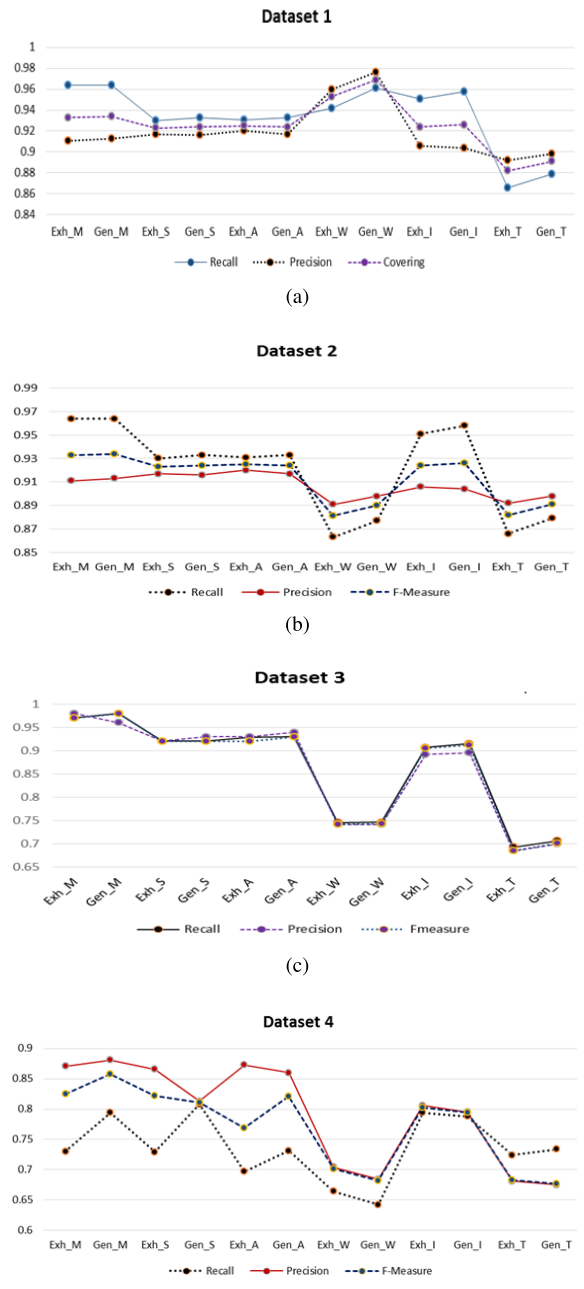

FIGURE 5. The positive predictive value (precision) and sensitivity (recall) followed by F-measure (the weighted harmonic mean of precision and recall) evaluate the classifiers' performance on dataset (1-4) where (a), (b), (c) and (d) reflects the preciseness and robustness the targeted algorithms. F-measure reaches its worst value at 0 & best value at 1.

existing number of samples of used datasets have created difficulty when using small datasets for average performed (i.e., SMOTE, ADASYN and ICOTE) and worst performed (MWMOTE and TRkNN) algorithms in this study. In order to completely cover the information gap in datasets (e.g., Tables 1 and 2) and avoid the over estimation of samples, MTDF more appropriately substituted the required samples with the help of both data trend estimation and mega diffusion [80]. On the other hand, MTDF is based on normal distribution which is a compulsory condition in the statistical data-analysis [11]. Therefore, MTDF is best techniques for more systematically estimating the existing samples or data in the dataset.

C. STATISTICAL TEST

In order to compare different classifiers and algorithms, this study supports our comparative assumptions through statistical evidence. For this work, a non-parametric test was used to provide statistical comparisons of some classifiers according

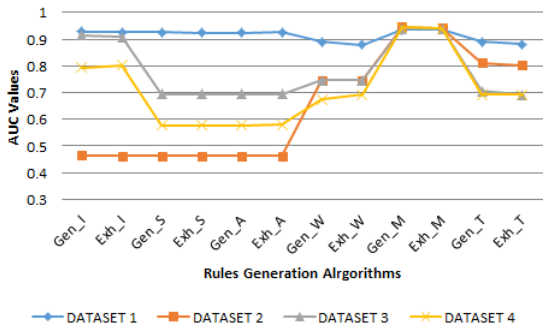

FIGURE 6. Comparison of selected algorithms based on AUC values.

to recommendation suggested by [4], [70], and [81]. The reasons why this study uses a nonparametric statistical test as follows [81], [82]: (i) these tests can handle both normally and non-normally distributed data while parametric test, usually apply on normally distributed data only, (ii) non-parametric test are guaranteed the reliability of the parametric test, and (iii) parametric tests are more closed to reject the null-hypothesis than the non-parametric tests unless their assumptions are violated. Therefore, Demsar [81] recommended using the non-parametric tests than using parametric test. These tests may not be satisfied causing the statistical analysis to lose its credibility. Furthermore, McNemar's test [83] was applied to evaluate the classifiers' performance by comparing the results of best-performing algorithms (e.g. Gen_M, Gen_S, Gen_A, Gen_I, Gen_W and Gen_T).

Under the H_0 (e.g., null hypothesis), different algorithms should have the same error rate, which means that classifier A = Classifier B. McNemar's test is based on the chi statistic distribution χ^2 test for goodness-of-fit that compares the distribution of the counts expected under the H_0 to the observed values. it is sufficed that to reject the H_0 in favor of the hypothesis that the multiple algorithms have different performance when trained on the targeted data set. McNemar's Test value can be calculated using the following formula given in equation (22) [83], [84]:

$$M = \frac{(|n_{01} - n_{10}|)^2}{n_{01} - n_{10}} > \chi^2_{1,\alpha} \tag{22}$$

The probability for the quantity of M is larger than $\chi^2_{1,0.95} = 3.841459$ is less than 0.05 for 95% confidence test with 1 degree of freedom [84]. If the null hypothesis that the debate had no effect were true then Gen_M = Gen_S = Gen_A and so on, as chi statistic χ^2 where the degree of freedom is 1. In case the H_0 is correct, then the probability that this quantity is greater than the $\chi^2_{1,0.95} = 3.841459$ is less than 0.05 for 95% confidence test [84]. McNemar's Test value can be calculated using a formula (i.e. equation (22)). We can reject the H_0, as these classifiers have the same error rate which may also consider in favor of the null hypothesis that these algorithms have different performance when trained on the same datasets (e.g. dataset 1, 2, 3 and 4). Table 14 reflects the performance of classifiers.

TABLE 14. p-value (P), M values obtained from McNemar's test.

Algorithms		Dataset 1	Dataset 2	Dataset 3	Dataset 4	Avg. Values
Gen_M	P	0.012	**0.078**	**0.003**	**0.0008**	**0.0234**
	M	7.04878	**3.84845**	**9.30769**	**8.27944**	**7.01359**
Gen_S	P	0.542	0.199	0.41	0.656	0.5092
	M	0.58139	1.86452	0.92452	0.05581	0.8565
Gen_A	P	0.661	0.326	0.322	0.0007	0.32742
	M	0.0088	0.0127	-0.0117	8.27751	2.07183
Gen_W	P	**0.0001**	0.9000	0.9980	0.851	0.79975
	M	**7.70916**	0.03937	0.0397	0.29388	0.27044
Gen_I	P	0.018	0.235	0.508	0.9800	0.43525
	M	7.2867	1.52039	0.73529	0.01207	2.38861
Gen_T	P	0.526	0.815	0.851	0.473	0.66625
	M	0.59346	0.17234	0.08707	0.59838	0.36281

"M" values is greater than chi statistic $\chi^2 = 3.841459$ reflecting to reject the null hypothesis with 95% confidence and 1 degree of freedom while "P" values is lower than 0.05 indicating that the performance difference between classifiers is statistically significant. The overall best average performance between the classifiers is shown in bold.

D. THREATS TO VALIDITY

- **Open source tools and public dataset:** To investigate the performance of the proposed solution, this study has used four publicly available datasets from different sources related to the telecom sector. Also open source tools for evaluation and classification process were used; therefore, the results may not be generalizable to closed-source tools or proprietary data, which may lead to variance in performance.
- **Distribution of artificial samples:** this study has also applied the randomization of artificial samples to avoid the biases in the distribution of samples population over the decision classes through the adoption of the following steps: (i) producing a unique number, used as sequence for each class label using a rand-between statistical function, then (ii) sorting function is used to arrange the generated values, (iii) finally, the normalization procedure using standardized statistical function particularly on "VMail_Message and Intl_Plan". Applying different function or method for normalization may not lead to consistent results.
- **Evaluation methods:** our work has used K-fold cross validation to avoid biases in the proposed work. Applying other validation methods (e.g., hold-out validation method) might not provide the same results. Different splits may also yield different performances.

VI. CONCLUSION AND FUTURE WORK

This study has addressed the class imbalance problem (CIP) and has focused on comparing the performance of six oversampling solutions for dealing with CIP—Mega-trend Diffusion Function (MTDF), Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling

approach (ADASYN), Majority Weighted Minority Oversampling Technique (MWMOTE), immune centroids oversampling technique (ICOTE) and Couples Top-N Reverse k-Nearest Neighbor (TRkNN)—on four publicly available datasets for the telecommunications sector. We examined the performance of these oversampling techniques with four (4) different rules-generation algorithms based on the Rough Set Theory (RST) classification approach—i.e., Learning from Example Module, version 2 (LEM2), Covering, Exhaustive and Genetic algorithms. The experiments performed show that the Genetic algorithm using MTDF for oversampling yielded the best performance in dealing with the CIP. Future studies might delineate more specific issues such as an extension to address the multi-class learning problems instead of considering only two class problem, analyzing time complexity of targeted oversampling techniques (i.e. applied in proposed study). This will further help in investigating the performances of the over-sampling techniques not only on two-class but also multi-class problems. Moreover, another natural extension of this study is to analyze the Receiver operating characteristic (ROC) curves obtained from the two-class & multi-class classifiers. This might provide us with a deeper understanding of the behavior of balancing and data preparation of large imbalanced datasets. Finally, it is also important to investigate the nature of the outliers before oversampling because none of the methods consider removing outliers before oversampling.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their insightful comments and suggestions, which helped improve the quality of this paper.

REFERENCES

[1] W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2354–2364, 2011.

[2] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4626–4636, 2009.

[3] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 40–49, 2004.

[4] 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. 2012.

[5] S. Maheshwari, J. Agrawal, and S. Sharma, "New approach for classification of highly imbalanced datasets using evolutionary algorithms," *Int. J. Sci. Eng. Res.*, vol. 2, no. 7, pp. 1–5, 2011.

[6] 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, no. 1, pp. 321–357, 2002.

[7] J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of imbalanced data by using the smote algorithm and locally linear embedding," in *Proc. 8th Int. Conf. Signal Process.*, vol. 3. 2006, pp. 1–4.

[8] C. S. Ertekin, "Adaptive oversampling for imbalanced data classification," in *Proc. 28th Int. Symp. Comput. Inf. Sci.*, vol. 264. Sep. 2013, pp. 261–269.

[9] 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.

[10] D.-C. Li, C.-S. Wu, T.-I. Tsai, and Y.-S. Lina, "Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge," *Comput. Oper. Res.*, vol. 34, no. 4, pp. 966–982, 2007.

[11] C. C. Teck, L. Xiang, Z. Junhong, L. Xiaoli, C. Hong, and D. Woon, "Hybrid rebalancing approach to handle imbalanced dataset for fault diagnosis in manufacturing systems," in *Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jul. 2012, pp. 1224–1229.

[12] J.-C. Deville and Y. Tillé, "Efficient balanced sampling: The cube method," *Biometrika*, vol. 91, no. 4, pp. 893–912, 2004.

[13] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Int. Conf. Artif. Intell.*, 2000, pp. 1–7.

[14] A. Amin, F. Rahim, I. Ali, C. Khan, and S. Anwar, "A comparison of two oversampling techniques (smote vs mtdf) for handling class imbalance problem: A case study of customer churn prediction," in *Proc. 3rd World Conf. Inf. Syst. Technol. (WorldCIST)*, Apr. 2015, pp. 215–225.

[15] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach," *Eur. J. Oper. Res.*, vol. 218, no. 1, pp. 211–229, 2012.

[16] D. Van den Poel and B. Lariviere, "Customer attrition analysis for financial services using proportional hazard models," *Eur. J. Oper. Res.*, vol. 157, no. 1, pp. 196–217, 2004.

[17] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.

[18] M. Karnstedt, M. Rowe, J. Chan, H. Alani, and C. Hayes, "The effect of user features on churn in social networks," in *Proc. 3rd Int. Web Sci. Conf.*, 2011, p. 23.

[19] X. Long *et al.*, "Churn analysis of online social network users using data mining techniques," in *Proc. Int. Multi Conf. Eng. Comput. Sci.*, vol. 1. 2012, pp. 551–556.

[20] W. Verbeke, D. Martens, and B. Baesens, "Social network analysis for customer churn prediction," *Appl. Soft Comput.*, vol. 14, pp. 431–446, Jan. 2014.

[21] P. D. Kusuma, D. Radosavljevik, F. W. Takes, and P. van der Putten, "Combining customer attribute and social network mining for prepaid mobile churn prediction," in *Proc. 23rd Annu. Belgian Dutch Conf. Mach. Learn. (BENELEARN)*, 2013, pp. 50–58.

[22] J. Wang, C. Jiang, T. Q. S. Quek, X. Wang, and Y. Ren, "The value strength aided information diffusion in socially-aware mobile networks," *IEEE Access*, vol. 4, pp. 3907–3919, Aug. 2016.

[23] U. D. Prasad and S. Madhavi, "Prediction of churn behavior of bank customers using data mining tools," *Bus. Intell. J.*, vol. 5, no. 1, pp. 96–101, 2012.

[24] K. Chitra and B. Subashini, "Customer retention in banking sector using predictive data mining technique," in *Proc. 5th Int. Conf. Inf. Technol.*, 2011, pp. 1–4.

[25] J. Bloemer, K. de Ruyter, and P. Peeters, "Investigating drivers of bank loyalty: The complex relationship between image, service quality and satisfaction," *Int. J. Bank Marketing*, vol. 16, no. 7, pp. 276–286, 1998.

[26] M. A. H. Farquad, V. Ravi, and S. B. Raju, "Churn prediction using comprehensible support vector machine: An analytical CRM application," *Appl. Soft Comput.*, vol. 19, pp. 31–40, Jun. 2014.

[27] C.-S. Lin, G.-H. Tzeng, and Y.-C. Chin, "Combined rough set theory and flow network graph to predict customer churn in credit card accounts," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 8–15, 2011.

[28] K. Lee, N. Chung, and K. Shin, "An artificial intelligence-based data mining approach to extracting strategies for reducing the churning rate in credit card industry," *J. Intell. Inf. Syst.*, vol. 8, no. 2, pp. 15–35, 2002.

[29] M. Suznjevic, I. Stupar, and M. Matijasevic, "MMORPG player behavior model based on player action categories," in *Proc. 10th Annu. Workshop Netw. Syst. Support Games*, 2011, Art. no. 6.

[30] J. Kawale, A. Pal, and J. Srivastava, "Churn prediction in MMORPGs: A social influence based approach," in *Proc. Int. Conf. Comput. Sci. Eng. (CSE)*, vol. 4. 2009, pp. 423–428.

[31] M. L. Kane-Sellers, "Predictive models of employee voluntary turnover in a north American professional sales force using data-mining analysis," Ph.D. dissertation, Texas A&M Univ., College Station, TX, USA, Aug. 2007.

[32] V. V. Saradhi and G. K. Palshikar, "Employee churn prediction," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1999–2006, 2011.

[33] M. Saron and Z. A. Othman, "Academic talent model based on human resource data mart," *Int. J. Res. Comput. Sci.*, vol. 2, no. 5, p. 29, 2012.

[34] G. Dror, D. Pelleg, O. Rokhlenko, and I. Szpektor, "Churn prediction in new users of Yahoo! Answers," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 829–834.

[35] M. Jaudet, N. Iqbal, and A. Hussain, "Neural networks for fault-prediction in a telecommunications network," in *Proc. 8th Int. IEEE Multitopic Conf. (INMIC)*, Dec. 2014, pp. 315–320.
[36] A. Hawalah and M. Fasli, "Dynamic user profiles for Web personalisation," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2547–2569, 2015.
[37] N. Ahad, J. Qadir, and N. Ahsan, "Neural networks in wireless networks: Techniques, applications and guidelines," *J. Netw. Comput. Appl.*, vol. 68, pp. 1–27, Jun. 2016.
[38] C. Fang, J. Liu, and Z. Lei, "Fine-grained HTTP Web traffic analysis based on large-scale mobile datasets," *IEEE Access*, vol. 4, pp. 4364–4373, Aug. 2016.
[39] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, May 2013.
[40] R. A. Soeini and K. V. Rodpysh, "Applying data mining to insurance customer churn management," *Int. Proc. Comput. Sci. Inf. Technol.*, vol. 30, pp. 82–92, Feb. 2012.
[41] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 313–327, 2008.
[42] J. Burez and D. Van den Poel, "CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services," *Expert Syst. Appl.*, vol. 32, no. 2, pp. 277–288, 2007.
[43] N. Chawla, N. Japkowicz, and A. Kolcz, "Special issue on learning from imbalanced datasets," *ACM SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.
[44] S. Visa and A. Ralescu, "Issues in mining imbalanced data sets—A review paper," in *Proc. 16th Midwest Artif. Intell. Cognit. Sci. Conf.*, 2005, pp. 67–73.
[45] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*. Springer, 2005, pp. 853–867.
[46] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 7–19, 2004.
[47] F. He, X. Wang, and B. Liu, "Attack detection by rough set theory in recommendation system," in *Proc. IEEE Int. Conf. Granular Comput. (GrC)*, 2010, pp. 692–695.
[48] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 20–29, 2004.
[49] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. Eur. Conf. Principles Data Mining Knowl. Discovery*, 2003, pp. 107–119.
[50] B. Liu, Y. Ma, and C. K. Wong, "Improving an association rule based classifier," in *Proc. Eur. Conf. Principles Data Mining Knowl. Discovery*, 2000, pp. 504–509.
[51] K. P. Satyasree and J. Murthy, "An exhaustive literature review on class imbalance problem," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 2, pp. 109–118, May 2013.
[52] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 4th Int. Conf. Mach. Learn.*, vol. 97. Nashville, TN, USA, Jul. 1997, pp. 179–186.
[53] C. X. Ling and C. Li, "Data mining for direct marketing: Problems and solutions," in *Proc. KDD*, 1998, pp. 73–79.
[54] Y. Tang, S. Krasser, D. Alperovitch, and P. Judge, "Spam sender detection with classification modeling on highly imbalanced mail server behavior data," in *Proc. Artif. Intell. Pattern Recognit.*, 2008, pp. 174–180.
[55] G. Wu and E. Y. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 786–796, Jun. 2006.
[56] P. Foster, "Machine learning from imbalanced data sets 101," in *Proc. AAAI Workshop Imbalanced Data Sets*, 2000, pp. 1–3.
[57] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 30–39, 2004.
[58] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[59] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE—Majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.
[60] M.-F. Tsai and S.-S. Yu, "Distance metric based oversampling method for bioinformatics and performance evaluation," *J. Med. Syst.*, vol. 40, no. 7, pp. 1–9, 2016.
[61] X. Ai, J. Wu, V. S. Sheng, P. Zhao, Y. Yao, and Z. Cui, "Immune centroids over-sampling method for multi-class classification," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2015, pp. 251–263.
[62] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, Oct. 1982.
[63] Z. Pawlak and A. Skowron, "Rough sets and conflict analysis," in *E-Service Intelligence* (Studies in Computational Intelligence), vol. 37. Springer, 2007, pp. 35–74.
[64] J. G. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wróblewski, "Rough set algorithms in classification problem," in *Rough Set Methods and Applications* (Studies in Fuzziness and Soft Computing), vol. 56. Springer, 2000, pp. 49–88.
[65] J. G. Bazan and M. Szczuka, "The rough set exploration system," in *Transactions on Rough Sets III* (Lecture Notes in Computer Science), vol. 3400. Springer, 2005, pp. 37–56.
[66] H. Nguyen and S. Nguyen, "Analysis of STULONG data by rough set exploration system (RSES)," in *Proc. ECML/PKDD Workshop*, 2003, pp. 71–82.
[67] J. Wróblewski, "Genetic algorithms in decomposition and classification problems," in *Rough Sets in Knowledge Discovery 2* (Studies in Fuzziness and Soft Computing), vol. 19. Springer, 1997, pp. 471–487.
[68] J. W. Grzymala-Busse, "A new version of the rule induction system LERS," *Fundam. Inform.*, vol. 31, no. 1, pp. 27–39, 1997.
[69] J. W. Grzymala-Busse, "LERS—A system for learning from examples based on rough sets," in *Intelligent Decision Support* (Theory and Decision Library), vol. 11. Springer, 1992, pp. 3–18.
[70] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013.
[71] (Aug. 2015). *Data Source*, accessed on Aug. 1, 2015. [Online]. Available: http://www.sgi.com/tech/mlc/db/
[72] (Oct. 2015). *Data Source*, accessed on Oct. 15, 2015. [Online]. Available: http://lamda.nju.edu.cn/yuy/dm07/assign2.htm
[73] (Oct. 2015). *KDD'06 Challenge Dataset*, accessed on Oct. 15, 2015. [Online]. Available: http://www3.ntu.edu.sg/sce/pakdd2006/
[74] (Jan. 2016). *IBM Telecom Dataset*, accessed on Jan. 1, 2016. [Online]. Available: https://www.ibm.com/analytics/watson-analytics/community/predictive-insights-in-the-telco-customer-churn-data-set/
[75] G. Holmes, A. Donkin, and I. H. Witten, "WEKA: A machine learning workbench," in *Proc. 2nd Austral. New Zealand Conf. Intell. Inf. Syst.*, Nov./Dec. 1994, pp. 357–361.
[76] Microsoft. (2014). *Standardize Function*, accessed on Nov. 9, 2014. [Online]. Available: http://office.microsoft.com/en-001/excel-help/standardize-function-HP010342919.aspx
[77] R. Bellazzi and B. Zupan, "Predictive data mining in clinical medicine: Current issues and guidelines," *Int. J. Med. Informat.*, vol. 77, no. 2, pp. 81–97, 2008.
[78] H. S. Almuallim, "Concept coverage and its application to two learning tasks," Ph.D. dissertation, Oregon State Univ., Corvallis, OR, USA, Apr. 1992, pp. 10–30.
[79] M. R. K. Dhurjad and M. S. Banait, "A survey on oversampling techniques for imbalanced learning," *Int. J. Appl. Innov. Eng. Manage.*, vol. 3, no. 1, pp. 279–284, 2014.
[80] N. H. Ruparel, N. M. Shahane, and D. P. Bhamare, "Learning from small data set to build classification model: A survey," in *Proc. IJCA Int. Conf. Recent Trends Eng. Technol. (ICRTET)*, May 2013, pp. 23–26.
[81] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[82] Minitab. (2015). *Choosing Between a Nonparametric Test and a Parametric Test*, accessed on Dec. 19, 2015. [Online]. Available: http://blog.minitab.com/blog/adventures-in-statistics/choosing-between-a-nonparametric-test-and-a-parametric-test
[83] B. S. Everitt, *The Analysis of Contingency Tables*. Boca Raton, FL, USA: CRC Press, 1992.
[84] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, 1998.
[85] K. Dashtipour *et al.*, "Multilingual sentiment analysis: State of the art and independent comparison of techniques," *Cognit. Comput.*, vol. 8, no. 4, pp. 757–771, 2016.

[86] Z. Tang, J. Lu, and P. Wang, "A unified biologically-inspired prediction framework for classification of movement-related potentials based on a logistic regression model," *Cognit. Computat.*, vol. 7, no. 6, pp. 731–739, 2015.

ADNAN AMIN received the M.Sc. degree in computer science from the University of Peshawar, and the M.S. degree (with distinction) in computer science with major databases from the Institute of Management Sciences, Peshawar, Pakistan, in 2008 and 2015, respectively. He is currently a Ph.D. Scholar and a Lecturer with the Department of Computer Science, Institute of Management Sciences, Peshawar. His research interests include data mining, databases, big data, and machine learning.

SAJID ANWAR received the B.Sc. and M.Sc. degrees in computer science from the University of Peshawar in 1997 and 1999, respectively, and the M.S. degree in computer science and the Ph.D. degree in software architecture from the University of NUCES-FAST, Pakistan, in 2007 and 2011, respectively. He is currently an Assistant Professor of Computing Science, and a Coordinator of the B.S.-Software Engineering with the Institute of Management Sciences, Peshawar, Pakistan. His research interests are concerned with software architecture, software requirement engineering, searched-based software engineering, and mining software repository.

AWAIS ADNAN received the Ph.D. degree from IMSciences, Peshawar, Pakistan, and the M.S. degree from NUST, Islamabad, Pakistan. He is currently an Assistant Professor and a Coordinator of Master Program with the Department of Comptuer Science, Institute of Management Sciences, Peshawar, where he is also a Manager of ORIC. His major areas of interest are multimedia and machine learning.

MUHAMMAD NAWAZ received the M.Sc. degree in computer science and the M.S. degree in information technology from the University of Peshawar, Pakistan. He was a Lecturer with the University of Peshawar, and also a Computer Programmer with Khyber Teaching Hospital, Peshawar. He is currently an Assistant Professor in multimedia with the Institute of Management Sciences, Peshawar, he is also the Head of Ph.D. and M.S.-computer sciences with IMSciences, Peshawar.

NEWTON HOWARD is currently an Associate Professor of Computational Neuroscience and Functional Neurosurgery with the University of Oxford, where he manages the newly formed Computational Neuroscience Laboratory. He is also the Director of the Synthetic Intelligence Laboratory, MIT, where he served as the Founder and the Director of the MIT Mind Machine Project from 2008 to 2012. He is also the Chairman of the Brain Sciences Foundation. He is an active member of several research laboratories worldwide, including the Descartes Institute, the Brain Physics Group, and INSERM, Paris.

JUNAID QADIR (SM'14) received the Ph.D. degree from the University of New South Wales, Australia, in 2008, and the bachelor's degree in electrical engineering from UET, Lahore, Pakistan, in 2000. He was an Assistant Professor with the School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology from 2008 to 2015. In 2015, he joined as an Associate Professor with the Information Technology University-Punjab, Lahore. He has 7+ years of teaching experience in which he has taught a wide portfolio of courses in systems and networking, signal processing, and wireless communications and networking. His primary research interests are in the areas of computer systems and networking and using ICT for development. He has served in the program committee of a number of international conferences and reviews regularly for various high-quality journals. He is a member of ACM. He is the award-winning teacher who has been awarded the highest National Teaching Award in Pakistan and the higher education commission's (HEC) Best University Teacher Award, in 2012 and 2013, respectively. He has been nominated for the HEC Best University Teacher Award twice from NUST in 2011, and 2012–2013. He is an Associate Editor for the IEEE Access, the Central's *Big Data Analytics journal* (Springer/BioMed), and the *IEEE Communications Magazine*.

AHMAD HAWALAH is currently the Vice Dean and the Head of IS Department with the College of the Computer Science and Engineering, Taibah University, Medina, Saudi Arabia. He received the M.Sc degree from Sussex University, Brighton, UK, and the Ph.D. degree in information retrieval and data mining from Essex University, Colchester, UK, in 2012. He is currently the Vice Dean and the Head of IS Department with the College of the Computer Science and Engineering, Taibah University, Medina, Saudi Arabia. His research interests include data mining, artificial intelligence, and information sciences. His research interests include data mining, artificial intelligence, and information sciences.

AMIR HUSSAIN received the B.Eng. degree (Hons.) and the Ph.D. degree in novel neural network architectures and algorithms from the University of Strathclyde, Glasgow, U.K., in 1992 and 1997, respectively. He is currently a Professor of Computing Science and the Founding Director of the Cognitive Big Data Informatics (CogBID) Research Laboratory, University of Stirling, U.K. He has conducted and led collaborative research with industry; partnered in major European research programs, and supervised over 30 Ph.D. students. He has authored over 300 papers, including over a dozen books and 100+ journal papers. He is the founding Editor-in-Chief of the journals: Cognitive Computation (Springer Nature), and Big Data Analytics (BioMed Central), and Chief-Editor of the Springer Book Series on Socio-Affective Computing, and Springer Briefs on Cognitive Computation. He is Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Systems, Man, and Cybernetics: Systems, and the IEEE Computational Intelligence Magazine. He is a Senior Fellow of the Brain Science Foundation (USA), Senior Member of the IEEE, a member of several Technical Committees of the IEEE Computational Intelligence Society, and Chapter Chair of the IEEE U.K. & RI Industry Applications Society.