



Optimum profit-driven churn decision making: innovative artificial neural networks in telecom industry

Ruholla Jafari-Marandi¹ · Joshua Denton² · Adnan Idris³ · Brian K. Smith⁴ · Abbas Keramati^{5,6}

Received: 11 March 2019 / Accepted: 14 March 2020 / Published online: 9 April 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Knowledge-based churn prediction and decision making is invaluable for telecom companies due to highly competitive markets. The comprehensiveness and action ability of a data-driven churn prediction system depend on the effective extraction of hidden patterns from the data. Generally, data analytics is employed to extrapolate the extracted patterns from the training dataset to the test set. In this study, one more step is taken; the improved prediction performance is attained by capturing the individuality of each customer while discovering the hidden pattern from the train set and then applying all the knowledge to the test set. The proposed churn prediction system is developed using artificial neural networks that take advantage of both self-organizing and error-driven learning approaches (ChP-SOEDNN). We are introducing a new dimension to the study of churn prediction in telecom industry: a systematic and profit-driven churn decision-making framework. The comparison of the ChP-SOEDNN with other techniques shows its superiority regarding both accuracy and misclassification cost. Misclassification cost is a realistic criterion this article introduces to measure the success of a method in finding the best set of decisions that leads to the minimum possible loss of profit. Moreover, ChP-SOEDNN shows capability in devising a cost-efficient retention strategy for each cluster of customers, in addition to strength in dealing with the typical issue of imbalanced class distribution that is common in churn prediction problems.

Keywords Artificial neural networks (ANNs) · Profit-driven churn prediction · Self-organizing map (SOM) · Self-organizing error-driven ANN (SOEDANN)

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s00521-020-04850-6>) contains supplementary material, which is available to authorized users.

✉ Ruholla Jafari-Marandi
rojafari@calpoly.edu

¹ Industrial and Manufacturing Engineering Department, Cal Poly, San Luis Obispo, CA 93407, USA

² Department of Marketing, Quantitative Analysis, and Business Law, Mississippi State University, Starkville, MS 39759, USA

³ Department of Computer Sciences and IT, University of Poonch, Rawalakot, Pakistan

⁴ Department of Industrial and Systems Engineering, Mississippi State University, Starkville, MS 39759, USA

⁵ Ted Rogers School of Information Technology Management, Ryerson University, Toronto, ON, Canada

⁶ School of Industrial and Systems Engineering, University of Tehran, Tehran, Iran

1 Introduction

Subscription-based firms prefer retaining customers to acquiring new ones due to the relatively higher cost of customer acquisition [18]. Telecom industry witnesses highly saturated markets around the globe; thus, retaining existing customers is a prime concern for the firms in order to maintain stable profits. The telecom industry has two characteristics that give rise to the competitive application of data analysis for churn prediction. First, companies store and utilize different types of customer data such as personal, demographic, billing, and usage behavior. Second, owing to the saturated telecom market, there is fierce competition to win customers. Many studies focus on improving the accuracy of churn prediction for telecom customers. For instance, Keramati et al. [31] reported above 90% accuracy for predicting telecom churners. While the major part of the literature focuses on improving the accuracy of churn prediction, it is essential that telecom companies aim to apply churn models from a financial

perspective [54]. Since profit is often the highest priority in business environments, churn models need to move from applying standard classification techniques and toward incorporating revenue—gain or loss depending on the action of the firm—in more comprehensive models [16].

Churn prediction, in different disciplines, depends on various statistical and data analytic methods. Although research efforts to develop more accurate data analytic models are valuable, it is also important to realize that data analysis can only discover the existing patterns in the data. The prediction performance of a learning system is dependent on the quality of learning induced from the training set. If the pattern that leads to perfect prediction does not exist in a training set, then achieving 100% accuracy is impossible. In other words, the prediction performance can only be as good as the hidden patterns in the dataset used for learning. Nevertheless, it is valid to argue that the existing methods might not be able to capture all the existing hidden patterns. However, the argument cannot be refuted unless a method is discovered that is more successful at finding all the hidden patterns. This paper emphasizes that it is valuable to explore other means to improve churn decision making rather than just seeking more effective pattern recognition.

An improved methodology can be developed by accepting the existence of misclassifications and instead targeting lower misclassification cost rather than focusing on obtaining fewer numbers of misclassifications. In other words, a churn classifier can be developed that prefers more misclassifications with lower cost over fewer misclassifications with higher cost. The existing data analysis techniques collectively apply the extracted patterns to make churn decision for all customers without taking the uniqueness of each customer into consideration. Therefore, systematically applying the knowledge of customer value before making final churn decisions has the potential to significantly decrease misclassification cost. For instance, the misclassification of a highly valuable customer as a non-churner is much costlier than the misclassification of a normal customer as a churner. Generally, mistaken churn classifications are less costly, as they only incur an unnecessary retention expenditure, compared to mistaken non-churn classifications that may lose a company's valuable customers. Having systematically calculated the misclassification costs (false positive or false negative), in the case of each customer, helps focus the prediction power of data analysis technique toward minimizing the misclassification cost.

This paper proposes a method that systematically leverages the information extracted from an unsupervised learning technique to calculate the value of each type of customer and employ those values in the supervised learning of a churn classification task. The proposed profit-

driven method employs both self-organizing and error-driven learning styles of ANNs for churn prediction (ChP-SOEDNN) to incorporate misclassification costs of customers into method training.

The general structure of the paper is as follows. Section 2 presents the relevant literature review of churn decision making and, more importantly, discusses where this study is contributing to the literature. Section 3 provides background for the presentation of the proposed system (ChP-SOEDNN), whereas details of the proposed system are presented in Sect. 4. Section 5 presents the results of the performed experiments. Section 6 discusses the results from different perspectives. Finally, conclusions are drawn in Sect. 7.

2 Literature review

An important part of the literature is the introduction of and improvement in different data mining techniques for better predictions [15]. Specifically, Wei and Chiu [56] use call pattern changes and contractual data for developing a churn prediction technique with a data mining approach. Similarly, Keramati et al. [31] propose a hybrid technique using decision trees, ANN, support vector machines (SVMs), and *K*-nearest neighbors (KNNs) to improve the quality of extracted patterns and accuracy of prediction. Amin et al. [2] propose a rough set theory-based approach, using genetic algorithm to devise decision rules for efficiently classifying churn and non-churn customers. Additionally, Chen et al. [10] and Chen and Fan [11] propose two methods naming hierarchical multiple kernel support vector machine (H-MK-SVM), and collaborative multiple kernel support vector machine (C-MK-SVM) for multi-level, multifaceted, and longitudinal behavioral datasets including telecom. Moreover, van Wezel and Potharst [53] incorporate “Bagging,” “Boosting,” and “MultiBoosting” as ensemble techniques with base statistical and machine learning methods such as logistic regression, and decision trees to improve customer choice predictions.

The literature also offers the researchers and practitioners specific guidelines toward feature selection and the relevant techniques. Specifically, Tang et al. [51] employ an orthogonal polynomial approximation analysis with the purpose of deriving hidden information before churn classification and then show that the inclusion of the derived information in a profit-hazard rate model can increase model accuracy and interpretability. Furthermore, Huang et al. [22] contribute to the literature of churn prediction by finding the best feature sets through experimenting with different feature subsets and seven prediction techniques. Their new suggested set of features includes aggregated call details, Henley segmentation, account

information, bill information, dial types, line information, payment information, complaint information, and service information. Similarly, Keramati et al. [31] apply a heuristic dimension reduction method to conclude that frequency of use, complaints, time of use, status, and subscription length are the best churn predictors. Moreover, the literature depicts that the integration of feature selection, sample selection, and a strong classifier can significantly increase the accuracy of churn prediction [24]. Idris et al. [25, 26] use a minimum redundancy and maximum relevance (mRMR) feature selection and genetic algorithm (GA)-based feature selection approach to improve the RotBoost-based ensemble and Adaboosting-based classifiers to tackle churn prediction challenges more effectively.

It is a common practice, not just for churn prediction but also for other applications of binary classifications [57], to employ clustering methods parallel to classification methods in the hope of creating more control in the decision-making process of churn management in other ways than just increasing the accuracy. A prime example of such data-driven studies is the effort of Liu and Zhuang [37], which incorporates customer segmentation and classification cost in improving the churn decision-making process. In this study, *K*-means is first used to segment the customers into three groups, and then, a decision tree is employed for churn prediction using the results of the grouping and their associated misclassification costs. Similarly, Lu et al. [38] employ a boosting algorithm to separate the customer base and increase the accuracy of logistic regression for churn prediction. The weight of the boosting resulted in separating the dataset into two clusters which has contributed to a better classification of the dataset. Huang and Kechadi [23] integrate supervised and unsupervised techniques to improve the accuracy and interpretability of existing models: decision tree, logit regression, KNN, SVM, OneR, and PART. Additionally, Verbeke et al. [54] develop a profit-centric performance measure with the purpose of including only the optimal fraction of customers with the highest predicted probabilities for a customer retention campaign. Furthermore, Bi et al. [8], with the goal of bringing marketing strategies at the level of data analysis, proposed a semantic-driven subtractive clustering method to improve customer churn management within a big data environment. Sivasankar and Vijaya [3] also present a hybrid clustering classification approach, which exploits the explanation power of probabilistic possibilistic fuzzy *C*-means clustering (PPFCM) and distinguishing capability of ANN. The authors show that PPFCM–ANN outperforms decision tree (DT), SVM, KNN, and ANN when evaluated by accuracy, true churn value, and false churn value.

The application of cost-sensitive classifiers on churn datasets is an effort to include the cost of misclassifications

for improving churn decision making. Glady et al. [16] propose a comprehensive churn study using cost sensitivity toward customer lifetime value. The loss function used in Glady's study does not assume equal misclassification costs for all customers leading to cost-sensitive classification experiments that show significant improvements. However, in this paper we have gone one step further by including the individuality of each customer toward optimum classification decisions. In contrast, Glady et al. [16] propose to adjust a cut-off point based on the maximized cumulative profit for all predictions, whereas our proposed system introduces a computationally frugal approach to adjust individual cut-off points for each customer. Bahnsen et al. [5] incorporate cost sensitivity in the impurity criteria and pruning method of DTs and create a classification that is capable of including the cost sensitivity in the process of training. Later on, Bahnsen et al. [6] apply the developed cost-sensitive techniques on churn prediction to show significant savings compared to cost-insensitive approaches. Moreover, another recent and DT-based profit-driven churn prediction [21] optimizes a specialized measure called expected maximum profit measure for customer churn (EMPC) [55]. Similarly, Stripling et al. [48] put forward a genetic algorithm that inculcates EMPC into the model construction of churn prediction.

In the contemporary literature, several studies use deep neural network architectures [1, 32, 43] for telecom churn prediction problem. However, their primary focus lies in improving pattern recognition capabilities and improving the prediction accuracy of proposed solutions. In a few techniques such as [32], some representational information is obtained to recognize the social dependence of customers for controlling churn in addition to churn prediction accuracy. However, none of the deep neural network-based algorithms have so far been used in building a profit-driven churn prediction solution.

In contrast, this paper contributes to the literature from different perspectives. (1) The paper improves and brings together two existing segments of the literature: the usage of cost-sensitive techniques for profit-driven churn prediction and the application of unsupervised learning to enhance churn classification. SOED has become profit-driven at the intersection of supervised (error-driven) and unsupervised (self-organizing) learning of the proposed method. (2) Because of this, unlike the cost-sensitive methods available in the literature, our method is not confined to having to adjust only one decision-making cut-off point for all customers. The method is capable of optimally specifying different cut-off points for separate clusters of customers through a line adjustment procedure. As pointed out by Lemmens and Gupta [35], the existing “profit-driven” churn decision-making techniques are limited to the profit-driven assessment method through

only cost-sensitive learning. However, the presented method is profit-driven at both the level of learning and assessment. That is to say, unlike the other presented profit-driven studies that the data analytic models only include cost sensitivity and a profit-driven feedback loop used to help the model move slightly toward increasing profit, this paper maximizes the profit by dealing with the model restriction for a complete embracement of profit drive. (3) While the success of artificial neural networks (ANNs) and other data mining techniques used in dealing with churn prediction are proven, a tailored self-organizing error-driven (SOED) ANN customized for churn decision-driven data analytics is applied to show its superiority. (4) The paper shows and discusses the capability of the line adjustment procedure to include an optimum selection of retention strategy for each cluster of customers. The available profit-driven churn prediction approaches allow for only one retention strategy for all types of customers; however, we argue this is too simplistic of an approach and offer a clear future research direction. (5) In the literature of profit-driven churn decision making, the definition of the cost function ignores the possibility of customers churning despite retention plans; however, this paper includes such possibilities in the cost function definition by incorporating a retention probability calculation. (6) Results suggest that ChP-SOEDNN has also dealt effectively with one of the challenges of telecom churn prediction due to the imbalanced ratio of churn and non-churn customers.

3 Background concepts and methods

This section introduces concepts and background that are employed in shaping theoretical and technical contributions in this paper: cost-sensitive classification, profit-driven churn decision making, and three variations of artificial neural networks (ANNs), naming multilayered perceptron (MLP), self-organizing map, and self-organizing error-driven (SOED) ANN.

3.1 Cost-sensitive classification

In most of the classification literature, the notion of misclassification is considered as uniform. This viewpoint is apparent in the definition of various performance metrics that compare the performance of different classifiers. Metrics such as the number of misclassifications and accuracy treat all misclassifications as equally undesirable. However, in reality, different types of misclassifications, in various cases, have different levels of undesirability. For instance, in case of churn prediction, because of the costly nature of customer acquisition, false positives (FPs)—predicting non-churning customers as churning—tend to be

less undesirable than false negatives (FNs)—predicting churning customers as non-churning. A classification method that takes the level of the undesirability of misclassifications into account is cost sensitive. Data scientists inculcate cost sensitivity at different stages of a classification experiment: sampling, thresholding, and learning [20]. Also, the literature shows the application of different cost-sensitive performance metrics: area under receiver operating curve (AUC), sensitivity, specificity, recall, precision, and *F*-score.

The application of cost-blind classifiers over a resampled train set creates a cost-sensitive classifier at the sampling stage [59]. For instance, in the case of churn prediction, if FNs are five times as undesirable as FPs, the resampled train set should have five times more rows of data with the label churn than non-churn. Also, a cost-blind classifier can be used for a cost-sensitive classification if the threshold of ultimate decision making for churn prediction is manipulated so that the costlier misclassifications happen on fewer occasions. The threshold can be chosen naively (in between), based on the studied ratio of different misclassifications (for instance 1–5), or optimized using preliminary experiments [16]. Moreover, the literature includes cases of alteration of the learning process to create cost-sensitive classifiers. For example, Bahnsen et al. [5] include cost sensitivity in the impurity criteria and pruning of the decision tree and thus create a cost-sensitive classification at the stage of learning.

3.2 Profit-driven churn decision making

While it may seem logical to infer that applying a cost-sensitive classification will lead to a profit-driven churn decision making, the cost sensitivity of classifiers only captures one dimension of a data- and profit-driven churn decision making. Although cost-sensitive classifiers can recognize and include the undesirability balance between FP and FN misclassifications, they do not capture the uniqueness of each customer with regard to this balance of undesirability. Cost-sensitive classifiers can only accommodate one fixed ratio that captures the balance undesirability of FNs and FPs. For instance, assuming a 5–1 ratio for FN to FP means every time a FN happens its misclassification cost is 5 times more than anytime a FP misclassification occurs. This does not accommodate for the management reality that each customer has different values and therefore different balance of misclassification ratio for FP and FN. The most successful profit-driven churn decision making only has been able to include a real profit-driven assessment of classical cost-sensitive classifiers. At best, this allows to adjust the parameters of the cost-sensitive methods to reach the best possible profit-driven assessment [54]. However, as also recognized by

Lemmens and Gupta [36], a data-driven churn decision-making model that includes profit at the level data analytic mode is necessary. This paper presents that inclusion of profit at the level of model, as suggested, indeed leads to better (more profitable) decisions. This paper exclusively has inculcated a profit drive at the level of ANN learning.

3.3 Artificial neural networks (ANNs)

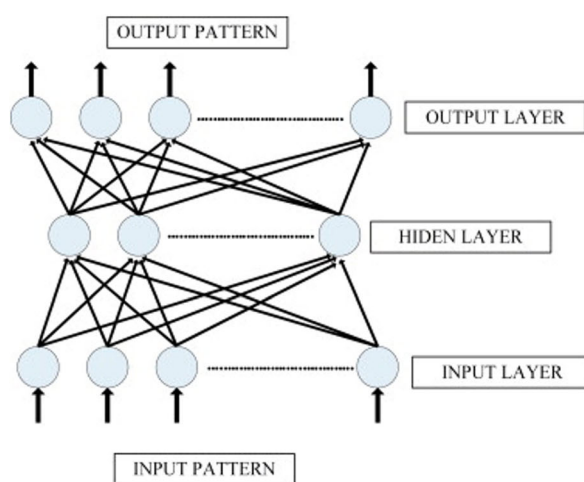
Artificial neural networks, as implied by the name, comprise the area of science and technology that uses simulations and models to hypothesize and investigate the human brain. Moreover, and especially in the case of artificial intelligence and data analysis, these methods use algorithms based on known brain structures to solve data analytic problems such as classification. This section introduces three variations of ANNs: multilayered perceptron (MLP), self-organizing map (SOM), and self-organizing error-driven ANN (SOED).

3.3.1 Multilayered perceptron (MLP)

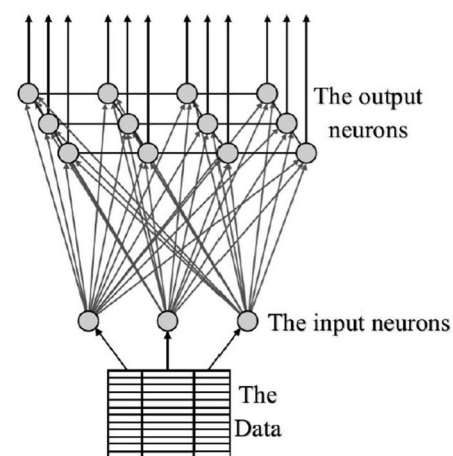
MLP, also known as feedforward ANN [19], is the variation of ANN that is mostly applied for supervised data analytic tasks and specifically for classification. Figure 1a illustrates the three most important parts of an MLP: the input, hidden, and output layers, with each layer composed of a number of neurons. Based on the complexity of the problem, the number of hidden layers is adjusted: More difficult classification tasks normally require more hidden layers and neurons. For a classification task, each neuron in the input layer represents one of the predicting (independent) attributes and each neuron in an output layer stands

for one of the targeting (dependent) attributes. The role of hidden layers is to find nonlinear relationships between independent and dependent attributes of a dataset. For instance, a churn prediction task with ten independent (predicting) attributes has ten neurons in the input layer and one in the output layer. For a churn prediction (classification) task, there is only one dependent (targeting) attribute, and that is if each customer churns or not.

As presented in Fig. 1a, all neurons of one level, with forwarding arrows, are connected exhaustively to the neurons of the next layer; hence, such an ANN is called feedforward. These connections are in fact MLP's means of finding the nonlinear relationship between input and output layers. There are a set of weights on these connections that relate any input to any output in a way that captures the complicated relationship. A process known as "training," which is also biologically motivated, finds these weights. An untrained network is not a network that has zeros for these weights, but is a network that has random weights between negative one and one assigned to each connection. So even for an untrained network, there is a relationship between input and output neurons. For each case of the data, an untrained network is capable of using the randomly assigned weights to calculate values for the output neurons. These predicted values will most likely be erroneous. In fact, the process of training an MLP calculates the error for later improvements. In each epoch of training, every case in a train set is presented to the network, and training happens by change of weights to reduce the prediction errors. Back propagation typically is the algorithm of choice for this purpose [29].



(a) MLP Structure



(b) SOM Structure

Fig. 1 MLP and SOM structures

3.3.2 Self-organizing map (SOM)

Self-organizing map (SOM) [33] is a unique and powerful variation of ANN. It is unique because unlike other ANNs the method solely relies on unsupervised training, and for that, it can address clustering tasks. Also, it is powerful owing to its effective graphical tool, which is capable of outputting a membership map for high-dimensional datasets. It offers more than just clustering techniques, as the position of each cluster in the map and its relationships with the neighboring clusters are meaningful.

Different from MLP, SOM has only two layers: input and output layers. Every attribute in a dataset has a neuron in the input layer. The number and formation of neurons in the output layer are not dependent of the dataset, but they can be adjusted to tailor the network based on the needs of different problems. The output layer of a SOM network is the membership map of SOM. There are two types of topology for this map: quadrilateral or hexagonal. Quadrilateral SOM topology houses each neuron, so it has at most four neighboring neurons. In the case of hexagonal, the number of possible neighbors increases to six. Figure 1b represents a SOM network for a dataset with three attributes. The output layer of Fig. 1b has 3×4 quadrilateral map.

The training process of SOM is both similar to and different from that of MLP. It is similar since the process presents the cases of data to the input layers, and weight changes happen for learning. However, it is different because SOM cannot take advantage of backpropagation due to the unavailability of a target value. Every time a network is exposed to a case, based on the current weights on the connections between neurons, one of the neurons in the output layer will have a greater value. The training process changes the weights to solidify the membership of the case to the particular neuron.

3.3.3 Self-organizing error-driven (SOED) ANN

Self-organizing error-driven (SOED) ANN [29] combines the unsupervised (self-organizing) learning drive of SOM with supervised (error-driven) learning of MLP to optimally use the different hidden pattern in a dataset for smarter classification. In SOED, MLP predicts the location of data points on SOM. When a simple dividing line is used to separate different parts of SOM for each class, SOED reports higher accuracy and reliability performance due to the fact that more hidden patterns of the data are extracted and used in prediction. However, when the segmentation of SOM for final classification is done with calculated consequence of mistakes, SOED will result in a set of decisions that are made by optimally using the extracted pattern of the data to minimize a quantified goal. For instance,

recently SOED has been used to successfully optimize decision-making goals in the areas of data-driven breast cancer diagnosis [28] and laser-based additive manufacturing microstructure fault detection [30].

4 Research framework

This section presents the structure of performed research to answer the research questions. The main research question is whether the proposed classification algorithm, namely ChP-SOEDNN, leads to decision-making improvement. A combination of many parts such as a recent literature review, a set of assumptions, several state-of-the-art contending algorithms, and detailed design of experiments provides the necessary understanding and comparison ground to evaluate the performance of the proposed algorithm. Because this section has many interconnected subsections whose understanding is important, Table 1 presents a summary of this section with a summary of each subsection.

4.1 Dataset

Generally, a dataset used for telecom churn prediction belongs to a specific socio-culture encompassing the detailed information about customers of a society [22, 26, 52]. This study also focuses on a dataset that is randomly collected from an Iranian telecom company's database over a period of 12 months [31]. A total of 3150 rows of data, each representing a customer, bear information for 12 columns. The attributes that are in this dataset are call failures, frequency of SMS, number of complaints, number of distinct calls, subscription length, age group, charge amount, type of service, seconds of use, status, and frequency of use. The dataset is randomly divided into three different sets: train set 2205 customers (70%), validation set 473 customers (15%), and test set 472 customers (15%). For achieving the research goals of this paper, the paper only uses a fixed randomization throughout. That means all of the reported performance metrics are only about the same 472 customers who are selected randomly. While this may compromise the validity evaluation of the generalizability of our proposals, it allows us to portray the importance and relevance of the assumptions needed for a profit-driven churn prediction. Moreover, the ratio of churn customers in the dataset is 15.7%, which is quite low. Such a low ratio of churn customers compared to non-churn customers leads to class imbalance problem [4, 59].

Table 1 The summary and structure of Sect. 4, research framework

Title: Short description	
4.1	Dataset: The description of the dataset that used for experimentation
4.2	Assumptions: The description of the included assumptions to allow for using classification algorithms as a decision-making model
4.3	Proposed churn decision-making framework: ChP-SOEDNN: The presentation of the proposed algorithm
4.4	Specific computations and calculations: The proposed decision-making model requires specific calculations and computations that are presented in four different subsections
<i>Subsections</i>	
1.	<i>Calculate customer value:</i> For the profit-driven decision-making problem, calculating a value for each customer is essential. This subsection presents the relevant literature and the formula that is used
2.	<i>Prediction preparation:</i> ChP-SOEDNN includes novel steps. This subsection attempts to illustrate them for better understanding of the method
3.	<i>Line adjustment procedure:</i> This procedure is the crux of ChP-SOEDNN's informed decision making. After having understood the method and customer value, this subsection presents a thorough description of the procedure
4.	<i>Misclassification cost calculation:</i> The new evaluation metric, namely misclassification cost, and its formulation
4.5	Significance of assumptions: Using the novelty of ChP-SOEDNN method and its specific line procedure, this subsection illustrates the significance of the presented assumption in Sect. 4.2
4.6	Comparative Analysis: the proposed algorithm is compared with other state-of-the-arts algorithms. Different subsections address various aspects such as the design of experiments and specific algorithms tuning
<i>Subsections</i>	
1.	<i>Grounds for Comparisons:</i> The foundation of classification experiment designs such as randomization, train and test sets, evaluation metrics, and assumptions
2.	<i>Methods tuning and Parameter Selection:</i> The general information about method tuning and parameter selection that are used for all of the methods
3.	<i>MLP:</i> The specific MLP tuning and parameter selection
4.	<i>SOED:</i> The specific SOED tuning and parameter selection
5.	<i>Decision trees:</i> The specific decision trees tuning and parameter selection
6.	<i>Cost-sensitive AdaBoost:</i> The specific AdaBoost tuning and parameter selection

4.2 Assumptions

Two assumptions are made to facilitate the calculation of misclassification costs in this study. First, when a customer is classified as churn, the telecom company is assumed to be willing to spend a fraction of the customer revenue as retention expenditure to prevent the churn. Retention expenditure ratio (RER) and retention steady expenditure (RSE) are the two names for this idea. RER uses a ratio of the value of customers to spend on retention offers, whereas RSE has a uniform retention plan for all customers. Customarily, it is assumed that customers will not churn if offered a retention plan [6, 16]. However, in reality not every churn retention plan will be successful. This paper formulates the second assumption to include the effectiveness of retention expenditures in order to be more realistic. The assumption is that retention expenditure has a constant success rate (SR). SR should correlate with the amount of retention expenditure. The more spent for the customer who is about to churn, the less probable that they will churn because retention efforts reduce the likelihood of a churn event [45]. The relationship is unlikely to be

linear, but we suspect that it exists. With the hope of devising an effective retention strategy, a company may opt to develop a proper mix of strategies [12, 34].

4.3 Proposed churn decision-making framework: ChP-SOEDNN

Churn prediction is a well-established and well-studied area of business [15]; however, the dimension of decision-making improvement, i.e., profit-driven churn decision making [6, 16, 54], remains underdeveloped. ChP-SOEDNN is developed to address this research gap by introducing a decision-making framework that infuses both the pattern recognition and risk analysis. While there are efforts that have bettered churn decision making by incorporating an informed predictive prescription [39], ChP-SOEDNN's strength is at infusing both pattern recognition and risk analysis. In other words, not only does the method in this paper use the extracted patterns to improve the decision-making, ChP-SOEDNN also finds the pattern that are more optimal in facilitating the best decision making.

This infusion is a possibility due to the power of SEOD that stems from combining two powerful ANNs, SOM and MLP, in a way that the two supplement each other for smarter classifications. This paper improves SOED classification method with tailoring its advantages toward a profit-driven churn decision-making framework. SOED only uses the extracted hidden patterns from both unsupervised and supervised learning and applies them toward better classifications; the improved method employs the extracted hidden pattern from unsupervised learning of ANN to capture similarities and dissimilarities of customers in a map and their calculated customer value. These captured relationships help the supervised learning of ANN to reach a set of decisions that lead to the highest profit. Figures 2 and 3 present ChP-SOEDNN's updated stepwise procedure and flowchart. Some elements of Fig. 2 are too small for reading; however, they are presented to give a general overview of the method; the smaller parts in Fig. 2 are presented later in the text. ChP-SOEDNN includes a profit drive at the level of ANN learning.

Step-0, which is the general advice when working with SOM [50], transforms every single column in the dataset to a range from negative one to one. Step-1 applies SOM on the transformed train set and validation set. It is worth mentioning that the target column, which is churn label in this study, is also included in the SOM training. Step-2 is a checkpoint, and it ensures that the SOM's output is clamped. In this paper, the output of SOM, which is basically a 2D membership map, also referred to as the map of decision (MOD), will be used for making the final decisions in Step-10. In the case of this paper, an MOD is clamped if the map has two distinctive parts for churn and non-churn; for example, in case of Fig. 4a, one can see that the bottom right part of the map is occupied only by churn customers and the rest of the map houses non-churn customers. If MOD has the explained distinction, the procedure can continue. If not, the increase in the clamping weight finally leads to a clamped map that has the explained distinction. Additionally, it is important to know the dataset is divided into three parts: train set, validation

set, and test set. Train set and validation set are applied to train SOM and ANN, whereas validation set is used to find the optimum dividing line (Step-9), and finally, the test set is saved for examining and evaluating the procedure. From another perspective, validation set is a part of train set as it helps SOED train better. However, validation set is identified differently for presentation purposes.

Step-3 finds a value for each cluster that, when compared to other clusters, stands for the worth of each customer in the cluster. Customer lifetime value (CLV) is a perfect candidate metric for this value. CLV is the worth of an entire customer lifetime with a business. Step-4 is concerned with the calculation of misclassification costs, and it is dependent on two distinct pieces of knowledge about the customers: a relationship between retention expenditure and retention success rate for each cluster, and a value that represents each customer's worth (provided in Step-3). The combination of the two leads to finding two types of misclassification costs for every cluster. This requires a way to calculate the probability of a potential churn customer being retained based on the cost we are willing to accept as retention expenditure. This relationship can be constructed using the data from previous retention plans of customers in each cluster. Steps-6 defines the classification target of MLP. Customarily, MLP is used to connect the independent variables of churn prediction (input neurons) to dependent variable which is the churn outcome (output neuron); however, in the Step-7 MLP connects the independent variables to the location of customers of MOD calculated in Step-6. Step-8 uses the trained MLP to predict the location of customers in the validation set on MOD. Step-5 tentatively separates MOD into two parts—churn and non-churn—and defines possibilities for a definite separation. Consequently, Step-9 uses the known churn outcome of validation set to find the possible separation which leads to minimum misclassification costs for the validation set. Step-10 uses MOD with the optimum separating line to decide the churn outcome of customers in the test set.

Fig. 2 ChP-SOEDNN pseudocode for optimum profit-driven SOED churn prediction system

ALGORITHM: ChP-SOEDNN procedure of churn prediction

Step-0	Transform all the attributes to be between -1 and 1; Initialize $W = 1$ (clamping weight)
Step-1	Use SOM to cluster the train set
Step-2	Check if the SOM output is clamped, NO: Increase W by 1 and go to Step-1, Yes: carry on
Step-3	Find (estimate) the customer value (revenue) for each cluster profile
Step-4	Formulate a relationship between retention expenditure and retention success rate for each cluster
Step-5	Identify the two regions of the SOM map and line possibilities: churn & not churn
Step-6	Define two new targets for train set: x and y coordinates of the cluster each data falls between
Step-7	Train an MLP with train set for the two new defined targets
Step-8	Use the trained ANN net to predict where each case in validation set would land
Step-9	Use the line adjustment procedure to find the best dividing line of the two regions of map of decision (MOD) using the information from Step-3 and step-4
Step-10	Use the trained ANN net to predict where each new case would land in the updated MOD and according to the MOD decide for their churn prediction

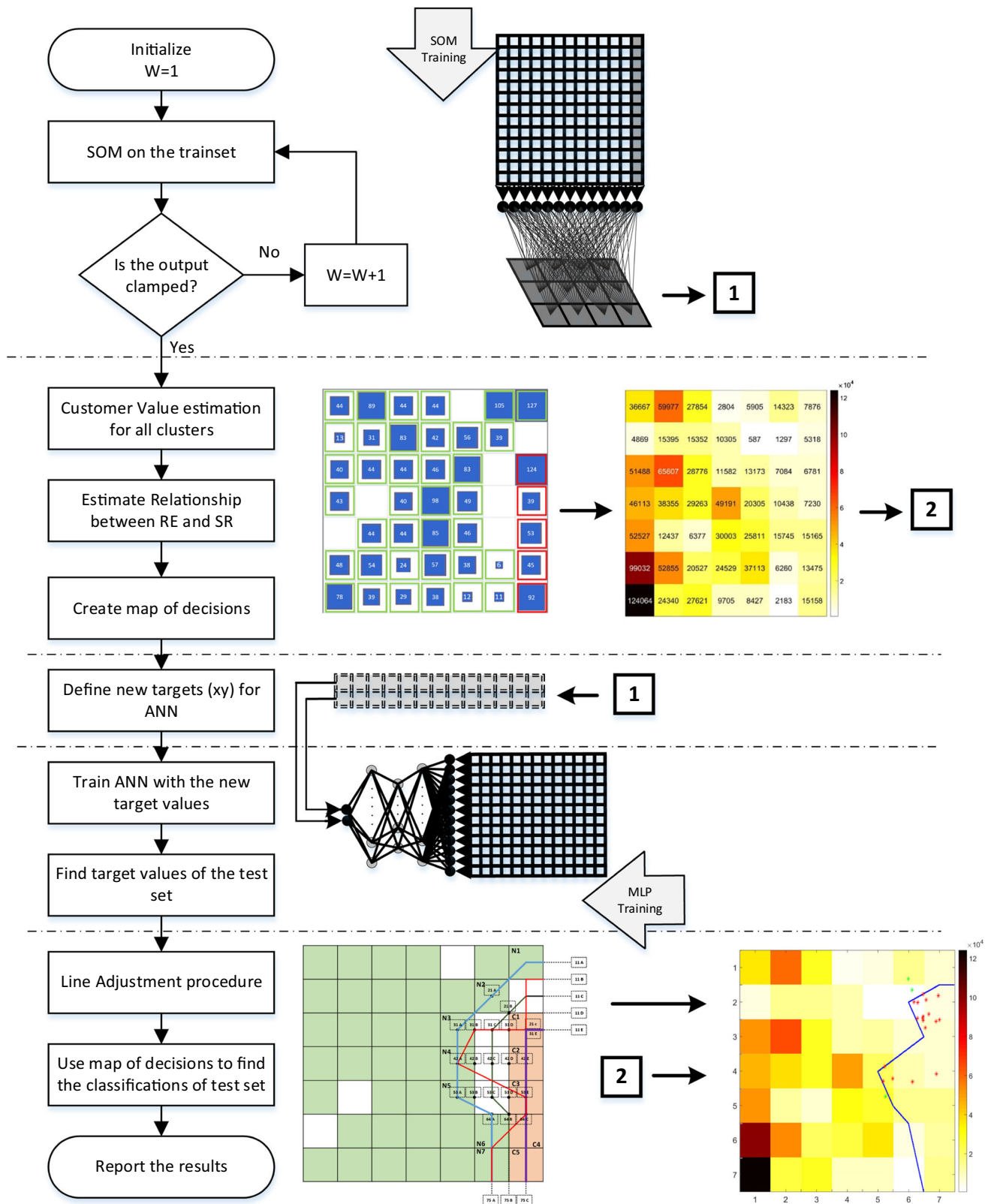


Fig. 3 ChP-SOEDNN Flowchart for optimum profit-driven SOED churn prediction system

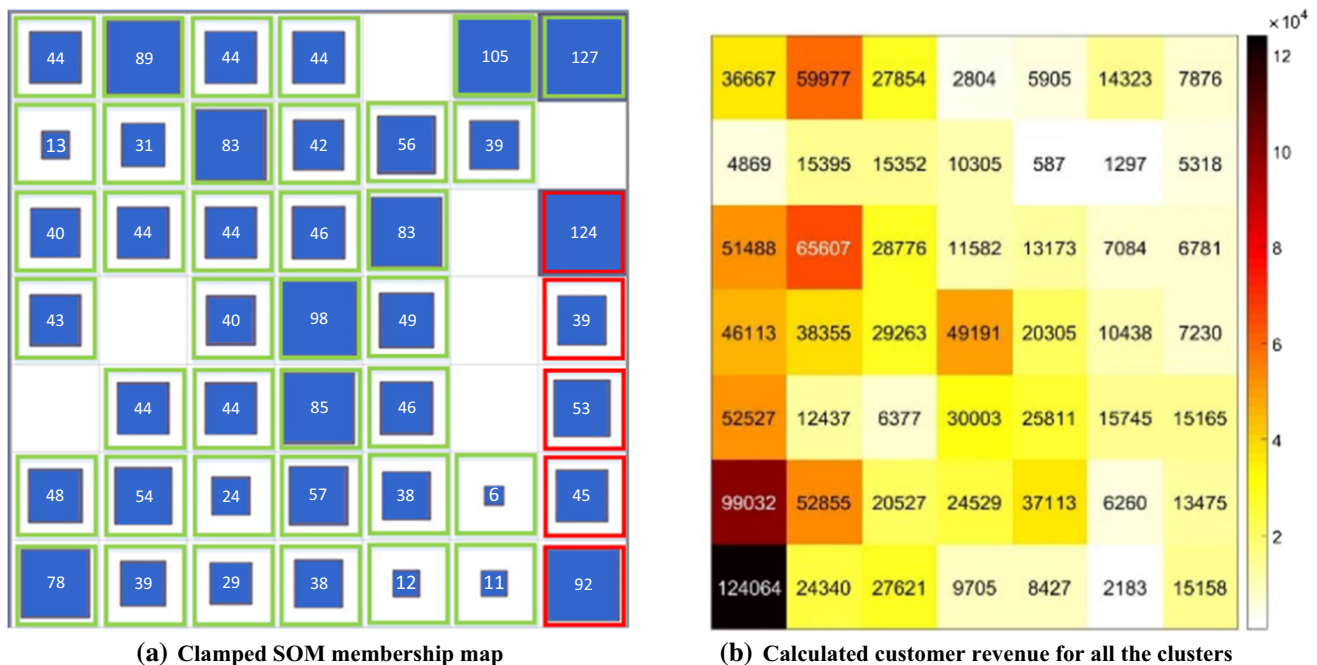


Fig. 4 SOM hit rate and customer revenues for all clusters

4.4 Specific computations and calculations

Figure 4a represents the output from Step-1 of the procedure. It is the SOM hit rate representation that shows the membership count of each neuron in the map. For instance, the neuron at the bottom left of the map has 78 members associated with it. After checking the types of customers residing in each cluster, we observe that in no cluster do both churn and non-churn customers reside. Also, all of the clusters with only churn customers and all of the clusters with only non-churn customers exist on a specific part of the map. The two regions shown in Fig. 4a correspond to 100% to each class in the dataset with the color green for non-churn and red for churn. In other words, Fig. 4a shows a clamped map (Step-2 in Algorithm 1).

4.4.1 Calculate customer value

To calculate customer lifetime value (CLV), one projects the net future cash flows of a given customer over time [7]. While having a well-defined CLV for all the customers of a company is invaluable, its correct calculation heavily hinges on the type of available data [13]. Typically, the time aspect of this calculation is determined, but since the dataset used in this paper does not allow for such calculations, customer lifetime revenue (CLR) substitutes for CLV. It is true that in a given application, CLV is a more precise and preferable metric; however, CLR is still valuable and is relevant for stock price and customer satisfaction [17]. CLR is calculated based on certain variables

from the dataset and pricing estimations informed by online telecommunications pricing. Equation 1 is employed to generate the average monthly revenue of a cluster in the Step-3. Equation 1 contains *FreSMS*, *SecUSE*, *PriSMS*, and *PriUSE* which, respectively, present the average frequency or the number of text messages sent per month, the average seconds of call usage, the price per text message, and the price per second of call usage, respectively.

$$\text{Customer revenue} = \text{FreSMS} \times \text{PriSMS} + \text{SecUSE} \times \text{PriUSE} \quad (1)$$

The time aspect of this calculation may be determined by few different methods [13]: average life of existing accounts as an expected time horizon [44], or an infinite time horizon [14, 18]. In telecom, the possible customer base ranges in age from quite young to past retirement age. Moreover, it is desirable for a company to retain customers as long as possible, barring any great costs somehow incurred. Therefore, an estimated time horizon as the time span over which future cash flows will happen is the average age of the cluster subtracted from 65. Age 65 is still slightly lower than many countries' average life expectancy [42]. Finally, to calculate the average remaining customer lifetime revenue in a given cluster, the monthly customer revenue average of each cluster is transformed into the average annual revenue and then multiplied by the calculated expected lifetime remaining for the cluster. Figure 4b presents the estimated customer revenues for all clusters. One may notice while there are

some clusters in Fig. 4a without members, Fig. 4b, regardless, has customer values for them. Those customer values are an average of the customer values of their non-empty neighbors. For instance, the average of 25,811, 6620, and 15,165 is 15,745 in Fig. 4b (from the bottom third row, from the left second column).

4.4.2 Prediction preparation

Step-5 identifies the different regions of the map. In Figs. 4a and 5a, b, the green squares are the clusters where customers have not churned, and the red squares are clusters of cases which have churned. As one can see in the map of clusters, there is a division: churn and non-churn. There are seven cells in the map that no case is occupying. Three of them are not in the dividing part of the map. The other four are actually the main focus of Step-9 because they give SOED a chance to adjust a dividing line to find the optimum profit-driven classifications.

Step-6 seeks to change the target column for classification of MLP. In a standard binary classification, MLP strives to map the predicting attribute of the dataset with the binary value of the target. However, SOED employs MLP to map the same predicting attributes to a location, as shown in the map in Fig. 5b. The new target columns assume the position of neurons in MOD. For example, for each of three cases at the bottom left neuron in the map, the two target columns are 0.5 and 0.5. Figure 5b depicts all the target columns for other neurons in the map.

Steps-7 and 8 apply MLP to predict the positions of the customers on MOD for the train and validation sets using the predicting attributes. Figure 6a shows the predicted coordination of cases of the validation set. Since the map is clamped, the plot could segregate between the two by filled green circles for non-churn and red stars for churn cases. Also, there is a basic dividing line between the two areas of the map. Based on this line, the misclassifications can be recognized. With this dividing line, the green circles on the

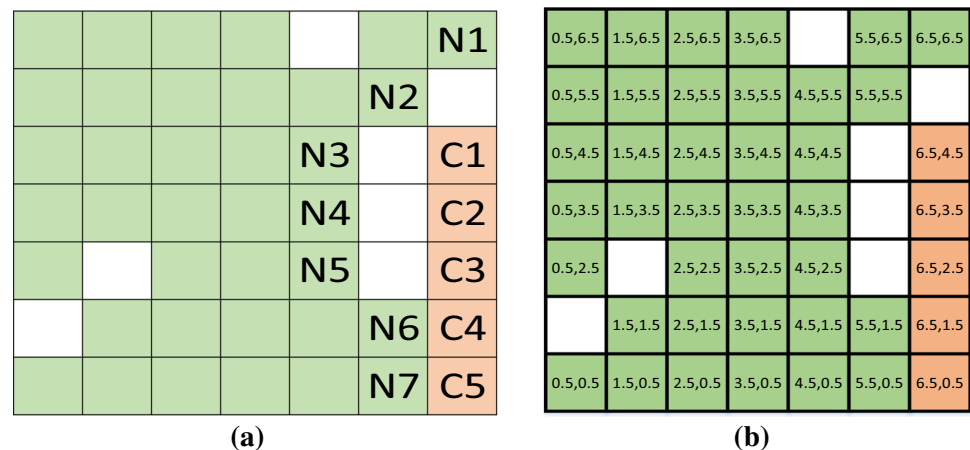
right side of the dividing line are false positives and the red stars before the dividing line are false negatives.

4.4.3 Line adjustment procedure

The application of the basic dividing line is the counterpart of using the same naive cut-off point for churn decision from churn probability prediction [16]. However, the mapping of SOED and the customer value associated with each cluster have created the foundation for an optimum profit-driven churn classification. The outputs of Steps-3 and 4, customer revenue and a relationship between retention expenditure and success rate afford a minimizing line adjustment for misclassification costs based on the cases in the validation set. Step-9 introduces a procedure to that end. Figure 7 shows different possibilities for the adjustment of different lines. The dividing line connects some chosen points that are named with numbers and a letter; the first number of the coding comes from the number with N (non-churn) clusters and the second number comes from C (churn) clusters, and the letter makes a distinction between points in the same situation. There are two distinct types of situations. The first type is when there is a gap between the two clusters that the dividing line needs to separate (Fig. 5a N1–C1, N3–C1, N4–C2, N5–C3). In these situations, five different points are devised starting from least false positive to least false negative. For instance, the dividing line for situation N1–C1 has to start from one of the points from the point 11A to point 11E. The 11C point is the middle ground, not preferring false negative or false positive. The second type of situation is where there is no gap between the two neurons. The same idea applies, here only for three points. These situations are N2–C1, N6–C4, and N7–C5.

The dividing line procedure contributes to a more intelligent and profit-driven prediction because of the mentioned foundation that hybridization of SOM and MLP has created. Two pieces of valuable information about each

Fig. 5 SOM's map coding



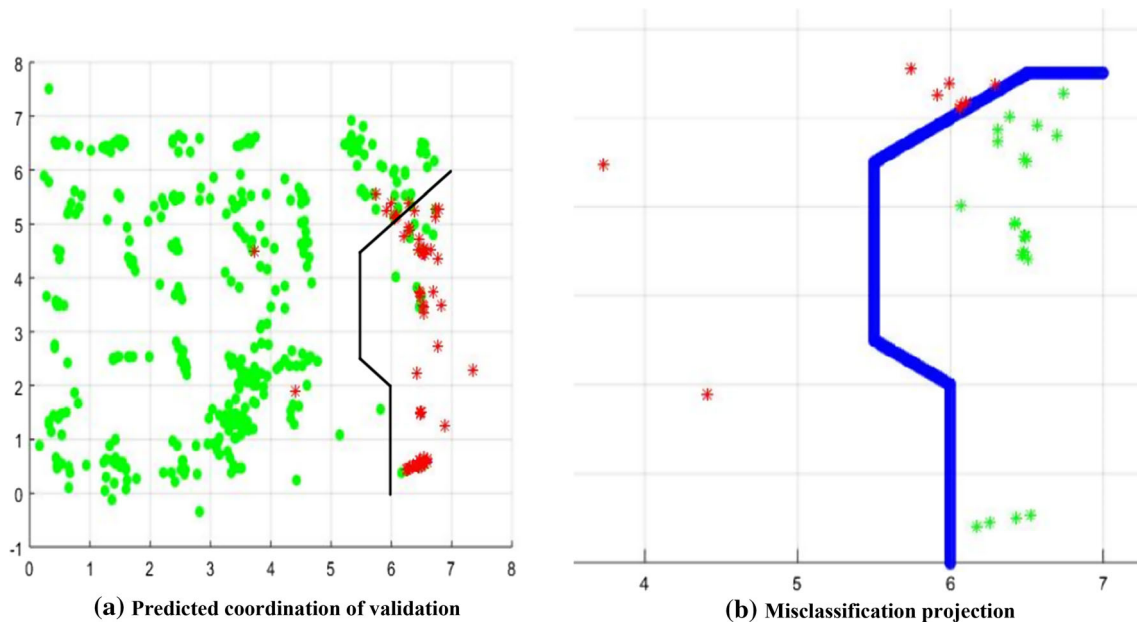
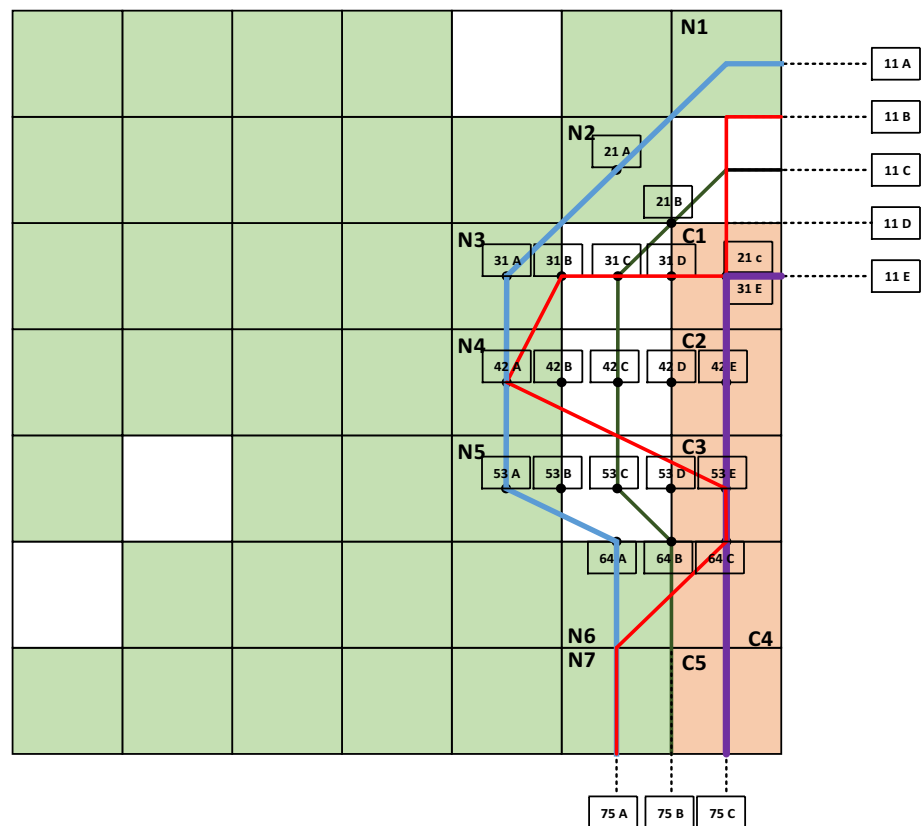


Fig. 6 The predicted coordination of validation set with their known labels (Step-8) and its misclassification projection

Fig. 7 Line adjustment possibilities



customer are accessible to the line adjustment procedure: the value of each customer and the conclusiveness level of MLP's prediction for each customer. The line adjustment procedure uses these two pieces of information to segment

MOD such that the misclassification costs of the classification will be minimum.

A dividing line forms by sequentially connecting to one of the points in the next situation. The same line as presented in Fig. 6a forms by connecting 11C, 21B, 31C, 42C,

53C, 64B, and 75B. This line is also in Fig. 7 with the green color. Since there are seven situations with four of them having five possible points and three having three possible points, there are 16,875 possible dividing lines— $5^4 \times 3^3$. Figure 7, beside the middle ground possible dividing line (green), also shows three of these possibilities. The blue line connecting all the A points in all the situations leads to the least number of false positives. The purple line connecting all the E or C points, on the other hand, if used, minimizes the number of false negatives. Finally, the red dividing line shows the possibility choosing different points and how that could appear.

4.4.4 Misclassification cost calculation

The procedure in Step-9 calculates the misclassification costs of the train and validation set for each of 16,875 line possibilities. The line that leads to the least misclassification cost is the best choice for the final MOD. Using the assumptions, laid out in Sect. 4.2, the misclassification cost for each dividing line alternative is calculated using Eq. 2 or Eq. 3. In these formulas i, j , NFP, NFN, CR $_i$, CR $_j$, RER, RSE, and SR, respectively, represent index for false positives, index for false negatives, number of false-positive cases, number of false-negative cases, customer revenue value of customer i (false positive), customer revenue value of customer j (false negative), retention expenditure ratio, retention steady expenditure, and success rate. The only difference between the two equations is the assumption behind the formula. Using Eq. 2 includes the assumption that the retention expenditure is a fraction of customer revenue for each cluster. On the other hand, Eq. 3 uses a steady retention expenditure which is independent of the customer revenue

$$C = \sum_{i=1}^{N_{FP}} CR_i \times RER + \sum_{j=1}^{N_{FN}} CR_j \times (SR - RER) \quad (2)$$

$$C = \sum_{i=1}^{N_{FP}} RSE + \sum_{j=1}^{N_{FN}} [CR_j \times SR - RSE]. \quad (3)$$

Figure 6b helps clarify the formulas. All the stars shown in the figure are cases that have been misclassified based on the line (one dividing line alternative): Red colors are false negatives and green colors are false positives. The misclassification costs of all the green cases are summed using the first sigma and the second sigma sums the misclassification costs of red cases. The reason that the second sigma of the above formulas is using success rate is that if the false-positive cases had been classified correctly, only the retention of so many of them would have been possible. The procedure recognizes the misclassification cases (false positive or false negative) for every possible dividing line

and calculates the misclassification costs. Supplementary Video 1 shows the coded procedure of Step-9 that recognizes misclassifications for different dividing lines.

4.5 Significance of assumptions

To observe the significance of the proposed method and the influence of assumptions, two sets of experiments investigate the impact of different values for RER, RSE, and SR. Figures 8 and 10a illustrate the result of the combinatory changes in RER and SR. Similarly, Figs. 9 and 10b show the result of experiments with changing of RSE and SR. Across all the experiments, SR may assume four different values: 0.05, 0.1, 0.3, and 0.6. For instance, SR having the value of 0.3 means the retention expenditure has 30% probability of convincing the churning customer to stay. The combination of some of the success rates and retention expenditures is not logical, for example, the combination of having 60% success rate for only spending 1% of customer revenue as retention expenditure. In other words, the cost of buying churning customers back is so low that even the slightest indication of a customer churning leads to retention prevention plans. Nonetheless, many of these combinations are considered only to show the flexibility of the proposed method and different projections of the dividing line based on the different settings.

Dividing lines resembling the blue line in Fig. 7 are indicative of problem settings that the optimum classification setting prefers the least number of false positives (wrong churn prediction). In these cases settings, usually, the misclassification costs of false positives are comparatively high so that the line is adjusted to prevent their occurrence. On the other hand, the dividing lines similar to the purple line in Fig. 7 are indicative of classification settings that false negatives (wrong non-churn prediction) are costlier. The higher misclassification cost of false negatives over that of false positives makes the classifier attempt to avoid them by pulling back the dividing line. Generally, in Figs. 7, 8, and 9, the more pushed-out lines (blue line in Fig. 7) lead to a fewer number of false positives, and the more pulled-back lines (purple line in Fig. 7) result in a fewer number of false negatives.

Keramati et al. [31] have studied these two extreme cases before. They devised ϕ variable manipulating which enables the hybrid methodology to move from fewer false positives (and therefore more false negatives) to fewer false negatives (and therefore more false positives). However, the contribution of this paper is where the method is capable of adjusting the dividing lines based on the different characteristics of each cluster. For instance, looking at Fig. 8 (SR = 0.6), we can see setting retention expenditure rate (RER) to be 0.01 when a success rate of retention strategy is 0.6, which yields a pushed-out

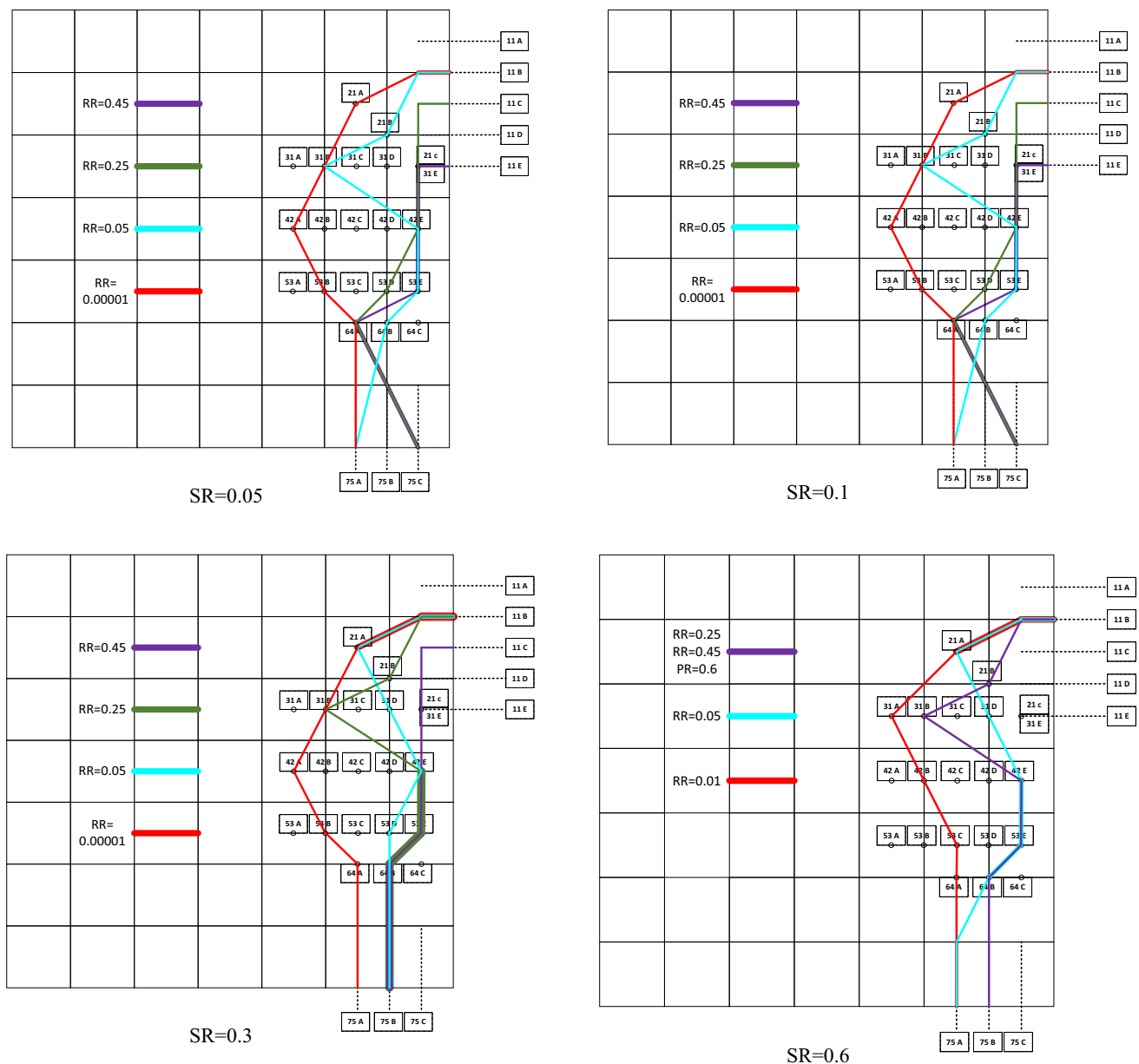


Fig. 8 Experiments with rate-based expenditure assumptions (RER)

dividing line; this is not surprising as retention expenditure is low. One can see in Fig. 8 ($SR = 0.6$), increasing RER to 0.05 makes the dividing line to pull back at certain spots. It is interesting to notice that increasing the rate to values 0.25–0.6 makes the dividing line push out in some spots and pull back for some others.

Figure 9 is related to experiments similar to that of Fig. 8. The difference is the experiments performed for Fig. 9 working under a different assumption—retention expenditures are not calculated based on the clusters (retention steady expenditure—RSE). Nevertheless, the behavior reflected in Fig. 9 has similarity to that of Fig. 8. Figure 9 ($SR = 0.6$) represents that increasing retention

expenditure from \$25 to \$5000 transforms the dividing line from being almost all the way pushed out to nearly all the way pulled back. The middle retention expenditures, although not completely, showed similar behavior [compare Fig. 8 ($SR = 0.6$) with Fig. 9 ($SR = 0.6$)].

Figures 8 and 9 reveal different settings based on the values of SR, RER, and RSE lead to different dividing lines. The projection of these differences is owing to the differences that exist between the customer revenue of each cluster. On the other hand, Fig. 10a, b shows the optimum total cost trend of these different settings. Figure 10a corresponds to Fig. 8, whereas Fig. 10b corresponds to Fig. 9. The trend of the total optimum cost is consistent.

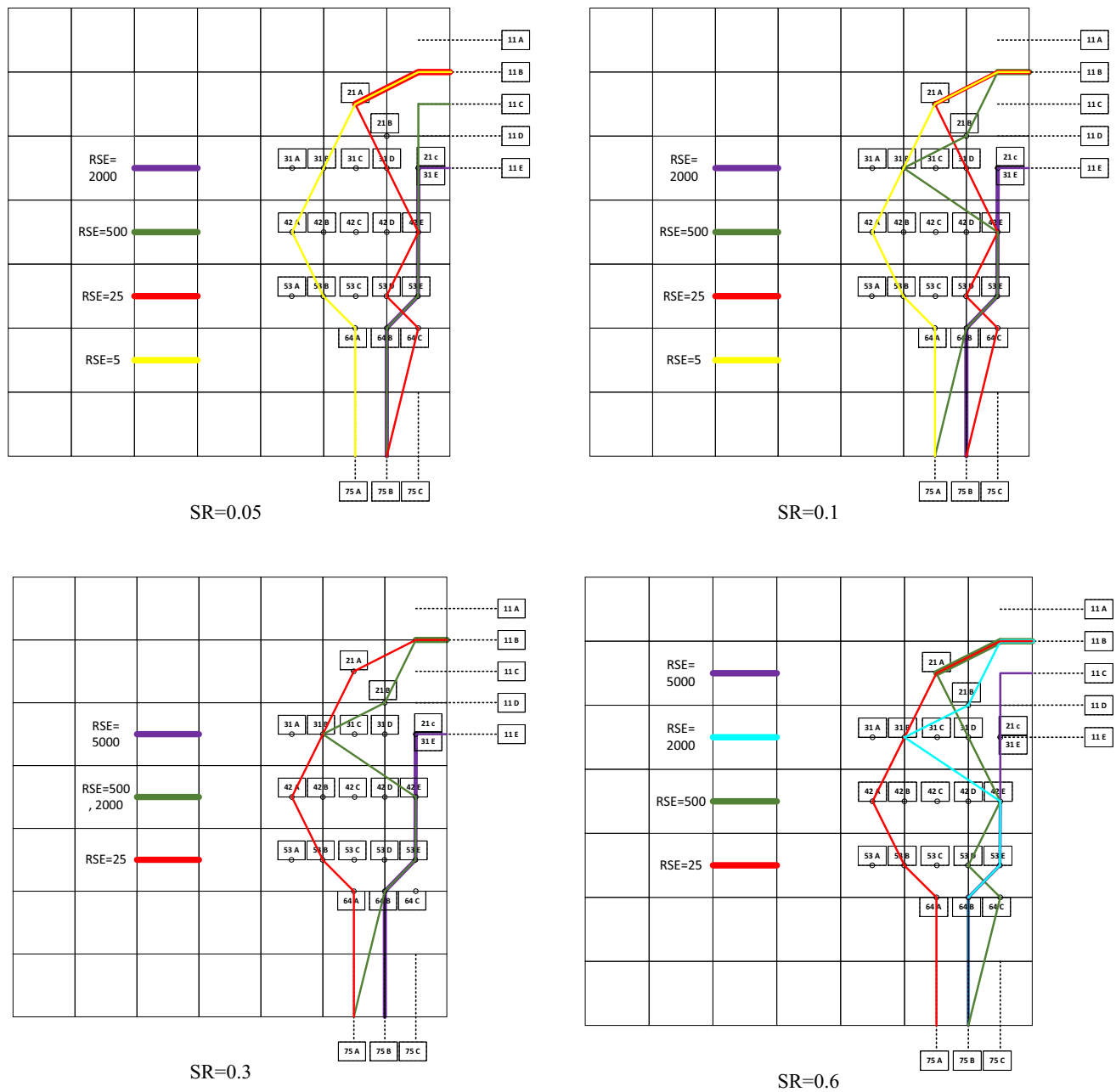


Fig. 9 Experiments with steady-based expenditure assumptions (RSE)

Increasing SR for both parts in Fig. 10 will lead to the rise of the total optimum cost. However, for higher success rates ($SR = 0.6$ and $SR = 0.3$), the slope of the increase as shown in Fig. 10 is greater.

4.6 Comparative analysis

To demonstrate the superiority of the proposals in this paper, they are compared with the recent state-of-the-art cost-sensitive classification techniques and the recent profit-driven churn decision-making efforts. The proposed

method is compared with a variety of different algorithms, namely decision trees (DT), cost-sensitive (CS) DT, CS AdaBoost, MLP, CS MLPs, accuracy-driven (AD) MLP, and F -score-driven (FSD) MLP. The assortment of the selected algorithms contains the recent successful profit-driven churn prediction developments [16, 21] and the most successful cost-sensitive algorithms [53]. Additionally, MLP, one of the most powerful classification algorithms, has been adapted to take advantage of thresholding [47] and resampling strategies [20] to lead to four different MLP-based cost-sensitive classification algorithms: CS

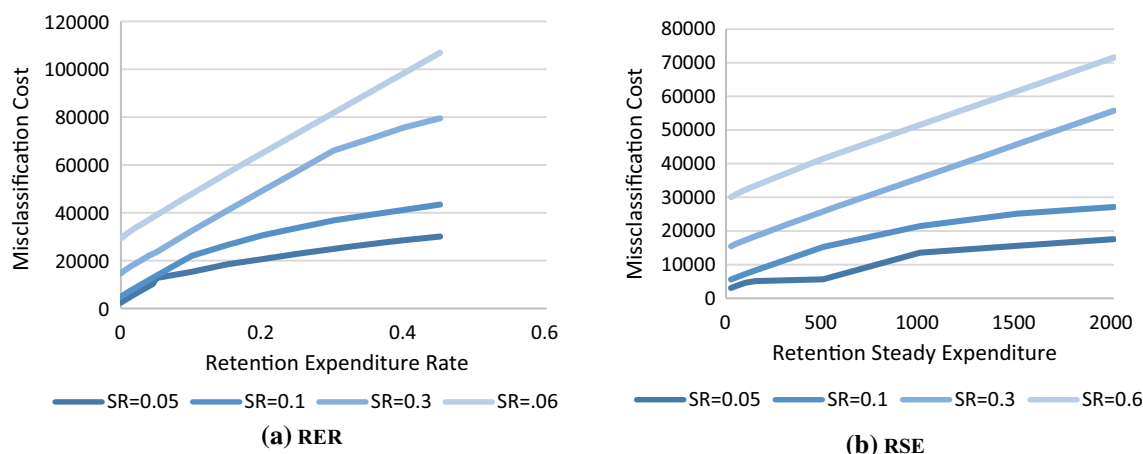


Fig. 10 Total cost variation of experiments with RER and RSE

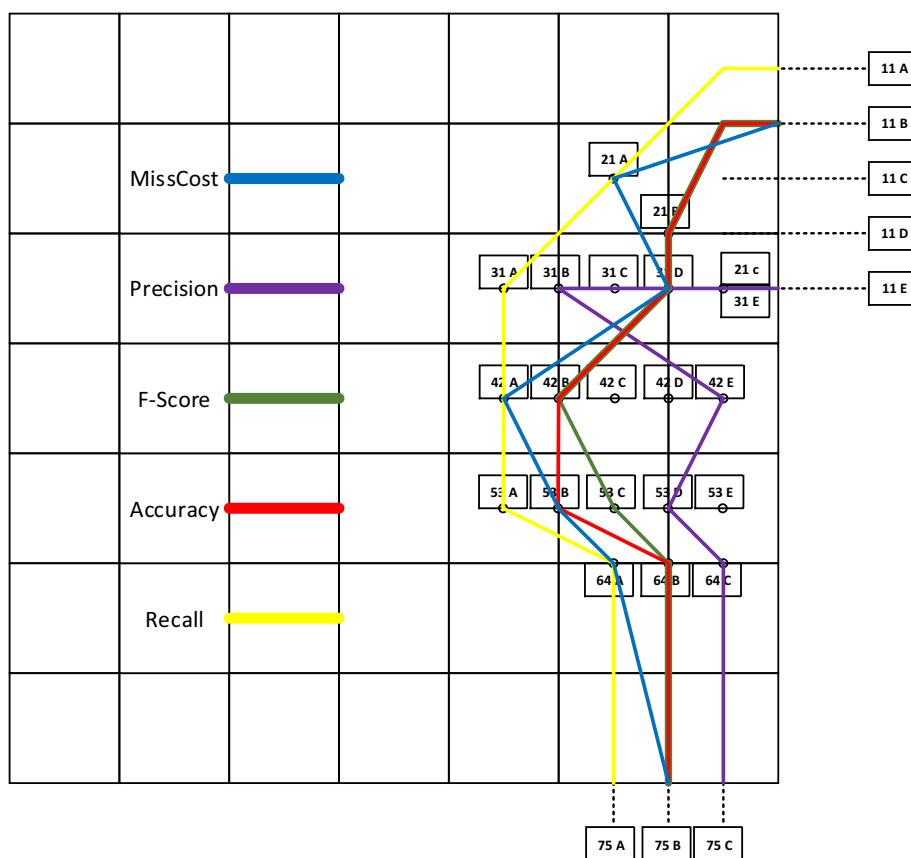
MLP 1–4. The reason for including profit-driven and cost-sensitive approaches among the contending algorithms is that this paper's proposed algorithm contributes to improving profit-driven churn decision making.

4.6.1 Grounds for comparisons

All the methods use the same proportions of the data for training and testing purposes. Train set (70%) and

validation set (15%) of the data are used to train and improve each method. All the methods, without having been exposed to the test set (15%), have predictions about its members. The computed comparison between the predictions and the actual churn occurrence based on the defined evaluation metrics is the performance of each method. There are three performance metrics employed: accuracy, *F*-score, and misclassification cost. Accuracy is a cost-blind, class-imbalance-blind metric. While *F*-score is

Fig. 11 SOED line adjustments based on accuracy, precision, recall, *F*-score, and misclassification costs



a cost-blind metric, it takes the class imbalance nature of churn datasets into account.

If we, respectively, define true positives (TPs), true negatives (TNs), false positives (FPs) and false negatives (FNs) as the number of times each case happens for a test set after prediction, precision, recall and F -score is defined as:

$$\text{Precision} = \frac{\text{TPs}}{\text{TPs} + \text{FPs}} \quad (4)$$

$$\text{Recall} = \frac{\text{TPs}}{\text{TPs} + \text{FNs}} \quad (5)$$

$$F\text{-score} = \frac{\text{precision} \times \text{recall} \times 2}{\text{precision} + \text{recall}}. \quad (6)$$

In the context of benefit-driven classification, comparisons require more assumptions. For instance, Glady et al. [16] assume a loss function which accumulates the costs for the occurrence of the two types of decision-making errors: false positive and false negative. The defined loss function is kept constant for all their experiments, thus giving the study a common ground for comparison. In this study, SOM output is the necessary assumption after being exposed to the train set and validations set. Since every single customer in all three sets (train, validation, and test sets) can have a cluster membership on the SOM map, the calculated customer value for each cluster is assumed to be the customer values of all the members. Training of SOM uses both train and validation sets. However, the assumption is that the cluster membership of the test set does not significantly change if the test set is also involved in the training of the SOM. “Appendix 1” section presents a hypothesis test for the relevance of this assumption.

Moreover, it is important to know a relationship between SR and RER or RSE for each cluster. We suspect that such estimated relationship would be different from one cluster to another. However, for experimentation, it is assumed that companies only can adopt 1% for RER and that leads to 30% customer retention success rate (SR). In reality, RER can be any percentage that a company wants to choose. However, consistency of this assumption throughout the study provides another ground for comparison.

4.6.2 Methods tuning and parameter selection

This section introduces the experimental settings of all the applied methods: multilayered perceptron (MLP), decision tree (DT), AdaCost, self-organizing error-driven ANN (SOED), and ChP-SOEDNN. There are common concepts used in the training of all classifiers: cost-sensitive (CS), accuracy driven (AD), and F -score driven (FSD). For instance, CS decision tree suggests that decision tree has been adapted to become cost-sensitive. Accuracy driven (AD) signifies that the method quests to perform the best regarding accuracy. Likewise, F -score driven (FSD) specifies that the method outputs trained classifiers with high F -score performance. It is noteworthy that even though only in the training or validation process of the classifiers which are cost-sensitive (CS) the calculated customer revenue of each member is used, for all classifiers the calculated customer revenue is utilized in the calculation of the performance measure cost in Table 2.

Except for ChP-SOEDNN that cost sensitivity and profit drive are imparted using the line adjustment procedure, other cost-sensitive experiments, such as decision tree, MLP, or AdaCost, may have used one of the renowned

Table 2 Comparison between MLP, DT, AdaBoost, and SOED

	Best cost (K\$)	Average cost (K\$)	Best accuracy (%)	Average accuracy (%)	Best F -score (%)	Average F -score (%)
DT	106.23	106.23	94.49	94.49	81.16	81.16
CS DT	23.01	56.75	91.95	86.29	77.65	66.93
CS AdaBoost	24.25	109.65	92.80	82.03	77.92	60.85
MLP	39.71	91.40	97.67	96.42	91.97	87.39
AD MLP	41.54	102.58	<u>98.09</u>	96.29	93.43	87.34
FSD MLP	39.71	93.08	97.67	96.18	93.53	88.62
CS MLP 1	31.44	75.55	98.31	96.24	94.20	88.57
CS MLP 2	3.10	32.78	97.25	89.52	91.50	74.83
CS MLP 3	<u>9.05</u>	26.34	97.46	88.36	91.78	74.26
CS MLP 4	10.72	38.51	95.55	82.57	86.62	65.53
SOED	34.23	34.23	95.13	95.13	82.96	82.96
AD SOED	20.01	<u>20.01</u>	97.03	97.03	90.00	90.00
ChP-SOEDNN	13.75	13.75	95.13	95.13	85.71	85.71

Table 3 Method specific training for best of their accuracy, *F*-score, and cost

	Network	Resampling ratio	Threshold	Cost ratio	Line adjustment
<i>MLP</i>					
Accuracy	[5, 10, 12]*	—	—	—	—
<i>F</i> -score	[5, 10, 12]*	—	—	—	—
Cost	[5, 11]	—	—	—	—
<i>AD MLP</i>					
Accuracy	[14, 16]	—	—	—	—
<i>F</i> -score	[6, 14]*	—	—	—	—
Cost	[6, 14]*	—	—	—	—
<i>FSD MLP</i>					
Accuracy	[4, 6, 9, 10, 16]	—	—	—	—
<i>F</i> -score	[11, 14]*	—	—	—	—
Cost	[11, 14]*	—	—	—	—
<i>SOED</i>					
Accuracy	[13, 14, 16]	—	—	—	[2, 3] Fig. 11: green line
<i>F</i> -score					
Cost					
<i>AD SOED</i>					
Accuracy	[13, 14, 16]	—	—	—	[2–4] Fig. 11: red line
<i>F</i> -score					
Cost					
<i>CS DT</i>					
Accuracy	—	2	0.2	—	—
<i>F</i> -score	—	2	0.2	—	—
Cost	—	7	— 0.05	—	—
<i>CS AdaBoost</i>					
Accuracy	—	1	—	4	—
<i>F</i> -score	—	1	—	6	—
Cost	—	9	—	1	—
<i>CS MLP 1</i>					
Accuracy	[5, 9, 10]*	—	—	—	—
<i>F</i> -score	[5, 9, 10]*	—	—	—	—
Cost	[5, 9, 10]*	—	—	—	—
<i>CS MLP 2</i>					
Accuracy	[3, 6, 9]*	2*	—	—	—
<i>F</i> -score	[8, 9, 12]	1	—	—	—
Cost	[3, 6, 9]*	2*	—	—	—
<i>CS MLP 3</i>					
Accuracy	[2, 13, 15]*	—	— 0.55*	—	—
<i>F</i> -score	[2, 13, 15]*	—	— 0.55*	—	—
Cost	[5, 13]	—	— 0.85	—	—
<i>CS MLP 4</i>					
Accuracy	[1, 9, 11]*	1*	— 0.2*	—	—
<i>F</i> -score	[1, 9, 11]*	1*	— 0.2*	—	—
Cost	[3, 4]	1	0.35	—	—
<i>OCS SOED</i>					
Accuracy	[13, 14, 16]	—	—	—	[2–5] Fig. 11: blue line
<i>F</i> -score					
Cost					

cost-sensitive strategies: Resampling ratio (RS) and thresholding (T). Table 3 presents the specific tuning of RS and T related for each method to reach the best accuracy, F -score, and cost. The asterisk sign (*) in Table 3 indicates that the best tuning under more than one evaluation metric has been the same. Equation 7 shows the definition of RS. N_{Churns} and $N_{\text{NonChurns}}$, respectively, stand for the number of customers that are labeled churn and the number of customers that are labeled non-churn in a resampled train set. Equation 8 presents how the churn decision is made based on a defined threshold (T). Here, predict_value is the value predicted by a regression-based predictor. In short, Eq. 8 is how the regression predictors are transformed to churn binary classifiers

$$\text{RS} = \frac{N_{\text{Churns}}}{N_{\text{NonChurns}}} \quad (7)$$

$$\begin{cases} \text{Churn} & \text{predict_value} \geq T \\ \text{NonChurn} & \text{predict_value} < T \end{cases} \quad (8)$$

4.6.3 MLP

The learning force of MLP originates from the weight changes between neurons due to the backpropagation of the errors. MLP is trained to output a churn expectation based on the input values of each customer. The difference between the expectation and the churn reality is the error. The churn reality either happens (1) or does not happen (-1). However, MLP outputs a value that is calculated solely based on the existing weights. In other words, MLP is not accuracy or cost driven but attempts to bring the output value of the network closer to the 1 or -1 of the churn reality. That is the reason behind the threshold definition in Eq. 7, so MLP can distinguish between distinct classes. Naively, in many MLP applications for binary classifications, where the classes are denoted by -1 and 1, zero is considered for T . However, the process of validation and power of randomization are used to give MLP more focused learning forces at the validation level. These learning forces are named accuracy driven (AD) and F -score driven (FSD).

On the other hand, CS MLP 1 to CS MLP 4 are cost-sensitive MLPs that at the level of validation use the calculated customer revenues; CLR is used to derive the misclassification cost of the trained network. In the validation process, CS MLP 1 will choose the network that outputs the smallest misclassification costs. CS MLP 2 assigns, at the same time, as randomly initiating a two- or three-layered network; a random resampling ratio (RS) between 0.1 and 10. RS can randomly take values 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. CS MLP 3, instead of a random RS, assigns an arbitrary

threshold (T). T can randomly take any value between -1 and 1 with 0.05 increments, for instance, -0.75 or 0.3 . CS MLP 4 is the random combination of CS MLP 2 and CS MLP 3 where each experiment is randomly assigned a two- or three-layered network, a resampling ratio (RS) and threshold (T).

For all the presented MLPs in Table 2, each row of data represents 1000 experiments. The best and average columns are over 20 best selected networks from a 50-run validation process. In each validation process, 50 random networks learn to show their performance when predicting the validation set. All the networks use Levenberg–Marquardt backpropagation and mean squared error and six validation checks. However, at each 50 network initiation, a random two- or three-layered network is assigned. The network that has performed the best among the 50 networks based on either one of the learning forces (AD, FSD) will learn using both train and validation sets with 500 validation checks. Setting the validation check of an ANN high gives confidence that the network reaches its fullest potential. The selected retrained network is used for churn prediction of the test set, and Table 2 reports the average and the best of all performance measures for all 20 best selected networks. Also, “Appendix 3” section presents the performance of all 20 best selected networks.

4.6.4 SOED

Table 3 includes three different rows related to the performance of SOED. SOED and AD SOED are not profit driven, i.e., CRV in the process of training, validation, or line adjustment is not used. While SOED is adopted from Jafari-Marandi et al. [29], AD SOED is the alteration of SOED where the line adjustment procedure has found the optimum line by which the best accuracy occurs. Using the red line in Fig. 11 instead of using the middle ground dividing line (the green line in Fig. 7) is the key difference between using a standard SOED and the AD SOED. Similarly, the blue line in Fig. 11 is employed to become profit driven (ChP-SOEDNN). The inclusion of CRV calculations and assumptions for the relationship between retention plan expenditure and customer retention success make it possible to adjust the line for minimum misclassification cost. In other words, ChP-SOEDNN uses the line adjustment procedure to find the line that leads to the least misclassification cost possible among all the presented line assortments.

4.6.5 Decision trees

Decision trees (DTs) are the most popular and transparent supervised learning techniques. Keramati et al. [31] reported 14 different decision tree induction methods in

three different software packages. After applying all of them on the same dataset as this paper, it was reported that MATLAB's decision tree is only second best to the random forest induction algorithm. However, under *F*-score metric MATLAB proved to be more stable than random forest.

This paper adopts two of MATLAB R2017a decision tree functions: `fitctree` and `fitrtree`. Function `fitctree` is a classification-based decision tree, and its branches only conclude in either one of the defined classes. On the other hand, function `fitrtree` is a regression-based decision tree with branches that have numeric values. DT row in Table 2 shows the result of the classification decision tree (`fitctree`) after being trained using train and validation sets. Since decision tree is not a random process, unlike many of Table 2's rows, DT is only representing one experiment.

CS DT employs the regression-based function of DT (`fitrtree`) in conjunction with a validation process tuning of resampling ratio and thresholding. Similar to other cost-sensitive rows in Table 2, the row of CS TD represents 1000 different experiments. The bests and averages are over 20 selected combinations of resampling ratio and adjusted thresholds from a 50-run validation process. Similar to CS MLPs, each validation process randomly assigns resampling ratio and thresholds and employs only the train set to train the decision tree by `fitrtree` function (regression-based decision tree) to predict the class of the validation set. The combination of resampling ratio and threshold that leads to the lowest misclassification cost of the validation set will be selected in the prediction of the test set using both the train and validation sets. The calculated performance of the test set will be one of the 20 experiments that CS TD represents ("Appendix 3" section).

4.6.6 Cost-sensitive AdaBoost

CS AdaBoost is an ensemble-based cost-sensitive algorithm where boosting a weak classifier turns into a strong one. The predefined cost of the two different types of misclassifications will lead the weight changes toward class predictions with less misclassification cost [49]. Therefore, for these types of classifications to be applied to churn prediction, there is a need for a misclassification cost ratio [9]. Equation 9 presents this ratio: the misclassification cost of a false negative, when a customer is wrongly predicted to churn, over the misclassification cost of a false positive, when a customer is wrongly predicted as non-churn. Since the cost of acquiring customers is usually higher than retaining them, we would expect a cost ratio greater than one to yield less misclassification cost

$$\text{Cost ratio} = \frac{\text{misclassification cost of false negative}}{\text{misclassification cost of false positive}} \quad (9)$$

The validation procedure adjusts the cost ratio along with a resampling ratio. Similar to CS MLPs and CS decision tree, CS AdaBoost in Table 2 represents 1000 experiments. The bests and averages are over 20 selected combinations of resampling ratio and cost ratio from a 50-run validation process. We utilized MATLAB R2017a `fitensemble` function for all the validation predictions and test predictions. The function is set with 50 as the number of boostings, decision trees as the weak learner, and classification as the task of learning with the cost ratio definition. "Appendix 3" section shows all of these validation runs.

5 Discussion and future research

The paper works under the assumption that misclassifications are a part of churn prediction. Our approach accepts their existence while trying to minimize their occurrence. Furthermore, the procedure employs the capability of SOED to incorporate other types of information such as customer revenue and costs of misclassification to work with the possibility of misclassifications instead of solely working against them. As such, the contribution of the paper is finding the best balance for misclassification costs of false positive and false negative for each cluster of customers. Figures 8 and 9 illustrate examples of these balances for different assumption settings. In different settings of RER, RSE, and SR, we can see how the dividing line between clusters might change with the results of the new balance between the two types of misclassification costs. Figure 12 portrays the significance of this contribution. The only difference the OCS SOED between AD SEOD is the fact that OCS SOED has used the blue line in Fig. 11 instead of the red line and that has led to around \$8000 cost savings.

In what follows, we will discuss the results shown in Tables 2 and 3 from different perspectives. In short, profit-driven ChP-SOEDNN shows significant superiority over all the cost-sensitive methods.

5.1 Accuracy-driven churn decision making

Keramati et al. [31] have shown statistically that ANN significantly outperformed decision tree, KNN, and SVM in dealing with the accuracy-driven classification task of this paper. In this study, the improved version of ANN which is self-organizing error-driven ANN [29] is also applied to show that it is a more accurate and consistent method. Table 2 shows while the best of AD MLP is higher than the best performance of AD SOED, its average could not stretch to the AD SOED's performance. AD SOED on

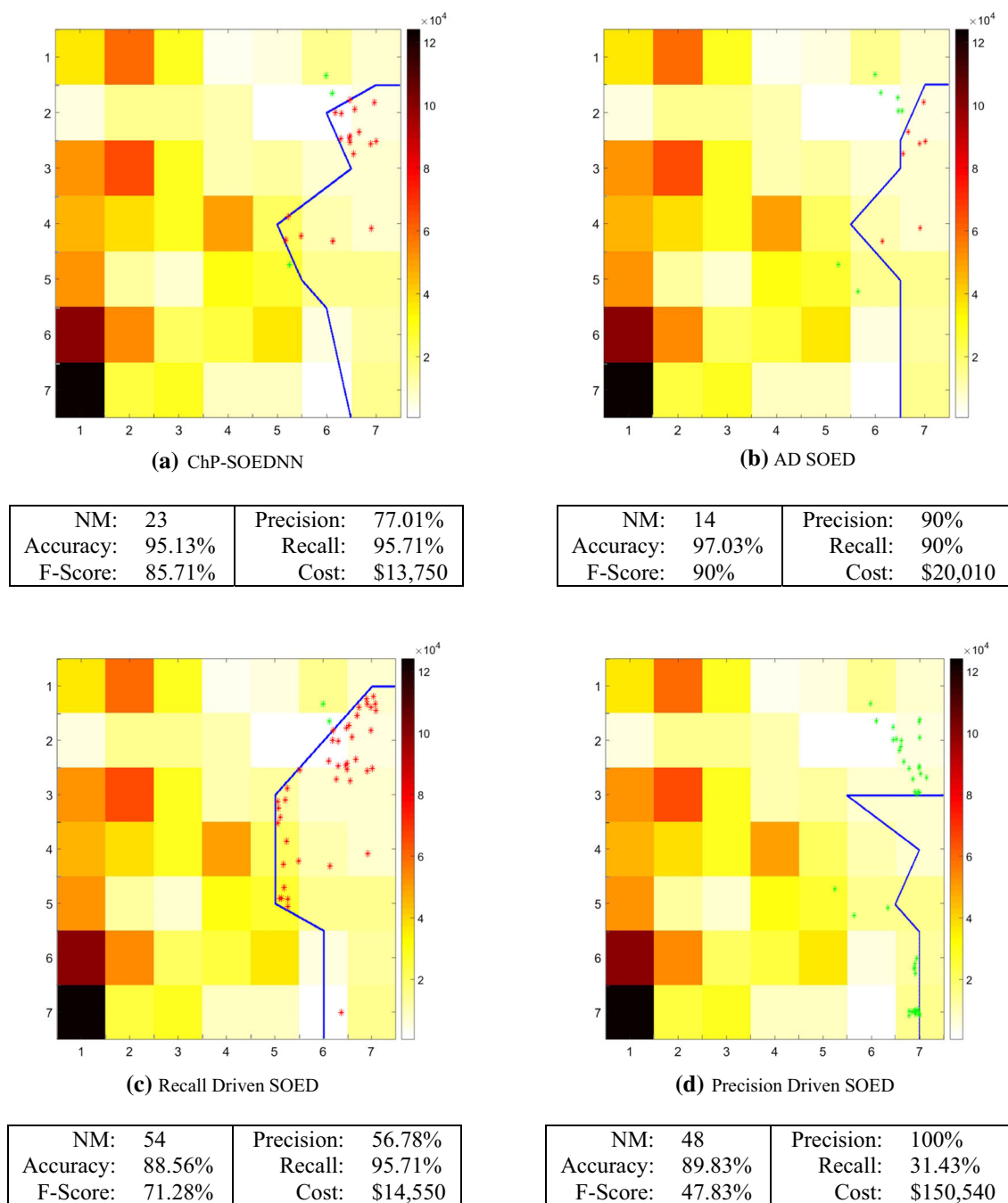


Fig. 12 Misclassification depiction of SOED based on different line adjustment strategies (Fig. 11)

average performs 0.72% better in the classification of 472 customers in the test set, i.e., AD SOED has four fewer churn misclassifications in comparison with MLP. The average of AD MLP is more important because, in reality, we are not able to know which one of the trained networks in the set of 20 will be performing the best. Besides that, the one best performance of AD MLP may have been a fluke hit. Although this shows that artificial neural network may still have room for improvements regarding creating a more accurate and reliable predictor for churn

classifications, the proven consistency of SOED at this level of accuracy is more desirable.

5.2 Profit-driven churn decision making

Table 2 validates recent profit-driven churn decision-making efforts [6, 16, 54]. Cost-sensitive methods such as CS DT and CS AdaBoost outperform the best classification techniques (MLP), when the evaluation metric is misclassification cost. Based on our experience on the same data

presented in [31], we show MLP outperforms DT, SVM, and KNN regarding accuracy, precision, and recall measures. However, Table 2 shows that CS DT and CS AdaBoost perform better than cost-insensitive methods (DT, MLP, and SOED) when evaluated with misclassification cost. Furthermore, we have learned that the effort to create a profit-driven ANN has improved churn decision making. Based on the decisions made for the test set which includes only 472 customers, the best CS MLP 2 outperforms the best of CS DT and CS AdaBoost by approximately \$20,000 and on average ChP-SOEDNN outperforms CS AdaBoost approximately by \$96,000 and CS DT by \$43,000.

5.3 SOED line adjustment

Figure 12 illustrates another representation of Fig. 11. Figure 12 depicts the misclassification occurrences of different SOED line adjustment strategies. For instance, Fig. 12a shows a selected dividing line which is adjusted to drive the least misclassification costs using the customers in the validation set. For all parts of Fig. 12, the chosen dividing line and the misclassification occurrence are presented using MOD. Also, each cluster on MOD uses a color to show the value of its customers. Figure 12 a–d shows the selected dividing line when the line adjustment procedure has been tuned to move toward, respectively, lowest misclassification cost, highest accuracy, highest recall, and highest precision. In general, Fig. 12 expresses, first, that the line adjustment is powerful and effective as by moving from different extremes to the others, the value of performance metrics changes significantly; misclassification costs changes by \$137,000, accuracy 8%, recall 64%, and precision 44%. Second, comparing Fig. 12 a (cost-driven adjustment) and c (recall-driven adjustment), we can infer that profit-driven classification is more than just preferring one type of error over the other. The comparison shows, based on the value of the customer revenue, sometimes a false negative (wrongly predicted as non-churn) should happen so to prevent some false positives (wrongly predicted as churn). Lastly, the similarity and difference between parts a (cost-driven adjustment) and b (accuracy-driven adjustment) of Fig. 12 summarize the similarity and

the differences between classical churn prediction and recent profit-driven churn decision making. In Fig. 12b, the classifier is adjusted to have the fewest number of misclassifications possible. In Fig. 12a, the constraint of fewer number of misclassifications is relaxed to let more misclassifications happen, and a profit-driven constraint is imposed on the classifier to move toward less misclassification cost. The change in training focus has led to 9 more misclassifications but around \$6000 in savings.

It is a famously known that efforts to improve precision will lead to the loss of recall, and vice versa [46]. In short, this paper introduces a method that finds the right balance that is informed by decision-making goals, i.e., more profit. In previous research on the same dataset [31], a hybrid method could achieve 97.14% and 98.56%, respectively, for recall and precision. The hybrid methodology uses a control variable in conjunction with four different classification methods: decision tree, KNN, SVM, and ANN. The control variable can be adjusted to lead the hybrid classifier for the best precision or best recall. Table 4 compares the performance of SOED that uses line adjustment for the two extremes—precision and recall—with the performance of a previous study [31]. The values present in Table 4 for best hybrid methodology are the average over the five experiments. The comparison shows that SOED with line adjustment has been able to perform better for precision. Also for recall, SOED can result in 100% accuracy if it were not for the limitation of line adjustments possibilities. That is to say, if the line adjustment procedure gives more possibility of movements, the dividing line in Fig. 12c could be more pushed out so the two green misclassifications would not happen. It is noteworthy that it would significantly decrease the *F*-score.

While SOED shows better control over reaching desired recall or precision, the hybrid methodology [31] shows better maintenance of one while maximizing the other. This can be seen by comparing the best of the recall-driven classification by both methods. While the hybrid method [31] leads to the lowest of 78.20% while maximizing recall, SOED drops to 56.78% for the same effort. This shows if achieving the best on one end of the spectrum (maximizing recall or precision is the goal of a decision maker) a hybrid methodology will lead to a more reliable classification. At the same time, if the decision maker is interested in finding the right point between the extremes of the spectrum that leads to the best possible set of decisions, we recommend using the presented method in this paper. Moreover, this highlights another possible direction for future research: developing a classification method that allows for the intelligent and profit-driven adjustment of balance between recall and precision while keeping the explained maintenance property of the hybrid method.

Table 4 Precision and recall performance of SOED with the proposed hybrid methodology

	Recall (%)	Precision (%)
<i>Precision driven</i>		
Best hybrid methodology [31]	32.66	98.56
SOED	31.43	100
<i>Recall driven</i>		
Best hybrid methodology [31]	97.14	78.20
SOED	95.71	56.78

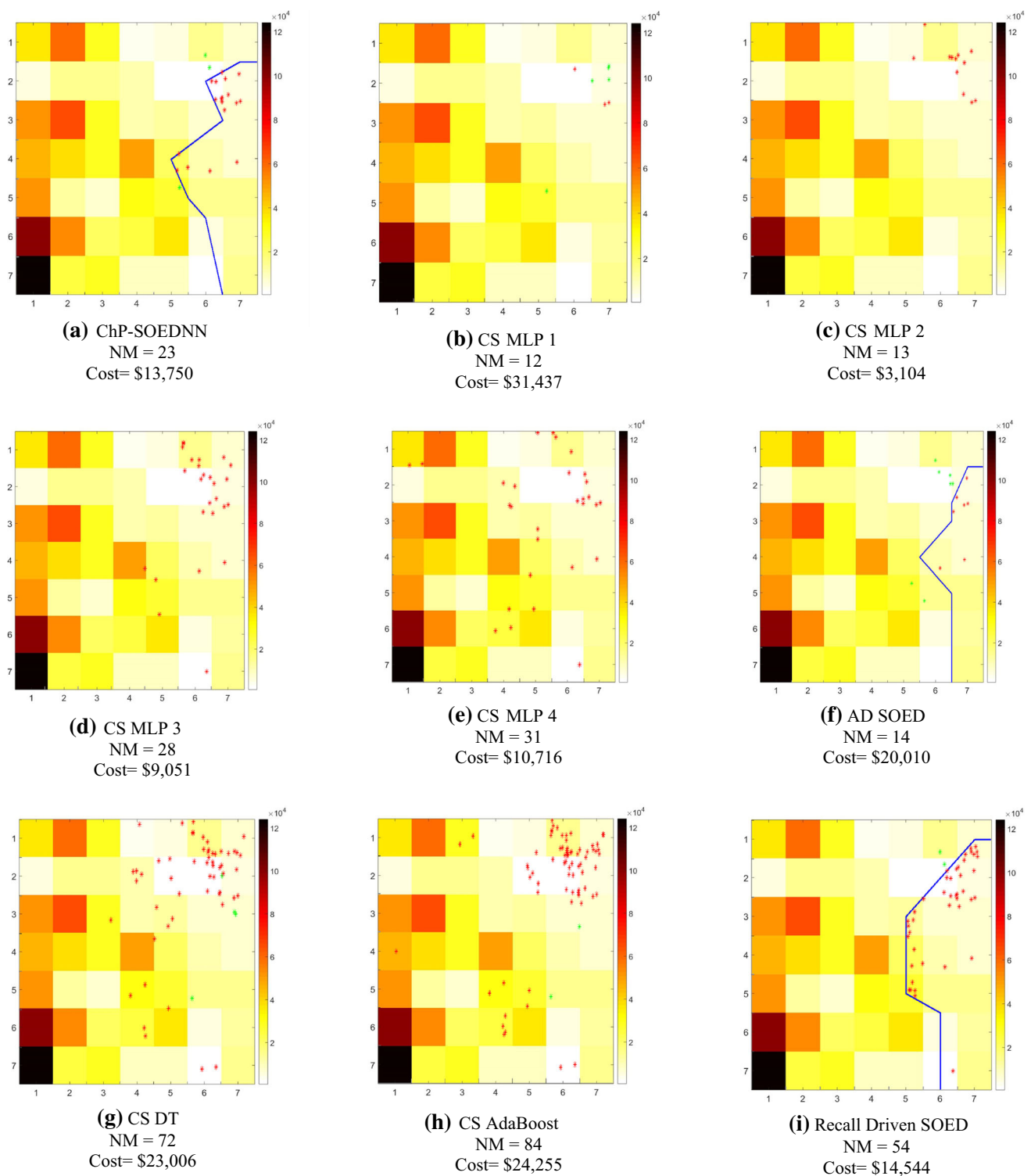


Fig. 13 Misclassification depiction of cost-sensitive methods using MOD of SOED

5.4 Profit-driven classifications

Figure 13 illustrates the misclassifications of all the cost-sensitive methods in Table 2. The misclassifications are depicted using MOD of SOED. The color of the clusters,

ranging from white to black, represents the estimated customer revenue for the cluster; colors red and black show higher customer values as opposed to colors white and yellow which show lower customer revenues. Figure 13 fully captures the importance and the essence of cost-

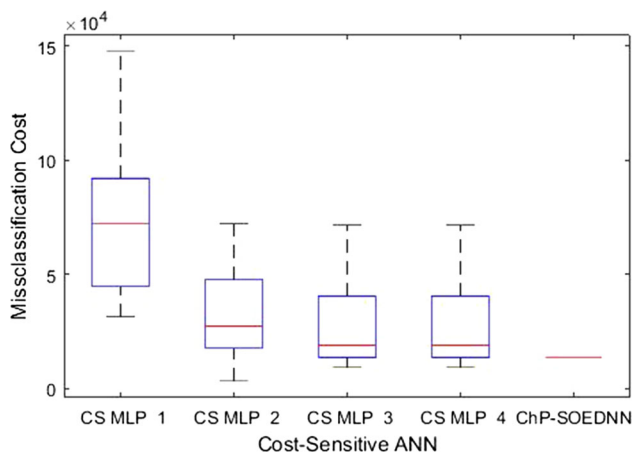


Fig. 14 Boxplot for cost-sensitive SOEDs

sensitive classifications. First, Fig. 13 reveals the number of misclassifications is not a realistic measure to evaluate classification decisions. For instance, even though CS MLP 2 has one more misclassification than CS MLP 1, CS MLP 2 has a significantly lower misclassification cost: approximately \$28,000. Furthermore, we can see that all the cost-sensitive methods tend to avoid false negatives (green stars: wrongly predicted non-churn) at the expense of allowing more false positives (red stars: wrongly predicted churn) to happen. While this logically corroborates with the fact that losing a valuable customer because their leaving was not predicted is more expensive than spending unnecessary customer retention expenditures on a customer that would not leave. The real reason is the assumption of 1% and 30% for RER and SR, respectively. In another level of logic, the real reason connects to the type of cost-sensitive strategies used in the training of classifiers. For instance, when a resampling of the train set gives more room for churn cases, the classifier will consequently prefer red misclassifications. Also, Fig. 13 ultimately demonstrates the misclassification cost of churn decision making is dependent of more than just the type of errors (false positive or false negative) but the type of customers about whom decision-making errors happen.

5.5 Cost-sensitive strategies

Three different strategies at the validation level are employed to impart cost sensitivity to SOED and MLP. Figure 14 displays a boxplot to compare the effectiveness of all these strategies. The lowest value of each method and the red line in the plot, respectively, indicate the best and average performance of each method (Table 2). CS MLP 2, which uses the resampling paradigm, has led to achieving the best possible performance of SOED. However, its average performance and its variation, which point to its

reliability, are lower than CS MLP 3 and CS MLP 4. CS MLP 3 employs the thresholding paradigm for cost sensitivity, and this suggests that while the thresholding is not as effective as resampling, it exercises better control over the reliability of the decision making. CS MLP 4, in which both resampling and thresholding paradigms are engaged, confirms the previous interpretation as CS MLP 4 shows more reliability in comparison with CSMLP 2 at the expense of losing some of its best performance virtue.

5.6 SOED's resilience against imbalanced classification tasks

One important observation from Table 2 is that normal SOED and AD SOED, which do not use the customer revenues of each cluster in their training, still perform better than the average of cost-sensitive methods (except for ChP-SOEDNN) which do use those values in their training. We suspect the reason for this reliability of SOED is the change in classification to the prediction of position on the map. As known in the literature, one of the challenges of churn classification is the class imbalance nature of churn datasets [59]. The challenge lies in the possible bias of classifiers because of churn datasets tending to have more non-churn cases than churn samples. Regardless of the nature of the dataset, SOED will never be biased because of the ratio of classes in any dataset. SOED ultimately predicts the location of each customer on MOD. As the coordinates of all customers on the map are spread out, the predictor cannot be biased based on the ratio of the classes. One may dedicate a study to proving this hypothesis using multiple imbalance churn datasets.

5.7 Role of retention strategies in ChP-SOEDNN

As discussed, to be able to compare different methods with the metric of misclassification cost, the relevance of our assumptions about retention strategies and assumed success rates are significant. The comparison assumes there is only one retention strategy, and it can only lead to the same retention success rate for each cluster. This assumption is necessary because of the lack of relevant data. A valuable future research direction ties with this research limitation. Collaborating with, or having better and more access to the data of a telecommunication company will afford researchers with more exploration possibilities. We were only able to show the significance of profit-driven SOED in striking a balance between false-positive and false-negative costs specific to each cluster. However, ChP-SOEDNN can even go beyond just decision making and find the set of optimized churn strategies for each cluster. In other words, if the data of previous churn attempts are available, the presented framework can find the best strategies for churn

retention for each cluster of customers. After estimating a relationship between all the retention strategies and the success rates unique to each cluster, SOED can achieve the optimum mix of retention strategies for each cluster alongside the adjustment of the dividing line. The requirements for such a study are access to a dataset of a company that includes both churn-related and retention-response-related data. Furthermore, the inclusion of SOED unsupervised (self-organizing) types of learning may create the possibility of taking advantage of the wisdom of the experts and data analysts by profiling the clusters and then introducing a set of retention strategies. “Appendix 2” section provides information on the profiling of clusters of customers using SOED.

5.8 Improve decision drive of SOED

We explored and discussed that resampling proved to be the most efficient and line adjustment demonstrated to be the most resilient cost-sensitive strategies. Valuable research can be conducted to explore how the coexistence of these two strategies could improve cost sensitivity of SOED. For instance, we suspect that a resampling strategy could be significantly successful if the probability of selecting a sample that will be used in the train set is relevant to how close the sample is to the part of SOM that the class label of members changes. In this paper, the resampling strategy creates a random train set given a resampling ratio. This suggestion will keep the benefits of the resampling paradigm while increasing its reliability. Recently, the success of more focused resampling using simple clustering methods has already been demonstrated in the literature [60].

Furthermore, there is room for improvement in the performance of proposed cost-sensitive ANN without line adjustment if one imparts cost sensitivity in the error calculation of MLP before backpropagation. In other words, instead of using mean squared error (MSE) to change the weights on the network in the backpropagation process the misclassification cost can be employed. The suggested substitution of error calculation is a proven challenge in the literature [40, 58], as MLP could only produce values than need recalculations before they lead to a churn prediction decision. That is the reason we are only able to inculcate the misclassification cost at the validation level of MLP training. Also, that is why Zhou and Liu [58] uses particle swarm optimization (PSO) to tune the thresholding and the resampling of the train set along with training MLP for each particle at every evolution of the PSO algorithm. A future effort could be training an MLP with the location of each case on MOD and, based on a selected dividing line, propagate back the misclassification cost of the prediction instead of simple MSE. We suspect improvements in the

performance of SOED as this forces MLP to focus on decreasing the misclassification cost. In theory, a cost-sensitive error calculation can create a profit-driven MLP.

5.9 Improve line adjustment procedure

The proposed line adjustment procedure in this paper, although effective, is time-consuming. Even though the procedure proved to be limited in reaching the extreme of 100% recall (Fig. 12c), each line adjustment procedure for different SR and RER needs to check 16,875 different possibilities of lines. The many numbers of possibilities proved to be computationally expensive. It is arguable that since we are only going to adjust the line one time for having a trained classifier, computational cost is not of a great concern. Nevertheless, from a research perspective, as research requires constantly running and testing experiments, it can be beneficial to create a less computationally expensive procedure. Also, if the resolution of SOM map or the line adjustment procedure is to increase, the number of possibilities will also be higher.

We suspect there can be some potential improvements. First, some heuristics can be proposed to give the search an intuitive advantage. Also, a meta-heuristic algorithm such as genetic algorithm (GA) may be used to find an acceptable line adjustment quicker than exhaustive search. In fact, the presentation of different lines in the last column of Table 3 may be a candidate for chromosome representation of the algorithm. After these computational improvements, an effort may study the impact of line adjustment resolution (the closeness of different situations in Fig. 7), which may create more or fewer possibilities of lines, on the performance of ChP-SOEDNN.

5.10 Improved method generalizability

Due to the selected method of optimization in this paper, a non-fixed randomized designed experiment to show the generalizability of our proposals was not possible. The optimization method is the line adjustment procedure. This choice provided the opportunity to study the influence and relevance of our assumptions in Sect. 4.2 (Assumptions). We have presented many interesting behaviors of the optimizing line due to the change in assumption parameters. For instance, now we know why including a profit drive at the level of ANN learning leads to improvement in profit-driven decision making; that is, the inconclusiveness nature of ANN prediction is handled informed by the uniqueness of the decision-making environment thorough an intelligent calculation of misclassification cost. While this discovery of knowledge is of importance—and not just for the literature of churn prediction, but also for any decision-driven data analytics—it is impossible to show the

generalizability of the presented method with validation methods such as k -fold cross-validation. This is due to the fact that SOM is a random visualization method; while it will lead to a consistent grouping of data points, it will output different visualization on each different run [29]. This fact makes creating a code-able line procedure nearly impossible or too time-consuming for practical application. We believe the presented decision-making framework is generalizable, and that has been shown in a different case study that uses a similar data mining method as this paper [28]. While this paper uses the line adjustment procedure, [28] use a meta-heuristic that allows for a code-able optimization at the expense of losing decision-making accuracy. Developing a more accurate and at the same time code-able optimization method for the method presented in this paper will be an important future contribution.

6 Conclusion

The paper presents a systematic profit-driven approach for telecom churn prediction, which is built using artificial neural networks based on error-driven and self-organizing learning approaches. ChP-SOEDNN has contributed to the phenomenon of churn prediction for telecom customers, through capturing the individuality of a customer for making a churn decision. In addition to churn accuracy, misclassification costs for false positives and false negatives are computed and incorporated for minimizing the cost.

The cost comparisons of ChP-SOEDNN performed prove the contributory role of the suggested perspective for churn prediction in the telecommunications industry. Specifically, the proposed method saves on average \$43,000 comparing with the best cost-sensitive approaches used in the literature in regard to churn decisions of only 472 customers. Thus, ChP-SOEDNN is proven to be more comprehensive and effective for providing a valuable support for strategic decision making to control the loss associated with the churn rate of a telecom firm.

The major limitation that authors faced in this research mostly relates to a lack of access to different types of data. The presented model in this paper is more comprehensive than just a churn prediction model in that it presents a management framework for decision and strategy making. Nevertheless, in this article, we had enough data access for churn prediction. For instance, we aspire we could have incorporated customer lifetime value if we had access to more data. However, we only managed to estimate customer revenue for the paper's clusters. Additionally, the limitation reduced the authors to making assumptions about success rates of different churn strategies to test the

effectiveness of the recommended decision-making framework.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix 1

Accepting the predicted membership of test set based on the SOM that is trained using train set and validation set as the test set membership is an important assumption. Therefore, it has been tested by comparing the similarity between the membership of test set cases when SOM learns with all customers in the dataset and when SOM learns only using train and validation sets.

In the literature, there are measures for comparing the similarities of two sets of clusterings. Clustering, unlike classification, does not have the luxury of class labels and this fact creates challenges to compare and evaluate the performance of different methods. There are two proposed approaches for comparing two sets of clusters (C and C') on the same set of data points: counting pairs and set matching [41]. We only introduce two common measures to compare clusters through counting pairs. When comparing C and C' , each pair of data points falls under one of four cases based on their memberships in C and C' . These four cases are denoted by $\bullet\bullet$, $\circ\circ$, $\bullet\circ$, $\circ\bullet$. Here, two filled circles denote that both of the data points are in the same cluster under both C and C' , and two hollow circles signify that under neither C nor C' is the pair in the same class. These two cases capture the similarities between C and C' . On the other hand, one filled circle and one hollow circle show that the pair has been the member of the same cluster under only one of C and C' , but not the other. Equation 10 and Eq. 11 represent, respectively, Fowlkes and Mallows and Rand's measures. In these formulas $N_{\bullet\bullet}$, $N_{\circ\circ}$, n , k , k' , nk , and nk' , respectively, stand for the number of pairs that fall under the same cluster under both C and C' , the number of pairs that are not the members of the same cluster under neither C nor C' , the number of data points, the number of clusters under C , the number of clusters under C' , the number of data points which are members of cluster k in C , and the number of data points which are member of cluster k' under C'

$$F(C, C') = \sqrt{\frac{N_{\bullet\bullet}}{\sum_k n_k(n_k - 1)/2} \times \frac{N_{\bullet\bullet}}{\sum_{k'} n_{k'}(n_{k'} - 1)/2}} \quad (10)$$

Table 5 SOMC, and Fowlkes and Mallows (FM) measures for different SOM outputs

	CD		TD		10% TD		B1		B2	
	SOMC	FM	SOMC	FM	SOMC	FM	SOMC	FM	SOMC	FM
Complete Data (CD)	1	1	0.5922	0.7761	0.5377	0.6922	− 0.0296	0.0257	− 0.8670	0.1748
Train Data (TD)	0.5922	0.7761	1	1	0.4794	0.6818	− 0.0135	0.0259	− 0.8560	0.1775
10% TD	0.5377	0.6922	0.4794	0.6818	1	1	− 0.0135	0.0267	− 0.8732	0.1878
Benchmark 1 (B1)	− 0.0296	0.0257	− 0.0135	0.0259	− 0.0135	0.0267	1	1	− 0.8282	0.1435
Benchmark 2 (B2)	− 0.8670	0.1748	− 0.8560	0.1775	− 0.8732	0.1878	− 0.8282	0.1435	1	1

$$R(C, C') = \frac{N_{\bullet\bullet} + N_{\infty}}{n(n-1)/2}. \quad (11)$$

Although these measures are capable of comparing two different SOM outputs [27], there is another dimension in the output of SOM more than clustering that these measures do not capture. In a normal clustering, there is not a defined relationship between two clusters; however, if each neuron in SOM is assumed to be a cluster, each cluster has neighboring clusters. In fact, this is the reason why in Step-6 of the SOED procedure (Algorithm 1) XY coordinates are assigned to the members of each cluster. Measures introduced in Eqs. 10 and 11 cannot capture this added dimension. The base of both of these equations is how similar the clustering technique has segmented all pairs of data points into different clusters. The proposed comparison measure, expressly designed for SOM outputs with the same topologies, captures the same similarity while including the neighboring dimension. In Eq. 12, n , α , $\text{Loc}_C(i)$, $\text{dist}(P1, P2)$ are, respectively, the number of data points, a constant parameters which falls between zero and the maximum distant possible between any two clusters in the SOM output, the location of the cluster which has data point i as a member, and the distant between $P1$ and $P2$. Preliminary experiments determine α , so SOMC returns 0 for the input of two random clusterings. Here, α may differ based on the different topology of SOM outputs. α is set to be 1.97 in the case of 7×7 SOM output

$$\text{SOMC}(o, o') = \frac{\sum_{i=j}^n \sum_{j=1}^n (\alpha - \text{dist}(\text{Loc}_C(i), \text{Loc}_C(j)))}{\alpha \times n \times (n-1)/2}. \quad (12)$$

The comparisons between different SOM made by Keramati and Jafari-Marandi [27] revealed that Rand's measures (Eq. 11) could not be as distinguishing as Fowlkes and Mallows (Eq. 10). Table 4 presents the Fowlkes and Mallows (FM—Eq. 10) and the proposed measure in Eq. 12 (SOMC) for the output of different SOM settings: Complete Data (CD), Train Data (TD), 10% TD, Benchmark 1 (B1), Benchmark 2 (B2). Complete Data and Train Data stand for the experiments that are the main

point of comparison in Table 5. Complete Data (CD) is the SOM setting that uses the complete data (train set, validation set, and test set) to drive the membership vector of the test set. Train Data (TD) only uses the train set and the validation set for the same output. Also, 10% TD only uses 10% of the train set for the same output. Moreover, Benchmark 1 (B1) is a simulation procedure of SOM output which creates a random membership vector of the test set, whereas Benchmark 2 (B2) is an intentional membership vector of the test set in which all the cases are the member of cluster 1. B1 and B2 serve as points of reference for the understanding of the range of the applied measures. FM ranges between 0 and 1: zero being no similarity and one perfect matching. SOMC ranges between − 1 and 1: negative one being negative matching, zero random matching, and one complete matching. Every cell in Table 4 is the average of ten experiments to control for the random nature of SOM. On the other hand, every cell in Table 5 is the value of the SOMC or FM for the presented example.

The comparison between the membership of test set records based on the CD and TD cases shows there is a meaningful similarity. Based on this similarity, we assume the validity of the predicted membership of test set based on the SOM that is trained using train set and validation set. Misclassification cost of the members of the test set is calculated based on the customer revenue calculated for each cluster in MOD.

Appendix 2

Profiling of clusters of customers is valuable because it creates insights on the kind of customers that are in the dividing part of SEOD map. Figure 5a shows these regions within the employed SOED map. $N1$ to $N7$ are the clusters residing non-churn cases, and $C1$ to $C5$ are the clusters of churn cases whose profiling are worthwhile for SOED's procedure. It is helpful to look at the average value of each attribute with the help of a color-coded map based on all

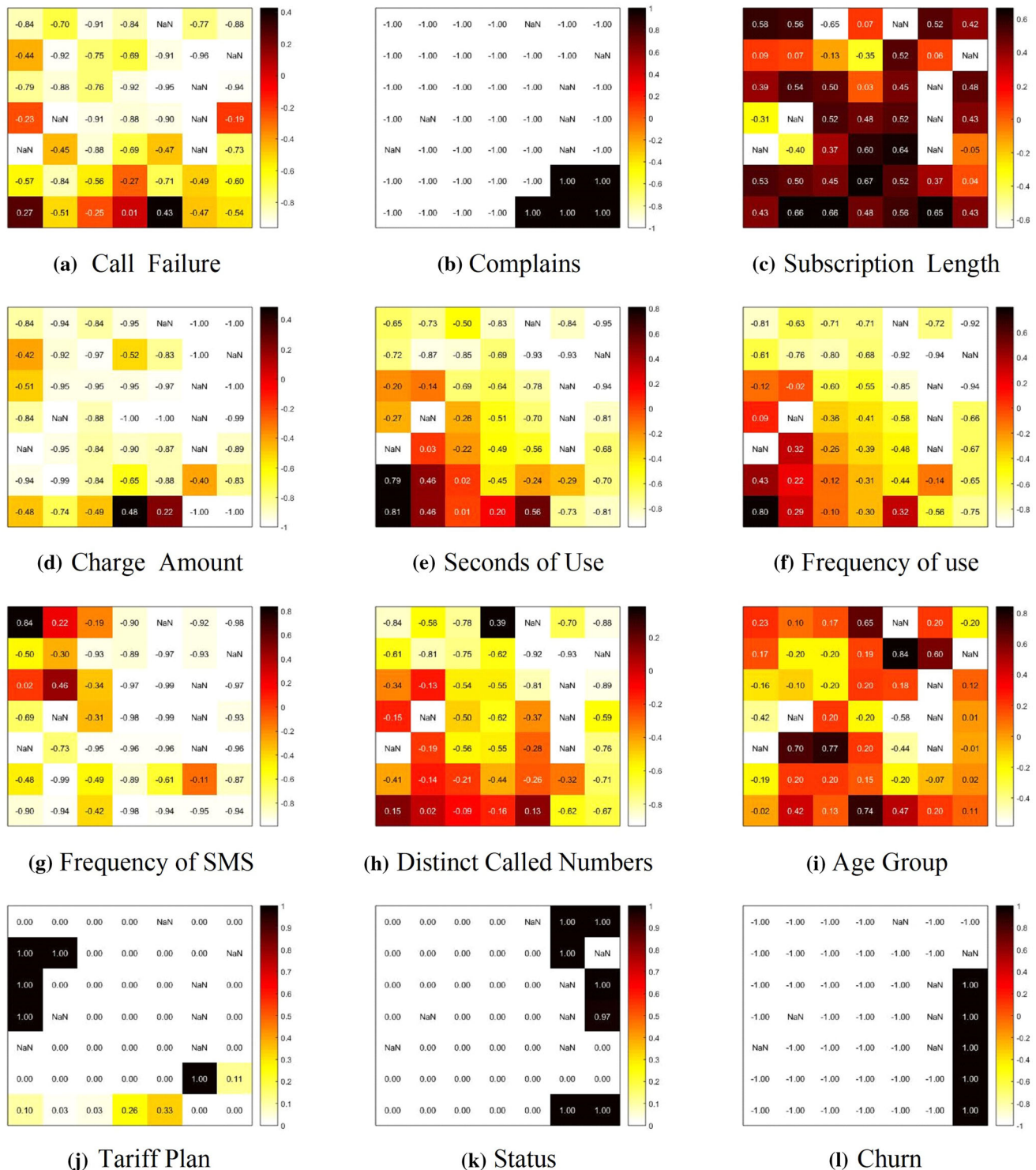


Fig. 15 SOED colored map for all the dataset attributes

the clusters' scaled average. Figure 15 presents these for all the attributes in the dataset.

Appendix 3

See Tables 6, 7, 8, 9, 10, and 11, respectively, representing the 20 validation runs for CS MLP 1–4, CS DT, and CS AdaBoost (Table 12).

Table 6 SOM hit rate examples for Table 4 experiments




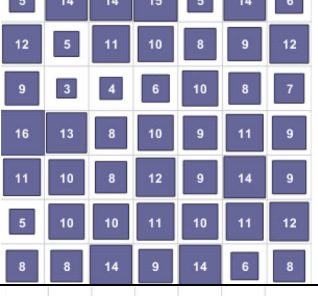

		CD		TD		10%TD		B1		B2	
		SOMC	FM	SOMC	FM	SOMC	FM	SOMC	FM	SOMC	FM
	CD	1	1								
	TD	.5184	.7810	1	1						
	10%TD	.3798	.7298	.4196	.7275	1	1				
	B1	.0288	.0312	.0402	.0323	.0009	.0277	1	1		
	B2	.8898	.1779	.9199	.1929	.8342	.1863	.8340	.1428	1	1

Table 7 CS MLP 1: 20 validation runs

No.	Cost	No. miss	Net	Accuracy	Recall	Precision	<i>F</i> -score
1	85,144.11	12	‘9,8,7’	0.974576	0.953125	0.871429	0.910448
2	74,807.47	13	‘11,7’	0.972458	0.925373	0.885714	0.905109
3	78,294.5	15	‘7,7,14’	0.96822	0.936508	0.842857	0.887218
4	31,437.5	12	‘14,3,1’	0.974576	0.881579	0.957143	0.917808
5	38,690.18	8	‘14,10,7’	0.983051	0.955882	0.928571	0.942029
6	45,030.4	21	‘7,3’	0.955508	0.810127	0.914286	0.85906
7	143,786.2	26	‘15,6’	0.944915	0.87931	0.728571	0.796875
8	129,183.8	21	‘8,9,11’	0.955508	0.929825	0.757143	0.834646
9	84,639.84	13	‘8,1’	0.972458	0.967213	0.842857	0.900763
10	68,286.46	11	‘15,1’	0.976695	0.927536	0.914286	0.920863
11	148,108	20	‘3,4,13’	0.957627	0.962963	0.742857	0.83871
12	46,218.07	17	‘8,12,6’	0.963983	0.853333	0.914286	0.882759
13	84,935.24	15	‘12,2’	0.96822	0.876712	0.914286	0.895105
14	48,232.32	17	‘12,2’	0.963983	0.853333	0.914286	0.882759
15	44,629.46	18	‘13,9,10’	0.961864	0.842105	0.914286	0.876712
16	107,696.1	13	‘6,14’	0.972458	0.952381	0.857143	0.902256
17	44,080.39	15	‘6,6’	0.96822	0.876712	0.914286	0.895105
18	38,468.28	10	‘7,11’	0.978814	0.928571	0.928571	0.928571
19	99,125.77	21	‘11,1’	0.955508	0.888889	0.8	0.842105
20	70,251.2	15	‘14,11,9’	0.96822	0.876712	0.914286	0.895105

Table 8 CS MLP 2: 20 validation runs

No.	Cost	No. miss	Net	RS	Accuracy	Recall	Precision	<i>F</i> -score
1	46,858.86	25	‘8,6’	1	0.947034	0.777778	0.9	0.834437
2	46,763.94	25	‘8,6’	1	0.947034	0.777778	0.9	0.834437
3	60,008.16	68	‘6,7,13’	6	0.855932	0.507692	0.942857	0.66
4	63,203.72	89	‘9,9,1’	5	0.811441	0.437908	0.957143	0.600897
5	31,157.41	43	‘4,8,9’	5	0.908898	0.623853	0.971429	0.759777
6	19,326.28	13	‘8,7,11’	1	0.972458	0.860759	0.971429	0.912752
7	40,847.19	53	‘10,4’	5	0.887712	0.575221	0.928571	0.710383
8	72,397.66	36	‘10,3’	2	0.923729	0.680851	0.914286	0.780488
9	8719.276	29	‘12,10’	4	0.938559	0.707071	1	0.828402
10	23,503.18	95	‘4,3,3’	8	0.798729	0.424242	1	0.595745
11	48,711.96	67	‘10,7’	10	0.858051	0.511111	0.985714	0.673171
12	9019.875	28	‘9,4,5’	1	0.940678	0.714286	1	0.833333
13	18,283.28	36	‘3,4’	1	0.923729	0.67	0.957143	0.788235
14	20,310.61	82	‘4,10’	7	0.826271	0.460526	1	0.630631
15	3104.054	13	‘6,8,3’	1	0.972458	0.843373	1	0.915033
16	12,729.76	18	‘6,10’	2	0.961864	0.809524	0.971429	0.883117
17	16,792.42	65	‘5,12,1’	4	0.862288	0.518519	1	0.682927
18	48,427.66	88	‘12,1,2’	10	0.813559	0.440789	0.957143	0.603604
19	43,267.73	28	‘11,7,4’	1	0.940678	0.733333	0.942857	0.825
20	22,177.93	88	‘8,8,7’	5	0.813559	0.443038	1	0.614035

Table 9 CS MLP 3: 20 validation runs

No.	Cost	No. miss	Net	T	Accuracy	Recall	Precision	F -score
1	43,674.44	161	'14,7'	− 0.95	0.658898	0.30303	1	0.465116
2	38,789.85	147	'8,2'	− 1	0.688559	0.322581	1	0.487805
3	45,547.07	27	'6,10'	− 0.7	0.942797	0.736264	0.957143	0.832298
4	28,981.55	80	'13,12'	− 0.9	0.830508	0.466216	0.985714	0.633028
5	9051.222	28	'12,13,10'	− 0.45	0.940678	0.714286	1	0.833333
6	15,600.01	23	'9,12,6'	− 0.5	0.951271	0.764045	0.971429	0.855346
7	14,065.22	47	'11,4,8'	− 0.85	0.900424	0.598291	1	0.748663
8	13,017.79	44	'9,5'	− 0.95	0.90678	0.614035	1	0.76087
9	12,026.86	20	'9,6'	− 0.7	0.957627	0.784091	0.985714	0.873418
10	71,931.05	13	'7,14'	− 0.35	0.972458	0.890411	0.928571	0.909091
11	10,170.49	30	'12,10,8'	− 0.75	0.936441	0.7	1	0.823529
12	21,560.49	73	'13,9,15'	− 0.95	0.845339	0.48951	1	0.657277
13	42,341.36	40	'14,14'	− 0.9	0.915254	0.641509	0.971429	0.772727
14	15,361.64	40	'15,15,5'	− 0.75	0.915254	0.638889	0.985714	0.775281
15	21,945.04	74	'13,15'	− 0.75	0.84322	0.486111	1	0.654206
16	9785.728	38	'12,5'	− 0.85	0.919492	0.648148	1	0.786517
17	36,099.81	114	'11,13'	− 0.95	0.758475	0.379121	0.985714	0.547619
18	16,119.87	53	'12,12'	− 0.75	0.887712	0.569106	1	0.725389
19	16,452.04	12	'14,12,2'	− 0.55	0.974576	0.881579	0.957143	0.917808
20	44,352.8	35	'7,6'	− 0.65	0.925847	0.676768	0.957143	0.792899

Table 10 CS MLP 4: 20 validation runs

No.	Cost	No. miss	Net	T	RS	Accuracy	Recall	Precision	F -score
1	56,061.96	174	'12,10'	− 1	2	0.631356	0.285124	0.985714	0.442308
2	95,825.87	58	'9,7,13'	0.8	4	0.877119	0.552632	0.9	0.684783
3	36,127.73	77	'7,2,8'	− 0.8	4	0.836864	0.475524	0.971429	0.638498
4	28,780.88	106	'12,12,2'	− 0.65	10	0.775424	0.397727	1	0.569106
5	45,986.42	36	'10,8'	− 0.65	2	0.923729	0.666667	0.971429	0.790698
6	10,716.39	31	'4,3'	0.35	1	0.934322	0.693069	1	0.818713
7	18,632.33	73	'9,5'	− 0.35	6	0.845339	0.48951	1	0.657277
8	20,827.27	73	'6,11,1'	− 0.65	3	0.845339	0.48951	1	0.657277
9	22,945.93	21	'10,8,1'	− 0.2	1	0.955508	0.781609	0.971429	0.866242
10	51,156.06	79	'12,11'	0.9	9	0.832627	0.469388	0.985714	0.635945
11	21,712.07	55	'8,11'	− 0.2	2	0.883475	0.560976	0.985714	0.715026
12	18,257.84	71	'8,8'	0.75	9	0.849576	0.496454	1	0.663507
13	43,472.11	142	'3,8'	− 0.75	1	0.699153	0.330189	1	0.496454
14	60,169.97	84	'6,4'	− 0.9	1	0.822034	0.452055	0.942857	0.611111
15	35,048.17	53	'9,7,5'	− 0.3	3	0.887712	0.571429	0.971429	0.719577
16	37,409.32	69	'12,5,3'	0.2	8	0.853814	0.50365	0.985714	0.666667
17	18,834.71	69	'9,7'	− 0.55	4	0.853814	0.503597	1	0.669856
18	35,744.42	80	'5,2,4'	0.45	2	0.830508	0.465278	0.957143	0.626168
19	41,862.92	26	'7,7'	0.9	2	0.944915	0.755814	0.928571	0.833333
20	70,651.84	268	'4,1,15'	− 1	3	0.432203	0.207101	1	0.343137

Table 11 CS DT: 20 validation runs

No.	Cost	No. miss	T	RS	Accuracy	Recall	Precision	F -score
1	55,461.37	49	0.65	8	0.896186	0.598131	0.914286	0.723164
2	58,616.98	60	0.7	10	0.872881	0.541667	0.928571	0.684211
3	62,318.93	67	0.95	10	0.858051	0.512	0.914286	0.65641
4	90,078.33	42	0.4	4	0.911017	0.642857	0.9	0.75
5	75,794.23	66	0.1	8	0.860169	0.516393	0.9	0.65625
6	71,379.09	83	0.65	7	0.824153	0.453237	0.9	0.602871
7	55,948.35	75	0.1	8	0.841102	0.482014	0.957143	0.641148
8	33,129.11	82	0.3	8	0.826271	0.459459	0.971429	0.623853
9	74,450.92	70	0.45	9	0.851695	0.5	0.857143	0.631579
10	32,188.91	59	0.1	9	0.875	0.545455	0.942857	0.691099
11	62,793.52	95	– 0.45	9	0.798729	0.419355	0.928571	0.577778
12	64,587.08	74	– 0.35	10	0.84322	0.484615	0.9	0.63
13	52,720.97	62	0.45	9	0.868644	0.535088	0.871429	0.663043
14	26,936	53	– 0.05	10	0.887712	0.57265	0.957143	0.716578
15	45,971.74	62	0.65	10	0.868644	0.532258	0.942857	0.680412
16	72,650.29	66	0.3	8	0.860169	0.516393	0.9	0.65625
17	65,760.49	68	0.6	6	0.855932	0.507937	0.914286	0.653061
18	65,811.32	51	0.55	5	0.891949	0.587156	0.914286	0.715084
19	45,493.64	38	0.3	2	0.919492	0.66	0.942857	0.776471
20	23,005.69	72	– 0.05	7	0.847458	0.492857	0.985714	0.657143

Table 12 CS AdaBoost: 20 validation runs

No.	Cost	No. miss	T	Cost ratio	Accuracy	Recall	Precision	F -score
1	272,775.8	39	– 1	6	0.917373	0.942857	0.471429	0.628571
2	42,672.35	161	2	4	0.658898	0.30131	0.985714	0.461538
3	274,828.2	39	0	10	0.917373	0.897436	0.5	0.642202
4	52,106.24	51	4	1	0.891949	0.588785	0.9	0.711864
5	60,284.8	54	0	9	0.885593	0.575472	0.871429	0.693182
6	48,547.48	73	3	0.333333	0.845339	0.488722	0.928571	0.640394
7	60,410.02	215	4	0.166667	0.544492	0.243816	0.985714	0.390935
8	24,254.86	84	9	1	0.822034	0.453333	0.971429	0.618182
9	263,413.7	44	0	0.1	0.90678	0.809524	0.485714	0.607143
10	113,428.2	38	1	0.166667	0.919492	0.705128	0.785714	0.743243
11	42,123.64	149	3	2	0.684322	0.317972	0.985714	0.480836
12	255,897.6	37	1	0.2	0.92161	0.923077	0.514286	0.66055
13	67,622.81	34	0	6	0.927966	0.714286	0.857143	0.779221
14	29,767.05	68	5	2	0.855932	0.507692	0.942857	0.66
15	36,670.15	131	– 1	0.1	0.722458	0.346734	0.985714	0.513011
16	36,728.4	130	1	5	0.724576	0.348485	0.985714	0.514925
17	233,334	37	2	5	0.92161	0.866667	0.557143	0.678261
18	199,762.7	34	0	4	0.927966	0.846154	0.628571	0.721311
19	51,310.32	187	2	0.5	0.603814	0.270588	0.985714	0.424615
20	27,104.8	91	– 2	0.111111	0.807203	0.433121	0.971429	0.599119

References

- Ahmed U, Khan A, Khan SH, Basit A, Haq IU, Lee YS (2019) Transfer learning and meta classification based deep churn prediction system for telecom industry. Preprint [arXiv:1901.06091](https://arxiv.org/abs/1901.06091)
- Amin A, Al-Obeidat F, Shah B, Adnan A, Loo J, Anwar S (2019) Customer churn prediction in telecommunication industry using data certainty. *J Bus Res* 94:290–301
- Amin A, Anwar S, Adnan A, Nawaz M, Alawfi K, Hussain A, Huang K (2017) Customer churn prediction in the telecommunication sector using a rough set approach. *Neurocomputing* 237:242–254
- Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, Hawalah A, Hussain A (2016) Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. *IEEE Access* 4:7940–7957
- Bahnsen AC, Aouada D, Ottersten B (2015) Example-dependent cost-sensitive decision trees. *Expert Syst Appl* 42(19):6609–6619
- Bahnsen AC, Aouada D, Ottersten B (2015) A novel cost-sensitive framework for customer churn predictive modeling. *Decis Anal* 2(1):5
- Berger PD, Nasr NI (1998) Customer lifetime value: Marketing models and applications. *J Interact Mark* 12(1):17–30
- Bi W, Cai M, Liu M, Li G (2016) A big data clustering algorithm for mitigating the risk of customer churn. *IEEE Trans Ind Inf* 12(3):1270–1281
- Burez J, Van den Poel D (2009) Handling class imbalance in customer churn prediction. *Expert Syst Appl* 36(3):4626–4636
- Chen Z-Y, Fan Z-P (2012) Distributed customer behavior prediction using multiplex data: a collaborative MK-SVM approach. *Knowl Based Syst* 35:111–119
- Chen Z-Y, Fan Z-P, Sun M (2012) A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *Eur J Oper Res* 223(2):461–472
- De Bock KW, Van den Poel D (2012) Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Syst Appl* 39(8):6816–6826
- Ekinci Y, Ülengin F, Uray N, Ülengin B (2014) Analysis of customer lifetime value and marketing expenditure decisions through a Markovian-based model. *Eur J Oper Res* 237(1):278–288
- Fader PS, Hardie BG, Lee KL (2005) RFM and CLV: using iso-value curves for customer base analysis. *J Mark Res* 42(4):415–430
- García DL, Nebot À, Vellido A (2017) Intelligent data analysis approaches to churn as a business problem: a survey. *Knowl Inf Syst* 51(3):719–774
- Glady N, Baesens B, Croux C (2009) Modeling churn using customer lifetime value. *Eur J Oper Res* 197(1):402–411
- Gruca TS, Rego LL (2005) Customer satisfaction, cash flow, and shareholder value. *J Mark* 69(3):115–130
- Gupta S, Lehmann DR, Stuart JA (2004) Valuing customers. *J Mark Res* 41(1):7–18
- Gurney K (2014) Multilayer nets and backpropagation. In: An introduction to neural networks, 1st edn. CRC Press, Boca Raton, pp 41–57
- Han S, Yuan B, Liu W (2009) Rare class mining: progress and prospect. In: CCPR 2009. Chinese conference on pattern recognition, 2009. IEEE, New York, pp 1–5
- Höppner S, Stripling E, Baesens B, vanden Broucke S, Verdonck T (2018) Profit driven decision trees for churn prediction. *Eur J Oper Res* 286(3):920–933
- Huang B, Kechadi MT, Buckley B (2012) Customer churn prediction in telecommunications. *Expert Syst Appl* 39(1):1414–1425
- Huang Y, Kechadi T (2013) An effective hybrid learning system for telecommunication churn prediction. *Expert Syst Appl* 40(14):5635–5647
- Idris A, Khan A, Lee YS (2012) Genetic programming and adaboosting based churn prediction for telecom. In: 2012 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, New York, pp 1328–1332
- Idris A, Khan A, Lee YS (2013) Intelligent churn prediction in telecom: employing mRMR feature selection and RotBoost based ensemble classification. *Appl Intell* 39(3):659–672
- Idris A, Rizwan M, Khan A (2012) Churn prediction in telecom using random forest and PSO based data balancing in combination with various feature selection strategies. *Comput Electr Eng* 38(6):1808–1819
- Jafari-Marandi AKR (2014) Webpage clustering—taking the zero step: a case study of an Iranian website. *J Web Eng* 13(3–4):333–360
- Jafari-Marandi R, Davarzani S, Gharibdousti MS, Smith BK (2018) An optimum ANN-based breast cancer diagnosis: bridging gaps between ANN learning and decision-making goals. *Appl Soft Comput* 72:108–120
- Jafari-Marandi R, Khanzadeh M, Smith BK, Bian L (2017) Self-organizing and error driven (SOED) artificial neural network for smarter classifications. *J Comput Des Eng* 4(4):282–304
- Jafari-Marandi R, Khanzadeh M, Tian W, Smith B, Bian L (2019) From in-situ monitoring toward high-throughput process control: cost-driven decision-making framework for laser-based additive manufacturing. *J Manufact Syst* 51:29–41
- Keramati A, Jafari-Marandi R, Aliannejadi M, Ahmadian I, Mozaffari M, Abbasi U (2014) Improved churn prediction in telecommunication industry using data mining techniques. *Appl Soft Comput* 24:994–1012
- Khan A, Sohail A, Ali A (2018) A new channel boosted convolutional neural network using transfer learning. Preprint [arXiv:1804.08528](https://arxiv.org/abs/1804.08528)
- Kohonen T (2013) Essentials of the self-organizing map. *Neural Netw* 37:52–65
- Lee H, Lee Y, Cho H, Im K, Kim YS (2011) Mining churning behaviors and developing retention strategies based on a partial least squares (PLS) model. *Decis Support Syst* 52(1):207–216
- Lemmens A, Gupta S (2017) Managing churn to maximize profits. Working paper
- Lemmens A, Gupta S (2017) Managing churn to maximize profits. Available at SSRN 2964906
- Liu Y, Zhuang Y (2015) Research model of churn prediction based on customer segmentation and misclassification cost in the context of big data. *J Comput Commun* 3(06):87
- Lu N, Lin H, Lu J, Zhang G (2014) A customer churn prediction model in telecom industry using boosting. *IEEE Trans Ind Inf* 10(2):1659–1665
- Maldonado S, López J, Vairetti C (2019) Profit-based churn prediction based on minimax probability machines. *Eur J Oper Res* 284(1):273–284
- Mazurovski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD (2008) Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. *Neural Netw* 21(2):427–436
- Meilă M (2007) Comparing clusterings—an information based distance. *J Multivar Anal* 98(5):873–895
- Organization WH (2010) World health statistics. World Health Organization, New York
- Prashanth R, Deepak K, Meher AK (2017) High accuracy predictive modelling for customer churn prediction in telecom

- industry. In: International conference on machine learning and data mining in pattern recognition. Springer, Berlin, pp 391–402
44. Reinartz WJ, Kumar V (2003) The impact of customer relationship characteristics on profitable lifetime duration. *J Mark* 67(1):77–99
 45. Risselada H, Verhoef PC, Bijmolt TH (2010) Staying power of churn prediction models. *J Interact Mark* 24(3):198–208
 46. Saito T, Rehmsmeier M (2015) The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* 10(3):e0118432
 47. Sheng VS, Ling CX (2006) Thresholding for making classifiers cost-sensitive. In: *AAAI*, pp 476–481
 48. Stripling E, vanden Broucke S, Antonio K, Baesens B, Snoeck M (2018) Profit maximizing logistic model for customer churn prediction using genetic algorithms. *Swarm Evol Comput* 40:116–130
 49. Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. *Pattern Recogn* 40(12):3358–3378
 50. Tan PN, Steinbach M, Kumar V (2016) Introduction to data mining. Pearson Education, India
 51. Tang L, Thomas L, Fletcher M, Pan J, Marshall A (2014) Assessing the impact of derived behavior information on customer attrition in the financial service industry. *Eur J Oper Res* 236(2):624–633
 52. Ullah I, Raza B, Malik AK, Imran M, Islam SU, Kim SW (2019) A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE Access* 7:60134–60149
 53. van Wezel M, Potharst R (2007) Improved customer choice predictions using ensemble methods. *Eur J Oper Res* 181(1):436–452
 54. Verbeke W, Dejaeger K, Martens D, Hur J, Baesens B (2012) New insights into churn prediction in the telecommunication sector: a profit driven data mining approach. *Eur J Oper Res* 218(1):211–229
 55. Verbraken T, Verbeke W, Baesens B (2013) A novel profit maximizing metric for measuring classification performance of customer churn prediction models. *IEEE Trans Knowl Data Eng* 25(5):961–973
 56. Wei C-P, Chiu I-T (2002) Turning telecommunications call details to churn prediction: a data mining approach. *Expert Syst Appl* 23(2):103–112
 57. Zhang C, Ni M, Yin H, Qiu K (2018) Developed density peak clustering with support vector data description for access network intrusion detection. *IEEE Access* 6:46356–46362
 58. Zhou Z-H, Liu X-Y (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans Knowl Data Eng* 18(1):63–77
 59. Zhu B, Baesens B, vanden Broucke SK, (2017) An empirical comparison of techniques for the class imbalance problem in churn prediction. *Inf Sci* 408:84–99
 60. Zhu H, Wang X (2017) A cost-sensitive semi-supervised learning model based on uncertainty. *Neurocomputing* 251:106–114

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.