PROFIT-DRIVEN NON-LINEAR CLASSIFICATION WITH APPLICATIONS TO CREDIT CARD FRAUD DETECTION, CHURN PREDICTION, DIRECT MARKETING AND CREDIT SCORING

A Thesis

by

Ashkan Zakaryazad

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
November 2015

PROFIT-DRIVEN NON-LINEAR CLASSIFICATION WITH APPLICATIONS TO CREDIT CARD FRAUD DETECTION, CHURN PREDICTION, DIRECT MARKETING AND CREDIT SCORING

Approved by:

Professor Ekrem Duman, Advisor
Department of Industrial Engineering
Özyeğin University

Assistant Professor Dilek Günneç
Department of Industrial Engineering
Özyeğin University

Assistant Professor Mustafa Ağaoğlu,
external examiner
Department of Computer Engineering
Marmara University

Date Approved: 6 November 2015

To my family and specially my mother and wife who have been a source of encouragement and inspiration to me throughout my life.

ABSTRACT

The rapid growth in data capture and computational power has led to an increasing focus on data-driven research. So far, most of the research is focused on predictive modeling using statistical optimization where it is aimed to minimize the number (or, the weighted number) of incorrectly classified records, while profit maximization has been given less priority. It is exactly the central issue that is addressed in this study by taking a profit-driven approach to develop a well-known non-linear classification technique (Artificial Neural Network) which maximizes the total profit earned by model implementation. Therefore, the focus is shifted from a statistical optimization to profit maximization.

Classification which is one of the most common prediction problems, have traditionally been tackled by the data mining (DM) algorithms. The objective taken in these algorithms is a statistical one where it is aimed to minimize the number (or, the weighted number) of incorrectly classified records. In traditional cost-sensitive classification, the error of mislabeling a minor class record (False Negative) could be larger than the error of mislabeling a major class record (False Positive). This approach is useful especially where there is a high imbalance between the classes. However, this does not cope for the situations where the costs of mislabeling the instances or the profits gained from correctly labeled instances are variable (i.e., changing from instance to instance).

The central objective here is to maximize the total net profit gained from applying the classification models using individual (case-based) profits and costs of each of the instances. This approach has been used in four application areas: Credit Card Fraud detection, Churn Prediction, Direct Marketing and Credit Scoring.

ÖZETÇE

Son yılllarda veri miktarlarında ve bilgisayarlı hesaplama güçlerinde yaşanan artışlar veri analizlerinin önemini artırmıştır. Şimdiye kadar daha çok betimsel analizler yapılmış, tahmine dönük analizler biraz daha az gündem bulmuştur. Bu çalışma daha eksik olan kısımda yani tahmin modellemesi cephesinde yer almaktadır.

Tahmin etme problemlerinin en tipik olanı sınıflandırma problemleri şimdiye kadar çoğunlukla veri madenciliği (VM) teknikleriyle çözülmeye çalışılmıştır. Bu algoritmalarda ele alınan amaç fonksiyonu genellikle istatistiki bir ölçünün eniyilenmesidir (doğru etiketlenen kayıtların sayısının veya ağırlıklı toplam sayısının yüksek olması gibi).

Bu algoritmalarda ele alınan amaç fonksiyonu istatistiksel olduğuna göre yanlış sınıflandırılmış örneklerin sayısını azaltmaktır. Eski maliyete duyarlı sınıflandırma algoritmalarında yanlış negatif (YN) hata bedeli yanlış pozitif (YP) hatasından fazla olabilir. Bu yaklaşım özellikle çok dengesiz veri kümlerinde faydalıdır. Halbuki, bu yaklaşım örneklerin yanlış sınıflandırılması maliyeti ya doğru sınıflandırılması karı değişken olduğu durumlarda kullanılamaz hale gelmektedir.

Bu çalışmada, esas amaç fonksiyonu sınıflandırma yapay sınır ağları kullanarak ve örneklerin değişken kar ve maliyetlerini göz önüne alarak toplam net kari maksimize etmektir. Bu yaklaşım dört farklı uygulama alanında kullanılmıştır: Kredi kart sahtekarlık tespiti, terk analizi, doğrudan pazarlama, ve kredi skorlama.

ACKNOWLEDGEMENTS

With a deep sense of gratitude I would like to thank Prof. Ekrem Duman for his patience, motivation, and immense knowledge who made the great opportunity for me to perform this research and be a member of TÜBİTAK project (Project No= 113M063). Besides, I would like to thank our group members Nader Mahmoudi and Azamat Kibekbaev whose guidance helped me in all the time of research and writing of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Statistical learning refers to understanding data by statistical and computer analysis and its results are detected patterns and knowledge in the data which can not be acquired by traditional statistical analysis. Statistical and particularly, machine learning boils down to two major learning types, supervised and unsupervised. Supervised learning deals with constructing a model to estimate a output using one or more inputs and their corresponding supervising target amounts. Unsupervised learning refers to learning structure and relationship from data without any supervising target value [1]. Classification problems which are one of the most common prediction problems in supervised learning, have traditionally been tackled by the data mining (DM) algorithms. The objective taken in these algorithms is a statistical one where it is aimed to minimize the number (or, the weighted number) of incorrectly classified records. For binary classification the two classes are named as positive (P) and negative (N). If a P record is correctly classified as P, it is a true positive (TP) and if an N record is correctly classified as N, it is a true negative (TN). The other two decisions are wrong decisions (false positive: FP and false negative: FN). In traditional settings of DM, the costs of both types of errors are taken as the same. However, by the advent of cost-sensitive learning, the cost of one type of error can be taken as a constant multiple of the other type of error. This was a big contribution for classification problems which made it one of the challenging problems in data mining research [2] because especially where there is a high imbalance between the classes, the error of mislabeling a minor class record (FN) could be larger than the error of mislabeling a major class record (FP). However, this does not cope for the situations where the costs of mislabeling the instances or the profits of correct labeling of them are variable (i.e., changing from one instance to another). In this research we have taken the individual costs and profits into account in both model building step and performance measurements.

In real-life we can find many problem domains where a profit-driven approach can be taken in the classification task. We have focused on four example domains while testing our theoretical developments: credit card fraud detection, customer churn prediction, direct marketing response models and credit scoring. Consider a credit card fraud detection problem where when classifying the incoming credit card transactions as fraudulent or legitimate, the cost of a FN (missing to label a fraudulent transaction as fraudulent) is much larger than the cost of a FP (false alert), which is often variable. This is because when fraudsters obtain a card (or its information) they typically use it until its remaining usable limit (which is varying from card to card) is depleted. Next, assume a company knows the CLV (customer lifetime value) of its customers which measures the present worth of all future profits expected from each customer. If the company is now developing a churn prediction model, then the reward of correctly labeling the actual churners will be different for each customer and it will be equal to their CLV values. In marketing models if a correct customer is targeted, depending on the profile of the customer, a variable incremental profit can be made. In credit risk modeling, the cost of mislabeling an indeed-good customer will be equal to the lost profits of the rejected credit. Therefore, the total net profit of each model depends on the individual costs and profits of each instance.

In the literature, numerous metrics have been proposed for measuring and comparing the performances of alternative classifiers. However, these performance measures are typically based on a statistical criterion without explicitly taking the costs and benefits into account. Hence, these often used performance metrics are not aligned with profit maximization, the main goal in a business context. On the other hand, a company needs to bare some costs if it wants to obtain some benefits using such classification models (e.g., to change the mind of a potential churner an incentive should be given). Whether the benefit will be obtained or not depends on the size of the incentive and other parameters which will be discussed in the next sections. Thus, there are some dependencies between the cost-benefit parameters and it is not easy to calculate the overall net profit. Such dependencies have not been studied in the literature before.

Although there has been some attention for cost-sensitive learning in literature, the profit-driven classification is at its infancy yet. It is exactly this gap we tried to fill with this research.

The most popular application of the supervised learning in business is classification of the data. In classification problems, the goal is to take an input vector X and assign it to one of the K disjoint classes C_k where $k = 1, 2, ..., K$. If there are just two classes for the data, the problem is called "binary classification". The input space is divided into *decision regions* whose boundaries are called *decision boundary*. If a decision boundary is a linear function of input vector x, the classification is linear classification and the data sets which can be separated by these classifiers are called "linearly separable" data sets. Fisher's discriminant, logistic regression and linear perceptron are examples from linear classification models [3]. Rule based classifiers such as Naïve Bayes and tree-based methods and some complex models such as support vector machines and artificial neural network are examples from non-linear classifiers which generate non-linear functions from input vectors and consequently non-linear boundaries for discriminating the data to different classes. Frequently used traditional statistical techniques in business-oriented classification problems are logistic regression, linear and quadratic discriminant analysis models. However, their pre-determined functional form and restrictive model assumptions limit their usefulness [3, 4].

Among nonlinear classification models, artificial neural network (ANN) has attracted a huge attention of data scientists because of its unique features which are briefly explained here. First, in contrast to traditional model-based algorithms, ANNs learn from examples and they are self-adapting models which capture subtle functional relationships among the data even if the underlying relationships are unknown or hard to describe. They have acceptable performance even if the sample data contains noisy information and they are universal functional approximators [6, 7, 8]. This means that they can approximate any non-linear function with arbitrary accuracy and this is a great advantage to ANNs as there are huge number of non-linear functions in real-word problems and ANNs are able to approximate them well.

Second, in most of the real-world problems it is difficult to determine whether the problem under study is linear or non-linear and ANNs can be good choices when there is a need to difficultly specified knowledge about the solutions but a sufficient data is available to learn from. Accordingly this non-parametric non-linear statistical method is suitable for the cases where it is difficult to gain knowledge about the underlying laws of the corresponding system but easy to collect sufficient experiences as data sets.

Finally, ANNs can generalize the learned relationship from the data and use it to correctly interpret about the unseen part of the data. This is a key attribute for a prediction model which will work on the new and regularly updating data and ANNs need not regular manipulation to cope with new data. These advantages have made the ANNs as promising models for approximation, regression and classification problems and it is proved that they have acceptable performance in broad applications in business, medicine, speech recognition and so on [9, 10, 11].

Plenty of applications have been proposed for neural network classifier such as bankruptcy prediction [12, 13, 14], credit card fraud detection [15, 16, 17], churn prediction [19], credit scoring [18, 20] and direct marketing [22, 62]. In most of these research the concentration is on the maximum accuracy of the classifier model to maximize the number of correct classifications, however, in this research, the focus has been shifted from statistical accuracy maximization to net profit maximization. Although accuracy maximization or equivalently prediction error minimization result in a model which indirectly maximizes the total profit of the financial institutions, taking the net profit of the model implementation into account may result in a model which is more robust in terms of net profit, the primary objective of management. We propose two kinds of neural network classifier which work based on profit maximization. In one of the approaches we modify the Sum of Squared Errors (SSE) and make it more sensitive to individual profit and cost of each instance and in the second approach, we use the total net profit as the objective of the model to be maximized using well-known metaheuristic algorithms; Particla Swarm Optimization (PSO), Genetic Algorithm (GA), Artificial Bee Colony (ABC) and Migrating Birds Optimization (MBO). Moreover, the metaheuristic algorithms are proved to reach better solutions than gradient-descent based algorithms which can not skip local optimums and find the global optimum point. We apply these models on four application areas, credit card fraud detection, churn prediction, direct marketing and credit scoring.

Another contribution of this research is to investigate the expected profit of churn prediction where calculating the profit is more complex than credit scoring and credit card fraud detection. This is because of the effect of customers and their decision on the total net profit of the model. In this application of data mining, we consider the probabilistic reaction of the customers and their corresponding profits and find the total expected profit of the model implementation.

We have applied our new models in all of the aforementioned application areas of classification and compare them with related works to show the contributions of the new models considering total net profit which is the primary objective in this study. We first introduce the ANN model and the studied application areas in chapter 1 and provide the related literature in chapter 2. After that, we discuss proposed models and performance metrics in chapter 3. Then, we compare the results in chapter 4 and finally reach the conclusion and future works in chapter 5.

CHAPTER II

RELATED WORK

In this chapter we provide the related works in class-imbalance problem and cost-sensitive ANN, and literature about four application areas where the cost-sensitive classifiers have been used.

Classification algorithms perform the classification assuming that there is a balance between each of the classes in the training set without any assumption about skewed data which is a common problem in real-world classification data sets. This problem arises when there is a significantly large difference between the number of classes in the training set [23, 24]. This problem can occur in both binary and multi-class classification problems [25, 26]. However, we just focus on imbalance problem in the binary classification case where one of the classes is in majority with huge number of instances and one of them is minority class with significantly low proportion of the number of instances. In this case, regular statistical classifiers tend to memorize the majority class and predict them with high accuracy (almost 100%) and mis-classify most of the instances in the minority class (with accuracy 0 to 10%) [28]. This problem can cause a big cost for prediction models where the minority class is of more importance than majority one like the diagnosis problem [29] where the class for cancerous is in minority and there are less instances from this class and a misclassification of cancerous as non-cancerous would cost a huge amount for the prediction model.

Business problems deal with skewed data problem as well when there is a between-class imbalance and the minority class is more important to be predicted correctly than the majority class [30, 31]. As an example, in CC fraud detection, if a fraudulent transaction of credit card is misclassified as legitimate, the result will be the loss of the usable limit of the card which is extremely more than the cost of misclassifying a legitimate transaction as fraudulent one. However, in the training set of this problem which is mostly skewed data, there is an out-presence of legitimate transactions and under-presence of fraudulent ones and regular classifiers tend to memorize the legitimate ones and detect them correctly. To deal with this problem, a huge amount of research have been done to make the classifiers more sensitive to minority class. In the following, we review well-known cost-sensitive approaches which are applicable on the neural network classifier.

Cost-sensitive learning has been widely used in neural networks classifier specially the most practical version of ANN which is Multi-Layer Perceptron (MLP) model [5]. In the literature, four well-known approaches have been introduced to develop cost-sensitive ANN [32]:

- **Adapting estimated probabilities.** The most straightforward approach is to modify the outputs of the network which represent the probability of belonging to a class. In this approach, all of the precesses and outputs of the network remain same as original ANN but the probability estimation is more intensified for the class with high cost of misclassification. If $P(i)$ represent the probability of belonging to class for an instance in the original ANN, the following $P'(i)$ represents the modified output for this cost-sensitive approach:

  $$P'(i) = \frac{CostVector[i]P(i)}{\sum_j CostVector[j]P(j)} \tag{1}$$

  The advantage of this approach is that all of the outputs and network structure are left same as original case but only the estimated probabilities are modifies to be more strong for costly class and the backpropagation learning works as original ANN. Same approach has been used in [33] with another type of modification in the estimated probabilities. Gracia (2009) [34] reached a better performance by applying this approach to ensemble methods and using cross-validation technique.

- **Adapting the output of the network.** In this approach, the output of the network is corrected directly and instances belonging to costly classes are given more attention in the backpropagation algorithm. The targeted outputs are changed based on the following equation:

$$o'_j = \frac{CostVector[j]o_j}{max_i CostVector[i]} \tag{2}$$

  Where o'_j is the changed output of an instance belonging to class j.

- **Adapting the learning rate.** Tan et. al [35] investigated that modification of learning rate (η) has significant effect on weight adjustment. Also this approach is closely following Breiman et. al [36] and intensifies the learning rate for weights related to more costly instances (instances belonging to costly classes) to make the error function more sensitive to the instances from a class with high expected misclassification cost. In other words, this approach assigns high weights for high-cost examples from a class. The modified variable learning rate can be found from the following equation:

$$\eta(p) = \frac{\eta.CostVector[class(p)]}{max_i CostVector[i]} \tag{3}$$

  Experimental results on this approach have represented a significant improvement of ANN over original ANN [32].

- **Minimization of misclassification cost.** Finally in this approach the prediction error function is modified in a way that represents the cost of misclassification for all of the instances belonging to each of the classes. The error function is corrected here by defining a cost factor for each of the classes following the equation:

$$E = \sum_{p \in Examples} \frac{1}{2} \sum_{i \in Outputs} ((y_i - o_i).K[class(p), i])^2 \tag{4}$$

The cost factor K is defined as such the backpropagation algorithm works in its regular procedure.

$$K[i,j] = \begin{cases} CostVector[i] & i = j \\ Cost[i,j] & i \neq j \end{cases}$$

K behaves as a constant in backpropagation algorithm and this approach has been proved to be more consistent and result in a better performance in terms of minimum cost [32].

Although these approaches have resulted in a better performance over the original ANN, all of them are developed for the cases where there is a class imbalance problem in the data set or a different cost of misclassification for different classes of data. In our study, there is specific cost of misclassification for each of the instances (not classes) and these approaches in their original developments can not be used as they have assumed same cost of misclassification for all of the members of each class. Moreover, in all of the cost-sensitive neural networks developed up to now, the main objective is to minimize the total cost of misclassification. However in our study, we perform classification on data which have specific kind of profit for correct classifications and also cost of misclassification for each instance. Accordingly, we introduce two kinds of neural networks in this study which aim to maximize the total net profit earned by classification model.

In all of the application areas considered in this study, there are few researches which have dealt with cost-sensitive classification problem and there are very few work on profit-based classification. All of these researches are reviewed as follows:

- **Credit Card Fraud Detection.** Oxford concise dictionary defines the word ”fraud” by introducing it as a ”wrongful or criminal deception intended to result in financial or personal gain”. In credit card world, there are two kinds of frauds one takes place with new accounts and the other with existing accounts via stolen or copied cards or some other ways. USA Federal Trade Commission has published a report about different kinds of thrift in year 2014 which implies that the second major fraud complained is credit card fraud (17%) which comes after government documents benefit fraud (34%) [37]. Billions of dollars are spent for credit card fraud in other countries as well. These imply the importance of decision support systems to identify the fraudulent transactions and stop the usable limit of the cards from vanishing.

  There are lots of studies in the field of statistical learning for CC fraud detection which have used supervised learning systems such as decision tree techniques (ID3, C4.5, and C&RT) and Logistic Regression or more complex models such as neural networks and support vector machines [38, 39, 40, 41, 42, 43, 44, 45]. Most of these studies aim to maximize the number of correct classifications and some of them have studied the effect of cost-sensitive classification on the performance of the algorithms. However most of them have assumed class-based costs for instances and also no profit for correct classification of instances to find the total net profit which are investigated in this study.

  From profit/cost-wise point of view, in most of initial studies in this field, there was equal cost for each type of misclassifications. In later studies, researchers used class-based costs to different types of misclassifications and the reason is that in CC fraud detection, misclassified fraudulent instances (false negative) charge significantly more cost than the mislabeled legitimate ones (false alarms) [33, 46]. In real-world fraud detection problems there are individual costs for misclassified instances, and also there is case-based profit for correctly detected fraudulent transactions which in our study, is the usable limit of the card of the corresponding transaction. This profit is different from one transaction to another and can be considered as the profit of classification. This issue has been studied in some fraud detection problems [47, 48]. We have introduced two kinds of neural network classifier for CC fraud detection which perform their tasks based on individual net profit of each transaction where one of them work based on backpropagation algorithm and the other performs the classification using well-known metaheuristic algorithms to maximize the total net profit earned by implementation of aforementioned decision support system.

- **Churn Prediction.** Churn occurs when a customer decides to stop consuming the product or service offered by one company and use the same product from another company usually because the newer one offers a better price or service. Customer churn usually occurs gradually and if the company detects churner customers, it can take appropriate actions to keep them and save their life time value. Nowadays, in most of the companies particularly in service-based industries such as bank [49, 50] and telecommunication [51, 52, 18], there is a large number of competitors with numerous services which makes the customer retention a challenging issue.

  Literature shows that retaining an existing customer is easier and more profitable than finding new customers because finding new customer yields five to six times more cost than staying in touch with existing one [53, 54]. Moreover, by losing customers, the company begins to incur a cost associated with the

decreased sales [55]. Accordingly, small effort in customer churn prediction and then performing an effective action can lead to a huge profit for companies [56]. It is obvious that traditional models which aim to minimize the error function or maximize the accuracy of the model, result in a suboptimal solution in terms of total net profit. A lot of researches have been done on imbalanced data and cost-sensitive classifiers in churn prediction [31, 49] where the central issue is the class-based cost of misclassification and the individual costs and profits of customers (CLV an action costs) are given less attention. A recent research has been published regarding profit-based churn prediction where the researchers worked on a profit-driven measure to select a model which maximizes the profit[57]. However, in our research, the model is constructed based on each customer's net profit and the focus is on the model building step. The objective of this study is to maximize the profits earned by implementation of customer churn prediction system considering customers' individual profits and the cost of action which can be taken to retain the leaving customers.
Moreover, we introduce a profit function which takes the customers' individual profit and churn probability into account as independent variables. Also we use a fitting tool to analyze the effect of different promotion offers on customers with different churn probabilities and then to find the optimal retention promotion, we maximize the constructed function and find the corresponding retention promotions subject to limited budget of the company assigned to churn customer retention project.

- **Direct Marketing.** In the traditional mass marketing most of the companies have assumed that, all of the potential customers will take an action if they are informed about the new product. However, managers have known recently that not all of the potential customers can be the real target customers who will take a reaction regarding advertisements. Availability of large volume of data on customers and by the advent of new information technology tools, the focus of the customer relationship managers shifted from mass marketing to direct marketing which deals with customers' individual (case-based) information [58].

Oxford concise dictionary defines the "Direct Marketing" as "the business of selling products or services directly to the public, e.g., by mail order or telephone selling, rather than through retailers". In the banking industry this word means to individually sell bank's products to the targeted customers using their individual informations. By the adventure of direct marketing, most of the efforts of the managers are focused on dividing the customers in different segments and treat each of the segments independently with different tools and techniques.Few researches have been done using data mining techniques to segment the customers to different catalogs and take variable actions to each of them [59, 60].

Classification of buyer and not-buyer customers is a well-known topic in data mining field. Direct marketing problems which use classification techniques, aim to find the most likely and most unlikely customers to purchase the product and for this purpose, the customer relationship managers use customers' variant informations such as their demographics and other types of information such as purchase history. There is a key question in direct marketing, "Is this person a potential customer for our product?". The classification model will answer this question with binary numbers. The output of one means the customer will most likely purchase the product and the output of zero means he/she is not potential buyer of the product. Then the customer relationship management (CRM) decides on using which of the advertisement techniques to attract the customers who have been identified as buyers by the classification model.

In this study, we use a well-known and promising non-linear classification technique which has been widely used in the context of direct marketing, Artificial Neural Network (ANN) [61, 62, 4, 63, 64]. However, the central contribution here is that the objective of classification has been shifted from accuracy maximization into profit maximization considering individual profits and costs of the customers. Although in the direct marketing problems, maximizing the profit of the company is the primal objective of the studied models, this objective has been pursued indirectly and most of the used classifier techniques aim to maximize the number of correctly classified customers without dealing with their potential profit for the company. As an example, consider a customer which is actually interested in company's product but regarding his/her available balance, purchasing new product is a difficult decision for him/her and it may not happen. In the other hand, assume that there is a potential customer for the new product who has enough money to purchase it and with an advertisement and introducing the product, he/she will be more likely to purchase it. If the budget for the advertisement is limited who is the best choice between these two customers to contact? Our classification model will response to these types of questions considering the balance, target value or other monetary attributes of the customers to classify more profitable and most-likely buyers from other customers.

- **Credit Risk.** One of the critical problems for the financial institutions is to decide about customers' loan. This decision making problem is known as a "Credit Scoring" which employs classification techniques to distinguish high-risk customers from low-risk ones and based on this classification, the institution decides to grant the loan to him/her or not [65]. Obviously the financial institutions grant their loan to more reliable customers who are most likely to pay the loan's amount and accordingly, generate a profit for the institution by the revenue of the loan.

This classification problem was solved by domain experts' opinions in last decades but by the adventure of information technology tools and mass data recording, this task is performed using statistical or machine learning algorithms. Different types of classification techniques have been developed to do this such as traditional statistical methods like logistic regression [66], classification trees [67, 70], non-parametric statistical methods like k-nearest neighbor [68]and more complex models such as neural networks [69, 71, 70, 9, 72, 20, 73]. We have focused on different versions of ANN in our research where the focus is to maximize the profit instead of accuracy of prediction.

Distinguishing between good and bad loan applicants (Non-defaulters and defaulters) is the main objective of the researches performed in the credit scoring. However, in most of the business problems, managers are interested in models which emphasize not only the accuracy of the results but also the total profit of the model implementation [74]. In other words, managers, especially in the financial institutions, prefer to analyze different models based on monetary issues and identify the costs and profits instead of just accuracy based analysis. Therefore, there is a necessity to define a profit-driven measurement in credit scoring as well and use it to rank the models based on total savings which will be throughly discussed in this study.

With the same reasoning, in the decision support model building issue, most of the business administrators emphasize the models which takes the profits and costs of model implementation into account. Pursuing this objective, a profit-driven approach is needed which deals with monetary issues in the model building step and aims to maximize the total profit of the company using classification models. We will do this by introducing high-performance classification algorithms such as ANN which has promising result in classification problems and change them in the way that they perform their task based on maximizing the total profit without significant decrease in the accuracy of the model. The main contribution of this study is not only profit-based approach in model building step but also taking individual profits and costs of loan applicants into account and perform a case-based profit-driven analysis on customers.

So far, credit scoring models have been used indirectly to save the profit of the financial institutions and by increasing the accuracy of the model, most of the researches have been done to maximize the savings [75]. However, in this study, we directly use profit-based approach to build a model based on individual profits and costs of customers and then reach the objective of maximum profit earned by model implementation. We then compare our proposed model with traditional accuracy-based classifiers in terms of both accuracy and profit.

CHAPTER III

METHODOLOGY

In this section we focus on the most popular class of ANNs which are called "Multi-layer Perceptrons" and their structure and learning process using back-propagation and metaheuristic algorithms.

3.1 Backpropagation-based Artificial Neural Network (ANN)

Artificial neural network is the mathematical representation of information processing in biological neural network in human body [76, 78, 77]. Neural networks have two major types: single-layer perceptron and multi-layer perceptron (MLP). Single layer perceptron is a linear discriminant and it is limited in mapping the feature space and they are the first generation of ANNs. In particular we restrict our attention to a specific class of neural networks which have been proved to result in a promising models in terms of statistical performance; multi-layer perceptron [3, 5].

Multi-layer perceptron which has been used in this study, has three kinds of layers where information is transferred between them with weighted connections: input layer, hidden layers and output layer. Input layer has some neurons which get the inputs from data set multiplied by their weights and transfer them to the hidden layer with some other set of weights. Hidden layers also have neurons which get the information from input layer and map them between zero and one by using sigmoid function in this study. Then, the hidden layers transfer their outputs to the output layer with weighted connections. In each layer, there is some bias for each neuron to make the model easier to predict the exact target. If the output neurons have linear function, the target will be a real number and it is an appropriate function for regression. Nonetheless, as the problem proposed in this paper is the prediction of two classes, cases (positives) and non-cases (negatives), the model has to be a classification model and here the sigmoid function which is represented below, is used instead of simple linear function.

$$y = \frac{1}{1 + e^{-z}} \tag{5}$$

Here y is between zero and one and z comes from the following formula:

$$z = b + \sum_i x_i w_i \tag{6}$$

Where b is bias, x_i is the input vector for instance i and w_i is its corresponding weight. Typically sigmoid function is used in neural network because it has nice derivatives which simplify learning procedure. The following figure (Figure 1) shows the behavior of sigmoid function as a transferring and learning function:

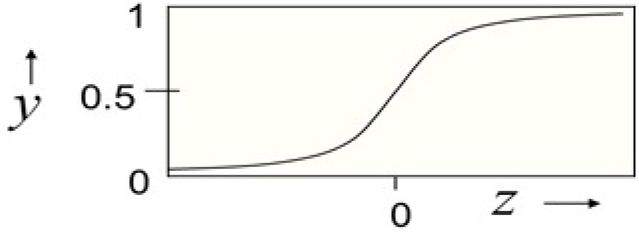

Figure 1: Sigmoid function

In neural network representations, the bias is often given the value of one to be able to write it in vector representation as follows:

$$z = \sum_i w_i x_i = \mathbf{w}^T \mathbf{x} \tag{7}$$

Here $\mathbf{x}$ is input vector and $\mathbf{w}$ represents the weight vector which the network is going to learn and predict the target as the error of the prediction is in its minimum amount.

The network uses the chain rule to get the derivatives needed for learning the weights of a logistic unit. To learn the weights which minimize the error function, we need the derivative of the output with respect to each weight.

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - y_n)^2 \tag{8}$$

$$\frac{\partial y}{\partial w_i} = \frac{\partial z}{\partial w_i}\frac{\partial y}{\partial z} = x_i y(1-y) \tag{9}$$

$$\frac{\partial E}{\partial w_i} = \sum_n \frac{\partial y_n}{\partial w_i}\frac{\partial E}{\partial y_n} = -\sum_n x_i^n y_n(1-y_n)(t_n - y_n) \tag{10}$$

x_n,y_n,t_n are respectively the n^{th} input, its corresponding output and its target value. Network without hidden units are very limited in the input-output mappings they can model hence, in complex data sets, adding a hidden layer makes them much more powerful. A neural network with just one hidden layer has been visualized in figure 2 which depicts some neurons in each of three layers. The hidden layer and output layers use sigmoid functions so the output of the network will be a value between 0 and 1 (the probability of taking one for the output).

The idea behind the back-propagation is using error derivatives with respect to hidden activities instead of using only desired activities to train the hidden units. Each hidden activity can affect many output units and can therefore have many separate effects on the error. These effects must be combined. Error derivatives can be computed for all of the hidden units efficiently at the same time. Once we have the error derivatives for the hidden activities, it is easy to achieve the error derivatives for the weights going into a hidden unit.

The following equation represents the general sum of squared errors (SSE) and its derivative which has been used in many research to back-propagate in the network using gradient descent. Each weight is updated using the following equations:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - y_n)^2 \tag{11}$$

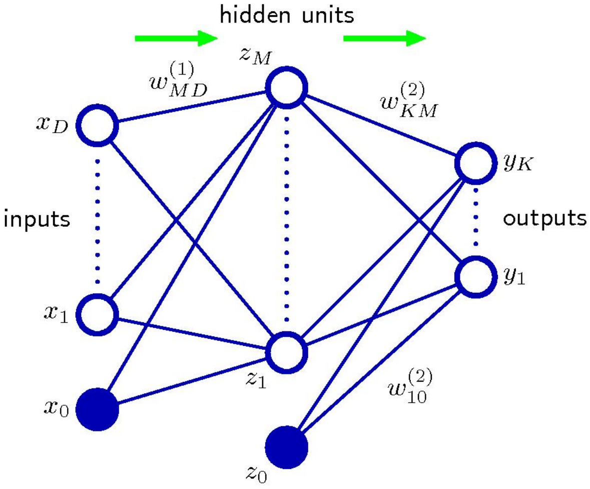

Figure 2: ANN with one hidden layer

$$\frac{\partial E}{\partial y_i} = -(t_i - y_i) \tag{12}$$

$$\mathbf{w}^{T+1} = \mathbf{w}^T - \eta \bigtriangledown E(\mathbf{w}^T) \tag{13}$$

Where $\mathbf{w}^T$ and $\bigtriangledown E(\mathbf{w}^T)$ indicate the vector of weights and vector of derivatives of the weights in T^{th} iteration .respectively, and η represents the learning rate.

3.2 Profit-Sensitive Neural Network

In our business problems, our main goal is to correctly classify the profitable instances as much as possible with minimum decrease in the accuracy of detecting other instances (i.e., less profitable ones). For this reason, an indicator has been used in the error function to make the algorithm more sensitive to high profitable instances. Accordingly, we used a multiplier to intensify the individual penalty of profitable false negatives (in CC Fraud, fraudulent misclassifications whose usable limit is more than average).

The indicator should indicate the profitable (important) instances which is the Usable Limit in the context of credit card fraud, the customer revenue (or balance) in direct marketing. Thus, indicator has been defined as:

$$I(A_j) = \begin{cases} 1 & \text{if } A_j \geq \overline{A} \\ \\ 0 & \text{otherwise} \end{cases}$$

Here A_j is the individual profit of instance j and $\overline{A}$ is the total average of instances' individual profits. In this scenario, multiplier in the error function (penalty) can be defined as profitability of i^{th} instance which is sensitive to its individual profit. Consequently, the error function can be defined as:

$$E(\mathbf{w}) = \sum_{j \in train} (\mathbf{t}_j - \mathbf{y}(x_j, \theta))^2 * (\frac{A_j}{\overline{A}})^{I(A_j)} \tag{14}$$

As low profitable instances are not going to be multiplied by a value, we can assume that they will be multiplied by one. We aim to make a connection between instances and their individual penalties to ensure that profitable ones will be classified as much as possible.

We can consider this modification from another point of view. A learning rate is a user-defined value to determine how much the weights of examples can be modified at each iteration. We can assume that the learning rate has been modified to assign an appropriate individual penalty for each example and penalize the misclassified important examples considering their individual importance. Moreover, a cost matrix (net profit matrix) can be represented to show the individual costs and profits (Table 3). Where A_i and A'_j are profits of true positive and true negative and C_i , C'_j are costs of false negative and false positive, respectively.

Table 1: Individual cost matrix (net profit matrix)

		Predicted class	
		case	non-case
Actual Class	case	A_i	C_i
	non-case	C'_j	A'_j

- **PNN using logarithm (LOG-PNN):**

  Neural network is very sensitive to any multiplier in error function and if it is a large value the error function will be unstable and if it is a small value, the rate of learning will be decreased considerably.

  As the ratio $\frac{A_j}{\overline{A}}$ in the previous version can give out large values it may cause instability in the model, so for the sake of making the multiplier not a very large value, we can use logarithm function in an alternative version. Hence, the penalty for each instance can be defined as:

$$P_j = \log(\frac{A_j}{\overline{A}} + 1) \tag{15}$$

  The value of one inside the logarithm guarantees that the output will always be positive as the ratio $\frac{A_j}{\overline{A}}$ is a positive real number. The penalty function and weight updating equations can be expressed as:

$$E(\mathbf{w}) = \sum_{j \in train} (\mathbf{t}_j - \mathbf{y}(x_j, \theta))^2 * (\log(\frac{A_j}{\overline{A}}) + 1)^{I(A_j)} \tag{16}$$

$$w_j^{T+1} = w_j^T - \eta \bigtriangledown E_n(w_j^T) * (\log(\frac{A_j}{\overline{A}}) + 1)^{I(A_j)} \tag{17}$$

  This scenario can make the network more stable when the profitability ratio has a large spread, for example, in the CC fraud data set used in this study, the usable limit attribute is considered as profit based metric. The range of this attribute is $[0, 99714]$ with average of 1816. Accordingly the profit ratio $\frac{A_j}{\overline{A}}$ has the range of which contains very large values as a multiplier to the error function. If we use logarithm function, the penalty will have the range $[0, \log(54.9)] = [0, 1.7474]$.

- **LOG-PNN without using indicator (LPWI):**

  In this version we use the modified version of SSE without the indicator. Therefore, not only misclassified profitable instances but also all misclassified ones are penalized proportional to their profitability. The error function is as follows:

$$E(\mathbf{w}) = \sum_{j \in train} (\mathbf{t}_j - \mathbf{y}(x_j, \theta))^2 * (\log(\frac{A_j}{\overline{A}}) + 1) \tag{18}$$

- **LOG-PNN without Average (LPWA):** This version uses the logarithm of each instance's profit as its multiplier in the error function:

$$E(\mathbf{w}) = \sum_{j \in train} (\mathbf{t}_j - \mathbf{y}(x_j, \theta))^2 * (\log A_j)^{I(A_j)} \tag{19}$$

- **Weights of modified Fisher (MF) [48]:**

  This version has been benchmarked from a recent profit-based study in fraud detection context which was the best choice to generate weights for Fisher Discriminant classifier. In this version, there is no indicator for profitable instances where all of the instances are given a weight related to their potential profit. We use this approach as a benchmark to compare its performance with other weight generation methods. The error function for this version of ANN is as follows:

$$E(\mathbf{w}) = \sum_{j \in train} (\mathbf{t}_j - \mathbf{y}(x_j, \theta))^2 * ((\frac{A_j}{\overline{A}}) + 1)^{\frac{1}{2}} \tag{20}$$

- **MAX-PNN:**

  This version uses the Kukar's way of weight generation [32] and it gives different weights for different instances considering their profit of correct classification (originally cost of misclassification). The only difference between this approach and Kukar's approach is that, here all of the instances have individual cost of misclassification where in the original paper they studied class-based costs.

$$E(\mathbf{w}) = \sum_{j \in train} (\mathbf{t}_j - \mathbf{y}(x_j, \theta))^2 * (1 + (\frac{A_j}{\max_j (A_j)})) \tag{21}$$

Note that as expressed previously, accuracy and other performance metrics based on accuracy are not suitable for cost-sensitive or profit-based classification models. We have used four different measures to compare the models, where two of them are accuracy-based measures which are: Accuracy and True Positive (TP) rate. Next two measures are cost/profit based measures. "Saving" measures the amount of profit in each model with threshold 0.5. The "Net profit in top n" (n is the number of actual positives in the test set) evaluates net profit when the cutoff point is the score of top nth instance. This measure has an advantage that doesn't care about the number of total positives in each classifier, but it gives more importance to the actual number of positives detected in the first top positives in each model and sums their net profits.

- **Class-based cost-sensitive ANN:**

  To compare performance of our presented model with traditional cost-sensitive approach, we have constructed a class-based cost and give more cost to the minor class. The minor class will receive more cost than major class to let the model reach a balance between the two classes. The magnitude of the cost used for the minor class is inversely proportional to its ratio regarding the major class. For example if in one of the data sets, the positive instances ratio is 1/10 and negative ratio 9/10, the cost of misclassification of a positive instance (false negative) will be 10 times the cost of misclassifying a negative instance (false positive).

  In this chapter, six data sets have been used to investigate how the proposed models work and all of the experiments have benefited from the Levenberg-Marquardt algorithm to train the neural networks. Note that we used one hidden layer ANN with three hidden units and there is just one output unit in the output layer. Also we used default value of 30 for the number of epochs. Two of data sets are new real-life bank credit card data which have been acquired from two well-known Turkish banks. One of them is bank direct marketing data sets which has been gathered from a Turkish bank. The fourth dataset is a bank churn data and the last one is bank credit risk data both acquired from a Turkish bank. Description of each data set and the comparison of the models in the aforementioned data, has been throughly discussed in the related application section.

In the empirical study of each data, the data set has been divided in a way that 0.7 proportion is used to train the model and 0.3 is used to test the trained model. In all of the data sets, we have a profit-based attribute which individualize the penalties in each case. The ANN model automatically extracts a validation data set (15%) from training data set in order to avoid over-fitting.

Empirical results of testing the seven presented models, class-based cost-sensitive ANN and also the original neural network has been analyzed thoroughly. In all of the models the train sets and test sets are the same, however, as the initial weights are generated randomly in neural network, each of the models has been run thirty times and the average of runs plus their standard deviations are considered as classifiers' final performance. Moreover, following expression demonstrates how to calculate the amount of net profit for each model:

$$NP = \sum_{i=1}^{N_{TP}} (A_j - c) + \sum_{k=1}^{N_{FP}} (-c) \tag{22}$$

Where c is the fixed cost of action (cost of contacting the customer) and N_{TP} and N_{FP} indicate the number of true positives and false positives, respectively. As mentioned above, A_i is the amount of profit gained when the instance i is classified correctly. In context of credit card fraud prediction it is the usable limit of the corresponding card used to make the transaction i. Furthermore, there is a fixed cost of applying the decision support system, but since it is fixed and has to be added for all models, this is not taken into consideration in the comparison procedure. The threshold is the number of cases (positives) in test set to show that in the top most probable instances, which of the classifiers is successful. We have benchmarked two well-known classifiers one from tree-based methods (C4.5) and other from rule-based classifiers (Naive Bayes) to compare the results of PSNN models with these classifiers as well.

3.3 Profit-Based Neural Network trained by metaheuristic algorithms

In this section we benchmark well-known metaheuristic algorithms which have had promising results in most of the optimization problems to maximize the net profit function instead of minimizing the prediction error, which is the objective of regular ANN models. These metaheuristic algorithms are Particle Swarm Optimization (PSO) [79], Genetic Algorithm (GA) [47], Artificial Bee Colony (ABC) [80] and Migrating Birds Optimization(MBO) [81]. Then, we compare the results with original ANN in both accuracy-based and profit-based measures.

The feed-forward equations in metaheuristic-based ANN are similar to the ones of the original ANN. As we have used one hidden layer with three hidden units, all of the feed-forward equations are as following:

$$u_{11} = \sum_{i=2}^{n} \theta_{i1} X_{i1} + \theta_{01} \tag{23}$$

$$u_{12} = \sum_{i=2}^{n} \theta_{i2} X_{i} + \theta_{02} \tag{24}$$

$$u_{13} = \sum_{i=2}^{n} \theta_{i3} X_{i} + \theta_{03} \tag{25}$$

Equations 23 to 25 represent the first part of three hidden units. In this step, for each hidden unit, all of the inputs are multiplied by their corresponding weights and a bias is added to the result.

$$o_{11} = \frac{1}{1 + e^{-u_{11}}} \tag{26}$$

$$o_{12} = \frac{1}{1 + e^{-u_{12}}} \tag{27}$$

$$o_{13} = \frac{1}{1 + e^{-u_{13}}} \tag{28}$$

These set of equations 26 to 28 indicate the second parts of hidden units which gets the weighted average of inputs and apply the sigmoid function to calculate the output.

$$u_2 = \delta_1 + \sum_{j=1}^{3} \vartheta_j o_{1j} \tag{29}$$

Then as shown by equation 29, the outputs of hidden layers are multiplied by their corresponding weights and then summed up to determine the output value (u_2) of the output layer.

$$\hat{y} = \frac{1}{1 + e^{u_2}} \tag{30}$$

Finally sigmoid function is applied to determine the output (equation 30). By this equation we ensure that the output of the network will be a value between zero and one which corresponds to the probability of being positive for each instance. In statistical learning this output is called as score of the instance which is popular in most of the data science studies.

We apply the metaheuristic algorithms to solve the aforementioned equations with the objective of maximum net profit in each of the contexts. Metaheuristic algorithms have to maximize the objective function subject to the ANN feed-forward equations.

3.4 Dependencies between the parameters in profit-based Churn prediction

Profit of churn prediction models are different from other class of problems like fraud detection and credit scoring and there is an uncertainty in earning the profit of each of the customers. In customer relationship management problems, after the model is implemented and target customers receive the promotion offers, the company has to wait to receive a response from the customer and this response depends on different variables of market, company's action and customer's tendency. For example in churn prediction, if the classification model has correctly detected a churner customer, the company has to take an action to retain him/her. The churn probability of the customer depends on the offer and maybe some other issues like opponent's offer or a personal preference. Then, the profit of retaining that customer is a probabilistic issue which has to deal with the customer's churn probability. The same case is applicable in direct marketing problems. In this section we discuss about the expected profit that can be earned from implementation of churn prediction model.

Expected profit of churn prediction

In the Profit-based classification approach, the first important point to be considered is to have a base scenario which represents the system without using the recommended models. In the customer churn prediction, the base scenario is that there is no prediction system and churner customers are not detected by the company and leave. Consequently, the company loses all of the potential profit which could be earned from those customers. Accordingly, all of the profits and costs in this study have been calculated based on base scenario which assumes that there is no churn prediction system. For instance, for the correctly detected churner customers, we consider a profit which is a proportion of his/her life time value and also a cost which is related to the retention promotion made for that customer. These profits and costs are not counted in the base scenario. Take another instance and consider false negative instance in the system, those who are actual churners but predicted non-churners in the prediction system. Although we lose that customer's potential profit, we do not consider it as a cost because in the base scenario this customer's profit was lost as well and we have no added cost comparing with the base scenario.

According to this base scenario and individual profits and costs, we can assume a net profit matrix for each instance. Assume that instance is an actual churner (positive) instance and is an actual non-churner (negative) one. The individual net profit matrix is as following:

Table 2: Individual net profit matrix for churn prediction

		Actual	
		Instance i (churner)	Instance j (non-churner)
Predicted	Churner	$CLV_i(S_{i,old} - S_{i,new}) - c_i$	$-c_j$
	Non-churner	0	0

CLV_i is the customer life time value for the instance (customer) i , $S_{i,old}$ is the score which comes from the model and it is assumed to be the predicted churn rate of the customer i, $S_{i,new}$ is the churn probability of the customer i after making an promotion offer to him/her and it is calculated based on equation 31. c is the fixed cost of offer which will be paid for all of predicted (true and false) positive instances. The profit will be the change in the churn rate multiplied by the CLV of the customer. We assume that in all of the cases. In other words, after the retention promotion is given, the customer's churn rate will be decreased or remain the same.

Calculating the amount of net profit for churn prediction needs a post-processing analysis and we have to consider the effects of each incentive offers on the predicted churner customers. For this purpose, we benefit from domain experts' opinion to quantify the effects of each incentive offer for different values of churn probability. For each of the possible churn probabilities, the expert gives the new changed customer churn probability assuming a specific kind of incentive offer. For different values of churn probabilities and also variable incentive offers, we made an approximation with fitting tools and found the relationship between them. The result revealed that the relationship is like a sigmoid function with different parameters for each of the churn probabilities. The new churn probability is calculated as:

$$S_{new} = \frac{2S_{old}}{1 + S_{old}^{-x}} \tag{31}$$

In this equation S and x represent the churn probability of each customer and cost parameter of incentive offer.

We consider different incentive offers as input of sigmoid function and the range as its output which shows the previous (initial) and new churn probability of the customer after receiving an incentive offer. The relationship for some examples with different churn probabilities are depicted in the figure 3:

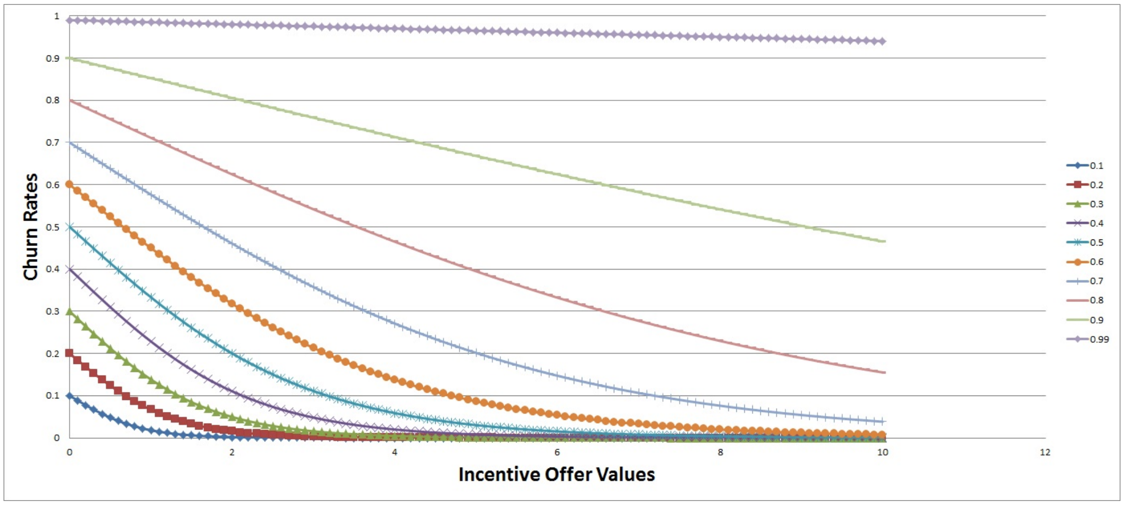

Figure 3: Sigmoid relationship between incentive offer cost and customer churn rate

The above figure is of interest in itself, because it represents the behavior of different types of customers in terms of churn probability regarding variable incentive offers. The reaction of customers with high churn probability shows that their churn probability is decreasing very slightly and slowly comparing with other ones and small offers cannot make significant change in their decision to leave or stay at company. On the other hand, more loyal customers who has lower churn probabilities has better reactions regarding even small incentive offers. This relationship helps us to appropriately find the net profit of churn prediction model regarding different types of incentive offers.

Moreover, to boost our approximation about customers' profitability, we made another approximation using different life time values of customers to show the behavior of profit function regarding the variable incentive offers. In this approach we not only use the churn probability of the customer as our input variable, but also consider his/her particular profit (life time value) to find the total profit of each customer when selecting a specific kind of incentive offer to make for him/her. The result is a relationship between incentive offers and customer's churn probability, but the difference here is that the relationship shows the amount of money earned for customers with particular churn probability when selecting different incentive offers. The relationship is depicted in the following figure for two example customers from data set with different churn probabilities and life time values. The first instance is a customer with churn probability 0.9 and life time value of 35000. The second one is a customer with churn probability 0.6 and life time value of 10000.

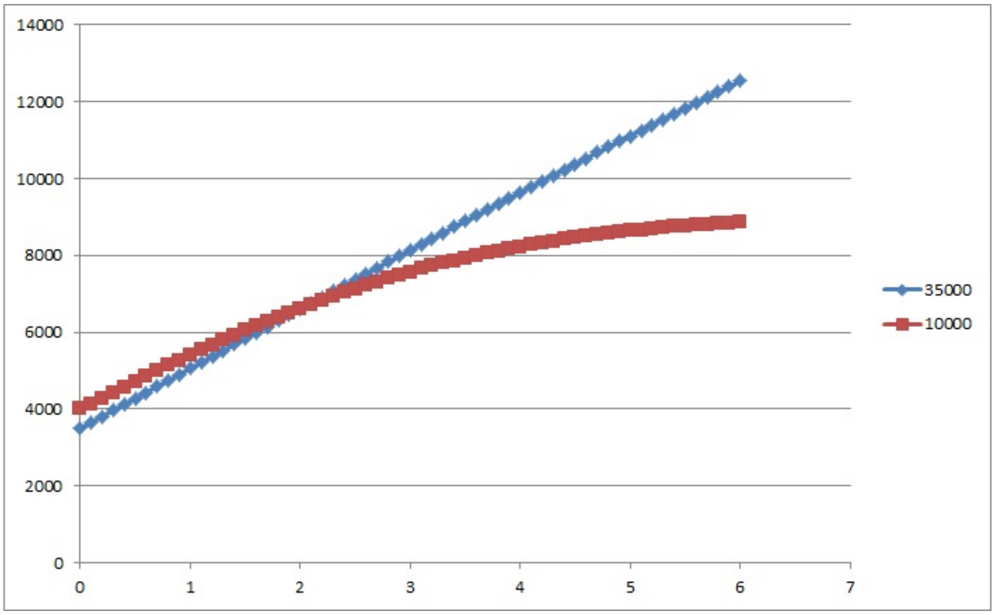

Figure 4: Relationship between incentive offer cost and customers retention profit

Figure 4 shows that, there is possibility to a less loyal customer to have more profit for company than a more loyal one for a specific incentive offer. For example in this comparison, if company gives offers which costs more than 2.1 unit money, the customer with higher churn probability will have more profit than the other one. This result confirms the results for Reinartz and Kumar (2000) which shows that loyal customers are not necessarily the most profitable customers to the company. Selecting an incentive offer depends on company's budget which has been assigned to customer relationship management projects and a budget constraint has to be considered for this purpose. The incentive offer policy is an issue in which the managers of the company have to make decision for. For example some companies prefer to give offers with fixed amount of money for all of the customers (fixed-incentive). In this scenario, it is necessary to find a point in the X-axes (incentive offer cost) in which the total profit is maximized. Some others prefer to give variable-cost incentive offers for each of the customers considering their profit and churn probability (variable-incentive). In the latter scenario, we have to find a maximum point in total profit for each of the customers and use the corresponding incentive offer for each of them.

3.4.0.1 One-to-all retention promotion

If the financial institution decides to give same retention promotion to all of the targeted customers, the total profit for customer retention has to be calculated and all kinds of promotion costs have to be considered and the promotion which maximizes the total profit has to be selected for all of the customers. The total profit here means the sum of all individual profits of each of the customers. Also, budget constraint has to be considered which means the total cost of retention promotions has not to exceed the budget assigned for this project. The formulation of this scenario is as following:

$$MAX(P) = \sum_{i=1}^{n}(S_i - \frac{2S_i}{1+(S_i)^{-y}})CLV_i - \sum_{i=1}^{n} c_y \tag{32}$$

$$\sum_{i=1}^{n} c_i \leq B \tag{33}$$

3.4.0.2 Stepwise Retention Promotion

In this scenario, there are finite kinds of retention promotions like the previous scenario but here, one customer can get different retention promotion than other one. Therefore, there are some types of promotions available for each of the customers and the type of offer selected for one customer depends on the total profit earned from all of the customers. Here we face a well-known type of optimization problem called integer programing and there are variations of algorithms to solve this problem. The formulation of the problem is as following:

$$MAX(P) = \sum_{j=1}^{J} \sum_{i=1}^{n} (S_i - \frac{2S_i}{1 + (S_i)^{-y}}) CLV_i - c_j y_{ij} \tag{34}$$

$$\sum_{j=1}^{J} \sum_{i=1}^{n} c_j y_{ij} \leq B \tag{35}$$

$$\sum_{j=1}^{J} y_{ij} \leq 1 \; for \; \forall i \in n \tag{36}$$

$$y_{ij} \in \{0, 1\} \tag{37}$$

$$y_{ij} = \begin{cases} 1 & \text{for customer } i \text{ the promotion } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

3.4.0.3 Continuous Retention Promotion

In this scenario, each customer can get a specific retention promotion depending on his/her total profit for the company. Therefore, to find the best promotion for each customer, we have to maximize his/her specific profit function subject to budget constraint.

$$MAX(P_i) = (S_i - \frac{2S_i}{1 + (S_i)^{-y}}) \times CLV_i - c_i y \ \forall i \in n \tag{38}$$

$$\sum_{i=1}^{n} c_i \leq B \tag{39}$$

But the important point is that the total cost of customer retention (sum of incentive offer costs for each of the customers) has not to exceed the budget limit.

In our churn prediction study, as the policy is to assign fixed offers for all of the predicted positive instances, there is no budget limit and assignment problem regarding this issue. However if the financial institutions or product companies have the policy of assigning variable offers for different customers, the importance of this modeling will be more obvious.

The other application where this approach can be used is to identify the threshold of churn prediction by maximizing the total profit subject to budget limit. This model can be written such that the threshold be the independent variable of the model.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Credit Card Fraud Detection

The CC Fraud data set has been gathered from a well-known Turkish bank and it includes 9388 transactions' information made by customers of a bank with a balance of 9 to 1 (legitimate transactions to fraudulent ones); 939 of them are fraudulent cases and the rest are legitimate ones. The number of attributes was 102 which is 27 after preprocessing. Note that we applied decision tree (C4.5) first and then trained ANN models with those variables used in DT. Accordingly, 27 variables are used for the training of the fraud data set.

By correctly detecting a fraudulent transaction the model can save its corresponding card's usable limit and by a false alarm there is fixed cost of action which is the cost of a short message or telephone call to the customer. UL_i indicates the usable limit of the card of the i^{th} transaction and c is the contact cost which is fixed for all cases. Considering this issue, benefit matrix is as shown below:

Table 3: Net profit matrix for fraud detection

		Actual	
		Case (fraudulent)	Non-Case (legitimate)
Predicted	Case (fraudulent)	$UL_i - c$	$-c$
	Non-Case (legitimate)	0	0

Table 4 represents all of the information about data sets and ANNs used in fraud detection problem.

In the test set of this data set we have overall 3130 instances, where 313 of them (10%) are fraudulent transactions and 2817 of them are legitimate ones. As the number of positive instances are 313 in the test set, the threshold here has been

Table 4: Fraud data set and ANNs used in this study

Name	No. of features	Size of samples	Training ratio	Testing ratio	Validation ratio	No. of hidden layers	No. of hidden units
Credit Card Fraud	27	9388	2/3	1/3	15% of training set	1	3

chosen the 313^{th} instance's score (top 10%) to analyze classifiers' performance. In the context of credit card fraud, the most important profit-based attribute is the usable limit of the card used for the corresponding transaction (instance). For transaction i, the usable limit of the corresponding credit card is recorded as UL_i. If we correctly detect fraudulent cases, we save their usable limit subject to a cost of contact. Let us consider the base scenario as the case where all transactions are supposed to be legitimate. It is a common approach for evaluating the profit of applying data mining algorithms and it considers the change in profit which the model makes. Then, the total net profit (TNP) that can be obtained from the implementation of the classification model will be:

$$TNP = \sum_{i=1}^{N_{TP}} (UL_i - c) + \sum_{k=1}^{N_{FP}} (-c) \tag{40}$$

Where, UL_i indicates the usable limit of the card of the i^{th} transaction and c is the contact cost which is fixed for all cases.

4.1.0.1 Profit-Sensitive ANN

Table 5 illustrates performance of PSNN models for the credit card fraud dataset.

As Table 5 demonstrates, considering true positive rate, original ANN and PNN models outperform others. High accuracy of the PNN model shows that this version of ANN represents that its weights overcome the imbalance problem of fraud data set even better than CNN. However, in terms of total net profit,LPWA (the version of LOG-PNN which has not the average function) has outperformed others. We confirmed that the superiority of LPWA over ANN and cost-sensitive ANN (CNN) and other models is statistically significant based on a t-test with $\alpha = 0.05$. Th results for

Table 5: The results of all PSNN models for the fraud data set

	Threshold=Top 50%					
Model	True positive rate %			Total Net Profit %		
	Min	Mean	Max	Min	Mean	Max
ANN	71.25	**73.42**	76.68	66.18	84.54	89.70
DW	17.89	63.83	73.80	35.30	78.73	90.25
PNN	71.57	**73.83**	75.08	62.09	81.33	89.89
LOG-PNN	67.09	71.63	74.76	65.16	82.71	91.16
LPWI	2.56	9.27	15.97	13.73	20.27	40.60
LPWA	67.73	72.33	74.76	74.44	**87.74**	91.41
MF	49.20	70.83	76.04	48.87	83.37	90.37
PNN-MAX	40.58	70.96	76.36	67.99	83.40	90.75
CNN	43.70	66.71	75.08	62.28	79.66	89.70
DT	70	70	70	84.4	84.4	84.4
NB	67	67	67	84.55	84.55	84.55

LPWI shows that without indicator for profitable transactions, the generated weights make error function very unstable. PNN, LOG-PNN, MF,PNN-MAX and CNN can reach an acceptable profit but these versions are very unstable and the range of the results for profit(i.e., $Max - Min$) are large. LPWA is the best trade-off between accuracy and profit as it has acceptable accuracy and also highest amount of profit for fraud detection problem.

4.1.1 Profit-based ANN trained by metaheuristic algorithms

In this subsection, the objective function of metaheuristic-based ANN is the total net profit that can be obtained in the test set by correctly classifying most of the important transactions. By the word of "Total Net Profit" we mean the sum over all usable limits of the cards of corresponding transactions which can be saved by the model if detected correctly minus action cost which is fixed cost for all of the actions.

$$TotalNetProfit = \sum_{k=1}^{N_{TP}} UL_k - c + \sum_{j=1}^{N_{FP}} c \ , \ k \in \{x_i | y_i = 1, t_i = 1\} \tag{41}$$

Here, UL_k is the amount of usable limit for transaction k which is a correctly detected positive instance where there are total N_{TP} actual positive instances in the test set and N_{FP} actual negatives which has been classified as positive. y_i and t_i are the i^{th} instance output and target value, respectively. We measure the total net profit of each model using the usable limit of all of the actual fraudulent instances. We divide the above mentioned amount by all of the usable limits of fraudulent transactions which could be detected by the perfect model (a model with 100% accuracy) to find the proportion of net profit which can be earned by the implementation of the model. The results for original ANN and four metaheuristic algorithms used to train the neural network and maximize the total net profit are presented in table 6. Moreover, for each of the pairs of models a t-test (after F-test) has been used to show the significance of the results.

Table 6: The results of all metaheuristic-based ANN models for the credit card Fraud data

	Threshold=Top 10%						P-Value				
Model	True Positive rate %			Total Net Profit%							
	Min	Mean	Max	Min	Mean	Max	**ANN**	**MBO-ANN**	**PSO-ANN**	**GA-ANN**	**ABC-ANN**
ANN	55.90	71.38	75.40	66.18	84.54	89.70	-	-	-	-	-
MBO-ANN	72.20	74.71	76.40	85.83	**89.04**	91.62	0.049	-	-	-	-
PSO-ANN	35.14	51.15	65.49	70.80	84.69	91.44	0.481	0.044	-	-	-
GA-ANN	60.38	67.95	71.56	82.43	87.70	90.91	0.120	0.125	0.130	-	-
ABC-ANN	69.53	**75.16**	77.03	64.37	77.06	82.30	3.11E-11	3.85E-16	2.54E-15	2.544E-15	-

Table 6 shows the average performance for all of the metaheuristic-based ANNs and original ANN with their maximum and minimum performance in the runs. If the accuracy is our concern, ABC-based ANN has the largest true positive rate and the second most accurate model is MBO-ANN with a little difference from the ABC-ANN. The neural network trained by the MBO and GA have the best performance in Total Net Profit (%). As the p-value amount between MBO-ANN and GA-ANN shows, there are no significant difference between them in $\alpha = 0.05$ level. However among these two models, MBO-ANN is more robust classifier in terms of profit as it has larger amount of Minimum TNP compared with GA-ANN's minimum performance and choosing MBO-ANN is more reasonable than others if the profit is our main concern.

Comparing best algorithms in tables 5 and 6, it can be seen that the ANN trained by the MBO could perform significantly better than the profit-sensitive neural network versions. Note that as in the metaheuristic-based ANNs the objective function has not an expression about the difference of predicted output and target value for each of the instances and all of these models assign zero and one for the outputs. But in CC fraud problem this can not be an important problem because after classification, we take action for all of the positive predicted instances and the amount of the score (output) is not important which is not the case in churn prediction and direct marketing. Accordingly, the most successful ANN in terms of profit is the MBO-based ANN which is the most robust model in terms of total profit for the CC fraud data set.

The proposed decision support system for credit card fraud detection is appeared as follows:

After training the CC fraud detection model, for the future transactions, the DSS gets information about transaction for data center and it sends the useful information for DSS model and its output is a score for the transaction. Decision center compares the score with threshold and if the score exceed the threshold it accept that the transaction is fraudulent and take the appropriate action. If it was not classified as fraudulent (score less than threshold) then there will be no action about the corresponding transaction.

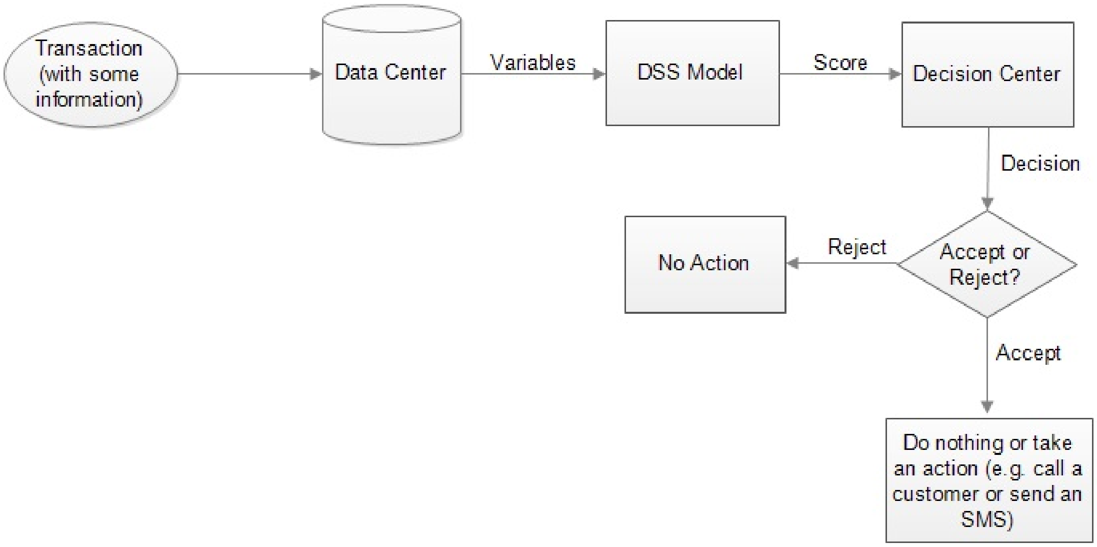
Transaction (with some information)
Data Center
Variables
DSS Model
Score
Decision Center
Decision
Accept or Reject?
Reject
No Action
Accept
Do nothing or take an action (e.g. call a customer or send an SMS)

Figure 5: System model for proposed Decision Support System (DSS) in CC fraud detection

4.2 Churn Prediction

In the churn prediction, the customer lifetime value can be considered as the most important profit-based attribute for all of the instances. Accordingly here, we use this attribute to represent the amount of profitability for each of the customers.

The profit of the churn prediction model can be calculated by considering a base scenario same as other profit-related business problems. The base scenario here is a CRM system without churn prediction model. By assuming the base scenario, we can calculate our model's profit as the change which it makes in the total profit of the company by performing the customer churn prediction. If we exclude the churn prediction model, some of the customers those were churners, will leave the company and then there is a cost of losing those customers in base scenario. Therefore, if the churn prediction model retain a customer which was actually a churner one, the model can save his/her lifetime value and this is a profit for the presented system. However misclassification of a churner is not actually a cost for our model because he/she is assumed to leave the company in the base scenario as well and there is no change in profit regarding this case. Also we have to assume a fixed cost for contacting the customers and a cost of retention promotion. Then the total net profit for our model will be the saved CLVs minus the total cost of contact for positive classified instances (true positives and false positives). Considering this issue, benefit matrix is as shown below:

Table 7: Net profit matrix for direct marketing

		Actual	
		Case (churner)	Non-Case (Non-churner)
Predicted	Case (Buyer)	$CLV(x) - c^*$	$-c^*$
	Non-Case (Non-churner)	0	0

The total Maximum net profit (TMNP) of implementation of churn prediction can be calculated as follows:

$$TMNP = \sum_{k=1}^{N_{TP}}(CLV_k - c) - \sum_{j=1}^{N_{FP}} c = \sum_{k=1}^{N_{TP}}(CLV_k) - \sum_{j=1}^{N_{TP}+N_{NP}} c \tag{42}$$

CLV_k represents the life time value of customer k where there are totally N_{TP} correctly detected churner customers and N_{FP} incorrectly non-churner customers who have been classified as churners. As can be seen in the above equation, there is a profit for each correctly classified churner (true positive) and also a cost of contact. There is contact cost for incorrectly classified non-churners (false positive) as well. Why we call this measure as ”Total Maximum Net Profit”? We use the word ”Maximum” because we are not sure that a customer will remain if receive a promotion offer and this measure assumes that all of the customers will remain with the retention promotion offers. We have discussed about the actual profit in churn prediction in the Section 3.4. The equation can be written as such, there is a profit for true positives and cost of contact for all of the positive predicted instances, both true and false ones. Note that there is no profit for true negatives and also there is no cost for false negatives because the net profit has to show the difference between the base scenario and proposed model. To make the comparison easy to understand in terms of net profit, we calculate the ”TMNP percent” for all of the models. It is calculated as following:

$$TMNP\% = \frac{(\sum_{k=1}^{TP} CLV_k) - \sum_{j=1}^{FP+TP} c}{\sum_{k=1}^{P}(CLV_k)} \tag{43}$$

P shows the actual number of churners in the test set. TMNP% shows the proportion of net profit which we have saved among total profit by implementing our model and shows the difference between model and a perfect model which can classify the instances with 100% accuracy.

The data used in churn prediction study has been gathered from a well-known Turkish bank. There are 20000 instances totally. We use 70% as training data and 30% as testing data. Accordingly, there are 14000 instances in the training set, 7000 instances from each of the classes (positives and negatives). In the test set there are 3000 instances from each of the classes, and then the threshold will be top 50% which is the 3000^{th} instance's score in the test set. We have benefited from the customer life time value for all of the instances in the data set and used it as our profit-based attribute and to calculate the total amount of profit which can be earned by implementing the presented churn prediction system.

The information of churn data set is shown in the following table.

Table 8: The information of churn data set

Data Set	Total No. of instances	Ratio of positive instances	No. of instances in training set	No. of instances in the testing set	No. of instances in the validation set	No. of attributes before preprocessing	No. of attributes after preprocessing
Churn Data set	20000	1/2	11900	6000	2100	41	23

4.2.1 Profit-Sensitive ANN

To compare the performance of the profit-sensitive models with original ANN, Decision tree and Naïve Bayes, the result of the testings have been presented in the following table. Note that as there was no imbalance in the data, class-based cost is not meaningful in this data set. The results of classification of all PSNN models have been presented in table 9.

Table 9: The results of all PSNN models for the churn prediction data set

	Threshold=Top 10%					
Model	True positive rate %			TMNP %		
	Min	Mean	Max	Min	Mean	Max
ANN	78.13	**78.94**	79.63	57.61	62.90	72.53
DW	61.27	75.16	78.90	50.58	62.27	78.71
PNN	78.07	**78.72**	79.10	56.96	60.11	63.82
LOG-PNN	78.60	**78.72**	79.10	56.14	61.50	75.59
LPWI	32.13	51.89	64.27	18.12	49.12	67.87
LPWA	69.57	76.23	78.53	59.34	**66.99**	76.51
MF	77.23	78.41	79.20	57.41	60.58	64.33
PNN-MAX	47.60	72.43	79.13	38.80	59.00	75.08
DT	75.00	75.00	75.00	64.52	64.52	64.52
NB	76.40	76.40	76.40	38.15	38.15	38.15

Table 9 shows that original ANN, PNN and LOG-PNN has competitive performance in term of true positive rate, the LPWA model has the significantly (considering t-test) best performance in terms of total net profit saved by the model. LPWI shows instability in both TP rate and profit. DW version of ANN, the version which uses direct weights(CLVs) in the prediction error, has a large range in profit and this shows that CLV values are not suitable to be used as weight even if the maximum profit has been earned in this version but the instability is a problem in this version. LPWA has a trade-off between accuracy and profit which makes it a considerable model specifically when there is a need for a model that generate accurate scores considering the profit of the instances. This model is needed in churn prediction when we assign promotion offers using each instance's score in the test set.

4.2.2 Profit-based ANN trained by metaheuristic algorithms

In the profit-based churn prediction, the classifiers aim to maximize the total net profit by correctly classifying churners and non-churners and giving the priority to the customers with high potential profit for the company. In this subsection, we use the metaheuristic algorithms, GA, PSO, ABC and MBO to maximize the objective function which is the maximum total net profit function. The objective function can be written as:

$$TMNP = \sum_{k=1}^{N_{TP}}(CLV_k - c) - \sum_{j=1}^{N_{FP}} = \sum_{k=1}^{N_{TP}}(CLV_k) - \sum_{j=1}^{N_{TP}+N_{FP}} c \qquad (44)$$

N_{TP} and N_{FP} represent the number of true positives and false positives classified by the model, respectively. CLV_k is the lifetime value of customer k. c is the cost of promotion offer selected by the customer relationship manager and it will be discussed in the next section throughly. As the fixed cost depends on decision of manager and also the budget limitation, we can exclude it from our objective function and just calculate the ratio of the maximum profit which can be earned by the model. In other words, we exclude the fixed cost in the aforementioned equation and divide the calculated value with total CLVs of the actual churner customers which can be correctly classified by an ideal model with 100% sensitivity. Then we can calculate the percentage of the Maximum Net Profit Value.

Table 10: The results of all metaheuristic-based ANN models for the churn data

	Threshold=Top 10%						P-Value				
Model	True positive rate %			TMNP%							
	Min	Mean	Max	Min	Mean	Max	**ANN**	**MBO-ANN**	**PSO-ANN**	**GA-ANN**	**ABC-ANN**
ANN	78.13	**78.94**	79.63	57.61	6289.52	72.53	-	-	-	-	-
MBO-ANN	67.90	72.78	84.73	80.98	82.56	84.73	2.9E-07	-	-	-	-
PSO-ANN	66.30	72.50	76.30	65.23	73.57	81.54	1.7E-04	0.0002	-	-	-
GA-ANN	64.30	70.58	74.30	80.57	**84.25**	86.42	6.2E-08	0.0100	0.0001	-	-
ABC-ANN	69.53	75.16	77.03	64.37	77.06	82.30	2.6E-05	0.0132	0.1051	0.0036	-

Table 10 shows the results for metaheuristic-based ANN models and the original ANN performances on churn prediction data. In terms of sensitivity (TP rate) which is an statistical performance metrics, the original ANN based on back-propagation algorithm has better performance than metaheuristic-based ANNs and the difference is statistically significant based on statistical t-test with $\alpha = 0.05$. In terms of total maximum net profit (TMNP%) GA-ANN has significantly better performance comparing with others. MBO-ANN is significantly better than PSO-ANN and ABC-ANN in terms of profit but PSO and ABC have competitive performance in terms of profit while ued for training the ANN. GA-ANN and MBO-ANN has same minimum performance in profit but as the maximum performance of the MBO-ANN is better and this shows that MBO-ANN is a good candidate when we are trying to classify churners and non-churners subject to maximizing the total profit.

For all of the models the total maximum net profit percent has been shown but in the previous chapter we have discussed about calculating the real profit of churn prediction models. Moreover, as these models perform classification based on aforementioned objective function which has no mention about target values in it, the metaheuristic models result in scores which do not represent the probability of being churner for each instance and their scores have to be calibrated. Therefore, if the probabilities are needed to company to assign the promotions based on scores from the model, these models will not be useful and the error modification is a better approach. But if the task is just to classify the customers into two classes, churners and non-churners, metaheuristic-based ANNs with profit as objective function will be the best alternative if the total profit is the main concern.

4.3 Direct Marketing

In this application area, the aim is to classify the customers into two groups of buyers (cases) and not-buyers (non-cases) by considering the maximum profit that can be obtained by selecting a set of customers as target rather than using prediction error in classification. So by classifying the buyers truly, a percentage (commission percentage or CP) of targeted amount value (TAV) is profit obtained from the customers, the base scenario is lack of existence of direct marketing system. The matrix below shows profit and cost related to classifying a customer x in direct marketing. Moreover, c^* is the action cost of the campaigning for a specific product.

Table 11: Net profit matrix for direct marketing

		Actual	
		Case (Buyer)	Non-Case (Not-buyer)
Predicted	Case (Buyer)	$CP \times TAV(x) - c^*$	$-c^*$
	Non-Case (Not-buyer)	0	0

The expression to calculate the saving amount or the change which model can make is shown as follows:

$$TMNP = \sum_{i=1}^{N_{TP}} (CP \times TAV(x) - c^*) + \sum_{k=1}^{N_{FP}} (-c^*) \tag{45}$$

This expression means that if the model is implemented, there will be such benefits and costs gained from all instances compared to the case of not implementing the model. Same as churn prediction model, we calculate the maximum profit because we are not sure that a customer will take reaction if receive a promotion offer and this measure assumes that all of the customers will take reactions with the advertisements.

The direct marketing data is a real-life data set gathered from a Turkish bank and there are totally 20000 instances in the data set and the size of positive instances are equal to negative ones. Therefore, there is no imbalance in data set and we randomly chosen 70% if data to train the proposed models and 30% for testing from each classes. As there are same number of each classes in the test set, class-based cost is meaningless here and also the threshold is top 50% which means that in the test set the 3000^{th} instance's score represents the value of the threshold for top 50%. The information of used data set is shown in the table 12.

Table 12: The information of direct marketing data set

Data Set	Total No. of instances	Ratio of positive instances	No. of instances in training set	No. of instances in the testing set	No. of instances in the validation set	No. of attributes before preprocessing	No. of attributes after preprocessing
Marketing Data set	20000	1/2	11900	6000	2100	26	10

4.3.1 Profit-Sensitive ANN

The results of all of the profit-sensitive models are represented in the following table for this data set.

Table 13: The results of all models for direct marketing data set

	Threshold=Top 50%					
Model	True positive rate %			TMNP %		
	Min	Mean	Max	Min	Mean	Max
ANN	87.10	88.98	91.13	55.33	67.57	94.67
DW	44.17	61.83	66.67	40.62	79.84	91.71
PNN	59.40	63.77	64.77	55.84	**83.70**	88.53
LOG-PNN	75.20	86.28	91.33	49.60	74.81	94.74
LPWI	24.07	55.30	99.93	0.00	9.49	79.84
LPWA	40.07	55.64	64.90	40.07	55.64	64.90
MF	38.40	49.61	63.37	13.16	54.41	89.45
PNN-MAX	62.93	64.42	65.83	38.23	79.82	87.42
DT	90.43	**90.43**	90.43	80.33	80.33	80.33
NB	68.47	68.47	68.47	69.76	69.76	69.76

Table 13 shows the performance of proposed models and the two benchmark classifiers, Decision Tree (DT) and Naive Bayesian (NB), in both accuracy based and profit-based measurements where the threshold is top 50%. The results show that for this data set PNN model has best performance in terms of total maximum net profit. However, Decision tree (C4.5) has outperformed other models in true positive rate. Original ANN has same performance in true positive rate but its stability is considerably less than DT.

4.3.2 Profit-based ANN trained by metaheuristic algorithms

Like Churn Prediction, the objective is classifying the customers into two groups of Buyers and Not-buyers and taking actions is depending to the organization in order to satisfy them to buy the products. Therefore, we took c^* zero here and the objective function turns to:

$$\text{Maximum Possible Profit} = \sum_{i=1}^{N_{TP}} CP \times TAV(i) \tag{46}$$

This means that without considering the fixed action cost, these models try to give priority to instances which have more profit.

Table 14: The results of all metaheuristic-based ANN models for the direct marketing data

	Threshold=Top 50%						P-Value				
Model	True positive rate %			Total Maximum Net Profit (%)							
	Min	Mean	Max	Min	Mean	Max	ANN	MBO-ANN	PSO-ANN	GA-ANN	ABC-ANN
ANN	88.00	**89.00**	90.00	67.42	67.57	67.72	-	-	-	-	-
MBO-ANN	56.17	59.61	62.03	88.71	91.79	94.15	0.0005	-	-	-	-
PSO-ANN	55.10	58.71	62.73	44.63	86.49	93.27	0.0065	0.1453	-	-	-
GA-ANN	55.97	58.97	61.60	89.99	92.03	93.84	0.0005	0.3793	0.1343	-	-
ABC-ANN	55.77	59.75	63.03	90.54	92.74	94.44	0.0004	0.1096	0.1078	0.1314	-

Table 14 represents the results for metaheuristic-based ANNs and the original one. All of the ANN models which use metaheuristic algorithms to maximize the net profit, outperform the original ANN in terms of total maximum net profit as p-value of t-tests represent. However all of these metaheuristic-based ANNs have same performance in total profit due to t-tests' results but among these models, PSO-ANN has less stable considering its minimum and maximum performance in TMNP. Original ANN has the highest TP rate and it is again the most accurate model but suboptimal one in terms of profit. Regarding the minimum performance of the models in total maximum net profit, the most robust model is ABC-ANN as it has the largest minimum TMNP among all models.

In the direct marketing classification, if there is no need for scores of the instances in the test set as instance probabilities of purchasing, the second approach (PBNN) is the optimal alternative in terms of total net profit. However if the companies need to assign the offers depending on the probabilities of the instances, the PSNN model can be better choice because PBNN models' scores do not represent the probabilities and they have to be calibrated.

4.4 Credit Risk

In banking credit risk problems there are two kinds of cost-benefit-wise terms which can be considered to calculate the saved net profit by classification model, default value (DV), return of investment (ROI) of given credit and fixed cost of situation evaluation or model implementation (c). The bank claims that if a customer is defaulter, they can receive the half amount of default value with their lawyers but they miss the revenue of the credit. The base scenario in credit risk is the lack of credit scoring system and defaulters will default and non-defaulters will give back the credit with a revenue. Considering the base scenario, for the classification model, there is a profit for true positive detected cases (defaulters) which is the half amount of default value which couldn't be received even with the bank lawyers and also there is a cost which is the opportunity cost of missing a ROI of given credit plus a model implementation cost. For false positive detected customers, there is a cost of missing a ROI. There is no cost and profit for negative detected instances because the base scenario assumes all of the instances as negative and classification model will not make any change in their profit and cost.

In this study, we will use proposed profit based ANN classifiers on credit risk analysis. The main objective is to detect profitable customers to grant the loan and prevent to do so with the costly defaulters as much as possible to maximize profit. The most important variables that are commonly used in our investigation include profit (Return on Investment of a granted loan) and Default Value (DV). As in credit card fraud detection, these two variables play main role in introducing profit based models.

In order to give reliable priority to important credit requests (i.e., credits with high risk), a measure which is actually inferred from the available dataset is needed. The matrix below shows profit and costs of classifying a customer as defaulter or not-defaulter. The profit is half of the default value as lawyers may get it back by setting a complaint against the defaulter customer. The base scenario is absence of model for predicting the defaulters. Comparing the perfect model with capability of classifying all customers correctly with respect to the base scenario, it can increase the profit at most 24% of the sum of half of DVs minus ROIs. In the results section, the performance of the developed algorithms in this research will be evaluated with regard to the capability of the perfect model.

Table 15: Net profit matrix for Credit Scoring

		Actual	
		Case (defaulter)	Non-Case (Non-defaulter)
Predicted	Case (defaulter)	$\frac{DV(x)}{2} - ROI(x) - c$	$-ROI(x) - c$
	Non-Case (Non-defaulter)	0	0

Based on the aforementioned matrix, the total net profit of credit scoring model can be calculated by the following equation:

$$TNP = \sum_{i=1}^{N_{TP}} (\frac{DV_i}{2} - ROI_i - c) + \sum_{k=1}^{N_{FP}} (-ROI_i - c) \qquad (47)$$

Where, DV_i indicates the default value of the i^{th} instance (customer), ROI is the return of investment for each of credits and c is the implementation cost which is fixed for all cases.

The credit risk data has been acquired from a Turkish bank as well and there are total 39857 instances in the data set with 36 attributes for each of the instances in the start. the proportion of 70% has been chosen to train the model and 30% for testing. The validation data is selected from training data and it is the 15% of its size.

The information of data has been shown in table 16.

Table 16: The information of credit risk data set

Data Set	Total No. of instances	Ratio of positive instances	No. of instances in training set	No. of instances in the testing set	No. of instances in the validation set	No. of attributes before preprocessing	No. of attributes after preprocessing
Credit risk Data set	39857	1/10	23715	11957	4185	35	33

4.4.1 Profit-Sensitive ANN

Like fraud detection, in credit scoring problems, the most important objective is to detect high-risk (bad) credit applicants with high potential savings and the financial institutions will take action to all of the positive detected instances. Accordingly, the scores which come from the prediction model for each of the instances in the data set are not important same as fraud detection case and PSNN models have not significant contribution in total net profit(TNP). Therefore we do not present the PSNN models' results in this section because the ANN models which work based on error minimization, had not acceptable results in this data set.

4.4.2 Profit-based ANN trained by metaheuristic algorithms

In this model we maximize the total net profit of credit scoring model which comes from equation 47. We solve this maximization problem using four well-known metaheuristic algorithms, MBO, PSO, GA and ABC.

The results are presented in table 17:

Table 17: The results of all metaheuristic-based ANN models for the credit risk data

	Threshold=Top 50%									P-Value			
Model	True Positive rate %			TNP %			TNP % compered to perfect model						
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	MBO-ANN	PSO-ANN	GA-ANN	ABC-ANN
MBO-ANN	18.58	**19.33**	20.06	13.20	15.23	18.09	55.01	63.45	75.36	-	-	-	-
PSO-ANN	14.42	19.10	21.16	10.77	**17.14**	19.82	44.88	**71.43**	82.57	0.0322	-	-	-
GA-ANN	19.13	**19.49**	19.87	11.96	13.90	15.00	49.85	57.94	62.52	0.0167	0.00196	-	-
ABC-ANN	14.79	17.23	18.58	4.29	8.67	12.86	17.87	36.11	53.60	0.0000	0.00000	3.3E-05	-

Table 17 represents that GA-ANN and MBO-ANN outperform other metaheuristic-based ANNs in terms of true positive rate but as the GA-ANN has better minimum statistical performance, it can be a good candidate when statistical performance is the main concern. In terms of Total Net Profit (TNP), PSO-ANN is significantly superior to other models based on the results of t-tests and it is able to find a better or maybe global solution in maximizing the total net profit. Note that as the perfect model with no prediction error just can save 24% TNP, we added another row to this table which shows the saving TNP of each model comparing with the perfect classifier. In other application domain the perfect model could save 100% of the profit and their profit percentages shows their performance comparing with the perfect classification model as well.

CHAPTER V

CONCLUSION

In the literature, there are a lot of studies which deal with class imbalance problem and developed class-based cost-sensitive classifiers to minimize the total cost of misclassification by assigning different costs to each of the minor and major class. However, in most of the classification problems in business context, there are variable cost of misclassification for each of the instances in the data set, moreover, there are some profits for correct classification of each of them. This study aims to bridge this gap by introducing two approaches to make the classifiers sensitive to the net profit of each instance.

In the first approach, we modified the prediction error function and made it sensitive to the net profit of each instance and the resulting models were ANN models which try to keep a trade-off between accuracy and total net profit. These models have better performance in terms of total net profit comparing with the traditional ANN model with a little decrease in the accuracy. The importance of these models is revealed when companies need to have a profit-sensitive model which generates probabilities (scores) for each of the instances because these probabilities are needed when companies give offers to customers by considering their scores and profits.

In the second approach, we trained the ANN models with metaheuristic algorithms when the objective was the total net profit earned by the model. In this approach we directly take the total net profit into account and the classifier aimed to maximize it. These models have significantly superior performance in terms of total net profit comparing with both traditional ANN and the first approach (profit-sensitive ANNs).

We called these models as "profit-based" models as their main objective is to maximize the total profit function. The resulting models of this approach are the best candidates for the cases when the objective is to maximize the net profit by taking action to all of the predicted positive instances because this models' scores do not represent the probabilities of belonging to a class for each instance and the normalization may be required. For this purpose, the scores of the model have to be calibrated. This issue can be a disadvantage in churn prediction and direct marketing problems which would need the probabilities to assign the appropriate actions for each of the instances. In these cases, profit-sensitive approach can be a better alternative.

The third contribution of this research is the development of an accurate profit of churn prediction considering the variable effects of the different promotion offers and customers' possible reactions regarding these offers. In churn prediction calculating the profit of the model is more complex than fraud detection and credit scoring because the total profit depends on the reaction of the customer. We have throughly analyzed this issue and give the formulation to find the appropriate offers for customers considering the budget limit of the company. This formulation maximizes the total net profit of churn prediction model using each customer's churn probability and profitability. Then we have formulated this problem for different policies of promotion offer selection. In our churn problem the policy was to give fixed promotion offer for all of the predicted churners and this formulation was not needed to be considered. However, this approach can be benefited in all of the customer-centric classification problems when there are probabilities and some profit-based attributes for each of the customers in the data set.

Bibliography

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani, An introduction to statistical learning. Springer, 2013.

[2] Q. YANG and X. WU, 10 CHALLENGING PROBLEMS IN DATA MINING RESEARCH, International Journal of Information Technology & Decision Making, vol. 05, no. 2006. pp. 597604, 2006.

[3] C. Bishop, Pattern recognition and machine learning. 2006.

[4] B. Baesens, S. Viaene, D. Van den Poel, J. Vanthienen, and G. Dedene, Bayesian neural network learning for repeat purchase modelling in direct marketing, Eur. J. Oper. Res., vol. 138, no. 1, pp. 191211, Apr. 2002.

[5] S. Haykin, Neural Networks: A comprehensive foundation, 2nd ed. Prentice-Hall, 1999.

[6] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control. Signals, Syst., vol. 2, no. 4, pp. 303314, Dec. 1989.

[7] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. ”Multilayer feedforward networks are universal approximators.” Neural networks 2, no. 5 (1989): 359-366.

[8] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks, vol. 4, no. 2, pp. 251257, Jan. 1991.

[9] Edward I. Altman, Giancarlo Marco, Franco Varetto, Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience), *Journal of Banking & Finance*, Volume 18, Issue 3, May 1994, Pages 505-529, ISSN 0378-4266

[10] W. G. Baxt, Use of an Artificial Neural Network for the Diagnosis of Myocardial Infarction, *Ann. Intern. Med.*, vol. 115, no. 11, p. 843, Dec. 1991.

[11] R. P. Lippmann, Review of Neural Networks for Speech Recognition, *Neural Comput.*, vol. 1, no. 1, pp. 138, Mar. 1989.

[12] K. Tam, Neural network models and the prediction of bank bankruptcy, Omega, vol. 19, no. 5, pp. 429445, Jan. 1991.

[13] H. Jo, I. Han, and H. Lee, Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis, Expert Syst. Appl., vol. 13, no. 2, pp. 97108, Aug. 1997.

[14] P. C. Pendharkar, A threshold-varying artificial neural network approach for classification and its application to bankruptcy prediction problem, Comput. Oper. Res., vol. 32, no. 10, pp. 25612582, Oct. 2005.

[15] S. Ghosh and D. L. Reilly, Credit card fraud detection with a neural-network, in Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences HICSS-94, 1994, vol. 3, pp. 621630.

[16] R. Brause, T. Langsdorf, and M. Hepp, Neural data mining for credit card fraud detection, Proc. 11th Int. Conf. Tools with Artif. Intell., pp. 103106, 1999.

[17] M. Syeda, Parallel granular neural networks for fast credit card fraud detection, in 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE02. Proceedings (Cat. No.02CH37291), 2002, vol. 1, pp. 572577.

[18] P. C. Pendharkar, Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services, Expert Syst. Appl., vol. 36, no. 3, pp. 67146720, Apr. 2009.

[19] C.-F. Tsai and Y.-H. Lu, Customer churn prediction by hybrid neural networks, Expert Syst. Appl., vol. 36, no. 10, pp. 1254712553, Dec. 2009.

[20] D. West, Neural network credit scoring models, Comput. Oper. Res., vol. 27, no. 1112, pp. 11311152, Sep. 2000.

[21] J. Zahavi and N. Levin, Applying neural computing to target marketing, J. Direct Mark., vol. 11, no. 1, pp. 522, 1997.

[22] F. Kaefer, C. M. Heilman, and S. D. Ramenofsky, A neural network application to consumer classification to improve the timing of direct marketing activities, Comput. Oper. Res., vol. 32, no. 10, pp. 25952615, Oct. 2005.

[23] M. Kubat, R. C. Holte, and S. Matwin, Machine Learning for the Detection of Oil Spills in Satellite Radar Images, *Mach. Learn.*, vol. 30, no. 23, pp. 195215, 1998.

[24] R. Pearson, G. Goney, and J. Shwaber, Imbalanced clustering for microarray time-series, *Proc.* ICML, 2003.

[25] Y. Sun, M. Kamel, and Y. Wang, Boosting for learning multiple classes with imbalanced class distribution, Data Mining, 2006. ICDM06. Sixth , 2006.

[26] N. Abe, B. Zadrozny, and J. Langford, An iterative method for multi-class cost-sensitive learning, *Proc. tenth ACM SIGKDD* , 2004.

[27] Z. Zhou and X. Liu, ON MULTI-CLASS COST-SENSITIVE LEARNING, *Comput. Intell.*, 2010.

[28] N. V. N. Chawla, K. K. W. Bowyer, and L. O. Hall, SMOTE: Synthetic minority over-sampling technique, *J. Artif.* , vol. 16, pp. 321357, 2002.

[29] R. Rao, S. Krishnan, and R. Niculescu, Data mining for improved cardiac care, *ACM SIGKDD Explor.* , 2006.

[30] T. M. Padmaja, N. Dhulipalla, R. S. Bapi, and P. R. Krishna, Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection, *in 15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, 2007, pp. 511516.

[31] J. Burez and D. Van den Poel, Handling class imbalance in customer churn prediction, *Expert Syst. Appl.*, vol. 36, no. 3, pp. 46264636, Apr. 2009.

[32] M. Kukar and I. Kononenko, Cost-Sensitive Learning with Neural Networks., *ECAI*, 1998.

[33] Zhi-Hua Zhou, Xu-Ying Liu, Z. Zhou, and X. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *Data Eng. IEEE Trans.*, vol. 18, no. 1, pp. 114, Jan. 2006.

[34] E. A. Garcia, Learning from Imbalanced Data, IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 12631284, Sep. 2009.

[35] A. Tan, D. Gilbert, and Y. Deville, Multi-class protein fold classification using a new ensemble machine learning approach, *Genome Informatics*, 2003.

[36] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and Regression Trees. Taylor & Francis, 1984.

[37] ”Federal Trade Commission,” 2014. [Online]. Available: http://www.ftc.gov/. [Accessed: 09-Dec-2014].

[38] K. J. Leonard, ”Detecting credit card fraud using expert systems,” *Comput. Ind. Eng.*, vol. 25, no. 1, pp. 103106, 1993.

[39] P. Chan and S. Stolfo, Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection., *Am. Assos. Artif. Intel.*, 1998.

[40] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, Distributed data mining in credit card fraud detection, IEEE Intell. Syst., vol. 14, no. 6, pp. 6774, Nov. 1999.

[41] R. C. Chen, S. T. Luo, X. Liang, and V. Lee, "Personalized approach based on SVM and ANN for detecting credit card fraud," *in Neural Networks and Brain, 2005. ICNN&B05. International Conference*, 2005, vol. 2, pp. 810815

[42] R. C. Chen, M. L. Chiu, Y. L. Huang, and L. T. Chen, "Detecting credit card fraud by using questionnaire-responded transaction model based on support vector machines," in *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, Springer, 2004, pp. 800806.

[43] A. Shen, R. Tong, and Y. Deng, "Application of Classification Models on Credit Card Fraud Detection," in *2007 International Conference on Service Systems and Service Management*, 2007, pp. 14.

[44] P. Juszczak, N. M. Adams, D. J. Hand, C. Whitrow, and D. J. Weston, "Off-the-peg and bespoke classifiers for fraud detection," *Comput. Stat. Data Anal.*, vol. 52, no. 9, pp. 45214532, 2008.

[45] Y. Sahin and E. Duman, Detecting credit card fraud by ANN and logistic regression, *2011 Int. Symp. Innov. Intell. Syst. Appl.*, pp. 315319, Jun. 2011.

[46] J. Langford and A. Beygelzimer, "Sensitive error correcting output codes," *in Learning Theory*, Springer, 2005, pp. 158172.

[47] E. Duman and M. H. Ozcelik, "Detecting credit card fraud by genetic algorithm and scatter search," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 1305713063, Sep. 2011.

[48] N. Mahmoudi and E. Duman, "Detecting credit card fraud by Modified Fisher Discriminant Analysis," *Expert Syst.* Appl., Nov. 2014.

[49] Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, Customer churn prediction using improved balanced random forests, *Expert Syst. Appl.*, vol. 36, no. 3, pp. 54455449, Apr. 2009.

[50] N. Glady, B. Baesens, and C. Croux, Modeling churn using customer lifetime value, *Eur. J. Oper. Res.*, vol. 197, no. 1, pp. 402411, Aug. 2009.

[51] C.-P. Wei and I.-T. Chiu, Turning telecommunications call details to churn prediction: a data mining approach, *Expert Syst. Appl.*, vol. 23, no. 2, pp. 103112, Aug. 2002.

[52] W. Au, K. C. C. Chan, and X. Yao, A novel evolutionary data mining algorithm with applications to churn prediction, *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 532545, Dec. 2003.

[53] C. B. Bhattacharya, When Customers Are Members: Customer Retention in Paid Membership Contexts, *J. Acad. Mark. Sci.*, vol. 26, no. 1, pp. 3144, Jan. 1998.

[54] A. D. Athanassopoulos, Customer satisfaction cues to support market segmentation and explain switching behavior, *J. Bus. Res.*, vol. 47, no. 3, pp. 191207, 2000

[55] Rust, R.T., Zahorik, A.J. Customer satisfaction, customer retention, and market share (1993) *Journal of Retailing*, 69 (2), pp. 193-215.

[56] Dirk Van den Poel, Bart Lariviére, Customer attrition analysis for financial services using proportional hazard models, *European Journal of Operational Research*, Volume 157, Issue 1, 16 August 2004, Pages 196-217, ISSN 0377-2217

[57] T. Verbraken, A novel profit maximizing metric for measuring classification performance of customer churn prediction models, *Knowl. Data* , vol. 25, no. 5, pp. 961973, 2013.

[58] M. J. Shaw, C. Subramaniam, G. W. Tan, and M. E. Welge, Knowledge management and data mining for marketing, *Decis. Support Syst.*, vol. 31, no. 1, pp. 127137, May 2001.

[59] L. Bulysheva and A. Bulyshev, Segmentation modeling algorithm: a novel algorithm in data mining, Inf. Technol. Manag., vol. 13, no. 4, pp. 263271, Aug. 2012.

[60] A. Hiziroglu, Soft computing applications in customer segmentation: State-of-art review and critique, Expert Syst. Appl., vol. 40, no. 16, pp. 64916507, Nov. 2013.

[61] B. Wray, A. Palmer, and D. Bejou, Using Neural Network Analysis to Evaluate Buyer-Seller Relationships, *Eur. J. Mark.*, vol. 28, no. 10, pp. 3248, Oct. 1994.

[62] J. Zahavi and N. Levin, Applying neural computing to target marketing, *J. Direct Mark.*, vol. 11, no. 1, pp. 522, 1997

[63] Y. Kim and W. N. Street, An intelligent system for customer targeting: a data mining approach, Decis. Support Syst., vol. 37, no. 2, pp. 215228, May 2004.

[64] Y. Kim, W. N. Street, G. J. Russell, and F. Menczer, Customer Targeting: A Neural Network Approach Guided by Genetic Algorithms, Manage. Sci., vol. 51, no. 2, pp. 264276, Feb. 2005.

[65] L. Thomas, D. Edelman, and J. Crook, Credit Scoring and its Applications, SIAM Monogr. Math., 2002.

[66] A. Steenackers and M. Goovaerts, A credit scoring model for personal loans, Insur. Math. Econ., 1989.

[67] R. H. DAVIS, D. B. EDELMAN, and A. J. GAMMERMAN, Machine-learning algorithms for credit-card applications, IMA J. Manag. Math., vol. 4, no. 1, pp. 4351, 1992.

[68] W. Henley, Construction of a k-nearest-neighbour credit-scoring system, *IMA J. Manag. Math.*, 1997.

[69] V. S. Desai, J. N. Crook, and G. A. Overstreet, A comparison of neural networks and linear scoring models in the credit union environment, *Eur. J. Oper. Res.*, vol. 95, no. 1, pp. 2437, Nov. 1996.

[70] K. Tam and M. Kiang, Predicting bank failures: a neural network approach, *Appl. Artif. Intell. an Int.* , 1990.

[71] J. Brill, The importance of credit scoring models in improving cash flow and collections, *Bus. Credit*, vol. 1, pp. 1617, 1998.

[72] P. Coats and L. Fant, Recognizing financial distress patterns using a neural network tool, *Financ. Manag.*, 1993.

[73] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation, Manage. Sci., vol. 49, no. 3, pp. 312329, Mar. 2003.

[74] A. Savvopoulos, Consumer Credit Models: Pricing, Profit and Portfolios, J. R. Stat. Soc. Ser. A (Statistics Soc., vol. 173, no. 2, pp. 468468, Apr. 2010.

[75] H. Jensen, Using neural networks for credit scoring, Manag. Financ., vol. 18, no. 6, pp. 1526, 1992.

[76] McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115133. Reprinted in Anderson and Rosenfeld (1988).

[77] Rosenblatt, F. (1962). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan.

[78] Widrow, B. and M. E. Hoff (1960). Adaptive switching circuits. In IRE WESCON Convention Record, Volume 4, pp. 96104. Reprinted in Anderson and Rosenfeld (1988).

[79] M. Carvalho and T. Ludermir, Particle Swarm Optimization of Feed-Forward Neural Networks with Weight Decay, 2006 Sixth Int. Conf. Hybrid Intell. Syst., no. 1, pp. 55, Dec. 2006.

[80] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim., 2007.

[81] E. Duman and I. Elikucuk, Applying Migrating Birds Optimization to Credit Card Fraud Detection, Trends Appl. Knowl. Discov. Data Min., pp. 416427, 2013.

VITA

Ashkan Zakaryazad received the BSc degree in industrial engineering from Iran University of Science and Technology (IUST), Tehran, Iran in 2012. Since 2013, he has been working toward the MSc degree at the department of industrial engineering at the Özyeğin University, Istanbul, Turkey. Being a member of the Scientific and Technological Research Council of Turkey (TÜBİTAK), his main research focuses on data mining in business settings, such as credit card fraud detection, churn prediction, direct marketing and credit scoring. He is starting his PhD. in Industrial engineering at Georgia Institute of Technology by fall 2015 with major in Analytics and Big Data.