A NEURAL NETWORKS APPROACH FOR CHURN PREDICTION MODELING IN MOBILE TELECOMMUNICATIONS INDUSTRY

Ionut BRANDUSOIU, Gavril TODEREAN
Technical University of Cluj-Napoca, Romania
ionut.brandusoiu@gmail.com, toderean@pro3soft.ro

Keywords: business intelligence, telecommunications market, churn prediction, radial basis functions networks, multi-layer perceptrons.

Abstract: *Nowadays, organizations are facing several challenges resulting from competition and market trends. Customer churn is a real issue for organizations in various industries, especially in the telecommunications sector with a churn rate of approximately 30%, placing this industry in the top of the list. Because higher expenses are involved when trying to attract a new customer than trying to retain an existing one, this is an important problem that needs an accurate resolution. This paper presents an advanced methodology for predicting customers churn in mobile telecommunication industry by applying data mining techniques on a data set consisting of call detail records. The data mining algorithms considered and compared in this paper are Multi-Layer Perceptron and Radial Basis Function neural networks.*

1. INTRODUCTION

Customer retention in the telecommunications industry has become very important with the increasing competition and diversity of offerings on the market. In order to keep up in such a competitive marketplace, many telecommunication companies are using data mining techniques to overcome challenging issues such as: churn prediction, customer profiling, cross-selling and up-selling, lifetime value, and fraud detection [1].

The word "*churn*" which has been obtained by combining the English words "*change*" and "*turn*", describes the phenomenon of loss of a customer. In mobile telecommunications industry (also referred to as customer attrition or subscriber churning) it refers to the movement of subscribers from one provider to another [2]. It is measured by the rate of churn and is an important indicator for organizations. The churn rate represents the percentage of lost customers over a given period divided by the total number of customers at the beginning of that period. The churn process is usually happening due to better rates or services, or due to different benefits that a competitor company offers when signing up.

The average monthly churn among mobile operators in Europe varies between 8% and 12% [3]. The annual attrition rate varies from 20% to 40% in most mobile telecommunication operators [4]. In such a competitive market, a defensive marketing strategy is very important. Instead of trying to acquire new customers or to attract customers away from the competition, the defensive marketing is more interested in reducing the churning of its customers [4], especially when it is 5 times more costly to acquire a new customer than to keep one [5]. Being so expensive to acquire new subscribers, it is crucial to keep the existing ones. Building a churn prediction model will ease the customer retention process, and in this way the mobile telecommunication companies will success in this constantly increasing competitive marketplace.

2. LITERATURE REVIEW

The churn prediction modeling process is strongly dependent on the data mining process and techniques due to an increased performance generated by machine learning (ML) algorithms compared to the statistical techniques for non-parametric data [6].

This paper presents an advanced data mining methodology based on two well-known neural network models. Multi-Layer Perceptron neural network has been previously used to create a churn prediction model using Call Detail Records (CDR) in mobile telecommunications industry, but Radial Basis Function neural network is for the first time purposed and used in this research paper.

This paper is organized as follows: a short introduction to data mining is presented and theoretical aspects of Multi-Layer Perceptron and Radial Basis Functions neural networks, and in the next section, the methodology with its corresponding phases. Finally, the conclusion and future work are stated.

2.1 Data Mining

Data mining is the practice of digging data to find trends and patterns, and can provide you with answers to questions that you should have asked [7]. Data mining methods lie at the intersection of artificial intelligence, machine learning, statistics, and database systems [8]. Data mining techniques can help building prediction models in order to discover future trends and behaviors, allowing organizations to make smart decisions based on knowledge from data.

The methodology used is called modeling. Modeling is the process of creating a mining model, an implementation of specific data mining algorithms. These algorithms are mathematical functions that perform specific types of analysis on the associated data set [9]. Based on the business problems that need assistance, these data mining algorithms can be broadly classified into several categories, including classification, segmentation or clustering, association, regression or forecasting, sequence analysis and prediction, and deviation or anomaly analysis [10]. Data mining methods can be summarized in two main categories: *predictive methods* (use existing variables to predict values of other variables – classification, regression or forecasting, and anomaly analysis) and *descriptive methods* (reveal patterns and trends in data that can be easily interpreted by the user – segmentation or clustering, association, and sequence analysis) [10].

Classification analysis is the process by which a model is built to categorize pre-classified instances (training examples) into classes. This classifier (classification model) is then used to classify future instances. Some of the most used classification algorithms are Decision Trees, Naïve Bayes, Neural Networks, Support Vector Machines, Genetic algorithm, and K-Nearest Neighbor.

In the following section, the discussion is restricted to neural networks classification algorithm used later to create a predictive model in order to solve the subscribers churning problem. We decided to use the neural networks approach because they can build predictive models with minimal demands on model structure and assumptions, and have the ability to detect complex nonlinear relationships between the predictors and target variables.

2.2 Neural Networks

An artificial neural network (ANN), often just called neural network (NN), is a very popular machine learning algorithm based on modern mathematical concepts inspired by biological neural networks. The capacity of human beings to perceive and memorize new information through a learning process, motivated researchers to develop artificial systems which are able to perform certain functions based on a learning process (training) [11].

Artificial neural networks are obtained by connecting a number of elementary information processing units, called artificial neurons. When arranged and connected in parallel, these artificial neurons form a layer of a neural network. An ANN can have one or more layers of neurons [12].

Artificial neural networks' ability to learn, consists in their property to change their parameters (weights and thresholds) based on the input data used during training (learning) process. ANN training could be supervised and unsupervised [13]. In our particular case we will use the supervised learning task because we will provide a training set which consists of input signals (predictors) and the correct output (target) which is explicitly given. In this case, the weights and thresholds values of the neurons are determined to lead to the minimization of a specified criterion function.

There are multiple types of neural networks, but in order to identify nonlinear processes, we will use the following two: Multi-Layer Perceptron (MLP) and Radial Basis Functions (RBF) neural networks. In the following part we will explain how we mathematically modeled each neural network algorithm for our churn prediction model.

2.3 Multi-Layer Perceptron (MLP) Networks

The multi-layer perceptron network consists of one or more layers connected in series, and each layer contains one or more processing units called perceptrons. The perceptrons are interconnected in a feedforward way, passing the network layer by layer until it reaches the output (Figure 1). The perceptron is the simplest form of a neural network, and has a single neuron that sums its weighted inputs, sum that is followed by an activation function.

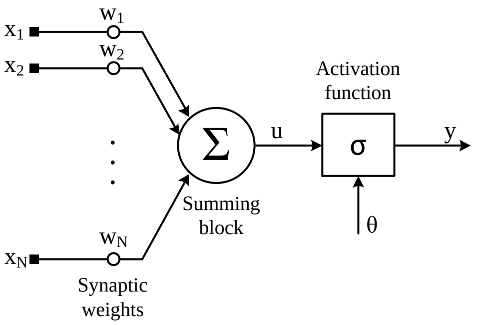

Fig. 1 Perceptron structure [14]

Where u is the transfer value and is given by equation (1), and θ is the threshold. Next, the nonlinear transformation that corresponds to the activation function is applied, thus obtaining the neuron output signal, y, given by (2) [14]:

$$u = \sum_{i=1}^{N} w_i x_i = w^T x \qquad (1)$$

$$y = \sigma(u) \qquad (2)$$

The activation function used is the sigmoid function with its equation and its first derivative in (3), respectively (4):

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x)) \qquad (4)$$

Arranging perceptrons in parallel in multiple layers connected in series forms a Multi-Layer Perceptron neural network. Typically, a MLP neural network has: an input layer, one or more hidden layers, and an output layer (Figure 2) [14]. The hidden layers confer the ability to learn to solve complex problems.

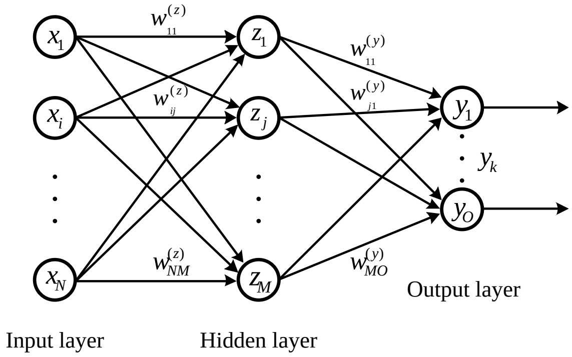

Fig. 2 Architecture of a MLP with one hidden layer

MLP neural network training is done in a supervised manner, using the Back-Propagation (BP) learning algorithm based on the generalized delta rule [15]. This learning algorithm is the most known training algorithm for neural networks. Approximately 95% of reported neural networks applications utilize multi-layer perceptron (MLP) neural network with the back-propagation learning algorithm [16].

This learning algorithm mechanism can be divided in two phases [14]:

- A forward pass – feedforward propagation of input signals. During this propagation, the neurons' weights and thresholds have fixed values;
- A backward pass – propagates the error (5) backwards through the network, starting at output units (where the error is the difference between actual (y) and desired output (t) values). During this back-propagation the weights and thresholds are adapted to obtain signals as closer as possible to the desired values.

$$e_k = t_k - y_k \qquad (5)$$

The sum of squares error function is defined by the following equation:

$$E = \frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{O}(t_k^{(l)} - y_k^{l})^2 \qquad (6)$$

Back-propagation algorithm seeks to minimize the error function defined in the space generated by the weights using the descent gradient optimization algorithm [14]. The set of weights that minimize the error function is considered the solution to the training (learning).

For each output node the local gradient is computed by using the formula:

$$\delta_k = y_k(1-y_k)(t_k - y_k) \tag{7}$$

For each hidden node the local gradient is computed by using the formula:

$$\delta_j = y_j(1-y_j)\sum_{k=1}^{O}\delta_k w_{jk} \tag{8}$$

The weights are updated bringing the weight in the opposite direction of the gradient by subtracting the learning rate (η) from the weight as in equations (9) and (10). The speed and quality of learning depend on this ratio.

$$w_{new} = w_{old} + \Delta w \tag{9}$$

$$\Delta w = -\eta \delta_l y_{l-1} \tag{10}$$

These two phases are repeated until the stopping rule is met. The algorithm stops if the total error for the training data computed at the end of each step does not decrease below E_1 (current minimum error) over the next n=1 steps.

2.4 Radial Basis Function (RBF) Networks

A radial basis function (RBF) network is a feedforward, supervised learning network with an input layer, only one hidden layer called the radial basis function layer, and an output layer [13]. Like MLP, RBF network can be used for prediction and classification problems. Compared to MLP networks, RBF networks take a different approach, in terms that the nodes from the hidden layer implement a set of radial basis functions. In our case we will use Gaussian function which has the equation (11). This function is localized, in the sense that for large values it tends to zero.

$$\phi(x) = \exp(-\frac{x^2}{2\sigma^2}) \tag{11}$$

Instead of every input data point (x_n) having its own influence on the Gaussian radial basis function, we are going to elect a number (K<<N) of important centers for the data, and have them influence the neighborhood around them. These centers (μ_k) will be the centers of the Gaussian functions, and are in the same space as the input data points, x_n.

The output equation for the j^{th} hidden node is:

$$z_k = \exp(-\frac{1}{2\sigma_k{}^2}\|x-\mu_k\|^2) \tag{12}$$

where, $\|x-\mu_k\|$ is the Euclidean distance between the input data vector x and the corresponding center μ_k.

Regarding the output nodes, they implement linear summation functions as in multi-layer perceptron networks [13]:

$$y = \sum_{k=1}^{K} w_k \exp(-\frac{1}{2\sigma_k{}^2}\|x_n-\mu_k\|^2) \tag{13}$$

The architecture of a RBF network is depicted in Figure 3.

The process of finding the w_k and μ_k parameters from equation (13) is called training. The RBF network is trained in two stages [17]:

- Determine the centers μ_k and their widths by using, in our case, the two-step clustering algorithm.
- Estimate the synaptic weights w_k given the radial basis functions.

Because of the simplicity of these two stages the RBF network is trained much quicker than MLP.

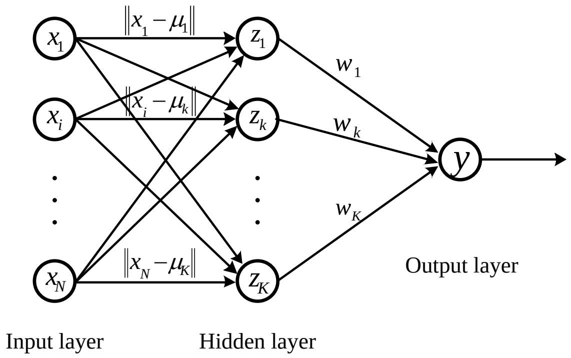

Fig. 3 Architecture of a RBF network [13]

The Two Step cluster algorithm [17] can handle very large data sets, and both continuous and categorical variables. It consists of two steps:

- Pre-cluster the input data into small sub-clusters, i.e., scans the input data and if the conditions are met the current input record is merged with the previously formed clusters or a new one is created based on the distance criterion [18].
- Cluster the sub-clusters into the desired number of clusters by using an

agglomerative hierarchical clustering method [17].

For each cluster the mean and standard deviation for each variable is computed [17]. The center μ_k of the k^th radial basis function is equal to:

- The k^th cluster mean of the i^th input variable, if it is a continuous variable: $\mu_k = \bar{x}_{ki}$.
- The proportion of the category of a categorical variable that the i^th variable corresponds to: $\mu_k = \pi_{ki}$.

The width σ_k of the k^th radial basis function is equal to:

- The k^th cluster standard deviation of the i^th input variable, if it is a continuous variable; where, $h>0$ is the RBF overlapping factor: $\sigma_k = h^{1/2} s_{ki}$.
- Equation (14), if the input variable is categorical:

$$\sigma_k = h^{1/2} \sqrt{n_{ki}(1-n_{ki})} \tag{14}$$

The error function used in RBF network's case is the sum of squares (6) and the activation function is the identity function ($f(x)=x$). The sum of squares error is minimized by using the ordinary least squares regression method [19].

3. METHODOLOGY

The data mining approach proposed in this paper is based on the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. This format is the most complete available and is applied in the present paper to extract information, interpret it and to propose solutions. The CRISP-DM major consisting phases are related in Figure 4 as it is applied to churn prediction modeling.

3.1 Business Understanding

Instead of the existing reactive retention program, where company representatives attempt to convince the customers to stay only when they are calling to leave the company, the mobile telecommunication companies want to implement a proactive retention program which should increase subscribers' loyalty before they decide to leave the company and target all their efforts towards clients who are at risk of churning.

The Churn Model analyzes the Call Detail Records (CRD) of subscribers that have left the company and of those that have remained and determines customers that are at the risk of churning.

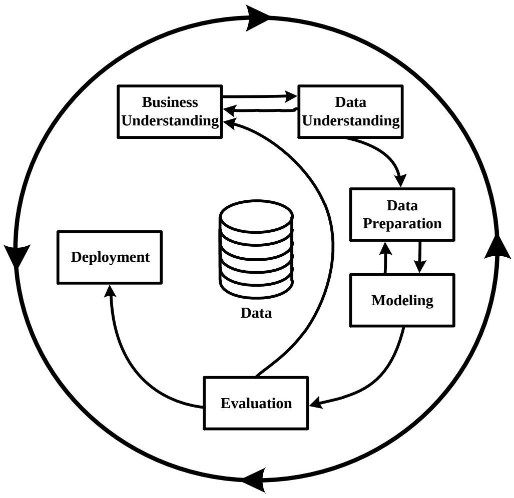

Fig. 4 CRISP-DM Methodology [20]

3.2 Data Understanding

The data set used in this article is from University of California, Department of Information and Computer Science, Irvine, CA [21]. This data set contains historical records of customer churn, how they turned out in hindsight, i.e., their previous behavior – if it turned out that they are churners or not.

The Call Detail Records data set used for this purpose has a total number of 3333 subscribers with 21 variables each. For each of these subscribers we can find information about their corresponding inbound/outbound calls count, inbound/outbound SMS count, and voice mail. The "*Churn*" variable will be used as the target variable, and all other 20 attributes will be used as input variables. In Table 1 we can see all attributes together with their corres-ponding type and value ranges.

Table 1 Data set variables

Variable Name	Type	Values
Account Length	Continuous	1.0 – 243.0
International Plan	Categorical	Yes/No
Voice Mail Plan	Categorical	Yes/No
# Voice Mail Messages	Continuous	0 – 51.0
# Day Minutes	Continuous	0 – 350.8
# Day Calls	Continuous	0 – 165.0
# Evening Minutes	Continuous	0 – 363.7
# Evening Calls	Continuous	0 – 170.0

# Night Minutes	Continuous	23.2 – 395.0
# Night Calls	Continuous	33.0 – 175.0
# International Minutes	Continuous	0 – 20.0
# International Calls	Continuous	0 – 20.0
Customer Service Calls	Continuous	0 – 9.0
Churn?	Categorical	Yes/No
Omitted		
State	Categorical	AK, AL, …
Area Code	Categorical	N/A
Phone	Categorical	N/A
# Day Charge	Continuous	0 – 59.64
# Evening Charge	Continuous	0 – 30.91
# Night Charge	Continuous	1.04 – 17.77
# International Charge	Continuous	0 – 5.4

3.3 Data Preparation

For the data preparation part and all the following we decided to use IBM SPSS (Statistical Product and Service Solutions), a statistical and data mining software package used to build predictive models [22].

Before proceeding to create the churn model, our data need to be cleaned, transformed and prepared in a proper format, suitable for analytical modeling. The first step taken was to audit the data in order to see if there are outliers, extreme or missing values for a particular variable, and to have a first visual contact about advanced statistics pertaining to the variables used. It showed that this is a complete data set, meaning that for each subscriber there is no attribute missing. This saves us additional work, because otherwise, we should have imputed them by choosing the proper method.

During data preparation phase we discovered that between some variables there is a perfect correlation with the R-squared statistic [23] precisely 1. The four *charge* variables are linear functions of the *minutes* variables, so we decided to arbitrarily eliminate all four of them to avoid incoherent results. The *area code* variable contains only three different values (from CA only) for all the records, while the *state* variable contains all 51 states, so we decided not include these two variables as well, as it can be bad data. The *phone* variable has been eliminated because it does not contain relevant data that can be used for prediction; it is useful only for identification purposes. We have therefore reduced the number of predictors from 20 to 13. The target variable is *Churn*, which has two values for each subscriber: *yes* and *no*, telling if a subscriber is a churner or not.

3.4 Modeling

The modeling phase involves choosing the modeling algorithms and the modeling architecture, and finally building the model. Because we intend to build a classification model based on neural networks we must partition the data set into two: a training set and a testing set [24]. The training set has been randomly partitioned to be 80% of the original data set, consisting of 2667 subscribers, whereas the testing set has been randomly partitioned to be 20% of the original data set, consisting of 666 subscribers, as shown in Figure 5.

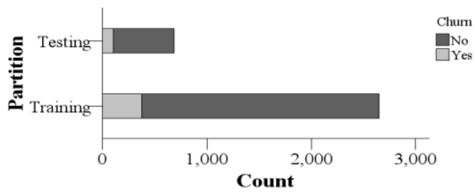

Fig. 5 Samples and churn distributions.

This figure also shows the churn distribution within both sets, and it is as follows: in the testing set we have 107 churners (15%) and 559 non-churners (85%), and in the training set we have 376 churners (14%) and 2291 non-churners (86%).

For implementing the churn predictive model a neural network approach was taken. The Multi-Layer Perceptron and Radial Basis Function neural network algorithms have been chosen to build the predictive models. After multiple configurations of the MLP and RBF neural networks the architectures that are listed in Table 2 and Table 3 are those that drove to better performance in the predictive models.

Table 2 MLP network architecture

MLP network architecture			
Input Layer	# of units	Factors	2
		Covariates	11
Hidden Layer	# of hidden layers		1
	# of units		6
	Activation function		Sigmoid
Output Layer	Dependent variables		1 (churn?)
	# of units		2 (yes/no)
	Activation function		Sigmoid
	Error function		Sum of squares
Optimization algorithm			Gradient descent

Table 3 RBF network architecture

RBF network architecture			
Input Layer	# of units	Factors	2
		Covariates	11
Hidden Layer	# of hidden layers		1
	# of units		11
	Activation function		Gaussian Softmax
Output Layer	Dependent variables		1 (churn?)
	# of units		2 (yes/no)
	Activation function		Identity
	Error function		Sum of squares

The stopping rule used, has stopped training the MLP neural network algorithm because there was no decrease in error after one consecutive step (IBM Corporation, 2011). When the RBF neural network algorithm is used these stopping rules are ignored.

3.5 Evaluation

In this part both models are evaluated in order to decide if the prediction model is a success or not. The performance of both algorithms used can be visualized by using the confusion matrix. The cells on the diagonal of the cross-classification of cases are correct predictions, whilst those off the diagonal are incorrect predictions. Table 4 and Table 5 represent the confusion matrices for MLP and RBF neural network algorithms used in creating the model.

When MLP network was used to create the model, 254 of the 376 subscribers who previously churned are classified correctly. 2253 of the 2291 non-churners are classified correctly. Overall, 94% of the customers from training data set are classified correctly, and 6% are classified incorrectly as shown in Table 4.

Table 4 Confusion matrix for MLP algorithm

Confusion matrix MLP network					
		Predicted			
Sample	Observed	No	Yes	% correct
Training	No	2253	38	98.3%
	Yes	122	254	67.6%
	Overall %	89.1%	10.9%	94. 0%
Testing	No	551	8	98.6%
	Yes	34	73	68.2%
	Overall %	87.8%	12.2%	93.7%

In the testing data set, 73 of the 107 subscribers who previously churned are classified correctly, and 551 of the 559 non-churners are classified correctly. Overall, 93.7% of the customers from testing data set are classified correctly, and 6.7% are classified incorrectly as shown in Table 4.

When RBF network was used to create the model, 195 of the 376 subscribers who previously churned are classified correctly. 2244 of the 2291 non-churners are classified correctly. Overall, 91.5% of the customers from training data set are classified correctly, and 8.5% are classified incorrectly as shown in Table 5 (has a lower performance than MLP model).

Table 5 Confusion matrix for RBF algorithm

Confusion matrix RBF network				
		Predicted		
Sample	Observed	No	Yes	% correct
Training	No	2244	47	97.9%
	Yes	181	195	51.9%
	Overall %	90.9%	9.1%	91.5%
Testing	No	544	15	97.3%
	Yes	49	58	54.2%
	Overall %	89.0%	11.0%	90.4%

In the testing data set, 58 of the 107 subscribers who previously churned are classified correctly, and 544 of the 559 non-churners are classified correctly. Overall, 90.4% of the customers from testing data set are classified correctly, and 9.6% are classified incorrectly (lower performance than MLP model) as shown in Table 5.

This suggest that, overall, our models will correctly classify 9 out of 10 subscribers. But, since our target is to identify customers that are at risk of churning, our MLP model has a percent of correct classification of 68.2% meaning that 7 out of 10 customers will be correctly classified as churners; while our RBF model a percent of 54.2%, meaning that 5 out of 10 customers will be correctly classified as churners. In our case, MLP model has a slightly better performance. The model summaries also tell that because we have almost equal percentages of incorrect predictions in both, training and testing samples, our models have not overtrained.

Another way of interpreting the results are the lift charts. The lift chart sorts the predicted pseudo-probabilities [25] in descending order and display the corresponding curve. There are two types of lift charts: cumulative and incremental. The cumulative lift chart shows how better the prediction rate produced by our models is, compared to the random expectation.

In Figure 6 we can see the curves corresponding to MLP and RBF models and by reading the graph on the horizontal axis, we can see that for the 20th percentile, both models have approximately a 4 lift index on the vertical axis.

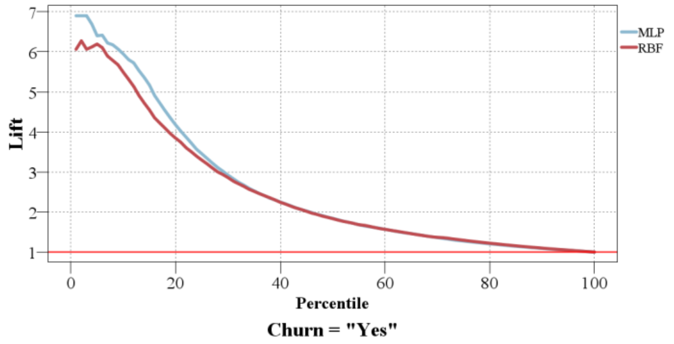

Fig. 6 Cumulative lift chart for MLP and RBF

The incremental lift chart displayed in Figure 7, shows the lift in each percentile [23] without any accumulation. The lift lines corresponding to the MLP and RBF models (blue and respectively burgundy line) descend below the random line (red line) at about 20th percentile, meaning that compared to random expectation our models achieve their best performance in the first 20% of the records.

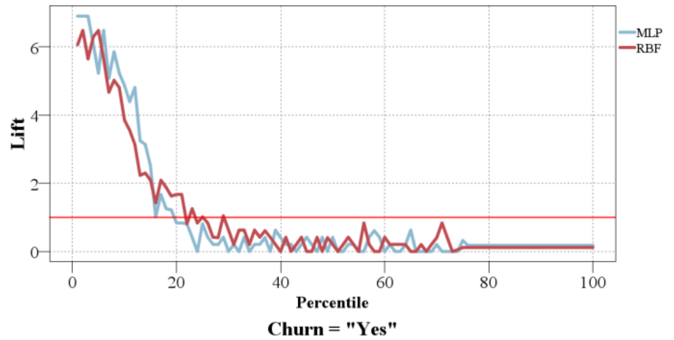

Fig. 7 Incremental lift chart for MLP and RBF

If, for instance, a mobile telecommunications company wants to send offers containing incentives to stay with them, they can easily select the top 20% subscribers in this sorted scored list, and expect to contact more than four times (lift index 4) the number of subscribers that are categorized as churners than normal.

3.6 Deployment

The deployment part of the CRISP-DM process involves organizing knowledge or information gained through data mining models in a way that stakeholders can use it in order to make decisions. The deployment phase could have different outputs, from a simple report to a more complex repeatable data mining process built in a business intelligence environment. The output obtained in this paper is summarized in a report that can be handled to decision making employees in a spread sheet or document format. Complex reports integrated in dashboards can be generated if a business intelligence approach is taken into consideration, where data used in the data mining process is extracted from data warehouse and the results are then observed by stakeholders using reporting tools.

4. CONCLUSIONS

In this paper we built two predictive models for subscribers churn that apply to mobile telecommunications companies, using MLP and RBF neural networks. By evaluating the results from the technical point of view, we observed that for predicting non-churners, MLP model has slightly the same performance as RBF, but for predicting churners it has a better performance, 68.2% compared to 54.2% in RBF's case. There is no need to neglect RBF neural networks, because the results are characteristic only for this data set. RBF networks can handle outliers better than an ordinary neural network, and for real-life extremely large data sets the computational performance is better and worth considering when using the neural networks approach.

From the practical point of view, these two models have a very good performance in predicting churners in a mobile telecommunications company. Decision making employees can build different marketing approaches to retain churners based on the predictors that have higher importance in scoring the model performance. These churn prediction models can be used in other customer response models, such as cross-selling, up-selling, or customer acquisition.

As future research we intend to study and create predictive models using other neural networks, such as Probabilistic neural networks (PNN) and General Regression neural networks (GRNN), as well as other machine learning algorithms, including Support Vector Machine, Random Forests, and Bayesian network.

5. REFERENCES

[1]. Shaw, M., Subramaniam, C., Tan, G.W., Welge, M. E., "*Knowledge management and data mining for marketing*", Decision Support Systems, Vol. 31, no. 1, pp. 127–137, Elsevier, 2001.

[2]. Wei, C. P., Chiu, I. T., "*Turning telecommunications call details to churn prediction: a data mining approach*", Expert Systems with Applications, Vol. 23, pp. 103–112, Pergamon, 2002.

[3]. Richeldi, M., Perrucci, A., "*Churn analysis case study*", Telecom Italia Lab., MiningMart, Torino, 2002.

[4]. Ahn, J. H., Han, S. P., Lee, Y. S., "*Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry*", Telecommunications Policy, Vol. 30, Issues 10–11, pp. 552–568, Elsevier, 2006.

[5]. Verbeke, W., Martens, D., Mues, C., Baesens, B., "*Building comprehensible customer churn prediction models with advanced rule induction techniques*", Expert Systems with Applications, Vol. 38, No. 3, pp. 2354–2364, Elsevier, 2011.

[6]. García, V., Marqués, A. I., Sánchez, J. S., "*Non-parametric statistical analysis of machine learning methods for credit scoring*", Advances in Intelligent Systems and Computing, Volume 171, pp. 263-272, 2012.

[7]. Whigham, S., "*What is data mining in SQL Server 2008/R2 Analysis Services?*", accessed 2011: www.youtube.com, min. 2:00.

[8]. Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., Wang, W., "*Data mining curriculum: A proposal*", Version 1.0, 2006.

[9]. Fouché, G., Langit, L., "*Foundations of SQL Server 2008 R2 Business Intelligence 2nd Edition*", Apress, 2011.

[10]. Gorunescu, F., "*Data Mining Concepts, Models and Techniques*", Springer-Verlag Berlin Heidelberg, 2011.

[11]. Cremene, M., Zahan, S., "*Inteligenta Artificiala, Retele neuronale – teorie si aplicatii in telecomunicatii*", U.T. Press, Cluj-Napoca, 2009.

[12]. Ayoubi, M., "*Nonlinear System Identification Based on Neural Networks with Locally Distributed Dynamics and Applications to Technical Process*", VDI Verlag, Düsseldorf, 1996.

[13]. Bishop, C. M., "*Neural Networks for Pattern Recognition*", Oxford University Press, New York, 2008.

[14]. Haykin, S. O., "*Neural Networks and Learning Machines 3rd Edition*", Prentice Hall, New York, 2009.

[15]. Rumelhart, D. E., McClelland, J. L., The PDP Research Group, "*Parallel Distributed Processing: Explorations in the Microstructure of Cognition*", Vol. 1: Foundations, Cambridge, MA: MIT Press, 1986.

[16]. Wong, B. K., Bodnovich, T. A., Selvi, Y., "*Neural network applications in business: A review and analysis of the literature (1988–1995)*", Decision Support Systems, Vol. 19, pp. 301–320, Elsevier, 1997.

[17]. Zhang, T., Ramakrishnon, R., Livny, M., "*BIRCH: An efficient data clustering method for very large databases*", Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 103–114, Montreal, Canada, 1996.

[18]. IBM Corporation, "*IBM SPSS Modeler 14.2 Algorithms Guide*", Chapter 25, 2011.

[19]. González-Camacho, J. M., De los Campos, G., Pérez, P., Gianola, D., Cairns, J. E., Mahuku, G., Babu, R., Crossa, J., "*Genome-enabled prediction of genetic values using radial basis function neural networks*", Theoretical and

Applied Genetics, 125(4), pp. 759–771, Springer, 2012.

[20]. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. et al., "*CRISP-DM 1.0*", Chicago, IL: SPSS, 2000.

[21]. Blake C.L., Merz, C.J., "*Churn data set*", UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, University of California, Department of Information and Computer Science, Irvine, CA, 1998.

[22]. IBM Corporation, "*IBM SPSS Modeler*", http://www-01.ibm.com/software/analytics/spss/

[23]. Brase, C. H., Brase, C. P., "*Understandable Statistics Concepts and Methods 10th Edition*", Cengage Learning, USA, 2012.

[24]. Sumathi, S., Sivanandam, S. N., "*Introduction to Data Mining and its Applications*", Studies in Computational Intelligence, Vol. 29, Springer, 2006.

[25]. Nedovic, L., Mihailovic, B., Ralevic, N. M., "*Some properties of pseudo-measures and pseudo-probability*", Intelligent Systems and Informatics, SISY 2007, 5th International Symposium on, pp.155-159, 2007.