Deep Ensemble Classifiers and Peer Effects Analysis for Churn Forecasting in Retail Banking

Yuzhou Chen[1], Yulia R. Gel[2(✉)], Vyacheslav Lyubchich[3], and Todd Winship[4]

[1] Southern Methodist University, Dallas, TX, USA
[2] University of Texas at Dallas, Richardson, TX, USA
ygl@utdallas.edu
[3] University of Maryland Center for Environmental Science, Cambridge, MD, USA
[4] Temenos, Surrey, BC, Canada

Abstract. Modern customer analytics offers retailers a variety of unprecedented opportunities to enhance customer intelligence solutions by tracking individual clients and their peers and studying clientele behavioral patterns. While telecommunication providers have been actively utilizing peer network data to improve their customer analytics for a number of years, there yet exists a very limited knowledge on the peer effects in retail banking. We introduce modern deep learning concepts to quantify the impact of social network variables on bank customer attrition. Furthermore, we propose a novel deep ensemble classifier that systematically integrates predictive capabilities of individual classifiers in a meta-level model, by efficiently stacking multiple predictions using convolutional neural networks. We evaluate our methodology in application to customer retention in a retail financial institution in Canada.

1 Introduction

Customer retention is crucial for company profitability and growth. Satisfied customers provide ongoing cross-sell and up-sell opportunities, and tend to refer a pool of new clients. Acquiring a new customer can be 5–25 times (depending on the industry) more expensive than retaining a current one [11]. In the saturated markets of retail banking, the intense competition pushes these costs toward the upper boundary. At the same time, there is a strong association between customer retention and profitability: long-term customers buy more and are less costly to serve, while new ones are likely to continue their churning behavior [16].

Loss of clients, also known as *churn* or customer attrition, is widely recognized as one of the most critical business challenges for a variety of companies, from telecommunication providers to financial institutions. While companies pursue new customers through acquisition marketing efforts, customer churn undermines the business growth. Voluntary turnover rates for banking and finance are the third largest (after hospitality and healthcare) among all industries [7]. Hence, analysis of customer characteristics, such as socio-demographics and

D. Phung et al. (Eds.): PAKDD 2018, LNAI 10937, pp. 373–385, 2018.
https://doi.org/10.1007/978-3-319-93034-3_30

activity patterns, is crucial for predictive identification of customers who are likely to churn, as well as for more efficient targeted application of marketing strategies for customer retention.

Through the theoretical and economic framework of customer retention strategy, [18] show that such reasons as purchase intention, proportion of category purchases and purchase regularity are strongly associated with loyalty decisions. Moreover, decisions of many customers tend to be strongly affected by customer's social neighbors. In the banking industry, 71% of customers turn to friends, family, and colleagues for information on bank products [10]. Still, most marketing tools primarily employ direct approaches, neglecting network effects and treating customers independently of their social network environment. As a result, banks lose invaluable information on the driving forces of customer's purchasing behavior and churn.

Despite the well-documented impacts of peer networks on customer behavior, still very few studies incorporate the network information of bank clientele in the retention models, and one of the reasons is the lack of the explicit network ties in the customer databases. For example, [3] use kinship information deliberately collected from bank customers for the study – this is a costly approach with a number of data quality and data privacy issues. As an alternative, [19] use information on bank transfers and joint loans to build customer networks. This approach, however, is applicable only for large banks, because a single bank with a moderate market share has a high portion of transfers being inter-bank transactions, where detailed information on the second customer is not available, thus, resulting in highly sparse data. In an attempt to enhance customer service and, possibly, the network database, HSBC has recently launched a social network for its business customers [14], which can be considered at this point as an experiment, rather than a standard practice. In this study, we adopt a different method of building the customer network, by taking advantage of information that is readily available at any bank – family name and address of each customer.

Furthermore, our customer dataset is highly unbalanced. That is, the number of non-churners is much larger than that of churners. In turn, most statistical and machine learning classifiers suffer from the inability to detect weak signals in such unbalanced datasets. In binary classification problems, such as customer retention, this phenomenon implies a low specificity or low sensitivity of a classifier. To address this issue, we propose a new deep ensemble classifier which harnesses powers of individual classifiers in a meta-level classifier, using convolutional neural networks. The rationale behind this novel framework is that convolutional neural networks can extract useful features by efficient stacking of multiple predictions. We demonstrate the performance of the new technique in binary classification tasks.

The main contributions of our study are as follows:

- We develop a novel predictive tool for customer churn in retail banking that accounts for the invaluable information on clientele social network effects.
- We introduce deep learning concepts to customer retention analysis in retail banking and propose a cost-effective way of building customer networks.

– We develop a novel deep ensemble classifier, which integrates predictive capabilities of single models in a meta-level classifier, using convolutional neural networks. Our studies indicate that the new deep ensemble classifier delivers a competitive performance, especially in largely unbalanced datasets, and hence has a potential for high utility in a wide variety of classification problems, well beyond customer retention.

The paper is organized as follows. Section 2 provides a background on the related work in social network analysis and customer retention modeling. Section 3 describes the data, and Sect. 4 presents the proposed methodology. Section 5 discusses the main results of the study. Section 6 summarizes the results and outlines directions for future research.

2 Related Work

Nowadays, there exists a plethora of machine learning approaches to customer data mining and retention modeling, ranging from classical regression to neural networks to random forests (e.g., see [12,17,20,23] for a general topic overview). The experiments in [21] showed that neural networks typically outperform logistic regression and decision trees in churn prediction. Nevertheless, the performance of neural networks noticeably deteriorates under a lower monthly churn rate (unbalanced data), that is, the problem that we address in this paper using a new deep ensemble classifier.

Peer networks are known to influence a variety of customer decisions. Applications of social network analysis to customer retention, however, are often limited due to poor availability of data on customer peer networks. Most progress in this direction has been achieved in telecommunication industry, where social networks are naturally observed from the call and message records (e.g., see [1,2,13,24,28]). Constructing networks of bank customers requires additional steps, such as targeted surveys [3], mining the databases of customers and their transactions [19], and, potentially, employing big data approaches for harnessing customer information from disparate sources, including online social media [22]. Overall, the analysis of the impact of peer networks on customer behavior in retail banking remains largely at its infancy, comparing with other industries. We address this challenge by introducing a cost-effective way to collect peer information in retail banking and integrate these data with high predictive utility into customer analytics solutions.

Deep learning (DL) methods continue to attract increasing interest in customer churn prediction, while being a relatively new tool in customer analytics. DL architectures, like the multi-layer feedforward architecture, can effectively capture features of the underlying customer data and learn hierarchical clientele data structures [4,25]. To our knowledge, this paper is the first one to introduce DL concepts into customer retention models in retail banking.

3 Data

The data used in this study comprise a database of all transactions, accounts, and (monthly) snapshots of a customer database of a retail financial institution in North America over a period of 3.5 years (2011.1–2014.6). The customer database contains information collected from the customers themselves (name, address, age, gender, etc.; some of these records are missing or outdated) and from the bank's records about each customer (tenure, number of accounts of each type, total amount owned, etc.; complete and up-to-date records). The customer data were redacted – first names completely removed, family names and addresses replaced by encrypted numeric IDs – so that customers' privacy was protected, but some information about their closeness (derived by matching family name and address IDs) was preserved.

From approximately 30 thousand customers in the sample dataset, we select customers who can make their own financial decisions (above 18 years old) and are likely alive (below 100 years old), then split the data into consecutive baseline and prediction periods. Information from a baseline period is used to predict whether a customer will churn in the nearest future, where *churn* is defined as inactivity (number of transactions is zero) during the prediction period. Hence, churn can be represented as a binary variable taking on the value 0 for customers who stay active in the prediction period and value 1 for those who churn. We use one-year baseline periods (2011, 2012, and 2013) with respective prediction periods of one year (2012 and 2013) and six months (2014.1–6).

3.1 Feature Engineering

Building a set of features for customers is an important step for capturing and quantifying nuances of customer behavior and achieving a superb predictive performance of the customer retention models. We use domain knowledge to create *individual features* (variables) that are potentially associated with customer's retention or churn: age, average time between transactions, time since last transaction, number of loans, number of past transactions, tenure, total savings and total credit balances. For example, middle-aged customers or those having large credit balances often are mortgage owners and will likely stay with the bank for some time. Conversely, older customers may become activity churners when they retire and direct their pension payouts to another bank.

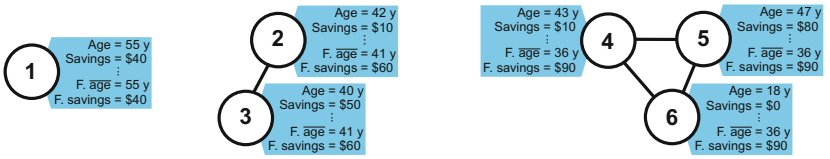

Fig. 1. An example network of six bank customers (nodes), where edges connect people from the same family. Each node has individual and family network features identified

The matching address and family name IDs allow us to create *family networks* (Fig. 1) that join customers who have the same address and family name (the IDs do not reveal any other details, such as neighbor or co-worker relationships). These family networks are cliques, because we consider each family member as connected to everyone else in that family. To capture the dependence of customers on their family members, we apply the egocentric network approach and define family network features for each customer. The *network features* include the individual features aggregated within a family (average age, tenure, total savings, etc.), family size, and two variables indicating whether a family has had churners in the baseline period ("Presence of churners") and how many ("Number of churners in the family").

4 Methods

4.1 Deep Convolutional Neural Networks

Convolutional neural networks (CNN) is a class of artificial neural networks that are based on translational invariance and are weight-shared. These two characteristics increase learning efficiency and make CNN less prone to overfitting than simple artificial neural networks. Hence, for multi-label (binary) datasets, CNN can be trained as a feature extractor and perform better than other classification techniques. An attractive property of CNN is that CNN trained on large datasets have demonstrated an ability to capture high-quality features describing data.

CNN are widely used for image and natural language processing because they can handle static content, like an image or a sentence, well. We make the first attempt to apply CNN for churn prediction in retail banking. In contrast to 2D inputs in image classification, churn input data are 1D. We create a supervised feedforward neural network for binary predictive classification of customer retention. Using the features listed in Sect. 3.1, we found that CNN are able to efficiently mine interesting classification rules.

Architecture. *Convolutional Filter Layer.* Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a high-dimensional input matrix and Y be the output vector. Deep learning can be treated as learning a function, F, mapping input to output:

$$Y = F(\mathbf{X}), \text{ where } X_i \in \mathbb{R}^p,\ Y_i \in \{0, 1\}. \tag{1}$$

In the convolution process, set a filter $\mathbf{w}$ of size k. Then, to obtain a feature m_i of the feature map $\mathbf{m} = [m_1, m_2, \ldots, m_{p-k+1}]$, where $m_i = f(\mathbf{w} \cdot X_{i:i+k-1} + b)$ and b is its corresponding bias offset, apply activation function f to $X_{i:i+k-1}$.

Activation Layer. The role of activation function is to transform the input space of each layer in neural network in such a way that output units become linearly separable. Commonly used activation functions are ReLU $f(x) = max(0, x)$, logistic $\sigma(x) = 1/(1 + \exp(-x))$, and hyperbolic tangent $tanh(x) = 2\sigma(2x) - 1$.

Pooling Layer. Pooling (downsampling) layer decreases the computational complexity and prevents overfitting by reducing number or dimensions in a previous layer. It is done by applying sum-pooling, average-pooling, or max-pooling [24,27]. In our study, we apply max-pooling: $\hat{m} = \max\{\mathbf{m}\}$.

Loss Function and Regularization. The goal of our study is to train the learner (1) using a loss function $\mathcal{L}(\mathbf{y}, \mathbf{o})$, where $\mathbf{o}$ is the output from learner and $\mathbf{y}$ is the true output label. Typically $\mathcal{L}_1 = \|\mathbf{y} - \mathbf{o}\|$ and $\mathcal{L}_2 = \|\mathbf{y} - \mathbf{o}\|^2$ are applied to the regression problem and training process of neural networks. In our case, we use cross-entropy $\mathcal{L} = -\sum_{i=1}^{k} \mathbf{y}_i \log(\mathbf{o}_i)$, because it delivers stable good performance with softmax layer, which is the last layer of our CNN:

$$f(x)_i = e^{x_i} \Big/ \sum\nolimits_{j=1}^{J} e^{x_j}, \quad \text{for } i = 1, \ldots, J.$$

When fitting a model on a relatively small training dataset, overfitting is always a problem for out-of-sample prediction. Neural networks have particularly many parameters that contribute together to building an excessively complex model, which may overfit the data. Dropout [26] is a regularization technique that helps to get an efficient final neural network architecture and to avoid overfitting. The dropout deletes some of the features in $\mathbf{X}$ and, in the training phase, sets the output of each hidden neuron to 0 with probability p. The feed-forward operation for layers $l = 1, \ldots, L-1$ [26]:

$$d_k^{(l)} \sim Bernoulli(p); \;\; \widetilde{y}^{(l)} = d^{(l)} \circ y^{(l)}; \;\; y^{(l+1)} = f(W^{(l+1)}\widetilde{y}^{(l)} + b^{(l+1)}) \;\; , \quad (2)$$

where d is a vector of Bernoulli random variables, $W^{(l)}$ and $y^{(l)}$ are the vectors of biases and outputs from layer l, and $\circ$ denotes the element-wise product.

4.2 Deep Ensemble Classifier

A single model cannot guarantee a uniformly optimal, or at least stable, performance in all cases for which we need to make predictions [29]. Some models are better than others in responding to specific patterns in the data, e.g., those mentioned in Sect. 3.1. A possible solution to this problem is training an ensemble of models and combining their results in some way to obtain more stable and accurate predictions. The stability of out-of-sample (generalization) errors is achieved in ensembles by aggregating information from many models that can potentially overfit the training data, but each model in its own way. Higher accuracy is often achieved even by simple averaging of the single model predictions, but more informed methods, which take into consideration specific strengths and weaknesses of each model, may lead to even better results.

The widely used ensemble methods include bagging, boosting, Bayesian model averaging (BMA), and stacking. Compared with stacking, BMA uses different posterior probabilities to weight each base-level model. The empirical

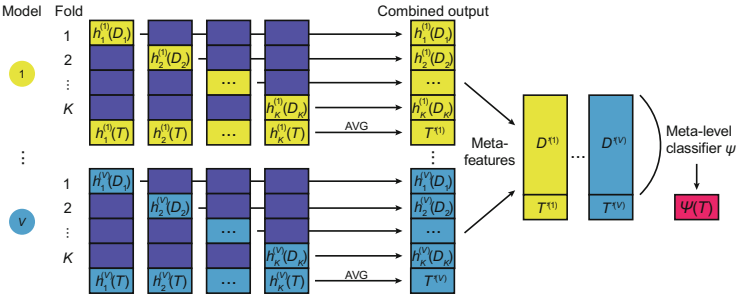

Fig. 2. Stacking with K-fold cross-validation

results in [6] showed that stacking consistently delivers more competitive performance than BMA. BMA works better only when the correct data generating model belongs to the set of model candidates and the noise is low, i.e., under the conditions that are very difficult to satisfy in applications. In turn, stacking outperforms other ensemble methods due to its ability to learn and flexibly account for the behaviors of other classifiers in a combining model [9].

The standard stacking technique is based on applying a logistic regression on the outputs of base-level models, which limits us to the case of monotonic relationships (also, with the same speed of approaching both asymptotes) between predictions from each base-level model and the response. We relax this condition and develop a new deep ensemble classifier for building the second layer of classifiers, based on more flexible machine learning methods. In particular, we propose and evaluate the performance of the following stacking approaches: stacking with CNN (StCNN); stacking with RF (StRF); stacking with XGB (StXGB); stacking with Extra-Trees (StET); stacking with NN (StNN), and stacking with KNN (StKNN). We also use K-fold cross-validation, which provides a good trade-off between variance and bias (see Algorithm 1 and Fig. 2).

5 Results

To compare the performance of single and stacked models and see the effect of adding network features, we design the following four scenarios: (i) single base-level model with individual features alone; (ii) single base-level model with both individual and network features; (iii) stacked models with individual features alone, and (iv) stacked models with both individual and network features.

We split the data into training ($\mathcal{D}$, 70%) and testing ($\mathcal{T}$, 30%) subsets and report results for predicting for the testing subset. On the dataset $\mathcal{D}$, we train five different single base-level models with the following methods: random forest (RF), extreme gradient boosting (XGB [5]), K-nearest neighbor algorithm (KNN), neural networks (NN), and CNN. Each of the considered five methods can provide several well-performing models with different tuning parameters, and we can use all of them when creating an ensemble (thus, each ensemble we

Algorithm 1. Stacking with K-fold cross-validation

INPUT: Training set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{m}$ and testing set $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ ($\mathbf{x}_i \in \mathbb{R}^p, y_i \in \{0, 1\}$), number of folds K, and V different base-level classifiers

OUTPUT: A meta-level classifier Ψ

1: Randomly split $\mathcal{D}$ into K equal-size subsets: $\mathcal{D} \leftarrow \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$
2: **for** v in 1 to V **do**
3: **for** k in 1 to K **do**
4: Train a classifier $h_k^{(v)}$ on $\mathcal{D} \setminus \mathcal{D}_k$
5: Let $h_k^{(v)}(\mathcal{D}_k)$ be the out-of-fold predictions for the set $\mathcal{D}_k$
6: Let $h_k^{(v)}(\mathcal{T})$ be predictions for the testing set
7: **end for**
8: Construct a new variable from out-of-fold predictions in the training set:

$$\mathcal{D}'^{(v)} \leftarrow \{h_1^{(v)}(\mathcal{D}_1), h_2^{(v)}(\mathcal{D}_2), \ldots, h_K^{(v)}(\mathcal{D}_K)\}$$

9: The new variable in the testing set is an average of K predictions:

$$\mathcal{T}'^{(v)} \leftarrow \text{AVG}\{h_1^{(v)}(\mathcal{T}), h_2^{(v)}(\mathcal{T}), \cdots, h_K^{(v)}(\mathcal{T})\}$$

10: **end for**
11: Train a meta-level classifier ψ on $\left\{\left(\mathcal{D}'^{(1)}, \mathcal{D}'^{(2)}, \ldots, \mathcal{D}'^{(V)}\right), y_i\right\}_{i=1}^{m}$
12: Return $\Psi(\mathcal{T}) \leftarrow \psi\left(\mathcal{T}'^{(1)}, \mathcal{T}'^{(2)}, \cdots, \mathcal{T}'^{(V)}\right)$

created had more than five members). In the stacking Algorithm 1, we use 4-fold cross-validation for the first two time periods and 7-fold cross-validation for the third period. The optimal parameters for each base-level model were chosen through a grid search.

The CNN architectures used in this study are shown in Table 1. In the CNN training, we use tanh as an activation function, and ReLU in the second layer. The advantages of ReLU include faster model training and smaller chance of the gradient to vanish. We apply dropout with the probability $p = 0.6$ at the second layer before pooling and insert a batch normalization layer [15] (eps = 0.00001, momentum = 0.99) before applying the activation function in the second layer. Stochastic gradient descent was chosen as the CNN optimizer, with the learning rate of 0.001 and momentum value of 0.9 as optimal parameters.

For each of the scenarios (i)–(iv), Table 2 layouts a confusion matrix $\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$ for the subset $\mathcal{T}$, where TN is the number of non-churners classified as non-churners, FP is the number of non-churners classified as churners, FN is the number of churners classified as non-churners, and TP is the number of churners classified as churners. Table 2 shows that improved churn predictions can be achieved by leveraging the CNN architecture, novel stacking approach (StCNN), and customer network features.

Figure 3 reports the misclassification rates $R = (FN + FP)/N$ (where $N = |\mathcal{T}|$ is the size of the testing set) delivered by various base-level models on the

Table 1. Architectures of CNN

Layer	Layer type	Size of base level	Size of meta level
1	Convolution + tanh	1×2 20 filters	1×2 30 filters
1	Max pooling	1×3, stride 1	1×2, stride 1
2	Convolution + ReLU	1×3 50 filters	1×2 50 filters
2	Max pooling	1×5, stride 1	1×2, stride 1
3	Fully connected + tanh	500 hidden units	500 hidden units
4	Fully connected + tanh	2 hidden units	2 hidden units
5	Softmax	2 ways	2 ways

test datasets. The base-level models XGB, RF, and CNN outperform NN and KNN in each period.

Remarkably, CNN that include both individual and network features perform noticeably better than other baseline models for the 1st and 2nd periods. We also observe that RF performs better in the 3rd period when using both individual and network features. The results in Fig. 3 prove that most of our single models

Table 2. Confusion matrices[a] of predictive classifying of bank customers

Baseline period	Prediction period	Model	Individual features		Individual & network features	
			Single model	Stacked models	Single model	Stacked models
2011.1–12	2012.1–12	CNN	6634 13 108 240	6628 19 101 247	6629 18 101 247	**6630 17** **99 249**
		XGB	6618 29 101 247	6621 26 102 246	6617 30 104 244	6616 31 97 251
		RF	6627 20 103 245	6626 21 105 243	6623 24 102 246	6630 17 105 243
2012.1–12	2013.1–12	CNN	8305 20 29 280	8304 21 26 283	**8305 20** **26 283**	**8305 20** **26 283**
		XGB	8304 21 41 268	8303 22 34 275	8304 21 37 272	8305 20 30 279
		RF	8304 21 27 282	8304 21 31 278	8303 22 29 280	8304 21 30 279
2013.1–12	2014.1–6	CNN	8227 14 78 749	**8219 22** **54 773**	8219 22 66 761	8221 20 61 766
		XGB	8207 34 57 770	8212 29 54 773	8214 27 61 766	8214 27 59 768
		RF	8220 21 62 765	8216 25 59 768	8217 24 58 769	8214 27 56 771

[a]Each cell is a 2×2 confusion matrix. For each period, matrices with minimal sum $FP+FN$ are highlighted.

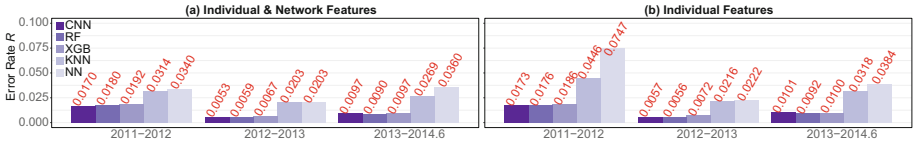

Fig. 3. Performance of base-level algorithms with different sets of features

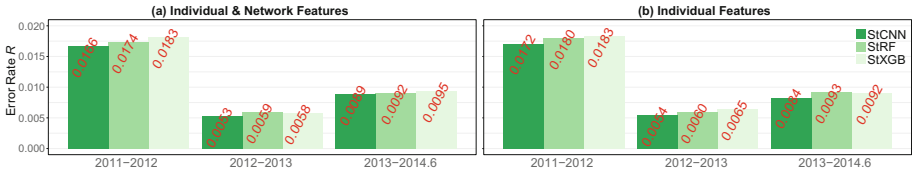

Fig. 4. Performance of meta-level algorithms with different sets of features

(especially CNN and RF) trained on both individual and network features are more accurate (lower R) than models trained exclusively on individual features.

Aggregation of results by stacking further improves the predictive performance. Among the six considered stacking algorithms, the best three are based on CNN, RF, and XGB – the algorithms that also show the best performance in the base-level scenarios (Fig. 3). Accuracy of these three methods is noticeably higher than of the other three (StET, StNN, and StKNN), while running time of XGB is considerably shorter. Figure 4 shows that StCNN always outperforms the other stacking schemes. Compared with Fig. 3, accuracy of the best-performing combinations changed as follows:

- improved from 98.30% (CNN with both individual and network features) to 98.34% (StCNN with both individual and network features), i.e., by 0.04 percentage points, for the period 2011–2012;
- stayed at about 99.47% (CNN and StCNN, each with both individual and network features) for 2012–2013;
- improved from 99.10% (RF with both individual and network features) to 99.16% (StCNN with individual features), i.e., by 0.06 points, for the period 2013–2014.6.

The results imply that the architecture of CNN can improve the performance of churn predictive classification with automatically capturing and extracting relevant features, especially after adding network features into the model. Furthermore, StCNN can simultaneously reduce false negative rates and yield the optimal true negative rate. Nevertheless, in the absence, to the best of our knowledge, of a formal statistical test applicable to a stacking scheme, more extensive experiments based on a cross-validation argument are needed to prove the statistical significance of the improvement of using StCNN.

In the StCNN, we use Adagrad [8] as the optimizer, and set learning rate, epsilon, and L_2 regularization coefficient (wd) to 10^{-2}, 10^{-10}, and 10^{-3} by tuning

with a grid search. The accuracy of the above results is high (i.e., errors R are low) in part due to a very low proportion of churned customers (below 6%). The dataset is unbalanced, as well as the costs of losing a customer. Various studies suggest that such costs can be 5–25 times higher than the costs of retaining an existing one [11], but the results above assume equal costs of FP and FN.

		2011 – 2012				2012 – 2013				2013 – 2014.6			
Features:		Individual		Individual & network		Individual		Individual & network		Individual		Individual & network	
Model:		Single	Stacked	Single	Stacked	Single	Stacked	Single	Stacked	Single	Stacked	Single	Stacked
Method	XGB	0.2165	0.2192	0.2212	0.2091	0.0857	0.0724	0.0782	0.0647	0.1087	0.1042	0.1161	0.1128
	RF	0.2229	0.2259	0.2198	0.2273	0.0588	0.0666	0.0627	0.0647	0.1185	0.1130	0.1114	0.1078
	CNN	0.2338	0.2198	0.2202	0.2171	0.0627	0.0568	0.0568	0.0568	0.1455	0.1048	0.1250	0.1169

Fig. 5. Misclassification rates R' (dark color shade means smaller) when churners weigh 20 times more than non-churners ($r = 20$)

In Fig. 5, we use cost ratio ($r = 20$) of FN to FP to upweight the errors in misclassifying churners ($FN + TP$):

$$R' = \frac{r \cdot FN + FP}{N + (r-1)(FN + TP)}. \tag{3}$$

Figure 5 shows that the egocentric network approach and model stacking, in particular, the StCNN, improve the churn predictions.

6 Conclusion

We have proposed a novel predictive tool for customer retention in retail banking by introducing deep learning concepts into churn analysis. Our approach allows to systematically and consistently integrate invaluable information on customer peer effects into the customer analytics process. We have developed a new deep ensemble classifier that fuses predictive powers of individual classifiers in a meta-level model, by efficiently stacking multiple predictions using convolutional neural networks. The proposed deep ensemble classifier delivers competitive performance in largely unbalanced customer data and, hence, has a potential for a wide applicability in classification problems well beyond customer analytics.

Acknowledgements. This research was partially supported by NSF IIS 1633331 & 1633355, NSF DMS 1736368, and Simons Foundation. The work of V. Lyubchich was supported by Mitacs Accelerate Internship Awards with contributions from Temenos Canada.

References

1. Backiel, A., Baesens, B., Claeskens, G.: Mining telecommunication networks to enhance customer lifetime predictions. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2014. LNCS (LNAI), vol. 8468, pp. 15–26. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07176-3_2
2. Backiel, A., Baesens, B., Claeskens, G.: Predicting time-to-churn of prepaid mobile telephone customers using social network analysis. J. Oper. Res. Soc. **67**(9), 1135–1145 (2016)
3. Benoit, D.F., Van den Poel, D.: Improving customer retention in financial services using kinship network information. Expert Syst. Appl. **39**(13), 11435–11442 (2012)
4. Castanedo, F., Valverde, G., Zaratiegui, J., Vazquez, A.: Using deep learning to predict customer churn in a mobile telecommunication network (2014)
5. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
6. Clarke, B.: Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. J. Mach. Learn. Res. **4**, 683–712 (2003)
7. Compensation Force: 2016 turnover rates by industry (2017)
8. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
9. Džeroski, S., Ženko, B.: Is combining classifiers with stacking better than selecting the best one? Mach. Learn. **54**(3), 255–273 (2004)
10. Ernst & Young: The customer takes control. Consumer Banking Survey (2012)
11. Gallo, A.: The value of keeping the right customers. Harv. Bus. Rev. **5**, 2–6 (2014)
12. Han, S.H., Lu, S.X., Leung, S.C.: Segmentation of telecom customers based on customer value by decision tree model. Expert Syst. Appl. **39**(4), 3964–3973 (2012)
13. Hill, S., Provost, F., Volinsky, C.: Network-based marketing: identifying likely adopters via consumer networks. Stat. Sci. **21**(2), 256–276 (2006)
14. HSBC: HSBC launches global 'social network' for business customers (2017)
15. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
16. Larivière, B., Van den Poel, D.: Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Syst. Appl. **29**(2), 472–484 (2005)
17. Li, D.C., Dai, W.L., Tseng, W.T.: A two-stage clustering method to analyze customer characteristics to build discriminative customer management: a case of textile manufacturing business. Expert Syst. Appl. **38**(6), 7186–7191 (2011)
18. Macintosh, G., Lockshin, L.S.: Retail relationships and store loyalty: a multi-level perspective. Int. J. Res. Mark. **14**(5), 487–497 (1997)
19. Mao, H., Jin, X., Zhu, L.: Methods of measuring influence of bank customer using social network model. Am. J. Ind. Bus. Manag. **5**(4), 155 (2015)
20. Miguéis, V.L., Van den Poel, D., Camanho, A.S., e Cunha, J.F.: Modeling partial customer churn: on the value of first product-category purchase sequences. Expert Syst. Appl. **39**(12), 11250–11256 (2012)
21. Mozer, M.C., Wolniewicz, R., Grimes, D.B., Johnson, E., Kaushansky, H.: Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. IEEE Trans. Neural Netw. **11**(3), 690–696 (2000)

22. NG Data: Predicting and preventing customer churn by unlocking big data (2013)
23. Ngai, E.W., Xiu, L., Chau, D.C.: Application of data mining techniques in customer relationship management: a literature review and classification. Expert Syst. Appl. **36**(2), 2592–2602 (2009)
24. Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6354, pp. 92–101. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15825-4_10
25. Spanoudes, P., Nguyen, T.: Deep learning in customer churn prediction: unsupervised feature learning on abstract company independent feature vectors. arXiv:1703.03869 (2017)
26. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
27. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of CVPR, pp. 1–9. IEEE (2015)
28. Verbeke, W., Martens, D., Baesens, B.: Social network analysis for customer churn prediction. Appl. Soft Comput. **14**, 431–446 (2014)
29. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2016)