Automated Feature Selection and Churn Prediction using Deep Learning Models

V. Umayaparvathi[1], **K. Iyakutti**[2]

[1] Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, Tamilnadu, India
[2] Professor-Emeritus, Department of Physics and Nanotechnology, SRM University, Chennai, Tamilnadu, India

Abstract – *In this competitive world, mobile telecommunications market tends to reach a saturation state and faces a fierce competition. This situation forces the telecom companies to focus their attention on keeping the customers intact instead of building a large customer base. According to telecom market, the process of subscribers (either prepaid or postpaid) switching from a service provider is called customer churn. Several predictive models have been proposed in the literature for churn prediction.*

The efficiency of any churn prediction model depends highly on the selection of customer attributes (feature selection) from the dataset for its model construction. These traditional methods have two major problems: 1) With hundreds of customer attributes, existing manual feature engineering process is very tedious and time consuming and often performed by a domain expert: 2) Often it is tailored to specific dataset, hence we need to repeat the feature engineering process for different datasets. Since deep learning algorithms automatically comes up with good features and representation for the input data, we investigated their applications for customer churn prediction problem. We developed three deep neural network architectures and built the corresponding churn prediction model using two telecom dataset. Our experimental results show that deep-learning based models are performing as good as traditional classification models, without even using the hand-picked features.

Key Words: Customer relationship management (CRM), Data mining, Churn prediction, Predictive models, and Deep learning.

1. INTRODUCTION

Today is the competitive world of communication technologies. Customer Churn is the major issue that almost all the Telecommunication Industries in the world faces now. In telecommunication paradigm, Churn is defined to be the activity of customers leaving the company and discarding the services offered by it due to dissatisfaction of the services and/or due to better offering from other network providers within the affordable price tag of the customer. This leads to a potential loss of revenue/profit to the company. Also, it has become a challenging task to retain the customers. Therefore, companies are going behind introducing new state of the art applications and technologies to offer their customers as much better services as possible so as to retain them intact. Before doing so, it is necessary to identify those customers who are likely to leave the company in the near future in advance because losing them would results in significant loss of profit for the company. This process is called Churn Prediction.

Data mining techniques are found to be more effective in predicting customer churn from the researches carried out during the past few years. The construction of effective churn prediction model is a significant task which involves lots of research right from the identification of optimal predictor variables (features) from the large volume of available customer data to the selection of effective predictive data mining technique that is suitable for the feature set. Telecom Industries collect a voluminous amount of data regarding customers such as Customer Profiling, Calling pattern, Democratic data in addition to the network data that are generated by them. Based on the history of the customers calling pattern and the behavior, there is a possibility to identify their mindset of either they will leave or not.

The efficiency of any churn prediction model depends highly on the selection of customer attributes (feature selection) from the dataset for its model construction. These traditional methods have two major problems: 1) With hundreds of customer attributes, existing manual feature engineering process is very tedious and time consuming and often performed by a domain expert: 2) Often it is tailored to specific dataset, hence we need to repeat the feature engineering process for different datasets. Since deep learning algorithms automatically comes up with good features and representation for the input data, we investigated their applications for customer churn prediction problem. We developed three deep neural network architectures and built the corresponding churn prediction model using two telecom dataset. Our experimental results show that deep-learning based models are performing as good as traditional classification models, without even using the hand-picked features.

The rest of the paper is organized as follows. In Section 2, we review the existing predictive models proposed in the literature for churn prediction. In Section 3, we present the details about deep-learning networks. In section 4, we present the architecture of the proposed deep-learning models. In Section 5, we present the experimental setup and results, followed by, in Section 6, we conclude the paper with future directions of this research.

2. RELATED WORK

There are a lot of researches being carried out in the area of customer churn prediction modeling. In this section, we survey some of the researches carried out in this area in the past few years.

Authors of [1] depict phases of a general churn prediction model such as data collection, preparation, classification and prediction. It also describes that identifying the right grouping of variables has significant influence in improving the percentage of true predictions (TP). A churn prediction model was proposed by [1], which works in 5 steps: i) problem identification; ii) dataset selection; iii) investigation of data set; iv) classification; v) clustering, and vi) using the knowledge. It seems to be a complete model. Classification techniques are used for distinguishing Churners. Clustering is used for model evaluation. For classification, Decision tree, Support vector machine and Neural Network are used. And for clustering Simple K-Means was used. It concluded that SVM was the best among the three methods in distinguishing churners from non-churners.

Building an effective customer churn prediction model using various techniques has become a significant topic for business and academics in recent years [2]. The identification of why customers give up their relationships has been focus of marketing research for the past few years [3]. Due to the enormous growth of customer related data and call detail data collected and maintained by the companies in the recent years, more sophisticated metrics have evolved to describe customer behaviour and better understand how behavioural characteristics can be linked to customer retention and firm performance [4].

Authors in [5] proposed a decision tree based Random Forest method for feature extraction. From the original data with Q features, exactly N samples with q < Q are randomly selected for each tree to form a forest. The number of trees depends on the number of features randomly combined for each decision tree from the whole Q features. Development of a predictive model based on data-centric approach for detecting the early warning signs of churn, and for developing a Churn Score to identify subscribers who are likely to end their relationship with the company was discussed in [6], where the customer's calling pattern played the major role in predicting churn.

An overview of socially grouped users and their behavioural pattern are elaborately identified in [7]. It also explores the impact of centrality features among customers. It concludes that when a customer in a group leaves the network, there is a high probability of others in that group to leave the network. It classifies the feature variables in two types: i) dependent variables (call duration of in-degree and out-degree), ii) independent variables (social forum like services and the customer's involvement in the forum). The level of customer's active participation was used as the measure of probability of churn. Customers who are members of multiple community forums are at high risk than those who are members in less number of forums.

In the recent past, deep learning algorithms have evolved to provide outstanding results in computer vision compared to the traditional classifiers. Authors of [8], applied deep convolutional neural networks and auto-encoders for building a churn prediction model. They transformed the temporal behavior of customers into images and then they modeled the churn prediction problem as image classification problem. Their experimental results show that their deep learning model outperforms decision trees. Authors of [9] also used deep learning models for churn prediction. They applied auto encoders, deep belief networks and multi-layer forward networks for churn prediction and showed that the deep learning models achieves better accuracy compared to random forests. In contrast with all these approaches, we designed three deep learning models with increasing complexity using simple and convolutional neural networks for the churn prediction task. We trained our networks over two public datasets and show that some datasets deep learning networks are better than traditional classifiers but not for all the datasets.

3. DEEP LEARNING MODELS

In this section, we first present an overview of deep learning networks and then present the proposed deep learning architectures for churn prediction task.

3.1 Overview of Deep Learning

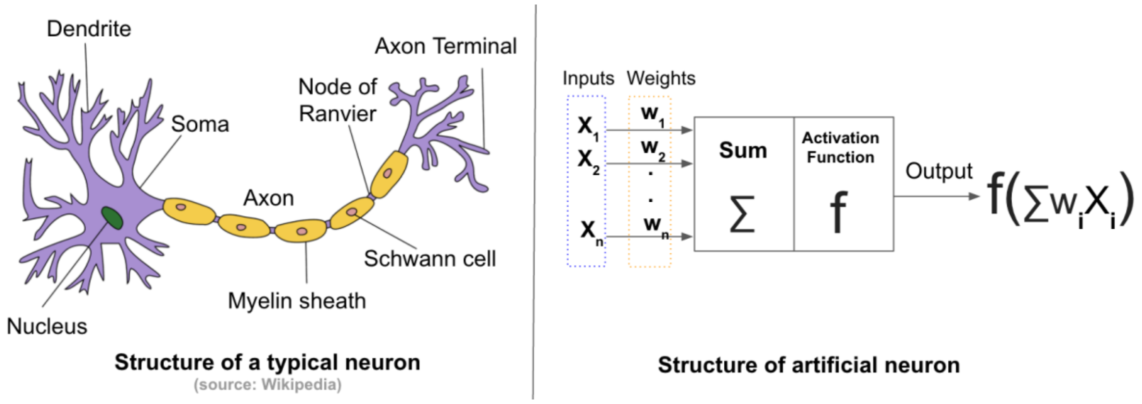
Deep learning refers to a class of artificial neural networks (ANNs) composed of many processing layers. ANNs existed for many decades, but attempts at training deep architectures of ANNs failed until Geoffrey Hinton's breakthrough work of the mid-2000s. In addition to algorithmic innovations, the increase in computing capabilities using GPUs and the collection of larger datasets are all factors that helped in the recent surge of deep learning.

3.1.1 Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) are a family of machine learning models inspired by biological neural networks.

Artificial Neural Networks vs. Biological Neural Networks

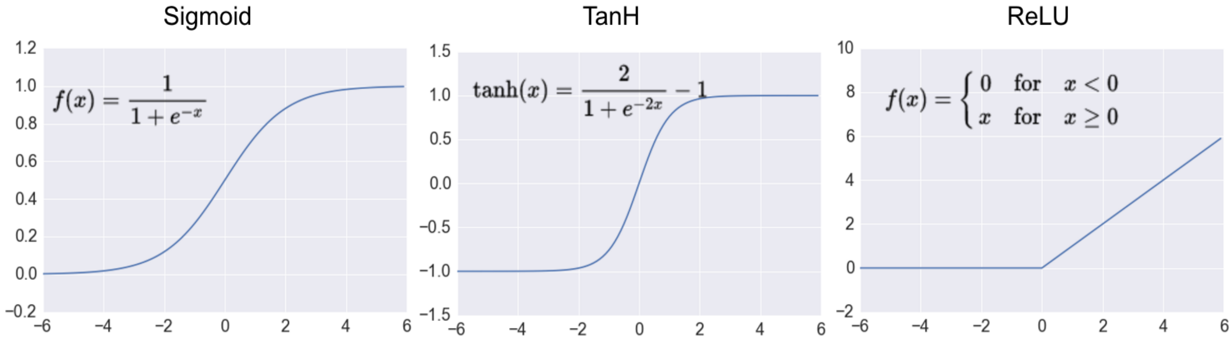
Biological Neurons are the core components of the human brain. A neuron consists of a cell body, dendrites, and an axon. It processes and transmit information to other neurons by emitting electrical signals. Each neuron receives input signals from its dendrites and produces output signals along its axon. The axon branches out and connects via synapses to dendrites of other neurons.



Structure of a typical neuron
(source: Wikipedia)

Structure of artificial neuron

Figure 1: Similarities between biological neurons and artificial neurons
Source: http://adilmoujahid.com


Sigmoid

TanH

ReLU

Figure 2: Commonly used activation functions such as Sigmoid, TanH and ReLU
Source: http://adilmoujahid.com

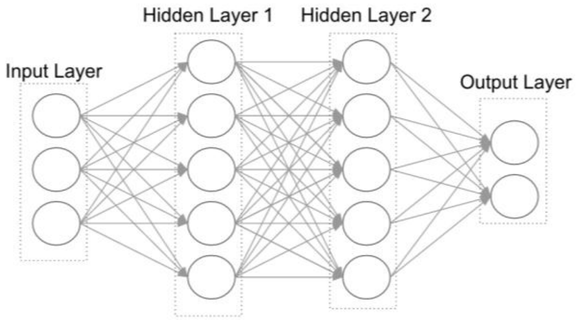


Figure 3: An example feedforward neural network with 2 hidden layers (Source: http://adilmoujahid.com)

A basic model for how the neurons work goes as follows: Each synapse has a strength that is learnable and control the strength of influence of one neuron on another. The dendrites carry the signals to the target neuron's body where they get summed. If the final sum is above a certain threshold, the neuron get fired, sending a spike along its axon.

Artificial neurons are inspired by biological neurons, and try to formulate the model explained above in a computational form. An artificial neuron has a finite number of inputs with weights associated to them, and an activation function (also called transfer function). The output of the neuron is the result of the activation function applied to the weighted sum of inputs. Artificial neurons are connected with each other's to form artificial neural networks. Similarities between biological neurons and artificial neurons are illustrated in Figure 1.

Feedforward Neural Networks

Feedforward Neural Networks are the simplest form of Artificial Neural Networks. These networks have 3 types of layers: Input layer, hidden layer and output layer. In these networks, data moves from the input layer through the hidden nodes (if any) and to the output nodes. A fully-connected feedforward neural network with 2 hidden layers is shown in Figure 3. Fully-connected means that each node is connected to all the nodes in the next layer. Note that, the number of hidden layers and their size are the only free parameters. The larger and deeper the hidden layers, the more complex patterns we can model in theory.

Activation Functions

Activation functions transform the weighted sum of inputs that goes into the artificial neurons. These functions should be non-linear to encode complex patterns of the data. The most popular activation functions are Sigmoid, Tanh and ReLU. ReLU is the most popular activation function in deep neural networks. Figure 2 shows the plots for these commonly used activation functions.

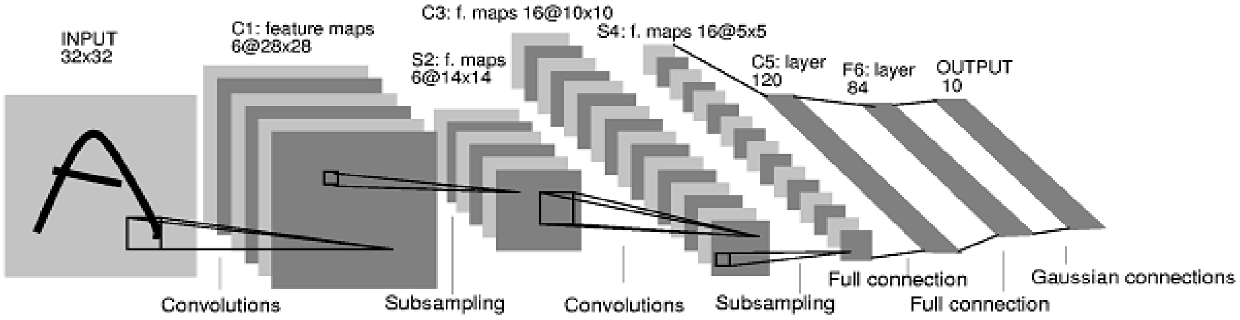


Figure 4: An example Convolutional Neural Network called LeNet by Yann LeCun (1988)

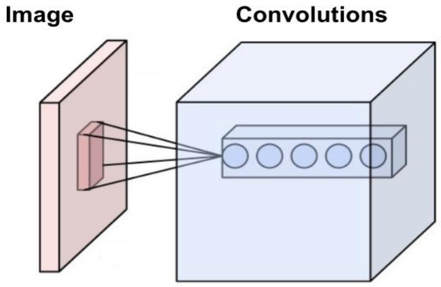


Figure 5: Neurons of convolutional layer, connected to their respective field (Source: Wikipedia)

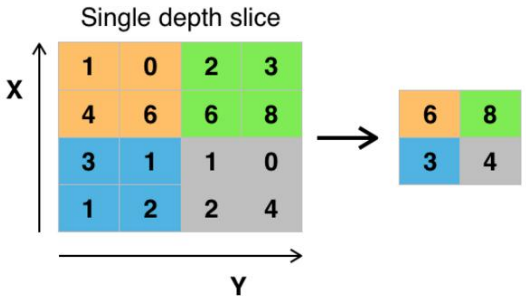


Figure 6: Max polling with a 2x2 filter and stride =2 (Source: Wikipedia)

Training Artificial Neural Networks

The goal of the training phase is to learn the network's weights. We need 2 elements to train an artificial neural network:

- Training data: In the case of image classification, the training data is composed of images and the corresponding labels.
- Loss function: A function that measures the inaccuracy of predictions.

Once we have the 2 elements above, we train the ANN using an algorithm called back-propagation together with gradient descent (or one of its derivatives).

3.1.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks are a special type of feed-forward networks, as shown in Figure 4. These models are designed to emulate the behavior of a visual cortex. CNNs perform very well on visual recognition tasks. CNNs have special layers called convolutional layers and pooling layers that allow the network to encode certain images properties.

Convolution Layer

This layer consists of a set of learnable filters that we slide over the image spatially, as shown in Figure 5, computing dot products between the entries of the filter and the input image. The filters should extend to the full depth of the input image. For example, if we want to apply a filter of size 5x5 to a colored image of size 32x32, then the filter should have depth 3 (5x5x3) to cover all 3 color channels. These filters will activate when they see same specific structure.

Pooling Layer

Pooling is a form of non-linear down-sampling, as shown in Figure 6. The goal of the pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control over fitting. There are several functions to implement pooling among which max pooling is the most common one. Pooling is often applied with filters of size 2x2 applied with a stride of 2 at every depth slice. A pooling layer of size 2x2 with stride of 2 shrinks the input image to a 1/4 of its original size.

Convolutional Neural Networks Architecture

The simplest architecture of a convolutional neural networks starts with an input layer (images) followed by a sequence of convolutional layers and pooling layers, and ends with fully-connected layers. The convolutional layers are usually followed by one layer of ReLU activation functions. The convolutional, pooling and ReLU layers act as learnable features extractors, while the fully connected layers acts as a machine learning classifier. Furthermore, the early layers of the network encode generic patterns of the images, while later layers encode the details patterns of the images. Note that only the convolutional layers and fully-connected layers have weights. These weights are learned in the training phase.

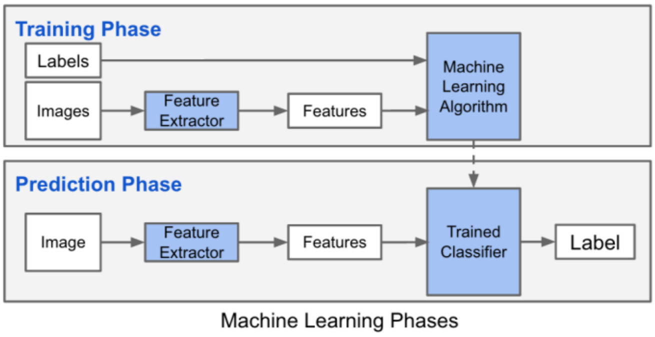


Figure 7: Traditional machine learning workflow for churn prediction task with manual feature engineering task.

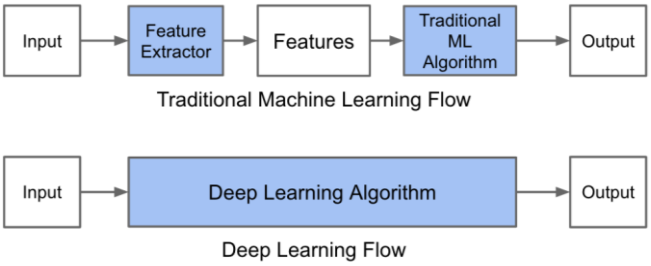


Figure 8: Deep learning based workflow for churn prediction task which does not involve manual feature engineering task

3.1.3 Classification using Traditional Machine Learning vs. Deep Learning

Classification using a machine learning algorithm has 2 phases, as shown in Figure 7:

1. Training phase:

   In this phase, we train a machine learning algorithm using a dataset comprised of the images and their corresponding labels.

2. Prediction phase:

   In this phase, we utilize the trained model to predict labels of unseen data.

The training phase for an image classification problem has 2 main steps:

1. Feature Extraction:

   In this phase, we utilize domain knowledge to extract new features that will be used by the machine learning algorithm..

2. Model Training:

   In this phase, we utilize a clean dataset composed of the customer data features and the corresponding labels to train the machine learning model.

In the predication phase, we apply the same feature extraction process to the new images and we pass the features to the trained machine learning algorithm to predict the label.

The main difference between traditional machine learning and deep learning algorithms is in the feature engineering. In traditional machine learning algorithms, we need to hand-craft the features. By contrast, as shown in Figure 8 in deep learning algorithms feature engineering is done automatically by the algorithm. Feature engineering is difficult, time-consuming and requires domain expertise. The promise of deep learning is more accurate machine learning algorithms compared to traditional machine learning with less or no feature engineering.

4. METHODOLOGY

We designed two neural network architectures for the churn prediction task. Before that we will summarize the neural network layer details

Dense Layer: A dense layer is simply a layer where each unit or neuron is connected to each neuron in the next layer. It can be initialized with different initialization parameters and activation functions

Dropout Layer: Dropout is a regularization technique, which aims to reduce the complexity of the model with the goal to prevent over-fitting. Using “dropout", you randomly deactivate certain units (neurons) in a layer with a certain probability p from a Bernoulli distribution (typically 50%, but this yet another hyper-parameter to be tuned). So, if you set half of the activations of a layer to zero, the neural network won’t be able to rely on particular activations in a given feed-forward pass during training. As a consequence, the neural network will learn different, redundant representations; the network can’t rely on the particular neurons and the combination (or interaction) of these to be present. Another nice side effect is that training will be faster.

Embedding Layer: It turn positive integers (indexes) into dense vectors of fixed size. Embedding layer is a simple matrix multiplication that transforms words into their corresponding word embedding. The weights of the Embedding layer are of the shape (vocabulary size, embedding dimension). For each training sample, its input are integers, which represent certain words. The integers are in the range of the vocabulary size. The Embedding layer transforms each integer i into the ith line of the embedding weights matrix.

4.1 Feedforward Neural Network (FNN)

The feedforward neural network was the first and simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. We created two FNN architectures with small and large number of hidden layers to learn both higher level and lower level details of the customer attributes.

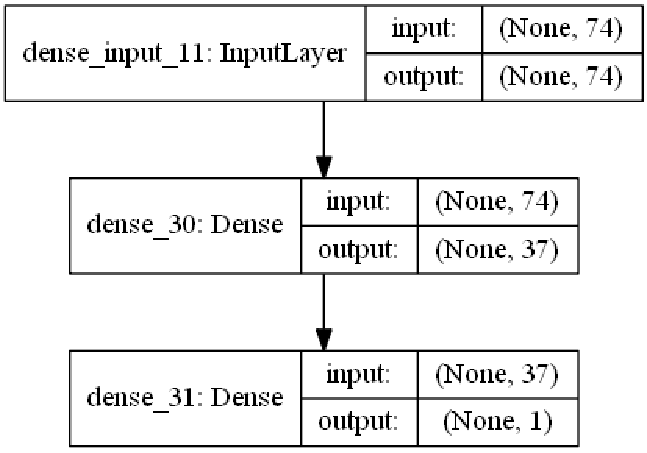


Figure 9: Small FNN network architecture for the Cell2Cell dataset

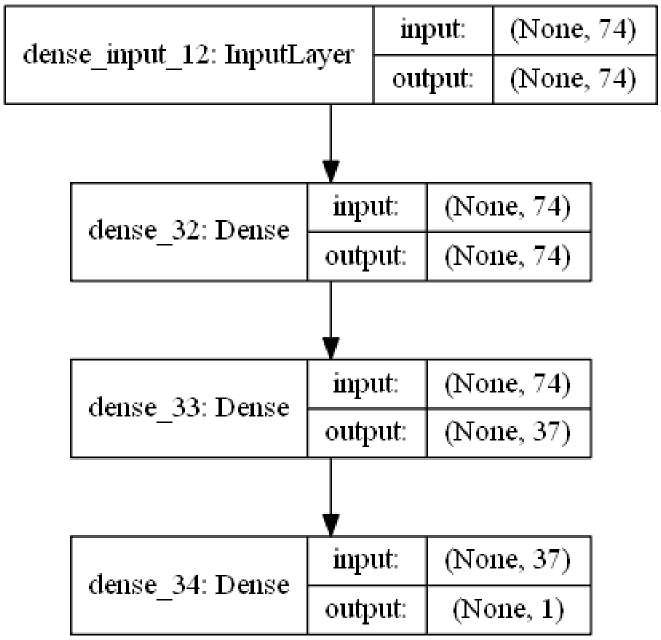


Figure 10: Large FNN network architecture for Cell2Cell dataset

4.1.1 Small Feedforward Neural Network (SFNN)

The small FNN consists of three layers. Their layer details are as follows as shown in Figure 9.

Layer1: Input

Layer 2: Dense

Input dimension: 74

Number of Neurons : 37

Initialization: Normal

Activation: Relu

Layer3: Dense

Number of Neurons : 1

Initialization: Normal

Activation: SigMoid

4.1.2 Large Feedforward Neural Network (LFNN)

The large FNN consists of three layers. Their layer details are as follows as shown in Figure 10.

Layer1: Input

Layer 2: Dense

Input dimension: 74

Number of Neurons : 74

Initialization: Normal

Activation: Relu

Layer 3: Dense

Input dimension: 74

Number of Neurons : 37

Initialization: Normal

Activation: Relu

Layer4: Dense

Number of Neurons : 1

Initialization: Normal

Activation: SigMoid

4.2 Convolutional Neural Network (CNN)

In addition to the above mentioned two FNN we created one convolutional neural network with below layer details which is shown in Figure 11.

Layer 1: Input

Layer 2: Embedding

Layer 3: Convolution1D

Layer 4: GlobalMaxPooling1D

Layer 5: Dense

Layer 7: Dropout

Layer 7: Dense

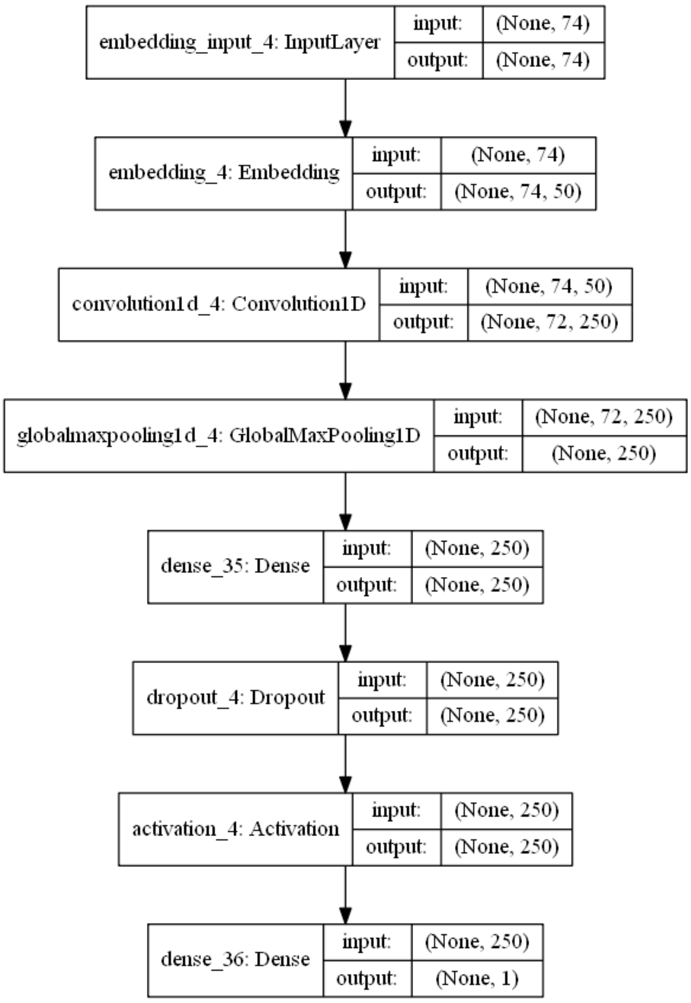


Figure 11: CNN architecture for the Cell2Cell dataset

5. DATASETS

We evaluated the proposed churn prediction methodology on two different real telecom datasets. Dataset details are given in Table 1.

5.1 Cell2Cell

We conducted the experiments on a large churn dataset from the Teradata Center for Customer Relationship Management of Duke University . This real dataset was collected from customers of Cell2Cell Telecom Company. Cell2Cell is one of the largest wireless companies in the USA with more than 10 million customers and its average monthly churn rate is 4%. For churn prediction, Cell2Cell collected several data about its customers including 1) customer care service details, 2) customer demography and personal details, 3) customer credit score, 4) bill and payment details, 5) customer usage pattern, and 6) customer value added services, totaling to 75 variables from 71, 047 customers. The dataset is divided into calibration and validation set. The calibration set (training data) contains 50% churners, 20,000 among 40,000 customers. Whereas, the validation set contain approximately 2% churners, 609 among 31,047 customers. For our experiments, we used the entire dataset in which 29% of customers are churn.

Table 1: Description of dataset used for our experiments

Dataset/Properties	Cell2Cell	CrowdAnalytix
Total no. of consumers	70,831	3,333
Total no. of variables	75	20
No. of non-churners	50,326	2,850
No. of churners	20,505	483
% of non-churners	71	86
% of churners	29	14
No. of features	74	18

5.2 CrowdAnalytix

This public dataset is provided by the CrowdAnalytix community as part of their churn prediction competition. The real name of the telecom company is anonymized. It contains 20 predictor variables mostly about customer usage patterns. There are 3333 records in this dataset, out of which 483 customers are churners and the remaining 2850 are non-churners. Thus the ratio of churners in this dataset is 14%.

6. EXPERIMENTAL SETUP AND RESULTS

In this section, we explain the experimental setup, implementation details and analyze the results.

6.1 Implementation

We implemented all the steps in the proposed churn prediction method in Python programming language. Python has several inbuilt libraries such as scikit-learn, pandas, numpy for various data mining tasks. Further, we implemented the entire workflow in IPython Notebook which runs in a browser for easy interaction. We used Keros library for developing proposed deep neural network architectures.

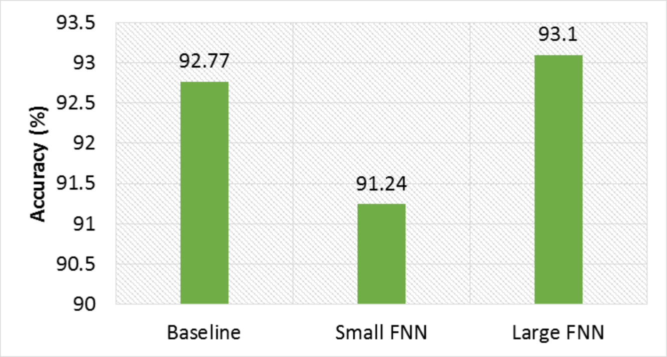


Figure 12: Churn prediction results for CrowdAnalytix data using three neural networks

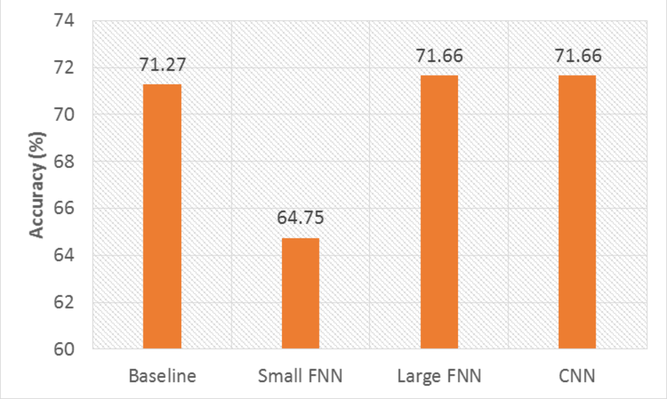


Figure 13: Results for the Cell2Cell dataset using baseline, small FNN, large FNN and CNN

6.2 Dataset Preparation

There are 216 customers in the Cell2Cell dataset found to have missing values for more than seven variables. So we removed those customers from the dataset. Further, there are some customers who had missing values for one or two variable. We used a forward filling method, propagating the previous valid value to the missing field, to those variables. Out of 75 predictor variables, we removed the 'customer service area' attribute which is irrelevant for churn prediction. We used the remaining 74 variables as the input to the proposed deep learning models. In summary, there are 70831 customer records each with 74 attributes, in which 20505 are churners (29%). In contrast with Cell2Cell dataset, CrowdAnalytix dataset is fairly clean with no missing values. Out of 20 predictor variables we excluded Area code and phone no, resulting to 18 predictor variables for each customer for constructing the churn model.

6.3 Model Construction and Cross Validation

After extracting the required features from the datasets, the next step is model construction. We model the churn prediction problem as a two-class classification problem. We trained the three networks for the both the datasets. For validating the performance of these classifiers, we use a stratified 10-fold cross-validation method over the given datasets. We chose stratified cross validation to avoid over fitting as both the datasets are unbalanced, e.g. unequal number of churner and non-churners. Stratified cross validation method ensures that the percentage of samples for each class is similar across folds (if not same).

6.4 Experimental Results

In this section, we present the experimental results using the presented deep neural network models and compare how they perform with respect to other models.

6.4.1 CrowdAnalytix Dataset

Figure 12 shows the results for the CrowdAnaytix dataset using a baseline neural network and two deep neural networks namely Small FNN and Large FNN. Using a baseline neural network, we achieved an accuracy of 92.77%. As we can see from this figure, large FNN improves the accuracy compared to the baseline model. But the small FNN accuracy is lower than the baseline model.

6.4.2. Cell2Cell

Figure 13 shows the results for the Cell2Cell dataset using a baseline neural network and three deep neural networks namely Small FNN, Large FNN and CNN. Using a baseline neural network, we achieved an accuracy of 71.27%. As we can see from this figure, large FNN improves the accuracy compared to the baseline model. But the small FNN accuracy is lower than the baseline model.

7. CONCLUSIONS

There are simple decision rules based models and complex classification models for churn prediction task has been proposed in the literature. While these methods are efficient in performing the churn prediction task, they require manual feature engineering process which time consuming and error-prone. In this paper, we presented a methodology of using deep-learning models to eliminate the manual feature engineering process. We created three deep neural network architectures for the churn prediction task. Experiments were conducted using two real world datasets CrowdAnalytix and Cell2Cell. Out experimental results show that deep learning models performing equally as good as traditional classifiers such as SVM and random forest.

ACKNOWLEDGEMENT

Authors would like thank Teradata Center for Customer Relationship Management (CCRM) of Duke University and CrowdAnalytix community for providing the churn prediction competition dataset for our research.

REFERENCES

[1] Liao, Shu-Hsien, Pei-Hui Chu, and Pei-Yuan Hsiao. "Data mining techniques and applications–A decade review from 2000 to 2011." Expert Systems with Applications 39, no. 12 (2012): 11303-11311.

[2] Kamalraj, N., and A. Malathi. "A survey on churn prediction techniques in communication sector." International Journal of Computer Applications 64, no. 5 (2013).

[3] N.Hashmi, N.ButtandM.Iqbal. Customer Churn Prediction in Telecommunication A Decade Review and Classification. International Journal of Com-puter Science Vol.10(5),2013

[4] V. Umayaparvathi, K. Iyakutti, " Applications of Data Mining Techniques in Telecom Churn Prediction", International Journal of Computer Applications, Vol. 42, No.20, 2012

[5] V. Umayaparvathi, K. Iyakutti,, "Attribute Selection and Customer Churn Prediction in Telecom Industry", Proceedings of the IEEE International Conference On Data Mining and Advanced Computing, 2016 (to be appeared).

[6] Huang, Bingquan, Mohand Tahar Kechadi, and Brian Buckley. "Customer churn prediction in telecommunications." Expert Systems with Applications 39, no. 1 (2012): 1414-1425

[7] Shaaban, Essam, Yehia Helmy, Ayman Khedr, and Mona Nasr. "A proposed churn prediction model." IJERA 2 (2012): 693-697.

[8] Wangperawong, Artit, Cyrille Brun, Olav Laudy, and Rujikorn Pavasuthipaisit. "Churn analysis using deep convolutional neural networks and autoencoders." arXiv preprint arXiv:1604.05377 (2016).

[9] Castanedo, Federico, Gabriel Valverde, Jaime Zaratiegui, and Alfonso Vazquez. "Using deep learning to predict customer churn in a mobile telecommunication network." (2014).