Empirical Evaluation of Telecommunication Customer Churn Prediction Models

Curtis Stewart, Haoming Yuan, Jonathan Rasmussen
SYDE 675 – Group 11, Option B
Systems Design Engineering Department
University of Waterloo, Canada
{cvstewart, h34yuan, jrasmussen}@uwaterloo.ca

***Abstract*—As the Canadian population continues to grow, there is increased pressure on telecommunication companies to reach more customers while maintaining their current customer base. As new technologies such as 5G are introduced, companies are finding it more challenging to retain their previous customers. As the cost of retaining current customers is fifty times less than acquiring new customers, it is imperative that telecommunication companies seek accurate prediction methods to detect when a customer may churn from their services. This paper conducts an empirical evaluation of five state of the art models and their application to customer churn prediction in the telecommunication industry. The five models, logistic regression, random forest, support vector machines, XGBoost, and artificial neural networks, are applied to two different datasets, namely the Cell2Cell and the IBM telecom datasets. Specific classification metrics and sampling methods were determined in order to maximise the efficacy of the models. The models were trained and tested on the two datasets with no sampling and with a 1:1 undersampling. The results of this analysis show that artificial neural networks are most effective for customer churn prediction when there is no sampling, based on the F1 score. When there is sampling involved, the more effective choice is a random forest model. Overall, no model could be chosen as the clear winner for customer churn as it is heavily dependent on the specific dataset, the sampling method chosen, and the metric of interest.**

***Index Terms*—Churn, Random Forest, ANN, XGBoost**

I. Introduction

During the Covid-19 pandemic, the everyday life of the average Canadian citizen went online. The Canadian populace became even more reliant on their telecommunication providers then ever before. The Canadian telecommunication industry is vast, with an estimated 98.9% of all Canadians having telecom coverage at any given time. This industry brought in an estimated $27.1 billion in revenue with mobile service subscribers alone in 2019. This market value should continue to increase as the Canadian population continues to grow 0.9% annually. Evidently there is huge growth and revenue potential in the telecommunications industry. In fact, over 90.7% of this market is shared by three service providers. These are the companies known as Bell Canada, Rogers Communications, and the Telus Corporation. These three companies are the apex of the industry and provides telecommunication services to the vast majority of Canadians. [1]

Although these companies vie for each others' customers and market shares, the services they provide are more or less interchangeable. As such, the telecommunications industry is prevalent to a business problem known as churn. Customer churn is a phenomenon where businesses lose a percentage of their customer base to competitors. This is particularly prevalent in the financial, utilities, and telecommunication industries where the services provided are interchangeable. The top three companies mentioned above experiences on average a monthly churn rate of 1.44%. A company can employ two strategies to combat customer churn loss. They could either acquire their competitors' customers to retain their bottom line or they could prevent their own customers from leaving to their competitors. Both Bell and Telus reported that their approach using a churn prevention strategy costs them on average fifty times less then their new customer acquisition strategies. This report shows that an optimal strategy for telecommunication companies is to minimize the churn rates of their own customers while also maximizing profitability by seeking this type of strategy. [1]

As the telecommunications industry brings cutting edge technologies such as fiber optic and 5G networks to the public, it is paramount for them to retain their customers while entering these new markets. Extensive research has been conducted in developing customer churn prediction models in the telecommunications industry. With accurate detection of customers with a high propensity to attrite, companies can target the customers most likely to churn and minimize the limited resources for this endeavour. [2] This report seeks to evaluate several state of the art machine learning algorithms and their effectiveness to this customer churn problem. The background of the five state-of-the-art algorithms, Logistic Regression, Random Forest, Support Vector Machines, XGBoost, and Artificial Neural Networks, will be discussed and literature on their applications to this problem will be explored. The background of several sampling strategies that were investigated will also be outlined. The methodology will describe the datasets used, namely the Cell2Cell and the IBM telecom datasets, and the subsequent preprocessing required. Here the results metrics, hyper-parameter tuning and sampling strategies will be provided. The results and empirical evaluation of the five models will be conducted on the two datasets. The performances will be compared and commented upon. Finally, the ideal technique will be identified and conclusions will be made upon this method as a solution to the churn prediction problem.

II. Background

A. Logistic Regression

Logistic regression (LR) is a supervised learning algorithm used for solving binary classification problem. It is based on the sigmoid function where the function produces a S shaped curved with the output values being between 0 and 1. The output from the sigmoid function represents the estimated probability of the input belonging to one class which is then assigned to a class by comparing with a threshold value. [3]

B. Random Forest

Random forest (RF) is an ensemble method used for classification or regression problems. The strategy of random forests is to grow a set of decision trees each from a random subset of the features and the trees are trained on a bootstrap sample of the dataset. The random forest will then make a prediction based on the aggregation of the predictions from all the trees. Random forest does not always work well for problems with extremely unbalanced datasets such as customer churn predictions [4]. To handle the class imbalances, two methods were proposed by Chen et al. (2004). The first technique is the balanced random forest (BRF) where a sampling technique is applied on the dataset to artificially make the minority and majority class priors equal before model training. The second technique is the weighted random forest (WRF) where weights are assigned to each class. The minority class is given a larger weight in order to penalize misclassification of the minority class more heavily. From the study by Chen et al. (2004), it was found that both BRF and WRF performed better than other techniques. However, there was no clear winner between BRF and WRF. [5]

C. SVM

Support vector machines (SVM) is considered one of the most widely used clustering algorithms in industrial applications and as such was chosen as one of the potential techniques. SVMs were originally proposed for linearly separable binary classification problems. However they can be adapted for multi-class and non-linear problems as well. For multiclass problems the one-vs-one or one-vs-rest schemes can be used to separate the different classes in the feature space. This feature space can be adapted by utilizing a kernel function to operate in a high dimensional implicit feature space, essentially allowing SVM to be applied to none linearly separable problems. The final parameter that could be investigated is the so called soft margin parameter. This parameter softens the separation between the classes and lessens the sensitive of the model to support vectors, improving the bias-variance trade-off. The performance of this model will be dependent on the separation of the customer classes and effective utilization of kernels. [6], [7]

D. XGBoost

Another technique of interest that will be investigated is the XGBoost technique. This technique is an implementation of gradient boosted decision trees. Gradient boosting is a technique where new regression tree models are created that predict the residuals on prior models [6]. The final prediction of this forest of regression trees can be averaged together to correct the errors of the individual trees, providing a more accurate prediction. It also introduces a "novel sparsity aware algorithm for handling sparse data and a theoretically justified weighted quantile sketch for approximate learning". [8] A key aspect of XGBoost is the construction of the forest in parallel, resulting in far faster training times and improved results. The XGBoost model is specifically designed for fast execution speeds and high model performance. It is currently the state-of-the-art for many data competitions, particularly for ones using structured data and as such is expected to perform well for this structured data problem. [9]–[11]

E. ANN and Hybrid Neural Networks

Tsai and Lu (2009) [12] proposed using hybrid artificial neural networks (ANN) for churn prediction. Hybrid models use two neural networks, where the first network is trained and tested on the training data and the incorrectly predicted training samples are then removed to produce a reduced training set, which is then used to train a second network for final prediction. This approach aims to remove outliers in the training. In [12], two hybrid models were tested, namely ANN + ANN, and self organizing map (SOM) + ANN. Only the ANN + ANN hybrid model is tested in this paper. The structure and parameters of both ANNs used in the hybrid ANN + ANN model were kept the same.

The ANN is a multilayer perceptron (MLP) which consists of a hidden layer of a set number of nodes, followed by a ReLU activation [13], and a final output layer consisting of a single node with sigmoid activation. The ANN is trained via backpropagation [14]. ANNs have many hyperparameters to tune, such as the optimization scheme and its parameters, number of epochs, batch size, and final threshold on the sigmoid output. In our implementation, stochastic gradient descent (SGD) with momentum is used for optimization. The sigmoid activation produces output in range [0,1] and a threshold is required to produce the final binary classification. This threshold is typically set to 0.5, however for highly imbalanced datasets, this can result in overwhelming prediction of the majority class. For imbalanced datasets, adjusting the sigmoid threshold to favour the minority class can result in better overall performance with class prediction ratios more similar to the class ratio of the dataset. In [12], networks of 8, 12, 16, 24, and 32 nodes in the hidden layer were tested, with the 32 node structure performing the best overall. In our testing, 32 nodes in the hidden layer was chosen and the other hyperparameters were tuned via cross-validation.

F. Sampling

In the presence of imbalanced data, the algorithms may present a bias towards the majority class as minimizing the predictions of the minority class will result in lower errors. To solve the class imbalance, data-level solutions are used to re-balance the training class distribution. Only four methods from Zhu et al. (2017) will be discussed in this paper. Random undersampling (RUS) creates a subset of the dataset by randomly removing majority class samples from the dataset until equal class distribution. Random oversampling (ROS) creates a superset of the dataset by randomly sampling from the minority class with replacement until equal class distribution. Synthetic minority oversampling technique (SMOTE) creates synthetic elements for the minority class based on existing minority elements until equal class distribution. SMOTE randomly picks points from the minority class and finds the k-nearest minority class neighbours to the selected point. The synthetic points are then generated through interpolation within neighbours and added to the dataset. Borderline-SMOTE (BSMOTE) is a modified version of SMOTE that only generates synthetic samples from minority points that are considered borderline (neighbours consists of points in the minority and majority class). These points on the border are more likely to be misclassified thus are considered more important for classification. [15]

III. Method

A. Datasets

For the purpose of this experiment, the purposed methodologies were evaluated on two different datasets to ensure the model performances were not biased towards any specific dataset.

1) Cell2Cell: The first dataset used was the Cell2Cell dataset developed by the Teradata Center for Customer Relationship Management at Duke University in 2003. The Cell2Cell dataset was retrieved from Kaggle where it consists of 71,047 unique rows of data and 71 numerical features representing the attributes of one customer between July 2001 to January 2002. The dataset consists of information about the customer which include service details, usage, personal information, demography, payment details and credit score. The churn attribute in the dataset classifies if the customer has left the service where 28% of customers in the dataset are churners. [16]

2) IBM: The second dataset is the IBM Telco Customer Churn dataset which was retrieved from Kaggle. The dataset was developed by IBM as a sample dataset for a fictional telecommunication company that provided Internet and home phone services to customers in California in Q3. The IBM dataset consists of 7043 rows of data where each row represents one customer. The dataset contains 20 features comprised of categorical and numerical features. The feature attributes describe the services that the customer has signed up for, the customers account information, and demographic information. The churn attribute in the dataset classifies if the customer has left the service where 26% of customers in the dataset are churners. [17]

B. Data Preprocessing

Before proceeding with the model construction, data preprocessing steps are performed. The first step was to drop columns that did not provide any useful information such as the customer ID. The next step was to handle missing values in the data where customers with 7 or more missing values were removed from the datasets. For the remaining customers with missing values, an imputation method was used to fill the missing values. Of the 3051 missing values in the Cell2Cell dataset, a 0-fill imputation method was used as 0 was the mode for the columns with missing values. Of the 9 missing values in the IBM dataset, a median-fill imputation was used since the missing values belonged to the same column. For the categorical features in the IBM dataset, categorical textual features were converted to numerical features with One Hot Encoding. A min-max scaler was applied to both datasets to standardize the data to be between 0 and 1.

C. Sampling Strategy

To find the optimal sampling strategy, the sampling methods RUS, ROS, SMOTE and BSMOTE were used to resample the dataset with a class ratio of 3:7, 2:3 and 1:1. For the least balanced ratio, 3:7 was chosen over the 1:3 ratio used in Zhu et al. (2017) as 3:7 was the smallest possible ratio from the chosen datasets [15]. Each sampling strategy was trained and tested on a random forest model with an 80:20 train test split. The sampling strategy with the best overall metrics was used for subsequent analysis and compared with the unsampled results.

D. Metrics

The following five metrics were used to analyze the performance of each churn prediction model.

1) ROC AUC: Receiver operating characteristic curve plots the True-Positive (TP) rates vs. the False-Positive (FP) rates. The area under the ROC curve (AUC) simplifies the curve into a single number that explains how well the model is capable at distinguishing between classes. An AUC score of 0.5 indicates no class separability while 1 indicates complete separability.
2) Accuracy: The total number of correct predicted samples over the total number of samples defined as:

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)}$$

where FP and FN are False-Positive and False-Negative, respectively. Accuracy may not provide the most information in the presence of imbalanced data as it does not describe the correct class predictions.
3) Precision: The number of relevant samples within the retrieved samples.

$$Precision = \frac{TP}{(TP + FP)}$$

4) Recall: The number of relevant samples retrieved over the total desired samples.

$$Recall = \frac{TP}{(TP + FN)}$$

5) F1-score: Measures the accuracy of a model by taking the harmonic mean of Precision and Recall where it provides a better measurement for incorrect classifications compared to Accuracy. F1-score is defined as:

$$F_1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

E. Model Construction and Cross-Validation

The ANN models were constructed using PyTorch and trained and tested on Google Colab for GPU access. Most other models were tested using their scikit-learn implementation, except for XGBoost which has its own python library. The sampling methods in imbalanced-learn were utilized for testing the various sampling strategies. To construct high performing churn prediction models, hyperparameter tuning is required to find the optimal parameters for each model. A grid search cross-validation technique is used to perform an exhaustive search over a set of hyperparameters for each algorithm. Each model constructed from the grid search is validated through a 5-fold stratified cross-validation (same class distribution across each fold) and the parameters of the best model was retrieved. With the completion of hyperparameter tuning, a 5-fold stratified cross-validation is applied to train and test each model with its optimal parameters. For each model, cross-validation is applied to the unsampled and randomly undersampled Cell2Cell and IBM datasets where the average result metrics over the 5 folds are reported.

IV. Results & Analysis

For the analysis of results, the metric determined to be most valuable for detecting customer churn was the F1 score. This is due to it being derived from both the precision and recall metric. The second most valuable metric considered was the AUC as it measures the trade-off of true positive and false positive rates. These two metrics were considered when appraising the results from both datasets as they provide more insight on the ability of the models to correctly identify churners. It was noted that in the results of the two datasets, the IBM dataset achieved higher metric values then that of the Cell2Cell dataset. This is attributed to the difference in data quality between the datasets. The resulting metric values were similar to that of the values found in the literature referenced.

A. Literature Results for Comparison

See Table I for sample results from literature on the Cell2Cell and IBM datasets for comparison with our testing results shown later. Note that the Recall and F1 shown in Table I are different than those reported in [18] as the paper reported the Negative Predictive Value (TN / (TN + FN)) as the Recall, whereas we are concerned with the Recall as Sensitivity (TP / (TP + FN)). As such, the Recall and F1 score were recalculated from the paper using their confusion matrix on our definition.

TABLE I
Sample results on chosen datasets from literature

Dataset Model Source	Cell2Cell Gradient Boost [18]	Cell2Cell Random Forest [19]	IBM Random Forest [20]
Accuracy	0.72	-	0.79
AUC	-	0.592	0.83
Precision	0.623	-	0.65
Recall	0.0678	-	0.49
F1	0.122	-	0.56

B. Sampling Strategy

From the results of the sampling strategies applied on to the Cell2Cell dataset and scored with a RF model shown in Table II, the RUS technique generally had the best scores across the five metrics. The RUS (1:1) sampling strategy had the best F1 score at 0.492 and the worst Accuracy at 0.607. RUS (1:1) was chosen as the sampling strategy to compare with no sampling in the empirical evaluation on the Cell2Cell dataset. RUS (1:1) was chosen due to it having the highest F1 score while having comparable AUC scores with other sampling strategies. Even though RUS (1:1) had the lowest Accuracy score, the importance of the high F1 score outweighs it as the F1 score better explains the class predictions. Similar analysis of the sampling strategies was applied to the IBM dataset where the RUS (1:1) sampling strategy was chosen in the same manner.

TABLE II
Cell2Cell sampling results using a baseline random forest

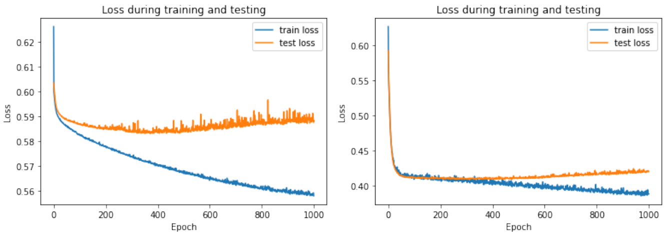
Sampling	Ratio	Accuracy	AUC	Precision	Recall	**F1**
RUS	(3:7)	**0.715**	0.672	0.576	0.086	0.149
	(2:3)	0.700	**0.679**	0.479	0.317	0.381
	(1:1)	0.607	0.671	0.395	**0.658**	**0.492**
ROS	(3:7)	0.714	0.673	0.577	0.077	0.137
	(2:3)	0.714	0.679	0.543	0.116	0.191
	(1:1)	0.712	0.674	0.517	0.163	0.247
SMOTE	(3:7)	**0.715**	0.672	**0.586**	0.079	0.139
	(2:3)	0.711	0.671	0.522	0.125	0.201
	(1:1)	0.704	0.665	0.482	0.182	0.264
BSMOTE	(3:7)	**0.715**	0.672	0.582	0.081	0.142
	(2:3)	0.711	0.671	0.521	0.122	0.198
	(1:1)	0.700	0.663	0.464	0.173	0.252

C. ANN Hyperparameters

When hyperparameter tuning the ANN for each dataset, the training and validation loss were plotted during training as shown in Fig 1. From this plot, it can be seen that training the ANNs for too long can result in overfitting on the training data and the test loss starts to increase. From testing, it was found that 300 epochs was sufficient for the Cell2Cell dataset, and 200 epochs was sufficient for the IBM dataset. A learning rate of 0.02 and momentum of 0.9 was chosen for the Cell2Cell dataset, and a learning rate of 0.01 and momentum of 0.5 was chosen for the IBM dataset. Batch sizes of 1000 and 100 were chosen for the Cell2Cell and IBM datasets respectively.

D. Cell2Cell Results

In the Cell2Cell without Sampling models, the highest performance was the Hybrid ANN with a F1 score of 0.463,



(a) Cell2Cell Dataset

(b) IBM Dataset

Fig. 1
Example ANN loss during training

as show in Table III. This and ANN, at a value of F1 score 0.458, out performed the other models. However, it is noted that the model with the highest AUC was XGBoost at a value of 0.677. Although this was the highest, the Hybrid ANN also had a relatively high AUC at 0.620. Therefore with this consideration, the Hybrid ANN appeared to perform the best for the Cell2Cell data without any sampling method.

TABLE III
Cell2Cell Results without Sampling

	XGB	SVM	LR	RF	ANN	Hybrid ANN
Accuracy	**0.720**	0.711	0.711	0.718	0.577	0.552
AUC	**0.677**	0.527	0.615	0.673	0.625	0.620
Precision	0.556	0.519	0.512	**0.611**	0.365	0.355
Recall	0.161	0.016	0.032	0.074	0.620	**0.668**
F1	0.250	0.032	0.059	0.132	0.458	**0.463**

When the models were trained on the Cell2Cell dataset with undersampling applied, the model with the highest F1 score was the RF model with a score of 0.4914, as seen in Table IV. This was slightly better then that of XGBoost with a F1 score of 0.4910. These two models were also very similar in terms of AUC with values of 0.671 and 0.672, respectively. This similarity is expected as both of these models were developed from decision trees. However as the F1 score was identified as the optimal metric, the RF model was considered the better model in this particular case.

TABLE IV
Cell2Cell Results with Undersampling

	XGB	SVM	LR	RF	ANN	Hybrid ANN
Accuracy	**0.617**	0.578	0.586	0.611	0.591	0.560
AUC	**0.672**	0.615	0.619	0.671	0.614	0.612
Precision	**0.399**	0.361	0.365	0.395	0.365	0.353
Recall	0.638	0.595	0.583	**0.649**	0.556	0.617
F1	0.491	0.450	0.449	**0.491**	0.440	0.447

E. IBM Results

For the IBM dataset, the results of the same models showed improvement compared to the cell2cell models. This is attributed to the data quality of the IBM dataset being vastly better then that of the cell2cell. When the models were trained without sampling on the IBM dataset, the model with the highest F1 score was the ANN at a value of 0.630, as shown in Table V. However, the highest AUC value achieved was from the RF model with a score of 0.847. The ANN achieved an AUC of 0.844, which was only slightly worse.

TABLE V
IBM Results without Sampling

	XGB	SVM	LR	RF	ANN	Hybrid ANN
Accuracy	0.791	0.799	**0.804**	0.801	0.750	0.745
AUC	0.828	0.834	0.845	**0.847**	0.844	0.840
Precision	0.630	0.645	0.655	**0.675**	0.518	0.511
Recall	0.515	0.536	0.545	0.479	0.805	**0.806**
F1	0.566	0.585	0.595	0.560	**0.630**	0.626

The models were then trained on the IBM dataset with undersampling, as described in Table VI. Similar to the Cell2Cell with undersampling results, the RF model had the highest F1 score at 0.635. It also had the best AUC score at 0.847. Overall, the RF had the highest metrics values in all except Recall where it was slightly outperformed by SVM.

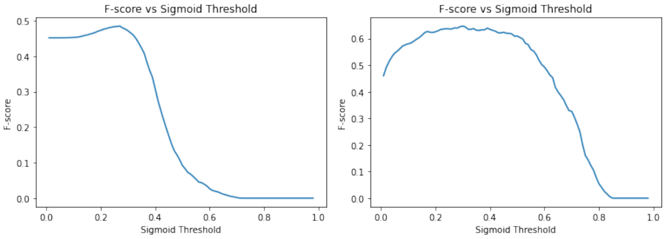
TABLE VI
IBM Results with Undersampling

	XGB	SVM	LR	RF	ANN	Hybrid ANN
Accuracy	0.730	0.700	0.750	**0.757**	0.749	0.747
AUC	0.822	0.830	0.845	**0.847**	0.842	0.840
Precision	0.493	0.463	0.518	**0.527**	0.516	0.515
Recall	0.759	**0.834**	0.800	0.799	0.794	0.794
F1	0.598	0.595	0.628	**0.635**	0.626	0.625

F. Overall Trends

Analyzing the results shows two interesting phenomenons. Without sampling, most methods rarely predicted churn resulting in low churn recall and F1-score. However, the ANN variant models showed the highest F1 score performance, which was high even without sampling. The Hybrid ANN had the highest F1 for the Cell2Cell dataset and the regular ANN showed the best F1 score for the IBM dataset. Due to the poor data quality in the Cell2Cell dataset, the Hybrid ANN network benefited from the sampling to remove outliers. However in the IBM case, the data did not need the Hybrid ANN to sample the data and as such the regular ANN structure outperformed the Hybrid ANN. The significant increase in F1 score between the ANN models and other models like SVM without sampling can be attributed to the chosen sigmoid threshold on the final ANN output. Without sampling, the sigmoid threshold can be adjusted to allow for increased prediction of the minority class, thus raising the recall of churn and the overall F1-Score. As can be seen in Fig 2, adjusting the threshold on the sigmoid output allows for optimization of the ratio of class predictions, thus allowing for maximization of the F1-score. Sigmoid thresholds of 0.29 and 0.26 were chosen for the Cell2Cell and IBM datasets respectively without sampling. When using undersampling, the sigmoid threshold was set to 0.5 for both datasets as the training data is no longer imbalanced.

The other phenomenon that was noted was the effectiveness of the RF model for both datasets. This was attributed to the decision tree style of logic used for this model that



(a) Cell2Cell Dataset

(b) IBM Dataset

Fig. 2
F-score vs sigmoid threshold without sampling

benefited from the undersampling processing. Note that the XGBoost model achieved a very similar F1 score as RF on the Cell2Cell with undersampling, but achieved higher in some of the other metrics. This is also evidence that the undersampling processing improves the performance of tree based models.

V. Conclusions

Churn prediction is an important approach for telecommunication companies as it is used to maximize profitability by minimizing the loss of existing customers to competitors. This paper investigated different algorithms and sampling strategies in finding the best modelling technique for predicting customer churn in the telecommunication industry. The five algorithms investigated are Logistic Regression, Random Forest, SVM, XGBoost, and ANNs. Models were trained and tested on the Cell2Cell and IBM datasets with no sampling technique and with the Random Undersampler with 1:1 class ratio technique. On both datasets with no sampling strategy applied, the Hybrid ANN had the best performance based on the F1-score at 0.463 for the Cell2Cell and 0.625 for the IBM dataset. Random Forest had the best performance on the datasets with the RUS (1:1) strategy with the F1-score at 0.491 for the Cell2Cell and 0.635 for the IBM dataset. From the analysis of the results, it can be concluded that there is no clear winner for the best model at predicting churn as it depends on the metrics and different models performed better in different situations. The performance of the models is highly dependent on the selected dataset, sampling technique and metric for evaluation. Thus, there was not a modelling technique analyzed in this paper that was good enough to fully solve the customer churn problem.

Note that in this paper, only one ANN structure with 32 hidden nodes was tested on. In the future, additional investigation on larger and more complex network structures with different techniques could be done to further improve the performance of the ANN models. Furthermore, the features from the Cell2Cell dataset did not provide enough relevant information due to outliers and missing data which resulted in poor churn prediction results. The IBM dataset contained more useful information than the Cell2Cell dataset, but it did not contain enough samples to simulate the scale of a telecommunication company. Therefore, larger and higher quality datasets are recommended for future analysis.

References

[1] Canadian Radio-television Government of Canada and Telecommunications Commission (CRTC). Communications monitoring report 2019, Feb 2020.

[2] Saad Ahmed Qureshi, Ammar Saleem Rehman, Ali Mustafa Qamar, Aatif Kamal, and Ahsan Rehman. Telecommunication subscribers' churn prediction model using machine learning. In *8th International Conference on Digital Information Management, ICDIM 2013*, pages 131–136, 2013.

[3] Bingquan Huang, Mohand Tahar Kechadi, and Brian Buckley. Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1):1414–1425, jan 2012.

[4] Yaya Xie, Xiu Li, E. W.T. Ngai, and Weiyun Ying. Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36(3 PART 1):5445–5449, apr 2009.

[5] Chao Chen and Leo Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 01 2004.

[6] Guo En Xia and Wei Dong Jin. Model of customer churn prediction on support vector machine. *Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice*, 28(1):71–77, jan 2008.

[7] Yu Zhao, Bing Li, Xiu Li, Wenhuang Liu, and Shouju Ren. Customer churn prediction using improved one-class support vector machine. In Xue Li, Shuliang Wang, and Zhao Yang Dong, editors, *Advanced Data Mining and Applications*, pages 300–306, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-Augu, pages 785–794, New York, NY, USA, aug 2016. Association for Computing Machinery.

[9] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, and K.Ch. Chatzisavvas. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55:1–9, 2015.

[10] Abdelrahim Kasem Ahmad, Assef Jafar, and Kadan Aljoumaa. Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6(1):1–24, dec 2019.

[11] J. Pamina, J. Beschi Raja, S. Sathya Bama, S. Soundarya, M. S. Sruthi, S. Kiruthika, V. J. Aiswaryadevi, and G. Priyanka. An effective classifier for predicting churn in telecommunication. *Journal of Advanced Research in Dynamical and Control Systems*, 11(1 Special Issue):221–229, jun 2019.

[12] Chih Fong Tsai and Yu Hsin Lu. Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, 36(10):12547–12553, dec 2009.

[13] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[14] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[15] Bing Zhu, Bart Baesens, Aimée Backiel, and Seppe K L M vanden Broucke. Benchmarking sampling techniques for imbalance learning in churn prediction. *Journal of the Operational Research Society*, 69(1):49–65, 2018.

[16] Pamina. telecom churn- new cell2cell dataset. https://www.kaggle.com/jpacse/telecom-churn-new-cell2cell-dataset, Jun 2019.

[17] BlastChar. Telco customer churn. https://www.kaggle.com/blastchar/telco-customer-churn, Feb 2018.

[18] V Umayaparvathi and K Iyakutti. Attribute selection and Customer Churn Prediction in telecom industry. In *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pages 84–90, mar 2016.

[19] Adnan Idris, Asifullah Khan, and Yeon Soo Lee. Intelligent churn prediction in telecom: employing mRMR feature selection and RotBoost based ensemble classification. *Applied Intelligence*, 39(3):659–672, 2013.

[20] J Pamina, J Beschi Raja, S Sam Peter, S Soundarya, S Sathya Bama, and M S Sruthi. Inferring Machine Learning Based Parameter Estimation for Telecom Churn Prediction. In S Smys, João Manuel R S Tavares, Valentina Emilia Balas, and Abdullah M Iliyasu, editors, *Computational Vision and Bio-Inspired Computing*, pages 257–267, Cham, 2020. Springer International Publishing.