2024 Fifth International Conference on Intelligent Data Science Technologies and Applications (IDSTA)

Predicting B2B Customer Churn using a Time Series Approach

1st Jim Ahlstrand
Telenor Sverige AB
Karlskrona, Sweden
jim.ahlstrand@telenor.se

2nd Martin Boldt
Department of Computer Science
Blekinge Institute of Technology
Karlskrona, Sweden
martin.boldt@bth.se

3rd Anton Borg
Department of Computer Science
Blekinge Institute of Technology
Karlskrona, Sweden
anton.borg@bth.se

4th Håkan Grahn
Department of Computer Science
Blekinge Institute of Technology
Karlskrona, Sweden
hakan.grahn@bth.se

***Abstract*—Preventing customer churn, i.e., termination of business commitments, is essential for companies operating in saturated markets, especially for subscription-based models such as telecommunication. Knowing when customers decide to terminate services is instrumental to effective churn prevention. In this study, we investigate how churn prediction performs in practice when training models on different time intervals of historic data (1-4 weeks back) and predicting churn at different numbers of weeks ahead (1-4 weeks). We use a real-world, time-series dataset of mobile subscription usage to examine churn prediction for business-to-business (B2B) customers. We utilize the time-series data at a higher temporal resolution than prior studies and investigate different forecasting horizons. Leveraging popular machine learning algorithms such as Random Forests, Gradient Boosting, Neural Networks, and Gated Recurrent Unit, we show that the best model achieves an average F_1-score of 79.3% for one-week ahead predictions. However, the average F_1-score decreases to 63.3% and 61.8% for two and four weeks ahead, respectively. A model interpretation framework (SHAP) evaluates the feature impact on the models' internal decision logic. We also discuss the challenges in applying churn prediction for the B2B segment.**

***Index Terms*—Customer churn prediction, Telecom B2B customers, Machine learning, Time-series data**

I. Introduction

Customer churn is a discontinued relationship between a customer and a business. Churn poses a significant challenge to maintaining sustainable growth in highly saturated and competitive markets like telecommunications [1]. However, churn prediction is challenging, as various factors, such as satisfaction, loyalty, preferences, and external events can influence customer behavior. In addition, customer churn is also slightly different depending on the industry and segment. The business-to-consumer (B2C) segment relies on high volumes compared to business-to-business (B2B), where a single customer can have a significant impact on revenue [2]. This paper focuses on mobile subscriptions in the B2B segment at a telecommunication company in Sweden.

Customer behavior is not static but dynamic and evolving, i.e., traditional segmentation methods without the temporal component would have problems modeling customer behavior [3]. Time series data can use temporal resolutions of individual days (which tends to be a too high resolution with low variance for telecom B2B customers), at individual weeks (which our data indicates is a suitable resolution), or at individual months (which seems to be a too low resolution [4]).

Modeling customers as time series enables businesses to learn more about their customers' behavior [5], identify behavior changes, and respond to evolving customer needs. Further, a higher temporal granularity in churn predictions can help businesses improve retention efforts and market agility [6].

Previous work in churn prediction has used various personal features (e.g., demographics, psychographics, and geographic data) to segment customers with increased churn probability [7]–[9]. Such features do not apply to the B2B segment since they relate to individuals. However, features such as contract status, call detail records (CDR), Frequency-Recency-Monetary (FRM), and customer lifetime value (CLV) apply to both B2C and B2B.

Churn prediction efforts have mainly focused on the business-to-consumer segment, indicating a gap in understanding churn predictors within B2B [10], [11]. Thus, we model B2B customers' temporal features using weekly aggregated data in this study. Next, different machine learning models are trained to predict B2B customers' churn based on changes in behavior. Since building individual models for each customer is costly and may disregard similarities between customers. We aim to learn from all customers so the models can generalize and identify common churn signals.

This study investigates how different temporal ranges (1-4 weeks) and forecasting horizons (1-4 weeks) affect the model performance in predicting B2B customer churn. Time series data of the customers' usage can reveal more timely signals of customer churn (e.g., sudden drops or spikes in activity or deviations from normal patterns) than data without the time

This work was partly funded by Telenor Sverige AB.

component.

The main contributions of this study are: (i) a thorough novel analysis of B2B customer churn using real-world time series data with a higher temporal resolution than prior studies, and (ii) a comparative evaluation of different ML models performed on various lengths of training history and prediction horizons.

II. Related Work

Previous works on churn prediction have mainly focused on using static features. However, these features may not fully capture the temporal dynamics of customer behavior, such as changes in usage patterns, frequency, or intensity [4], [8]. Moreover, most existing methods use coarse-grained time intervals, which may lose important information and the ability to apply timely and proactive countermeasures. One dataset on Kaggle [12] is used extensively in related works on customer churn. However, the data are based on a sample from an IBM dataset of a fictional telecommunication company and contain only static features, i.e. they do not change over time. This results in a customer segmentation problem that classifies segments with higher churn propensity. Our study instead uses real-world time series-based data and shows the benefits of dynamic data when identifying churning customer behavior.

It is common in the churn literature to aggregate data into monthly buckets, which means that changes in customer behavior within individual months are lost, as described by Alboukaey et al. [4]. The authors demonstrate how LSTM can improve telecommunication churn modeling by increasing the granularity from monthly to daily. However, they focus on the B2C segment, which is of limited relevance when considering B2B customers. The authors also do not investigate different forecasting horizons, focusing solely on churn during the next 30 days.

Mena et al. [3] evaluate the efficacy of an LSTM model against a non-sequential regularized Logistic Regression model for churn prediction using aggregated monthly time series data. The results indicate that the LSTM model, which can directly process sequential data, outperforms the Logistic Regression models' accuracy. Additionally, incorporating LSTM-derived probabilities as features into the Logistic Regression models further improves the overall performance up to an AUC of 0.78.

Jain et al. [13] investigate customer churn prediction within three domains (banking, telecom, and ICT) using the following four non-sequential learning algorithms: Logistic Regression, Random Forest, SVM, and XGBoost. In the telecom domain, the XGBoost model showed the best prediction accuracy at 82.9%, although the results were not statistically significant.

Tamaddoni Jahromi et al. [14] discuss several B2B-specific constraints on churn prevention campaigns. Their extensive study covers many topics relevant to this work, including individual customer lifetime value, campaign cost, and how potential B2B churners should be targeted. However, their churn prediction classifiers are based on transactional behavior without time-series data. Studies using time series data are scarce but not absent. Zhao et al. [15] used LSTM to analyze customer behaviors in the railway freight industry to determine the probability of stable, loss-prone, and lost customers. Although limited to a transactional-based business model.

In summary, prior research on customer churn in the B2B segment has received little attention. Furthermore, previous research on churn prediction using time series multivariate data to forecast churn is limited. Many similar studies have minimal coverage of temporal variables, focusing almost solely on static features such as demographics, which appear to have little influence on behavior. As a result, this work aims to fill this research gap by assessing the efficacy of machine learning models trained on multivariate time series data to forecast churn risk.

III. Method

A. Data

The weekly customer snapshots are first anonymized. The weekly aggregation was calculated using the sum, or the average, over seven days from Monday to Monday. Each week is numbered from 0 to 51. Week zero starts on the first Monday of the year. The selected variables are listed in Table I. We focused only on temporal features for customers in the Small and Medium Enterprise (SME) segment[1]. The dependent variable is *requested termination*, which is binary in this context. It is true if the requested termination amounts to at least 10% of the customer's subscription count. Note that the threshold could be defined arbitrarily, but it should be relative to the size of the customer.

As all features are based on aggregated volumes, there are no missing values, e.g., if the customer didn't make any calls, the value would be zero. We defined a maximum value to limit the range for unbound features, such as the number of calls and data used. The values were defined by visually inspecting the distribution to ensure minimal impact. It is reasonable to assume that the information gained from the exact value is inversely related to the value. At some point, it simply becomes a "high-volume customer".

B. Data Transformations

Sine / Cosine Transformation. The data includes weekly customer states sorted by a timestamp value that is constantly increasing. Because weeks are cyclic, the distance between weeks should be the same in relation to each other. We can transform week numbers into two sine/cosine signals to achieve this, as shown in Equation (1).

$$cos(\frac{week*2\pi}{52}), sin(\frac{week*2\pi}{52}) \qquad (1)$$

Normalization. Visual inspection of the data shows that normalization is necessary as the signals have vastly different scales. Using min/max scaling with the interval $[0,1]$, see

[1] The motivation is practical; large customers often have dedicated managers who regularly follow up with the customer, and a churn ranking is less useful in practice. Micro enterprises have too small a variance on the account level.

TABLE I: Variables extracted from the data warehouse, where "Requested term." is the dependent/target variable in the experiments, i.e., termination requests. All values are based on weekly aggregates.

Variable	Description
Week	Current week number
Subscriptions	Average number of subscriptions
Calls	Number of calls
Duration	Duration of calls
Data	Number of megabytes transferred
Avg reg	Average lifetime of active subscriptions
Recency	Weeks since latest purchase
Loyalty	Weeks since oldest purchase
ARPU	Average revenue per user
All cases	Number of support tickets
Churn cases	Number of tickets related to churn
Order cases	Number of tickets related to orders
Trouble cases	Number of tickets related to troubleshooting
Binding 30d	Average subscriptions with < 30 days of binding
Binding 90d	Average subscriptions with < 90 days of binding
Has binding	Average subscriptions that have binding
Requested term.	Number of requested subscription terminations

Equation (2) mitigates issues that can occur when, e.g., calculating the loss by squaring large numbers. Feature distributions with a very long tail were capped to avoid skewing the normalization range.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

C. Constructing the Train and Test Datasets

Each customer is represented as a multivariate time series. The variables are described in Table I, where the dependent variable is the number of termination requests. A sliding window approach was used to extract subsets as described in Figure 1a. A dataset was created by iterating all customers and extracting subsets with the target window size. The shape of the dataset is based on the number of previous weeks included and the number of variables, i.e. (sequence length and number of features). The dataset consists of $\approx 8,000$ B2B customers and results in $\approx 270,000$ time steps.

The dataset has a significant class imbalance between time frames with and without termination requests. There is a $48 : 1$ class ratio between the number of weeks with termination requests compared to without. As we aim to train a model to recognize events leading up to termination requests, we include all positive time frames and then sample an equal amount of negative time frames using uniform random selection, i.e., under-sampling the negatives. This balanced set of time frames was shuffled and then split into a train and a test set using a five-fold cross-validation strategy. A rolling forecasting origin approach to cross-validation is common in time series forecasting. However, in this case, such an approach would not work for two reasons: first, we aim to classify the time frame, and second, it does not handle the class imbalance well. Each time frame is treated as an independent instance. We train the model on the positive and negative outcomes from all weeks.

1) Special preprocessing for non-sequential models: To work with the non-sequential models, the dataset was sliced to only include a single week as the target (one-vs-all), as seen in Figure 1b. This slicing was performed for weeks one to four as input and each of the single weeks one to four as targets, i.e., 16 combinations. The input was flattened as shown in Figure 2 so that all features are ordered sequentially on the same dimension. Note that this drastically increases the input dimensionality when we increase the number of past weeks to include. The data were not flattened for the sequential model, and the sequential ordering was maintained.

D. Selected Learning Algorithms

The following learning algorithms have been chosen to be included in the study because they have performed well on churn prediction tasks in previous studies. It should be noted that since we use multivariate data, it is not possible to use univariate models, such as traditional ARIMA models. All selected models are mentioned in the bibliometric review by Bhattacharyya and Dash [8]. Ahmad and Aljoumaa [16] used similar methods for churn prediction and achieved 93% AUC using boosting trees. Alboukaey, Joukhadar, and Ghneim [4] achieved an AUC of 91% using LSTM for churn prediction. We aim to show how different non-sequential algorithms perform compared to the sequential GRU model in predicting churn and the difference in longer forecasting horizons. The ease of implementation with the scikit-learn framework allowed us to evaluate several classical machine learning models.

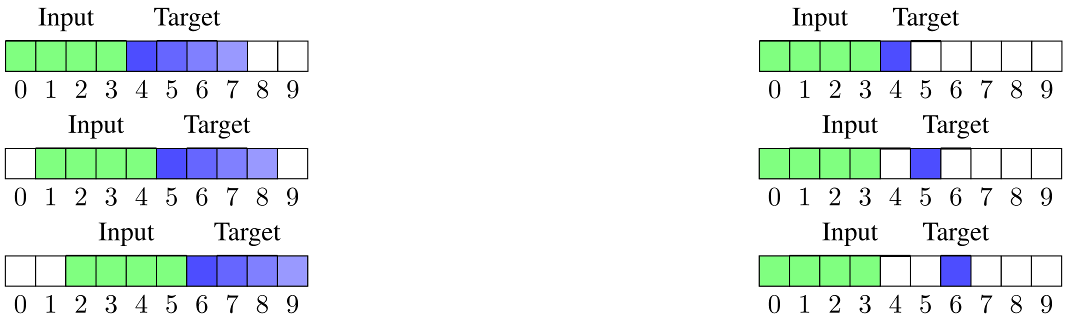
Random Forest (RF) [17] is an ensemble classifier consisting of multiple decision trees, where each tree is constructed from a sub-sample of the training data that under-samples the features. The hyper-parameters used were number of trees (500) and maximum depth (5 levels).

Support Vector Machine (SVM) [18], in a binary classification problem, tries to find the hyperplane that best separates the two classes by mapping the variables of the data in several dimensions. The SVM classifier was configured with a radial basis function kernel.

Multilayered Perceptron (MLP) [19] is a feed-forward neural network algorithm. In this study, Adam was used as a solver for weight optimization together with the ReLU activation function and two hidden layers of size 128.

Gradient Boosting (GB) [20] uses, similar to RF, a tree-based ensemble approach for training models. However, while RF trains the model by building trees in parallel using bagging, GB instead trains the model by building the trees in sequence using boosting, where each tree tries to correct the errors made by the previous tree in the sequence. The hyper-parameters used were: maximum tree depth of 5, maximum boosting rounds of 500, a learning rate of 0.01, and a binary logistic objective.

Gated Recurrent Unit (GRU) [21] has a similar architecture and performance as Long Short-Term Memory (LSTM) but consists of fewer parameters, thus making it easier and faster to train. The model comprises a GRU encoder-only layer and a fully connected neural network for classification. The



(a) Sliding window approach over a time series dataset spanning 10 weeks.

(b) Training slice of four weeks of data while predicting one, two, or three weeks ahead.

Fig. 1: Dataset creation with a sliding window.

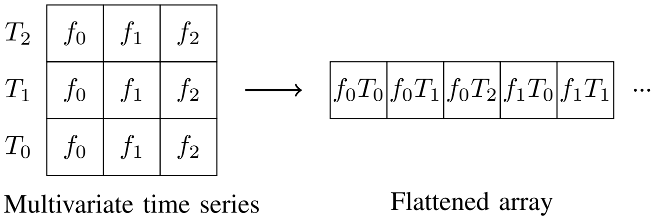


Fig. 2: Multivariate time series of features (f) per timestep (T) to a flat array.

GRU layer inputs 20 features over 16 timesteps in a batch of 64 sequences. The cells consist of 256 hidden weights and 3 unidirectional layers. The final hidden state of the cell connects to three fully connected layers with Parametric Rectified Linear Unit (PReLU) activation. Finally, the linear layers connect to a single output and a Sigmoid activation. All layers have a dropout of 25%. The network has, in total, just over one million parameters.

Random Guesser (RG) is a random, uniform classifier that predicts the possible class labels randomly according to a uniform distribution. The dummy classifier implementation from scikit-learn was used [22].

We used the scikit-learn implementation for all algorithms [22], except GRU, which was implemented using PyTorch [23].

E. Evaluation Metrics

We evaluate our models on a real-world dataset from a Swedish telecommunication provider. We used Precision, Recall, and F$_1$-score to measure the performance of our models [24]. The Precision and Recall are computed according to [24], and the positive class represents the customers that churn. Precision is calculated as the fraction of positive instances that are correctly classified:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where TP is the number of correctly predicted positives, and FP is the number of false positives. Recall is calculated as the ratio of the correctly predicted positives by all positives:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where FN is the number of false negatives. The F$_1$-score is the harmonic mean of the precision and recall [24]:

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN} = 2 \times \frac{Precision * Recall}{Precison + Recall} \tag{5}$$

We also evaluate the fairness of the algorithms by measuring the *Disparate Impact* [25]. Medium-sized customers with multiple subscriptions and regular churn might have a disproportional influence on the predictor compared to small customers with few subscriptions and irregular churn. The Disparate Impact is computed as the ratio of the proportion of favorable outcomes for the unprivileged group (i.e., medium-sized companies) to that of the privileged group (i.e., small companies), see Equation (6). We used Disparate impact since the data doesn't explicitly contain sensitive features [25].

$$DI = \frac{P(\hat{y} = 1 | X \in Unprivileged)}{P(\hat{y} = 1 | X \in Privileged)} \tag{6}$$

A value below 1 implies the privileged group benefits, and a value greater than 1 implies that the unprivileged group benefits. A value of 1.0 indicates that neither group benefits. In other domains, the wanted value is often above 0.8, indicating that the privileged group doesn't have too large benefits [25].

F. Statistical Tests

Several experiments will be conducted with various formats of time series data to compare the effects of the added temporal dimension. The non-parametric Friedman test [26] was used to investigate whether there were significant differences between the different temporal resolutions. If statistical differences were found, $p < 0.05$, the Nemenyi post-hoc test [27] was used for pair-wise tests between the candidates to determine between which there were significant differences.

G. *Explainability*

Presented by Lundberg and Lee in 2017 [28], SHapley Additive exPlanations (SHAP) analysis examines feature contribution to explain a model's output. The concept is derived from game theory, specifically from Shapley values, which assign a fair value to each characteristic according to its prediction value.

IV. Results

A. *Non-Sequential Algorithms*

The average F_1-score overall folds for each non-sequential model and combination of weeks are summarized in Figure 3. Most models gain a small increase in performance if the number of historical weeks is increased, although the effect is slight and diminishing. Each model performs better than the random guesser (RG) one week ahead, but the difference decreases as the week ahead increases. Using a Random Forest classifier with three weeks of history and a single week-ahead prediction gives the best results with an F_1-score of 79.3%. Table II and III present each model's Precision and Recall values.

RF and GB perform similarly regarding all three metrics, Precision, Recall, and F_1-score, and they also have similar standard deviation over folds. GB achieved the highest Precision metric, 0.836 when predicting one week ahead using two weeks of training data. Similarly, GB achieved the highest Recall at 0.739, using three weeks of training data. MLP performed worse than RF and GB, but most notably, it has a higher standard deviation across folds. SVM shows decent recall performance; however, due to its low precision, it receives an overall low F_1-score. Each model shows difficulties when making forecasts more than a week in advance, e.g., RF has an F_1-score of 0.793, which becomes 0.670, 0.648, and 0.626 for the targeted weeks-ahead of one to four.

B. *Gated Recurrent Unit*

The GRU network did not outperform the non-sequential models even with a longer input sequence of 16 past weeks, i.e., adding a deep neural network did not contribute to finding long-term churn signals. Table IV shows the results for the GRU model, which, similarly to the non-sequential models, shows the best performance when predicting churn one week ahead (F_1-score of 0.749) compared to two, three, and four weeks ahead (F_1-score of 0.571, 0.552, and 0.579). Contrary to the non-sequential models, GRU shows a more balanced recall and precision. Beyond forecasts made one week in advance, the predictive performance quickly decreases and becomes comparable to the random guesser.

C. *Statistical Tests*

To ascertain whether the performance difference between the number of weeks ahead is significant, a Friedman chi-squared test was used. The results show significant difference, $X^2(3) = 22.575$, $p < 0.05$. Therefore, we reject the null hypothesis and conclude that there is a difference between the number of future weeks to predict. A Nemenyi post-hoc test was conducted, and the results show a statistically significant ($p < 0.05$) difference between one and three weeks ahead and between one and four weeks ahead. No other statistically significant difference was found.

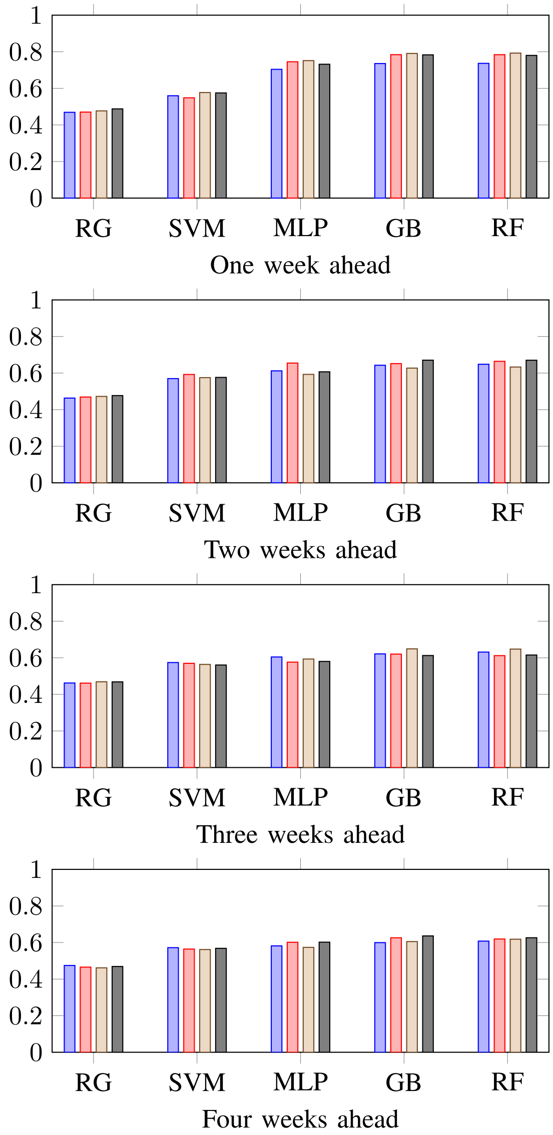


Fig. 3: F_1-score for non-sequential models for each week ahead. Blue, red, beige, and grey bar colors represent 1, 2, 3, and 4 past weeks as input.

D. *SHAP*

Figure 4 shows a summary plot of the SHAP analysis for Gradient Boosting. The analysis shows the contribution of each feature and its value to the outcome. The features are appended an index from $0 \rightarrow 3$ to indicate the number of weeks back, as explained in Section III-C1. Figure 4(a) shows the total number of support cases for the current and previous week and the number of churn cases for the current week, which are of high importance for the model. These features are less important if the predicted ahead horizon is increased from

TABLE II: Precision per model when predicting different numbers of weeks ahead and training on different numbers of past weeks. Best performance in bold font.

Week(s) ahead	Week(s) back	GB	MLP	RF	SVM	RG
1	1	.786 (.022)	.730 (.054)	**.814** (.017)	.554 (.025)	.471 (.000)
	2	**.836** (.025)	.765 (.034)	.829 (.046)	.546 (.012)	.472 (.000)
	3	.826 (.030)	.789 (.025)	**.832** (.021)	.575 (.008)	.478 (.000)
	4	**.823** (.017)	.783 (.016)	.819 (.010)	.566 (.020)	.488 (.001)
2	1	.654 (.017)	.664 (.064)	**.681** (.034)	.566 (.005)	.465 (.002)
	2	.675 (.032)	.663 (.014)	**.701** (.035)	.609 (.020)	.471 (.000)
	3	**.641** (.022)	.615 (.032)	.636 (.040)	.567 (.026)	.474 (.001)
	4	**.696** (.025)	.619 (.015)	.690 (.022)	.572 (.004)	.478 (.002)
3	1	.624 (.032)	.610 (.038)	**.642** (.029)	.570 (.016)	.464 (.000)
	2	**.624** (.017)	.595 (.018)	.618 (.022)	.561 (.013)	.464 (.001)
	3	**.653** (.023)	.605 (.027)	.647 (.007)	.567 (.023)	.471 (.000)
	4	.619 (.018)	.600 (.041)	**.638** (.020)	.553 (.012)	.470 (.000)
4	1	.601 (.017)	.585 (.066)	**.613** (.026)	.576 (.020)	.475 (.002)
	2	.629 (.047)	.611 (.063)	**.632** (.033)	.561 (.019)	.467 (.000)
	3	.608 (.027)	.595 (.041)	**.638** (.034)	.555 (.019)	.464 (.000)
	4	**.645** (.022)	.595 (.021)	**.645** (.018)	.565 (.024)	.471 (.000)

TABLE III: Recall per model when predicting different numbers of weeks ahead and training on different numbers of past weeks. Best performance in bold font.

Week(s) ahead	Week(s) back	GB	MLP	RF	SVM	RG
1	1	.654 (.025)	.671 (.057)	.624 (.030)	**.672** (.045)	.501 (.001)
	2	.710 (.026)	**.730** (.120)	.718 (.022)	.606 (.037)	.503 (.000)
	3	**.739** (.036)	.690 (.099)	.735 (.048)	.595 (.066)	.507 (.001)
	4	**.722** (.047)	.663 (.106)	.719 (.050)	.657 (.075)	.513 (.001)
2	1	.610 (.034)	.514 (.090)	.574 (.022)	**.619** (.069)	.493 (.001)
	2	.593 (.018)	**.635** (.062)	.584 (.015)	.526 (.036)	.500 (.001)
	3	.582 (.035)	.517 (.125)	.623 (.055)	**.664** (.026)	.505 (.001)
	4	.616 (.009)	**.669** (.049)	.625 (.012)	.635 (.044)	.507 (.001)
3	1	.611 (.026)	.595 (.040)	.591 (.025)	**.617** (.068)	.491 (.000)
	2	.603 (.053)	.555 (.069)	.585 (.024)	**.691** (.024)	.492 (.001)
	3	.631 (.044)	.588 (.120)	**.649** (.017)	.538 (.056)	.503 (.001)
	4	.587 (.033)	.586 (.126)	.541 (.028)	**.665** (.026)	.502 (.000)
4	1	.592 (.045)	**.610** (.064)	.584 (.048)	.564 (.024)	.497 (.001)
	2	**.616** (.029)	.565 (.103)	.575 (.034)	.599 (.048)	.493 (.000)
	3	.597 (.035)	.554 (.145)	.558 (.043)	**.671** (.037)	.493 (.000)
	4	.607 (.039)	**.661** (.058)	.571 (.040)	.613 (.029)	.500 (.001)

TABLE IV: Evaluation metrics for the GRU model.

Week(s) ahead	Week(s) back	F_1-score	Precision	Recall	Random
1	16	**0.749**	**0.749**	**0.749**	0.490
2	16	0.571	0.573	0.572	0.485
3	16	0.552	0.553	0.553	0.485
4	16	0.579	0.579	0.579	0.496

one to four weeks, as presented in Figure 4(b). It is important to note that this analysis does not explain why these values are important or the causal relationship.

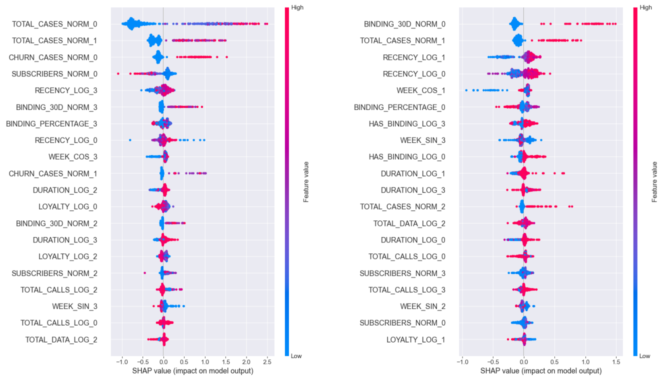
E. Fairness

As illustrated by Figure 5, the models do not exhibit unfair treatment of the defined privileged group. On the contrary, the models tend to favor the unprivileged group. We defined the privileged group as medium-sized companies and the unprivileged group as small companies. SVM stands out from the other models and provides the most equal treatment with low variation over the different training iterations. However, as demonstrated earlier, SVM also performs the worst, i.e., its almost random behavior is likely the reason.

V. Discussion

The multifaceted challenges with churn prediction for B2B customers are prevalent throughout this study. At the foundation, we have the dynamic definition of churn. A customer can show tendencies towards churning behavior if, for instance, they terminate faster than they add new subscriptions. It can be normal for one customer to request terminations every other week while it is not for another.

The problem of identifying churning behavior is based on the assumption that we can observe the behavior directly or



(a) One week-ahead predictions (b) Four week-ahead predictions

Fig. 4: SHAP summary of the Gradient Boosting classifier.

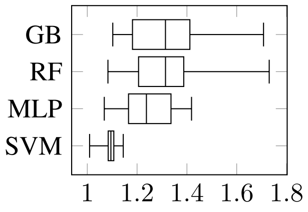


Fig. 5: Ratio of the proportion of favorable outcomes for the unprivileged over the privileged group.

indirectly from the data. However, many confounding variables affect the outcome and inhibit the ability to isolate churning behavior. It is a problem of imperfect information.

Furthermore, customer engagement is nuanced as the desired services and products vary between customers. Churn modeling is commonly motivated by the high customer acquisition cost compared to retaining existing ones. However, focusing on cross-selling and up-selling may increase the lifetime value more than preventing terminations. Terminations are unavoidable and, to some customers, normal. Identifying where long-term retention and the expected lifetime value can be improved might be more beneficial in practice.

A typical B2B telecommunication customer purchases several products and services over several months or years. Depending on their needs and desires, the engagement and reason for terminating subscriptions change over time. Forecasting trends and identifying causal variables that affect the trends may provide a more robust ranking. Such ranking may identify customers where retention failure is more likely, which reduces lifetime value.

The decision to view the customer as the sum of its parts was grounded in the business structure. Customer analysis for B2B is generally based on the entire customer base, a specific segment, or the aggregated engagement of a single customer. However, this approach limits the ability to draw more powerful conclusions regarding individual service usage. For instance, subscriptions with binding rarely register for termination until the binding period is almost over. Analyzing the individual parts may also reveal patterns related to the customers' changing needs that are difficult to spot from an aggregated view.

VI. Conclusion and Future Work

This study investigated the effect of different temporal resolutions and forecasting horizons for B2B customer churn predictions. The results indicate that the best predictions are closer in time than previously assumed. Significant differences were observed between forecasting horizons, where the best performance was achieved within one week. Employing deep learning models with the available data did not resolve this shortage of long-term linkage between behavioral patterns and termination requests. The results demonstrate the difficulties of churn predictions, which increase with the forecasting horizon.

Our experiments show that time series data can identify behavioral patterns of churning customers, which may allow a business to make informed decisions regarding anti-churn targeting. These results may adapt the retention strategies and allow more proactive interventions. However, it is essential to note that features relevant to predicting churn may not be appropriate to reduce churn. Future work should also investigate the effect of using time series data to predict customers' sensitivity to treatments.

Future work may also investigate forecasting as an alternative solution. If the number of subscriptions is used as a proxy for churn, a forecasting model may determine the future trend and thereby identify customers at risk of churn, e.g., using ARIMA models or transformer models for multivariate time series forecasting.

To the best of our knowledge, there is no publicly available multivariate customer churn dataset based on time series, so we could not compare these results to other research. The research gap in both time series and the B2B segment for customer churn made it challenging to find a common baseline overall. Future work may include the creation of such a public dataset.

References

[1] C.-P. Wei and I.-T. Chiu, "Turning telecommunications call details to churn prediction: A data mining approach," *Expert Systems with Applications*, vol. 23, no. 2, pp. 103–112, 2002. DOI: 10.1016/S0957-4174(02)00030-1.

[2] N. Gordini and V. Veglio, "Customers churn prediction and marketing retention strategies. an application of support vector machines based on the AUC parameter-selection technique in B2B e-commerce industry," *Industrial Marketing Management*, vol. 62, pp. 100–107, 2017. DOI: 10.1016/j.indmarman.2016.08.003.

[3] C. G. Mena, A. D. Caigny, K. Coussement, K. W. D. Bock, and S. Lessmann, *Churn prediction with sequential data and deep neural networks. a comparative analysis*, arXiv 1909.11114, 2019. arXiv: 1909.11114 [stat.AP].

[4] N. Alboukaey, A. Joukhadar, and N. Ghneim, "Dynamic behavior based churn prediction in mobile telecom," *Expert Systems with Applications*, vol. 162, p. 113 779, 2020. DOI: 10.1016/j.eswa.2020.113779. (visited on 11/10/2023).

[5] H. Abbasimehr and M. Shabani, "Forecasting of customer behavior using time series analysis," in *Data Science: From Research to Application*, M. Bohlouli, B. Sadeghi Bigham, Z. Narimani, M. Vasighi, and E. Ansari, Eds., ser. Lecture Notes on Data Engineering and Communications Technologies, Cham: Springer International Publishing, 2020, pp. 188–201, ISBN: 978-3-030-37309-2. DOI: 10.1007/978-3-030-37309-2_15.

[6] N. Jajam and N. Challa, "Dynamic behavior-based churn forecasts in the insurance sector," *Computers, Materials and Continua*, vol. 75, no. 1, pp. 977–997, 2023. DOI: 10.32604/cmc.2023.036098.

[7] A. S. Awate and S. K. Sharma, "Understanding customer behaviour: A comprehensive survey of segmentation and classification techniques in the age of big data," *Int'l J. of Intelligent Systems and Applications in Engineering*, vol. 11, no. 7, pp. 486–514, 2023.

[8] J. Bhattacharyya and M. K. Dash, "What do we know about customer churn behaviour in the telecommunication industry? a bibliometric analysis of research trends, 1985–2019," *FIIB Business Review*, vol. 11, no. 3, pp. 280–302, 2022. DOI: 10.1177/23197145211062687.

[9] H. Ribeiro, B. Barbosa, A. Moreira, and R. Rodrigues, "Determinants of churn in telecommunication services: A systematic literature review," *Management Review Quarterly*, 2023. DOI: 10.1007/s11301-023-00335-7.

[10] I. Figalist, C. Elsner, J. Bosch, and H. H. Olsson, "Customer churn prediction in b2b contexts," in *Software Business*, S. Hyrynsalmi, M. Suoranta, A. Nguyen-Duc, P. Tyrväinen, and P. Abrahamsson, Eds., ser. Lecture Notes in Business Information Processing, 2019, pp. 378–386, ISBN: 978-3-030-33742-1. DOI: 10.1007/978-3-030-33742-1_30.

[11] A. De Caigny, K. Coussement, W. Verbeke, K. Idbenjra, and M. Phan, "Uplift modeling and its implications for B2B customer churn prediction: A segmentation-based modeling approach," *Industrial Marketing Management*, vol. 99, pp. 28–39, 2021. DOI: 10.1016/j.indmarman.2021.10.001. (visited on 03/15/2024).

[12] "Telco customer churn." (), [Online]. Available: https://www.kaggle.com/datasets/blastchar/telco-customer-churn (visited on 08/08/2024).

[13] H. Jain, G. Yadav, and R. Manoov, "Churn prediction and retention in banking, telecom and it sectors using machine learning techniques," in *Advances in Machine Learning and Computational Intelligence*, S. Patnaik, X.-S. Yang, and I. K. Sethi, Eds., 2021, pp. 137–156, ISBN: 978-981-15-5243-4.

[14] A. Tamaddoni Jahromi, S. Stakhovych, and M. Ewing, "Managing b2b customer churn, retention and profitability," *Industrial Marketing Management*, vol. 43, no. 7, pp. 1258–1268, Oct. 1, 2014. DOI: 10.1016/j.indmarman.2014.06.016. (visited on 03/22/2024).

[15] F. Zhao, B. Dong, H. Pan, and A. Shi, "A mining algorithm to improve LSTM for predicting customer churn in railway freight traffic," *SIC*, vol. 32, no. 2, pp. 25–38, 2023. DOI: 10.24846/v32i2y202303.

[16] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," *Journal of Big Data*, vol. 6, no. 1, 2019. DOI: 10.1186/s40537-019-0191-6.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. DOI: 10.1023/A:1010933404324.

[18] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002. DOI: 10.1145/505282.505283.

[19] P. Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. USA: Cambridge University Press, 2012, ISBN: 1107422221.

[20] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, Dec. 11, 2014. arXiv: 1412.3555[cs]. [Online]. Available: http://arxiv.org/abs/1412.3555 (visited on 03/13/2024).

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in PyTorch," en, Oct. 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ (visited on 03/22/2024).

[24] I. Witten, E. Frank, M. Hall, and C. Pal, *Data Mining: Practical Machine Learning Tools and Techniques* (The Morgan Kaufmann Series in Data Management Systems). Elsevier Science, 2016, ISBN: 9780128043578. [Online]. Available: https://books.google.se/books?id=1SylCgAAQBAJ.

[25] D. Pessach and E. Shmueli, "A review on fairness in machine learning," *ACM Comput. Surv.*, vol. 55, no. 3, Feb. 2022. DOI: 10.1145/3494672.

[26] P. H. Kvam and B. Vidakovic, *Nonparametric statistics with applications to science and engineering*. Hoboken, NJ, USA: John Wiley & Sons, 2007.

[27] M. Hollander, D. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. Hoboken, NJ, USA: John Wiley & Sons, 1999.

[28] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/%20paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (visited on 03/18/2024).