Norwegian
University of
Life Sciences

Master's Thesis 2024 30 ECTS
Faculty of Science and Technology

Siamese Networks for Telecommunication Customer Churn Data in a Few-Shot Learning Context

Eljar Alihosseinzadeh
Data Science

Abstract

Accurate customer churn prediction is important for businesses seeking to retain their valuable customers who might churn. A new method to accurately predict which customer might churn is machine learning models. These models learn from prior labeled data to make informed predictions. However, many businesses, such as smaller- or startups businesses do not have access to an abundance of data for the model to learn from. Few-shot learning, a subfield of machine learning, presents a potential solution by enabling accurate predictions even with limited labeled training data.

Siamese networks is a machine learning model, typically known for Few-shot learning scenarios within image classification. This thesis investigates its potential to adapt its Few-shot learning capabilities into the realm of tabular data, specifically within telecommunication churn prediction. The thesis will aim to answer whether Siamese networks are a viable option in telecommunication churn prediction when using tabular data, as well as, how effective they are in improving the accuracy of Few-shot learning models when applied to telecommunication customer churn prediction.

The methodology taken in use involves feature pre processing, consisting of feature encoding, feature scaling and SMOTE. SMOTE addresses the common challenge of class imbalance usually experienced when working with churn prediction data. A specialized pairing function was also made to prepare the data for the Siamese network as pairs. The evaluation of the dataset was performed on two telecommunication churn datasets, Orange and IBM. The model was also put up against other traditional machine learning models in a comparative analysis to get a benchmark and provide context for the Siamese network performance relative to well-known alternatives.

Results from the evaluation showcased impressive results from the Siamese network on tabular data, it achieved 82.4% on the IBM dataset & 93.0% on the Orange dataset for the lowest sample size (5 churn, 25 non-churn) outperforming all other traditional models with a sizeable margin. It also had good results on the whole datasets reaching 83.6% & 94.4% for the IBM- and Orange datasets respectively (only surpassed by Random Forest on both instances). The study concludes that Siamese networks offer a new approach for tabular churn prediction, especially within the subfield of Few-shot learning. Thereby, the applicability of Siamese networks is extended beyond image classification.

Sammendrag

Nøyaktig kundeavgangsforutsigelse er avgjørende for bedrifter som ønsker å beholde sine kunder. Maskinlæringsmodeller presenterer en ny metode for å forutsi hvilke kunder som kan komme til å avslutte kundeforholdet. Disse modellene lærer fra tidligere merket data for å gjøre informerte beslutninger. Imidlertid har mange bedrifter, spesielt mindre bedrifter, oppstartsbedrifter eller bedrifter uten tilgang til store mengder data, også behov for å identifisere kunder som kan avslutte kundeforholdet. Few-shot learning, et underfelt innen maskinlæring, presenterer en potensiell løsning ved å muliggjøre nøyaktige forutsigelser selv med begrensede mengder merket treningsdata.

Siamesiske nettverk er en maskinlæringsmodell som vanligvis er kjent for Few-shot learning innen bildeklassifisering. Denne studien undersøker potensialet for å tilpasse dens Few-shot learning kapasiteter til tabulære data, spesielt innen telekommunikasjonens kundeavgangsprediksjon.

Studien tar sikte på å besvare hvorvidt Siamesiske nettverk er et potensielt alternativ for kundeavgangsprediksjon i telekommunikasjon når man bruker tabulær data. Samt hvor effektive de er til å forbedre nøyaktigheten til Few-shot learning modeller når de brukes på kundeavgang prediksjon innen telekommunikasjon.

Metodikken som er brukt inneholder forbehandling av egenskapsvariabler, inkludert SMOTE for å adressere den vanlige utfordringen med klasseubalanse som ofte oppleves når man arbeider med kundeavgangsdata. En paringsfunksjon ble også laget for å forberede dataene for det siamesiske nettverket i par. Evalueringen ble utført på to telekommunikasjonskundeavgangsdatasett, Orange og IBM. Modellen ble også sammenlignet med andre tradisjonelle maskinlæringsmodeller for å få et referansepunkt og gi kontekst for det Siamesiske nettverkets ytelse i forhold til kjente alternativer.

Resultatene fra evalueringen viste imponerende resultater fra det siamesiske nettverket på tabulære data. Det oppnådde 82,4% nøyaktighet på IBM-datasettet og 93,0% på Orange-datasettet for den minste utvalgsstørrelsen (5 avgang, 25 ikke-avgang), og overgikk alle andre tradisjonelle modeller med god margin. Det hadde også gode resultater på hele datasettene, og nådde 83,6% og 94,4% for henholdsvis IBM- og Orange-datasettene (kun overgått av Random Forest i begge tilfeller). Studien konkluderer med at siamesiske nettverk tilbyr en ny tilnærming for tabulær kundeavgangsforutsigelse, spesielt innen Few-shot learning, og utvider dermed anvendelsen utover tradisjonell bildeklassifisering.

acknowledgements

I would like to thank my supervisors Fadi Al Machot & Martin Thomas Horsch, for their continous support, insightful feedback and expert guidance throughout the master thesis process. Their knowledge and help was vital in shaping this research.

Contents

List of Figures

List of Tables

Abbreviations

SMOTE Synthetic Minority Oversampling Technique

SVM Support Vector Machine

RF Random Forest

LR Logistic Regression

ROC Reciever Operating Characteristics

AUC Area Under Curve

NB Naive Bayes

FN False Negative

FP False Positive

TN True Negative

TP True Positive

RBF Radial Basis Function

FPR False Positive Rate

TPR True Positive Rate

KNN K-Nearest Neighbors

CNN Convolutional Neural Network

DT Decision Tree

GBT Gradient Bossting Trees

XGBoost Extreme Gradient Boosting

ANN Artificial Neural Network

1 Introduction

1.1 Background

A customer within a subscription based service provider becomes a "churner" when the customer decides to discontinue the service. The customer can either churn to a competitor or churn by deciding they no longer require the service. i.e. churn is customer turnover. Churn is a key element for companies where a customer can easily switch to a competitors service [1].
Churning customers can be divided into two groups, where one group is defined as involuntarily churners, customers which are forced to terminate the subscription because of issues regarding payment or contractual violation. The other group is voluntarily churners, these churn as a result of themselves making a conscious decision to cancel the subscription with the company. Because they found a cheaper, more advanced or better quality subscription somewhere else. These kind of churners are more difficult to identify, but is also the group which is of interest of the company to identify [12].

Retaining existing customers through customer retention methods for businesses has a much lower cost, between 5 to 10 times, than the cost of selling to a new customer. Long lasting customers are of more value than new customers, i.e a customer who has a nine-year relationship would be more valuable than a new customer. The same goes for four-year relationship customer up against a nine-year relationship customer ([28] as cited in [22]).

In the telecommunication service sector loss of valuable customers through churn to competitors is a common phenomenon. There has been a liberalisation of the market, opening up for new competitors and several actors in the field, all competing for the same customers. Businesses present deals, new services and technologies to attract customers to their business rather than a competitor. Customer churn has become a central challenge for businesses within telecommunication services.

As the challenge for customer retention grows, the market gets more competitive and the use of advanced technologies to find a solution against customer churn becomes critical. The world is becoming more data based, opening the possibility of using advanced analytical tools to address these challenges effectively.

Machine learning has revolutionized the way data is analyzed and interpreted across different domains. In its essence, machine learning takes in use algorithms to parse data, use data to learn, and make informed decision based on what it has learned. By combining

computer science with statistics, machine learning enables predictive modeling without having to perform explicit programming for each and every task.
By having the ability to predict when or which customers are at risk of churning enables companies to implement customer retaining strategies, thereby reducing churn rates and increasing customer lifetime values. Machine learning has emerged as an important tool in identifying potential churners by analyzing patters in customer data and predicting customer behaviour.
Extensive research has been conducted on the application of machine learning techniques to identify customers at risk of churn, significantly benefiting the sector. This body of work, complemented by numerous surveys, underscores the potential of machine learning algorithms to predict customer behavior and inform retention strategies effectively [2] [22] [33].

1.2 Problem statement

The thesis will address the following statements:

- While Siamese networks have demonstrated efficacy in few-shot learning for image and text data, their application to tabular data remains underexplored.
- Predicting customer churn with high accuracy and remains a challenge for businesses, especially with limited labeled data.

1.3 Research questions

The thesis will address these problem statements and contribute to the field by exploring these research questions.

- How does the performance of Siamese networks compare to traditional machine learning models in churn prediction tasks using tabular data?

While Siamese networks have traditionally been applied to image and text data, their efficiency when dealing with tabular data remains underexplored. By evaluating its performance up against baseline models in customer churn prediction, it can establish a benchmark for its applicability in the tabular domain.

- How effective are Siamese networks in improving the accuracy and efficiency of few-shot learning models when applied to customer churn prediction?

Siamese networks have demonstrated notable success in Few-shot learning for image classification, its applicability to Few-shot learning for tabular data is underexplored. Will it be possible to transfer its success onto tabular data?

1.4 Objectives

Evaluate the effectiveness of Siamese networks using tabular data, by attempting to transfer its success in Few-shot learning in image classification to classification of tabular data. Assess its performance on churn prediction, evaluating its performance through several evaluation metrics. This includes comparing the performance of the Siamese network with different traditional machine learning approaches.
Investigate the positive impact of various feature pre processing techniques, on Siamese networks. Applying them to the model trying to improve its performance on churn prediction in a Few-shot learning environment. Specifically Synthetic Majority Oversampling Technique (SMOTE) and pair generation functions for the Siamese network.

1.5 Aim & scope & limitations

The aim of this thesis is to broaden the field of Siamese networks on tabular data. Through research to provide insights into the potential of Siamese networks on churn prediction, while focusing on a Few-shot learning environment.

This encompasses

- A performance review on IBM and Orange telecom churn prediction through evaluation metrics such as accuracy, F1-score, ROC/AUC and confusion matrix's.
- Exploring feature pre processing techniques to handle common challenges, such as class imbalance, within churn prediction.
- Analysis of the Siamese network on churn prediction in a regular and Few-shot learning environment. Compare the results up against other machine learning models to establish a baseline.

Limitations Some limitations within the study:

The models performance is within churn prediction and might not be applicable to other data types or industries.

The thesis explores a specific set of feature pre processing techniques (SMOTE etc), and

therefore does not cover all other beneficial methods which will not be covered in the study.

The performance difference on the two different datasets used in the thesis suggests the performance results, while performed on baseline models, does not directly apply to data collected by all telecom businesses and does not warrant the same results.

2 Literature review

The field of churn prediction is a rapidly evolving field, driven by both businesses need to retain their customers and advancements within machine learning. This literature review examines different methodologies employed in recent studies, highlighting their methodology, results and limitations. This is to provide a comprehensive foundation for understanding the field of churn prediction as well as emerging trends.

2.1 Review of related literature

The following table is made up of relevant studies on churn prediction between 2018 and 2023.

Table 1: Summary and key takeaways from various relevant churn prediction studies within the research area

TITLE	**DATA**	**PRE-PRO**	**MODEL**	**METRICS**	**LIMITS**	**RESULT**	**KEY TAKE-AWAY**	**YEAR**
Prediciting customer churn prediciton in telecom sector using various machine learning techniques [11]	Telco Customer Churn	Basic transformation and preprocessing	LR, SVM, RF, GBT	ROC, AUC		Gradient Boosting-84.57%	Boosting performed the best. SVM performed the worst.	2018
Churn Prediction in Telecommunication using Logistic Regression and Logit Boost [16]	Orange	Removed some features (Nan values & Redundant)	LR, LB	Kappa Statistics, MEA, RMSE, ROC, F1, Recall, Precision	Using standalone techniques rather than hybrid models	ROC LR-85.24% LB-85.19%	Recommends hybrid model with embedding in future. Wants to attempts SVM.	2019

Continued on next page

Churn Prediction: A Comparative Study Using KNN and Decision Trees [13]	Orange	Filters to remove noise	DT, KNN	Accuracy, Recall, Precision, F1, ROC, AUC	Over coming Raw data	Accuracy KNN-86.8% DT-92.6%	DT had slight better accuracy, but KNN had better TP, performing better for churn prediction.	2019
Customer churn prediction in telecom using machine learning in big data platform [17]	Orange (9 months of data)	Used pearson to remove features. Used K-fold for hyperparameter tuning. Used mean for Nan values.	XGboost DT, RT, GBM	AUC	Data imbalance	AUC XGBOOST Syriatel-93.301% XGBOOST orange dataset-89%	XGboost performed the best. Solved imbalance by under sampling & tree algorithms. Introduced Social Network Analysis using connection between customers as data.	2019

Continued on next page

A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector [30]	Orange	Removed noise Feature selection (Information gain and Correlation attributes)	RF	Accuracy, TP rate, FP rate, Precision, Recall, F1, ROC	Nan values in new data to predict on. Reason for churn is not given.	Accuracy 88.63%	Clusters churn customers into groups using cosine similarity opening up for group based retention methods (k means clustering).	2019
Customer churn prediction system: a machine learning approach [19]	Telco Customer Churn	Variance Analysis, correlation matrix, GSA for feature selection	LR, DT, AdaBoost, KNN, RF, NB, SVM, XgBoost	Recall, Precision, TP, TN, Accuracy, F1, AUC		Adaboost-81.71% XGboost 80.8% Were the two most consistent performers	In depth review of K fold cross validation & and confusion matrix.	2021

Continued on next page

Integrated Churn Prediction and Customer Segmentation Framework for Telco Business [33]	Telco Customer Churn	SMOTE, goes through several oversampling techniques on telecom churn. Relevant and well literature review.	AdaBoost	Precision, Recall, F1, ROC, AUC	Data imbalance	Model F1-63.11% AUC-84.52% Accuracy-77.19%	Customer segmentation after prediction providing likelihood of churning as well. SMOTE	2021
Telecom Churn Prediction System Based on Ensemble Learning Using Feature Grouping [36]	Orange	Equidistant grouping (sturges formula). Feature grouping, ensemble system & model stacking w soft voting.	Ensemble learning (soft voting) XgBoost, LR, DT, NB	Accuracy, Precision, Recall, F1		Model Accuracy Orange-96.12% Accuracy Newdata-98.09%	Equidistant grouping Stacking models, achieving 10% better performance (w. Soft voting) Feature grouping	2021

Continued on next page

B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM [34]	Bank data	Noise removal, feature importance, data transformation.	Model stacking, CNN layers → Logistic Regression prediction layer.	Precision, Recall, F1, Accuracy	No soft voting or weighted average aggregation.	Accuracy-97.10% Recall-94.39% F1-93.20% Accuracy-97.10%	Combination approach improves the performance of the model. In depth model evaluation and good results	2023
Deep Churn Prediction Method for Telecommunication Industry [29]	Orange Data & Large	Noise removal Regular preprocessing steps	Ensemble learning, ANN, DT, KNN, LR, CNN	Accuracy, Precision, F1, Sensitivity, Specificity	Data imbalance	ANN-99% accuracy	Impressive results, 98% for CNN and 99% for ANN. Properly preparing the data has a big impact	2023

The studies emphazises on the importance of proper pre-precossing of data to improve model performance. Comprising of noise removal, handling missing values and feature processing. Several studies handle the challenge of class imbalance which is common in churn prediction. Synthetic Minority Over-sampling Technique (SMOTE) and under sampling are some of the methods used to help balance the data.
The reviewed studies showcase a wide range of different model approaches, Logistic Regression, Support Vector Machines and Random Forest are some examples. More recent studies, particularly from 2021 and onwards, demonstrate a tendency to use deep learning techniques and combined models. These models often achieve higher accuracy.
To the best of our findings, only few works tried to tackle the problem of churn prediction as Few-shot learning.
The findings on relevant research studies on prediction through machine learning sets foundation for methodologies used in our thesis.

2.2 Limitations of Related Works

1. Data Preparation Challenges [13]: The process of overcoming raw data challenges and preparing data for analysis is a critical hurdle. This includes issues such as handling missing values, dealing with noisy data, and extracting meaningful features that accurately represent customer behavior and probability to churn.

2. While churn prediction has been extensively studied, there is limited exploration of Few-shot learning approaches for churn prediction.

3. Unbalanced Datasets [17] [33] [29]: A recurring challenge in churn prediction is the inherent imbalance in datasets. Typically, the proportion of customers who churn is significantly lower compared to those who do not. This imbalance poses a significant challenge in training predictive models, as it can lead to biased predictions and reduced model sensitivity to the churn class. Consequently, this highlights the importance of handling this problem in a Few-shot learning setting.

4. Handling of NaN Values and Unknown Churn Reasons [30]: The presence of NaN values in datasets, and the lack of explicit reasons for customer churn, add complexity to modeling churn. These factors can obscure underlying patterns and hinder the model's ability to learn effectively from the data.

3 Theoretical Framework

In this chapter we will explore important theory relevant to the thesis. We examine different advanced learning techniques, specifically One-shot learning, Few-shot learning and Zero-shot learning. These techniques allow for significant learning within machine learning models from minimal data input. Further on, we discuss different distance metrics. Distance metrics are a core part of Siamese networks and other similarity based learning methods. The theory within Siamese networks is explored to finish off the section, going through its inner workings, different data embedding techniques, loss functions and evaluation metrics relevant to itself.
By exploring the theory, we aim to provide a robust understanding of modern machine learning techniques and the core theory within Siamese networks.

3.1 Advanced Learning Techniques

Traditional machine learning often thrives on large, well labeled datasets. However this is not always the case in real-world scenarios. Many real-world scenarios have to deal with constraints such as limited data or limited labeled data. This could be attributed to factors such as, the cost of data acquisition, time sensitivity or the rarity of a certain event. We delve into one-shot learning where the focus is on achieving high accuracy with a single example, few-shot learning which extends this concept to slightly more but still limited data, and zero-shot learning that attempts to classify new classes without any labeled examples. These techniques not only enhance the model's ability to generalize from minimal data but also open new avenues for machine learning applications in fields where traditional data-hungry models would fall short.

3.1.1 One-Shot Learning

One-shot learning is a subfield within machine learning. It tackles a unique challenge of learning and classifying data with extreme scarcity of labeled data. Unlike traditional machine learning where models operate on large datasets, one-shot learning are for data limited conditions. Thus making it relevant for real-world scenarios where labeled data is scarce. Examples of such scenarios where One-shot learning is relevant are rare medical disease, new technologies or malfunctioning equipment. One-shot learning is when the model is trained on a single sample of each class, and then used to predict on unlabeled data.

One-shot learning focuses on estimating similarity between new unseen data, and the

limited labeled data it has trained on. Taking in use distance metrics to estimate how close unseen data is to already known examples, further discussed in section 3.2. There are also several challenges presented with One-shot learning. Limited data can cause the model to struggle with generalizability on unseen data. The importance of data augmentation and feature pre processing is paramount to extract maximum information from the limited labeled data available.

3.1.2 Few-Shot Learning

Few-shot learning is another subfield within machine learning, it addresses the challenge of learning and making predictions on data with a limited number of labeled data. This is a step up from the extreme data scarcity of one-shot learning, but still significantly less data than traditional machine learning approaches. Few-shot learning is valuable in situations where obtaining labeled data is possible, but still restricted due factors such as cost, time or rarity. Rare animals or in our case churn prediction with a limited amount of churning samples, are some scenarios where few-shot learning might come in handy.

In its core few-shot learning is similar to One-shot learning, it builds upon the similarity based learning presented in the One-Shot learning subsection, using a distance metric. It is also presented with the same challenges seen when using One-shot learning, it can struggle to generalize on unseen data. In our churn case, models can exhibit behaviour where they favor the majority class.
Few-shot learning allows an insight into models behaviour when working with fewer samples. This scenario, in our study, is presented as churn cases. As in real-world scenarios on churn will have fewer churn customers, than nonchurn. By employing several different data augmentation techniques, seen in section 4.2.1, we can test the benefits of similarity learning through Siamese networks for Few-shot learning.

3.1.3 Zero-Shot Learning

Zero-shot learning is a machine learning scenario, which challenges the traditional requirement of training examples for every class. Zero-shot learning models are trained to be able to recognize and categorize data which it has not yet observed through training.
This is done through auxiliary information, such as, textual description and attributes. For example if there is need to diagnose a novel disease based on the symptoms alone.

Zero-shot learning is a research area that still is vibrant. There is efforts on addressing these challenges, by improved embedding or improving the auxiliary information.

3.2 Distance Metrics

Distance metrics are important in machine learning, particularly for similarity based models such as Siamese networks. They are a way for the model to measure the similarity or dissimilarity between data, directly influencing performance in classification. In Siamese networks the distance metric decides how similarity between data points is calculated. Pairs which has a high distance metric value between them would be dissimilar, while a pair with a low distance metric between them would be similar. The measurement of distance directly affects the loss of the model, forming the backbone for training, driving the network to learn discriminative features that distinguish between churning and nonchurning customers. Distance metrics commonly used with Siamese networks include:

- **Euclidean distance**, represents the geometric distance in the n dimensional space. In lament terms it represents the straight line distance between two points in Euclidean space. It is given as: $d(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$, where $A = a_1, a_2...a_n$ and $B = b_1, b_2...b_n$ represents two points.
- **Cosine distance**, measures the cosine of the angle between the two vectors in the embedding space. Cosine similarity comes in handy when the magnitude of the vectors are not as important, but rather their directions. For two vectors $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_n)$ the cosine similarity is given by: $\cos(\theta) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2}}$
- **L1 distance**, measures the distance between two points. It is the sum of absolute differences of their corresponding features across n dimensions. Its formula is given by $D(A, B) = \sum_{i=1}^{n} |a_i - b_i|$, where A & B represents the data points.
- **Pearson distance**, measures the correlation between two variables, giving an indication on how much they are related. A high value, close to 1, indicates the variables move in the same direction, a negative value, close to -1,indicate they move in opposite directions. Lastly, values close 0 indicate no relationship. The pearson distance is calculated by: Pearson Distance $= 1 - \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$

Each distance metric being optimally suited for different application areas.

3.3 Siamese Network

A SNN is a neural network which is made up of two or more identical sub-networks. The sub-networks are referred to as "twins", as they have identical configurations, parameters and weights. Both sub-networks in a Siamese network are mirror images of each

other in terms of architectural design, operational parameters and updating of weights. both sub-networks (twins) share identical architectures and parameters. During the back-propagation process, when updating the network's weights, these updates are applied equally to both sub-networks. This ensures that any learning or adjustments made by the network in response to the input data are consistently reflected across both halves of the Siamese architecture.

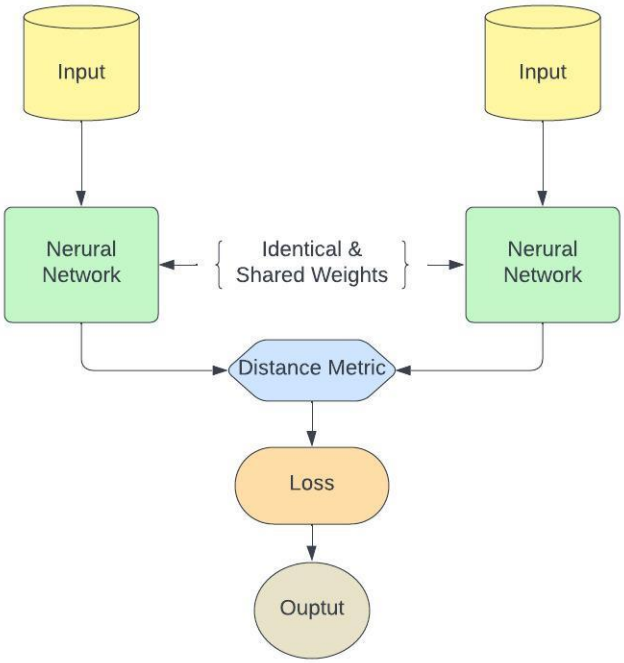


Figure 1: Showcasing a traditional Siamese network structure

Siamese networks are designed to learn embeddings such that similar items are closer together, and dissimilar items are farther apart in this space. The quality of embeddings directly influence the models performance. By capturing the essential characteristics of the input data, they can determine the level of similarity or dissimilarity. Once the inputs are transformed into embeddings, Siamese networks use a distance metric to define the similarity between the embeddings [9].

One of the key advantages of Siamese networks is its efficiency in learning from limited data. The network focuses on learning a similarity function, which makes it possible to generalize from a small number of examples, making them suitable for few-shot-learning or one-shot-learning scenarios. They are also able to classify new data without having to retrain the network [18].
Siamese networks are designed to learn a similarity metric or distance measure between

the inputs. They are known to excel and achieve results when working on tasks like image verification and authorship verification. It performs well in situations where it is required to compare and measure between pairs of data points, which is why it doesn't require large amounts of data to perform well [10] [18].

3.3.1 Data Embedding Techniques

Embeddings transform complex data types, in our case tabular data, into a high-dimensional vector. These vectors capture important features enabling algorithms to process similarities and differences more effectively. Embeddings provide a dense vector space where geometric distance correlate with the similarity between items.
Siamese networks use embeddings to compare input pairs. Each sub-network within the Siamese network has shared weights, processing inputs independently. The output of these sub-networks come out as embeddings. They are then compared using distance metrics, Cosine-, Euclidean-, Pearson- or L1 distance to analyze its similarities. Embeddings are optimized throughout training, using a loss function, ensuring similar items are close in vector-space while dissimilar items are distant. Enhancing the networks predictive performance.

3.3.2 Loss Function

Loss functions play a crucial role in guiding models towards optimal performance. Loss funtion has a few roles, for a pair of similar items, the loss function will penalize the model if their embeddings are far apart in the embedding space. This encourages the network to learn to bring embeddings of similar items closer.
Conversely, for a pair of dissimilar items, the loss function will penalize the model if their embeddings are too close together. This encourages the network to push embeddings of dissimilar items further apart.
There are several different loss functions applicable to Siamese networks. Some common loss functions for Siamese networks are

- **Contrastive Loss**, in the context of Siamese networks considers a pair. It takes the output of the Siamese network, learning embeddings from the output, by bringing similar samples closer together and pushing dissimilar samples further apart. In other words the loss is low if positive pairs are encoded to similar depictions and negative pairs are encoded to dissimilar depictions. Its is designed to handle pairs of items.
  The formula for contrastive loss with a distance d, a label y telling if they are similar

(0) or not (1) and m deciding how far apart dissimilar pairs should be pushed, can be formulated as $L(y, d) = (1 - y)\frac{1}{2}d^2 + y\frac{1}{2}\max(0, m - d)^2$

- **Triplet Loss**, extends the idea of contrastive loss considering triplets of samples at a time. It takes in an "anchor", a positive example (same label as anchor) and a negative example (different label than the anchor). The goal of the loss functions is to decreasing the distance from the anchor to the positive example while increasing the distance between the anchor and the negative sample. The formula for triplet loss where $d(a, p)$ and $d(a, n)$ represents the distances from the anchor (a) to the positive and negative sample respectively, $L = \max(0, d(a, p) - d(a, n) + \text{margin})$ (margin is a hyperparameter defining the minimum distance difference from the anchor to the positive and the negative sample)

- **Binary Cross Entropy**, is mainly used in binary classification tasks, like the one in our study (churn/nonchurn). The cross entropy function calculates the probability of the sample being a churn sample. If the probability of a churn sample is 1, we would need its loss to be as close to zero as possible, conversely if the probability is low, the loss would need to be high. The formula for binary cross entropy is as $L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$ where L is the loss, N is the number of samples in the batch, y_i is the true label, p_i is the calculated probability. In simple terms binary cross entropy measures the difference between the true label of the sample and the predicted label.

3.3.3 Evaluation Metrics

This subsection will detail the evaluation metrics used to evaluate the performance of the Siamese network developed in the thesis. We focus on the metrics used to evaluate the Siamese network in the results section 5. Encompassing classification accuracy, the area under the receiver operating characteristics curve (ROC/AUC), the F1 score and the analysis provided by the confusion matrix. Each metric provides different insights into the models performance, granting information on its specific areas of strength and weakness. Evaluating the performance and efficacy of the model, to understand its effectiveness and limitations.

Accuracy is the most intuituve performance measure. It is simply a ratio of correctly predicted observations up against the total number of observations. In some cases identifying the TP, in our case churn, more valuable than other correct predictions. Accuracy can therefore be a misleading metric in certain instances, where the minority class is more valuable to predict correctly. Accuracy can also be deceptive in imbalanced datasets where

the model achieves high accuracy by classifying all instances as the majority class. Even so, the metric is a good benchmark and the most common evaluation metric.
$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$.
Adressing the limitation of accuracy in imbalanced datasets or in datasets where TP is more important than TN, is F1 score. The F1 score combines the models precision and recall scores. Precision is the proportion of predicted positive cases that are actually positive, while recall is the proportion of predicted positive cases that are actually positive. If the model predicted all labels as 0, the negative class, while 50% of the dataset was positive, the F1 score would be 0, while the accuracy would be 0.5. The f1 score is calculated as $F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$
The confusion matrix provides a visual summary of the models performance on a classification task. It shows the number of true and false predictions for the classes, as True Positive (TP), False Positive (FP), False Positive (FP) and False Negative (FN).

Table 2: Table explaining the acronyms

Actual Class	**Predicted Class**	**Acronym**
Positive	Positive	TP
Positive	Negative	FN
Negative	Positive	FP
Negative	Negative	TN

The confusion matrix does not only give insight into metrics such as accuracy, precision, recall and f1 score, but also the types of errors made by the model.
The last evaluation metric is the ROC curve & AUC score. The ROC curve is a visual representation of the models performance on all different classification thresholds. It plots the TP rate on the y-axis and the FP rate on the x-axis. Depicting the trade-off between these two metrics. It is desired to achieve a high TPR rate with a low FPR rate. A good results on the ROC curve would be close to the upper left corner of the graph, indicating good TPR and FPR across all thresholds. The AUC score summarizes the performance of the model across all different classification thresholds. A AUC of 1 would mean a perfect model, while an AUC of 0.5 would be equal to random guessing in a balanced binary classification problem.

4 Methodology

The methodology includes the iterative development of a Siamese network model for similarity learning within customer churn prediction. Data preparation and metrics comprised of preprocessing, pairing functions, loss function and embedding generation.

4.1 Churn Datasets

In this section the main dataset used for this study is presented. This data contains an analysis of customer attrition within a fictional telecommunication company, examining customer attrition based on a multitude of different factors. The "churn" attribute is central, which signifies whether a customer has discontinued their services in the preceding month. Other attributes include demographic information (such as gender and dependants) and financial metrics (such as monthly charges) alongside details of the service portfolio the customer is subscribed to. The dataset provides a foundation, allowing research on the intricate dynamics of customer churn in the telecommunication industry.

Data Validation and Visualization The dataset is provided by IBM and made available through the web page Kaggle. The original dataset is made up of 21 columns and 7043 different customers.

Table 3: Identification of the Customer.

Column Name	**Type**	**Description**
Customer ID	VarChar	A unique sequence of integers and characters, representing identification for each customer.

Demographic information & social status.
The following attributes encompass demographic information about the customer, as well as their social and family status. These attributes can help identifying different patterns and preferences among genders, different ages. Customers with a partner or a dependant might have different service need, preferences and usage patterns.

Table 4: Demographic Information of Customers.

Column Name	**Type**	**Description**
Gender	Boolean	Details whether the customer is female or male
SeniorCitizen	Int	Details whether the customer is a senior citizen
Partner	Boolean	Details whether the customer has a partner
Dependants	Boolean	Details whether the customer has dependents

Subscription, service and payment information.

These attributes offer insight into the customers subscription details. Array of services, payment method and other key attributes. Understanding this data is essential to identifying customer engagement levels.

Table 5: Service and Billing Details of Customers.

Column Name	**Type**	**Description**
Tenure	Int	The number of months a customer has stayed with the company
PhoneServices	Boolean	Details whether the customer has a phone service or not (Yes, No)
MultipleLines	String (Categorical)	Details whether the customer has multiple lines or not (Yes, No, No phone service)
InternetService	String (Categorical)	Details which internet service provider the customer has (DSL, Fiber optic, No)
OnlineSecurity	String (Categorical)	Details whether the customer has online security or not (Yes, No, No phone service)
OnlineBackup	String (Categorical)	Details whether the customer has online backup or not (Yes, No, No phone service)
DeviceProtection	String (Categorical)	Details whether the customer has device protection or not (Yes, No, No phone service)
TechSupport	String (Categorical)	Details whether the customer has tech support or not (Yes, No, No phone service)
StreamingTV	String (Categorical)	Details whether the customer has streaming TV or not (Yes, No, No phone service)
StreamingMovies	String (Categorical)	Details whether the customer has streaming movies or not (Yes, No, No phone service)
Contract	String (Categorical)	Details the terms of the contract with the customer (Month-to-month, One year, Two year)
PaperlessBilling	Boolean	Details whether the customer has paperless billing or not (Yes, No)
PaymentMethod	String (Categorical)	Details the payment method used by the customer (Electronic check, Mailed check, Bank transfer, Credit card)
MonthlyCharges	Float	Details the monthly amount charged to the customer
TotalCharges	Float	Details the total amount the customer has been charged

Churn status is the dependant variable in the dataset which we will conduct research on.

Table 6: Customer Churn Status.

Column Name	**Type**	**Description**
Churn	Boolean	Details whether the customer has churned or not

The original dataset is an imbalanced dataset on the dependant variable "churn". 5174 out of the 7043 samples are non-churners, where merely 1869 of the samples are categorized as churners. This is a common phenomenon when dealing with churn data. Where it is typically observed that the number of customers who remain with a business substantially exceeds those who decide to depart [39].

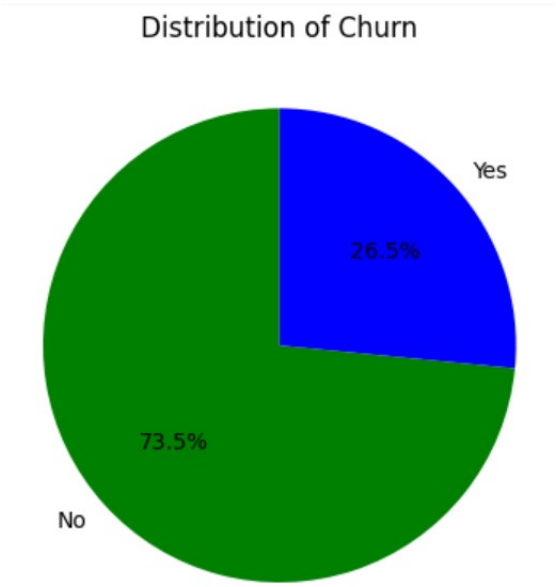


Figure 2: The distribution of the dependant variable "Churn"

The issue of class imbalance negatively affects the performance of conventional classification models. Models tend to exhibit favoritism towards the majority class. There is a risk that models assign all instances to the majority class in particularly skewed distributions, achieving a deceptively high overall accuracy while severely compromising precision for the minority class, which is often of greater interest. For example, in scenarios where the minority class makes up only 1% of the dataset, a model might attain an accuracy of 99% by indiscriminately predicting every instance as the majority class. This underscores the complexity of working with imbalanced data [39].
In churn prediction, various methodologies has been employed to deal with the challenge of imbalanced datasets. Xie et al. took in use an improved balanced random forest approach [35]. Chen et al. on the other hand, implemented random undersampling as a data preprocessing strategy for churn prediction [6]. Furthermore, SMOTE represents a valuable alternative approach, as demonstrated in the modeling work of Ali & Ariturk [40] and Wu et al. [33]. SMOTE is the solution taken in use for preprocessing in the thesis. Further discussed in section 4.2.1.

Data analysis and visualization form an important foundation of any study including data. This section outlines the techniques used to explore, understand and communicate key patterns and relations within the dataset.

Correlation Analysis Examining the correlation between the variables up against the dependant variable "Churn", will help identifying significant linear relations and potential areas to seek further insight. Prior to calculating correlations, categorical variables were transformed into numerical representations using one-hot encoding, a process that converts categories into binary columns, essential for assessing linear relationships with correlation measures. This technique is essential as standard correlation measures are designed to assess linear relationships between continuous variables [15].

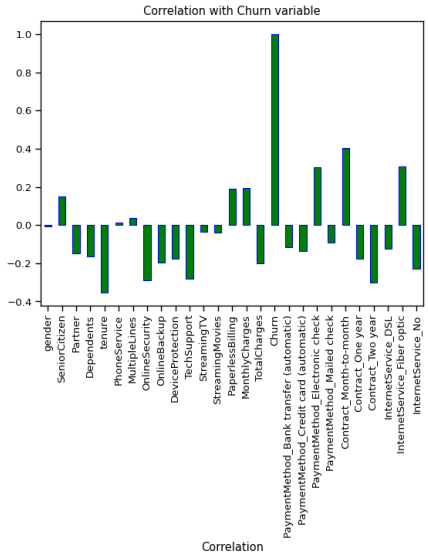


Figure 3: Correlation values of all variables up against the dependant variable "Churn"

As shown in the correlation bar chart Figure , the variables PaymentMethod_Electronic check, Contract_Month_to_month, and tenure are highlighted as potentially important predictors of the dependent variable 4.1. This suggests that these variables play a larger role in influencing the outcome of the dependant variable, warranting further insight.

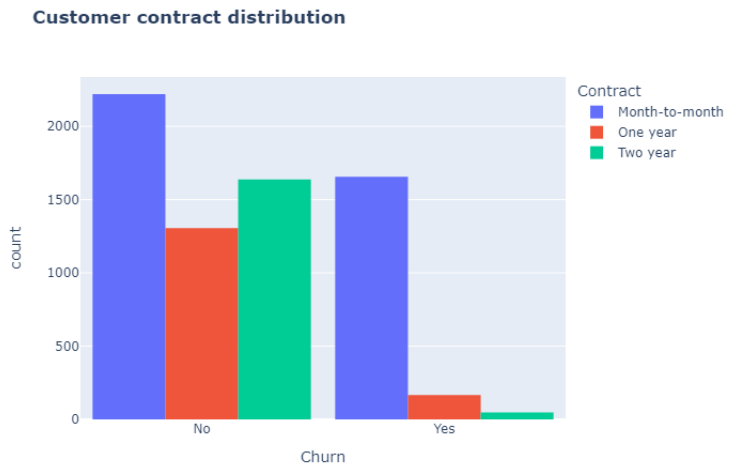


Figure 4: Customer contract distribution with regards to churn

Contract duration, by looking at the presented bar-chart Figure 4, one can see a significant discrepancy in the churn rates among customers with different contract duration's. The bar chart in Figure 4 shows that 89% of customers discontinuing services had month_to_month contracts, compared to just 9% with one-year contracts and 2% with two-year contracts. This disparity highlights the potential impact of contract length on customer retention in the telecommunications sector.

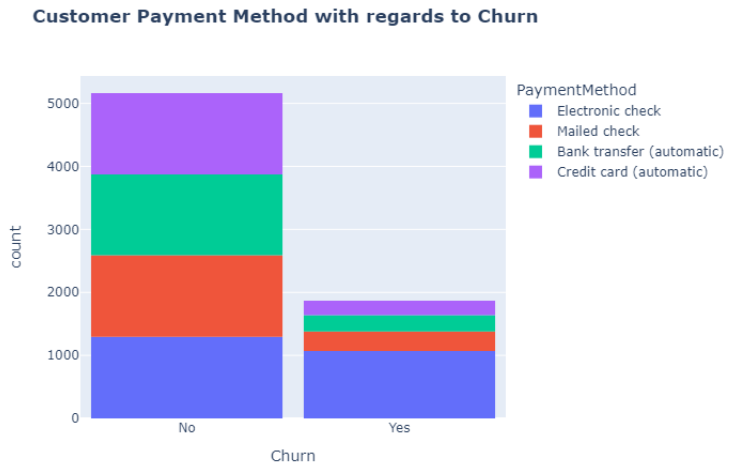


Figure 5: Customer payment method with regards to churn

Payment method, by looking further into the correlation between payment method and churn, one can see another notable trend on Figure 5. A predominant portion of

customers who had churned had chosen "Electronic check" as their method of payment (57%). On the other hand, customers who had opted for alternative payment methods, such as "Credit-Card Automatic Transfer" (12%), "Bank Automatic Transfer" (14%) or "Mailed Check" (17%) demonstrated a notable lower probability to discontinue their services. Similarly to customers who had month-to-month contract, there is a correlation between payment method and churn.

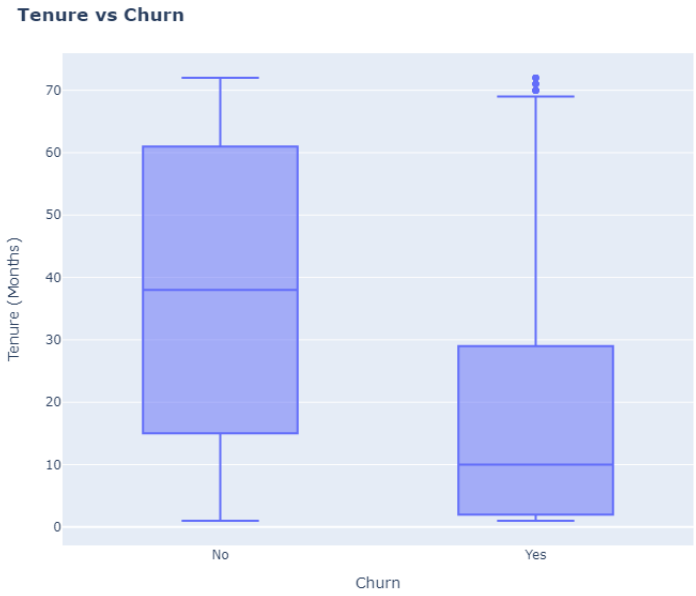


Figure 6: Box plot visualising tenure with regards to churn

Long-term and short-term customers, newly acquired customers have a higher probability for churn as presented by Figure 6. This trend suggests individuals who have recently commenced their relationship with the service provider are more inclined to discontinue their services in comparison to longer-standing customers. Such patterns underscores the importance of understanding the expectations of new customers to reduce churn rate.

These findings confirm the insights derived from the correlation plot 4.1, underscoring the importance of contract length, payment method and tenure as key predictors of churn. Guiding the embedding of the network, contract length and payment method will be represented as individual binary features using one-hot encoding. This approach is done so that the model retains maximum information from these categorical variables, as their importance has been emphasized in the plots 4 & 5. The encoding process is further explained in the following section 4.2.1.

Orange dataset To enhance the validity and generalisation of the findings, a separate dataset was used to serve as a validation dataset. This helps against the risk of overfitting to a specific dataset in the study, demonstrating the models performance is not dataset specific.

The Orange Telecom dataset was selected due to its extensive use in prior telecom churn studies, as shown in table 1. The dataset serves as a great benchmark for validating predictive models in churn analysis. This benchmark provides meaningful comparisons of the developed models performance up against previous models.

The Orange Telecom Churn dataset, is compromised of cleaned customer data, and a "churn" label indicating whether a customer has terminated their subscription. The dataset contains 21 features of various attributes and behaviours of the customer. It is made up of 3333 samples, class imbalance is evident in the dataset as in most churn datasets, with 483 churned customers and 2850 non-churned customers.

Table 7: Features describing customer behavior and their attributes.

Column Name	**Type**	**Description**
State	String (Categorical)	The US state where the customer resides
Account length	Int	The number of days the customer has been with the company
Area code	Int	The area code of the customer's phone number
Phone number	String	The phone number of the customer
International plan	String (Categorical)	Whether the customer has an international plan or not (Yes, No)
Voice mail plan	String (Categorical)	Whether the customer has a voice mail plan or not (Yes, No)
Number vmail messages	Int	The number of voice mail messages a customer has
Total day minutes	Float	Total number of minutes spent on day calls
Total day calls	Int	Total number of day calls
Total day charge	Float	Total charges for day calls
Total eve minutes	Float	Total number of minutes spent on evening calls
Total eve calls	Int	Total number of evening calls
Total eve charge	Float	Total charges for evening calls
Total night minutes	Float	Total number of minutes spent on night calls
Total night calls	Int	Total number of night calls
Total night charge	Float	Total charges for night calls
Total intl minutes	Float	Total number of minutes spent on international calls
Total intl calls	Int	Total number of international calls
Total intl charge	Float	Total charges for international calls
Customer service calls	Int	The number of customer service calls made

Table 8: Dependent variable.

Column Name	**Type**	**Description**
Churn	Boolean	Whether the customer has churned or not (True, False)

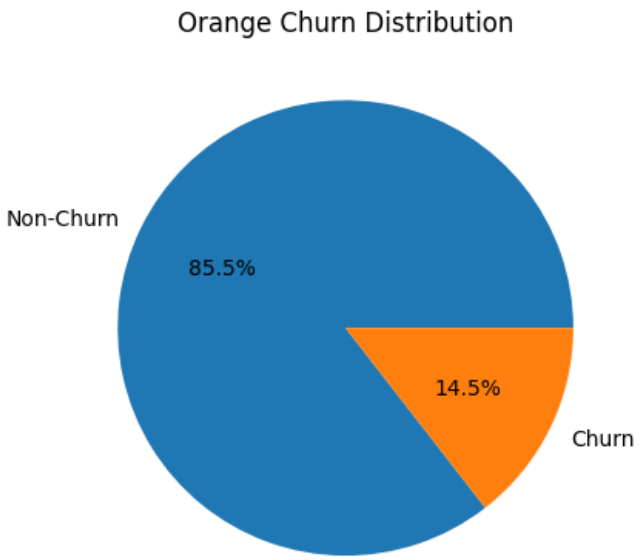


Figure 7: The distribution of the dependant variable "Churn" in the Orange dataset

The Orange dataset is also has a significant class imbalance, as visualized in the pie chart 7. The minority "churn" class represents 14.5% of the dataset, while "non-churn" represents the remaining 85.5%. This skewed distribution could cause implications, as many traditional classification algorithms struggle on minority classes. The Orange dataset has 12% fewer churn cases compared to the IBM dataset, while SMOTE is applied on both datasets the proportion of synthetic sampled generated for the Orange dataset could cause additional difficulties.

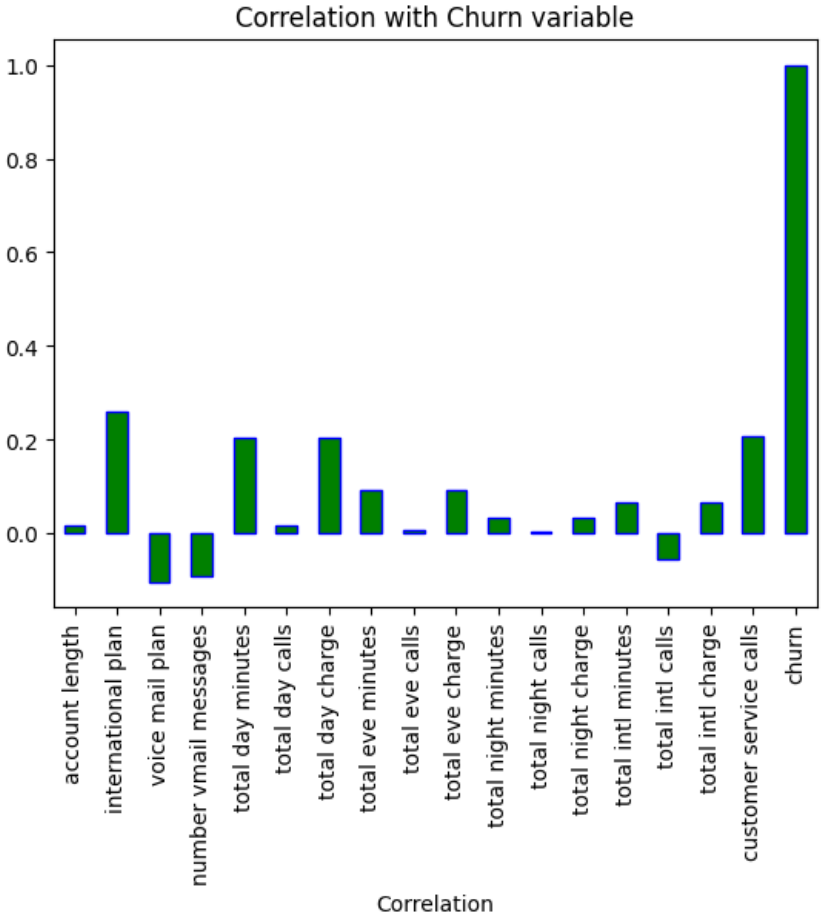


Figure 8: Correlation values of all variables up against the dependant variable "Churn" in the Orange dataset

The correlation plot highlights "international plan", "total day minutes" and "customer service calls" as the features exhibiting strong correlation with our dependant variable. The correlation with "customer service calls" is an interesting feature correlation, as it potentially suggests a link between customer dissatisfaction and their likelihood of churn.

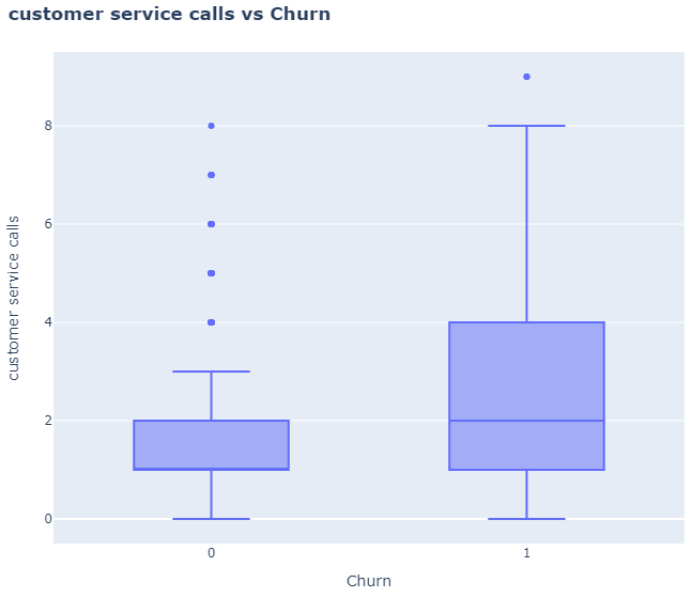


Figure 9: Distribution of customer service calls by churn status. Showcasing the association between frequent customer service calls and customer churn.

The box plot analysis in Figure 9 reveals a potential link between increased customer service interactions. Churning customer have a higher median number of customer service calls. This suggests dissatisfaction with service experiences could be a churn factor in the Orange dataset.

Having explored the characteristics, correlations and class imbalance within the Orange and IBM datasets, its crucial to pre-process and prepare the data for the model. Transforming raw data into useful information.

4.2 Research design

This research design section delves into the key components of the developed Siamese network model. It will cover the strategies used to prepare and pair input data, the architecture of the embedding generation sub-networks, the chosen loss function for optimization, and the overall integration of these elements within the Siamese framework.

4.2.1 Data Pre-processing

Before the development of the prediction model, extensive data pre-processing steps were performed to ensure data quality and making it suitable for the Siamese networks architecture. The aim is to enhance the data quality and thus making it more suitable for

the Siamese network. The inputs into the model are pre-processed by dealing with imbalance in the dataset, categorical data, scaling the data and pairing. The pre processing methodology outlined below adresses these challenges to improve the predictive power of the Siamese network.

Many machine learning models perform better when working with categorical data that has been transformed into numerical format which is more manageable for machine learning models. Two different encoding strategies were taken in use, label encoding and one-hot encoding.
Label encoding was used to address boolean data types in the datasets. The boolean columns were transformed into binary format to be more readable for the network. One-hot encoding was used for the categorical columns with more than two levels. One-hot encoding opposite to label encoder creates a new binary feature for each category in the original column, increasing the dimension of the dataset. This process ensured that there is no unintended relations between the categorical data, preventing misleading interpretation by the model.
Numerical features were standardized using StandardScaler from sklearn. The method fits to the data calculating the mean for each feature in the dataset with $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$. where n is the number of samples and x_i are the feature values. further on the method calculates the standard deviation for each feature using, $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$. These values are then stored within the object. To standardize the features, each feature value x_i is subtracted by the mean. The resulting value is then divided by the standard deviation, as shown $z_i = \frac{(x_i - \mu)}{\sigma}$. This ensures the columns have a mean of zero and a standard deviation of one, preventing biases to feature magnitude. Standardizing is important to models which are sensitive to the scale of input features, such as Siamese networks, so that all features contribute equally. The data transformer methods label encoder, one-hot encoding and StandardScaler were taken from sklearn.preprocessing [27].

It is typical for churn datasets to have an imbalanced dataset, as the number of churned customers is significantly lower than non-churned customers. Many machine learning models can struggle with imbalanced datasets, to handle this challenge Synthetic Minority Over-sampling Technique (SMOTE) from imblearn was employed [20]. SMOTE increases the representation of the minority class in a dataset by generating synthetic samples. It identifies the k nearest neighbours of the minority class, selects one, and creates a new sample along the line connecting them. The difference in feature values are multiplied by a number between 0 and 1, creating a new sample.

SMOTE balances the dataset, creating new synthetic samples rather than duplicating existing ones. Thus ensuring the model does not overfit to the majority class, improving its ability to generalize to new unseen data.

To further address the dataset imbalance, as SMOTE samples might have an effect on the Siamese network, a stratified splitting strategy was used. This aims to maintain the distribution of target classes across training, testing and validation sets. The train_test_split function from sklearn was used with the stratify = y parameter to ensure proportional representation within all the different subsets.
The full dataset was divided into a 70/15/15 split yielding a 70% training set, a 15% validation set and a 15% test set.
This approach provides sufficient data for model training, allows hyperparameter tuning and model selection based on the validation set, and offers a final performance evaluation on unseen data through the independent test set.
After the dataset had split into train, test and validation sets, the data had to be properly processed to fit as input into the Siamese network. As described in section 3.3, Siamese Networks operate on paired inputs to determine whether the pairs are similar or dissimilar. It therefore requires the input to be passed as pairs. A pairing function was used to generate the pairs, consisting of both similar and dissimilar instances. The specific methodology behind this pairing function is elaborated further in the subsequent section.

4.2.2 Siamese model pairing

Pairing functions are important in Siamese network architectures. They determine the selection of input data pairs, which are input into the networks twin branches. They play a critical role in teaching the network to distinguish similar and dissimilar data, allowing it to develop a meaningful similarity metric [5]. The paired input approach transforms the imbalance in the dataset into a validation problem. Since the model input will be negative and positive pairs, the problem of data imbalance is eliminated even if the classes themselves are imbalanced [37].

In the development of the Siamese network used in this thesis, two different pairing functions were implemented. The first pairing function, referred to as random_pairing, was designed to create pairs for the input data without increasing its the number of samples. The function takes a dataframe and a label series as input, along with a N parameter. The N parameter represents the number of pairs to be generated for each sample. To maintain the original dataset size and simplify the process, N has a default value of 1.

Algorithm 1 Random_pairing.
Input: dataframe, label series, N = 1.
Output: left input, right input, targets.

Ensure: *label series* = *panda series*
1: *similar indices* = 1
2: *dissimilar indices* ≠ 1
3: **for** index, row in dataframe **do**
4: **if** $N = 1$ **then**
5: **if** label at index is 1 **then**
6: **while** *Similar* ≠ *index* **do**
7: Repeat
8: *similar* ← *random*(*similar indices*)
9: **end while**
10: left input ← row
11: right input ← similar row
12: targets ← 1
13: **else if** label at index ≠ 1 **then**
14: **while** *Dissimilar* ≠ *index* **do**
15: Repeat
16: *dissimilar* ← *random*(*dissimilar indices*)
17: **end while**
18: left input ← row
19: right input ← dissimilar row
20: targets ← 0
21: **end if**
22: **end if**
23: **end for**

The algorithm operates by iterating over the dataset, randomly pairing data points from the same class (similar) or from different classes (dissimilar), based on the associated labels. These pairs are then used to train the Siamese network, distinguishing similar and dissimilar pairs. The pseudo-code provided outlines the functions steps 23.

The second pairing function, referred to as balanced_pairing, creates a balanced dataset of an equal amount of similar and dissimilar pairs for each sample. This function accepts the same inputs, but N tells how many pairs the algorithm should make, the default being 4.

Algorithm 2 Balanced_pairing
Input: dataframe, label series, N = 4.
Output: left input, right input, targets.

Ensure: label series is a pandas series with churn status

```
if N is odd then
    N ← N + 1                                   ▷ Ensure N is even for balance
end if
num similar ← N/2
num dissimilar ← N/2
for index, row in dataframe do
    for i from 1 to num similar do
        random index ← random(same label)
        left input ← row
        right input ← random index
        targets ← 1
    end for
    for i from 1 to num dissimilar do
        random index ← random(different label)
        left input ← row
        right input ← random index
        targets ← 0
    end for
end for
```

Unlike the random_pairing function. The balanced_pairing function generates $N/2$ similar and $N/2$ dissimilar pairs for each sample, ensuring the training process has an equal amount of similar dissimilar cases. If N is passed as an odd number, the function automatically adjusts it to the next even number to ensure a balanced training process. The detailed steps of the algorithm is described in the pseudo-code 19.

Throughout testing random_pairing outperformed balanced_pairings accuracy by a margin of 10%. This result was unexpected, considering the balanced nature of input pairs created by balanced_pairing, which was expected to provide better generalization. There are some factors which may contribute to this discrepancy, such as dataset imbalance and pairing value.

As the dataset has a class imbalance presented in section 4.2.1. Since balanced_pairing creates multiple pairs, this could lead to infrequent representation of the minority class, hindering the networks ability to learn from discriminative features. Even though SMOTE was used on the dataset, since SMOTE introduces synthetic minority examples. Synthetic samples might not perfectly replicate real world instances, making balanced pairing less beneficial, as the model might overfit on the synthetic examples.

Pairing value could also be a factor in this discrepancy, while an increased number of pairs could be beneficial, it could introduce less informative pairs or even noisy pairs. Such pairs risk compromising the model's ability to learn meaningful representations.

4.2.3 Embedding

Embeddings are a core aspect of Siamese networks. This subsubsection delves into the embedding process and its role in capturing the essence of the input data. At the heart of the Siamese network, sub-networks transform tabular data into low-dimensional vector representations known as embeddings. The network comprises multiple dense layers with ReLU and ELU activations. ReLU returns the element-wise maximum of 0 and the input tensor, $\max(x, 0)$. ELU returns x for inputs greater than 0 and $e^x - 1$ otherwise. Batch-normalization layers follow some dense layers, normalizing the outputs to stabilize learning. A dense layer includes L2 kernel regularization to prevent overfitting. The final layer, using sigmoid activation, outputs a 32-dimensional vector as the embedding for each input. This 32-dimensioned output retains enough information to distinguish between similar and dissimilar input pairs. These embeddings are then compared using the L1 distance, with distances near 0 indicating dissimilarity and those near 1 indicating similarity.

4.2.4 Loss Function

Our Siamese network employs the binary-cross entropy loss function, suitable for models that produce a single probabilistic output between 0 and 1, indicative of the similarity between inputs. This loss function measures the discrepancy between the predicted similarity, derived from the distance between embeddings, and the actual similarity in the training data. It penalizes the model when embeddings of similar items are distant or when those of dissimilar items are close. During training, this loss is back-propagated through the network, updating weights and parameters to minimize it, thereby enhancing the ability of the network to capture relevant similarities and differences.

4.2.5 Overall Model

In this section the model will be presented showcasing the result of tuning the embedding, loss and preprocessing. The final model employed in the study is a Siamese Neural network, designed for similarity learning, as showcased in figure 10. The models interior is compromised of two identical sub-networks, made up of sequential stack of layers.

- Input layer, 26-dimensioned preprocessed tabular data is passed onto the model as input (in the context of the IBM dataset).
- Hidden layers, multiple dense layers process the data, with both ReLU and ELU activation functions.
- Batchnormalization, normalization layers are implemented to accelerate training and improve the model.
- The final dense layer using sigmoid activation compresseses learned information into a 32-dimensional embedding vector.

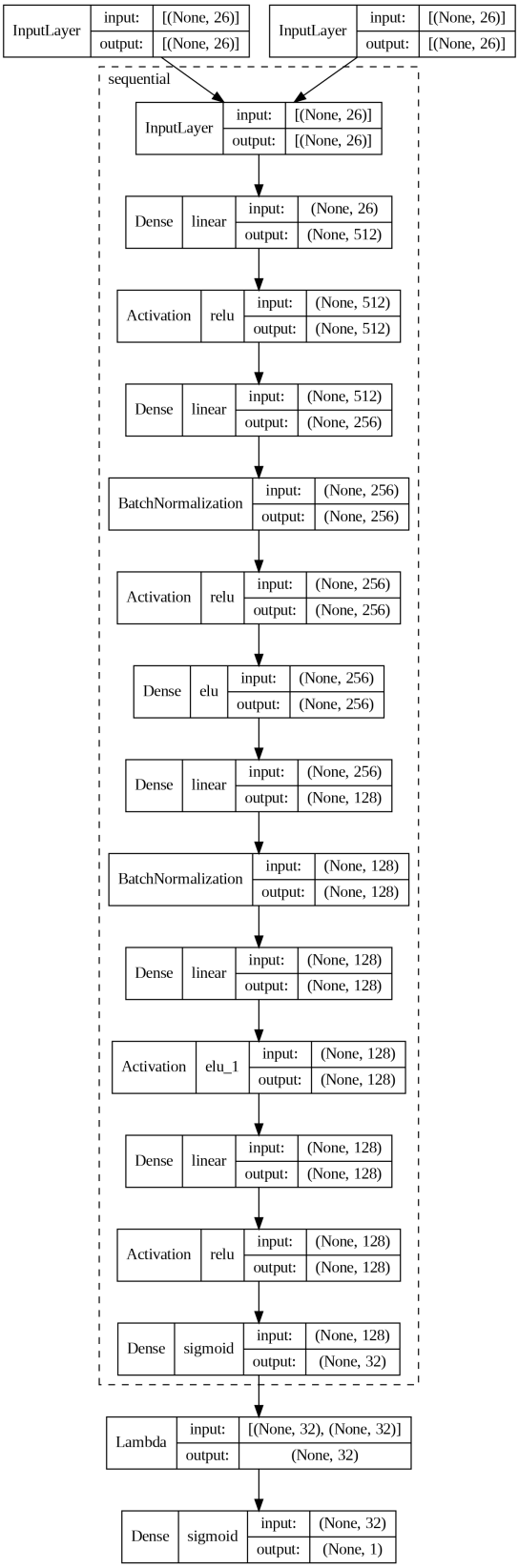


Figure 10: Model Design showcasing layers in one ”leg” of the Siamese network, the input layer, dense layers, activation function layers and Batchnormalization layers.

The Siamese network independently processes a pair of input data through the sub-networks. The embeddings provided by the sub-networks are compared using L1 distance, the absolute distance between points in all dimensions. A smaller L1 distance in the embedding space indicates greater similarity, which means a higher probability of the input pair being considered similar by the model, and vice versa.
The model is trained using Adam optimizer, using a exponentially decaying learning rate. Its robustness is beneficial for Siamese Network training. Adams efficiency in similar contexts has been documented [25] [3] supporting the choice of its use. Binary cross entropy serves as the loss function, a common choice in Siamese network research [25] [23] [24]. The loss is minimized by the network through iterations, improving the embeddings captured, showcasing the similarity and dissimilarity within the tabular dataset.

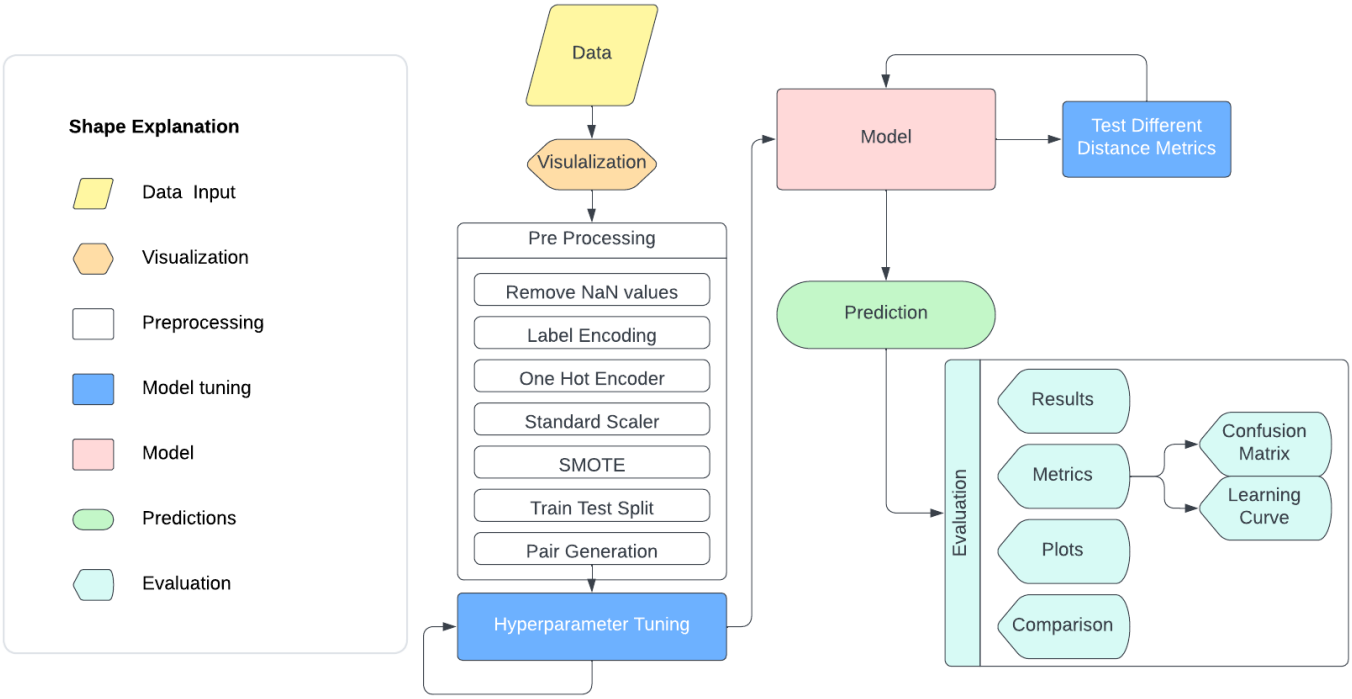


Figure 11: Flowchart showcasing the model development process with shape description

Figure 11 presents a visual depiction of the iterative workflow used for the development and evaluation of the Siamese network. The process starts with importing data, which is then visualized and analyzed in the visualization step to get an insight into the dataset. After visualization in depth pre-processing was performed to ensure it suitability for the Siamese network. The model was then tuned iteratively on the pre-processed data to find the best hyperparameters and metrics. After tuning the model, finding ideal hyperparameters and metrics, prediction and model evaluation commenced. The designed architecture

enables the Siamese network to effectively quantify the similarity between pairs of tabular data, directly addressing the core objectives of this research.

5 Results

This section presents the key findings from the evaluation of our Siamese Network. Initially, we assess the performance impact of various distance metrics. Following this, The outline of the hyper-parameter tuning process and its contribution to the optimization of the network. The final model's performance is then presented through accuracy measures, ROC curves, and confusion matrices. Further on comparative performance against other baseline models in regular as well as few-shot learning environments is performed.
The results display how the Siamese Network outperforms the other baseline models in the realm of few-shot learning. Not struggling with the challenges of class imbalance or few training samples as other models in customer churn prediction. They focus on learning relationship between data pairs, rather than needing large, labeled datasets to learn complex patterns. Giving an insight on how Siamese networks can reduce the need for large datasets, achieving good performance with smaller datasets in customer churn prediction.

The evaluation process will begin using the complete dataset described in section 4.1. The dataset being split into training, test and validation, with 70% of the dataset being allocated as training data, 15% for validation data and 15% for test data. The models in the first two subsubsections 5.1 & 5.2, are trained on the test data, with the help of validation data. Then tested on test data to get an insight on its generalizability on unseen data. Further in section 5.3 the models are trained with different few-shot learning subsets before being tested on the test data.

5.1 Siamese Model

Through the optimization of the model described in section 4.2.5, several different distance metrics were tested, to see which the Siamese model performed the best with. The distance metrics used in the thesis were:

- L1 distance
- Cosine distance
- Pearson distance
- Euclidean distance

Distance metric analysis showcased that L1 distance, also known as Manhattan distance, had the most effective performance for the Siamese model in this telecom churn prediction task. This was evident by the validation learning curves where L1 had the best validation

accuracy and lower final validation loss. While all of the models with different distance metrics exhibited some degree of overfitting in their learning curves, the L1 distance model's overfitting was not as strong and had better accuracy and F1 score.

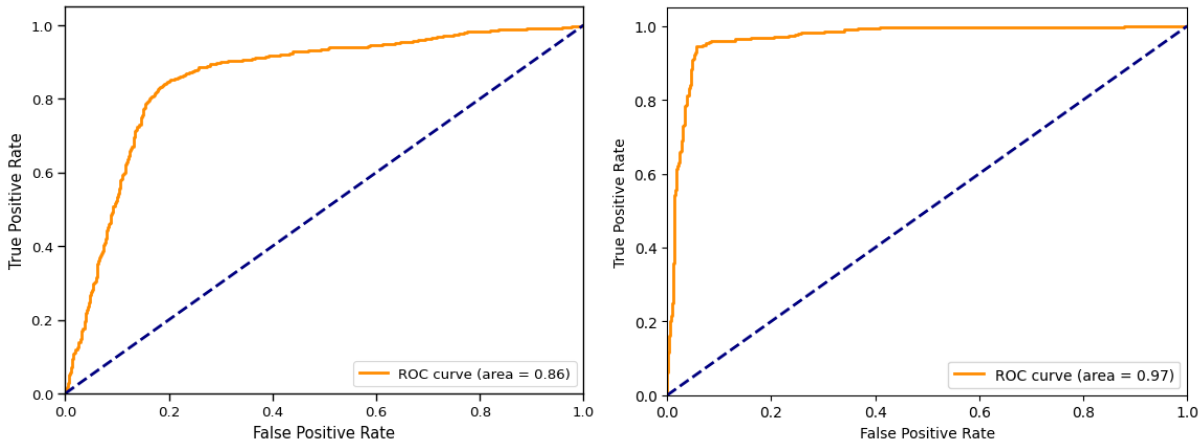
Table 9: Accuracy on the entire test data for the IBM dataset, for different distance metrics on the Siamese network described in section 4.2.5.

Distance Metric	**Accuracy**
L1 Distance	83.1%
Cosine Distance	79.9%
Pearson Distance	68.1%
Euclidean Distance	80.7%

The superiority of the L1 distance is further corroborated by its accuracy on unseen test data, once again having the best performance. As seen in table 9.

After finding of the ideal distance metric, an iterative hyperparameter tuning process was applied to the IBM dataset. Hyperparameter tuning is pivotal when tuning a machine learning model to achieve the best possible results within the given dataset.
Keras Tuner package from Keras was used [26]. Different amount of nodes for each layer was attempted, in the range of 2^3 to 2^9. Inclusion and exclusion of regularisation and dropout. Random Search was performed several times to find the ideal hyper parameters. After the search was finished manual editing of the model was performed to attempt to increase the accuracy even further. The final results of the hyperparameter process is shown in figure 10.

After completing the model, a series of evaluation tests were performed on the model to asses its efficacy. To get an insight on its inner workings and performance.



(a) ROC for IBM dataset

(b) ROC for orange dataset

Figure 12: The Siamese networks ROC curve for both IBM and Orange, on the full test datasets

Table 10: Performance (Accuracy and F1) of the Siamese network, on test data for both Orange and IBM datasets, having trained on respective training datasets.

Model	**Data**	**Accuracy**	**F1**	**Precision**	**Recall**	**ROC AUC**
Siamese Network	IBM dataset	83.64%	83.65%	83.68%	84.64%	82.61%
	Orange dataset	94.40%	94.45%	94.66%	94.40%	94.75%

The performance matrices for the Siamese network on the two different datasets exhibits the model performed significantly better on the Orange dataset. This suggests potential differences in difficulty or complexity between the datasets.

The Siamese network demonstrated a strong accuracy of 94.40% on the Orange dataset. The result outperforms benchmarks from comparable studies seen in table 1, such as Jain et al. [16], Hassonah et al. [13] and Ullah et al. [30], further discussed in the next chapter.

The metrics accuracy, f1, precision, recall and ROC AUC are all in very close proximity for the Siamese network. Usually there is a higher discrepancy between these values in a unbalanced dataset. In our case we use SMOTE to balance the datasets before predicting, explaining the results.

Confusion matrices provide a visual representation of the type of error made by the classification model. The matrix showcases instances of true positive, true negative, false positive and false negative, providing a view of where and how often the model correctly or incorrectly classifies classes. It helps highlighting where the model has trouble distinguishing

between the classes. Label 1 represents churn, while 0 represents nonchurn.

Table 11: Confusion Matrix on the IBM test set

		Predicted Labels	
		0	1
Actual Labels	0	1369	180
	1	160	589

Table 12: Confusion Matrix on the Orange test set

		Predicted Labels	
		0	1
Actual Labels	0	801	54
	1	18	413

160 & 18 false negative (FN) cases in each respective dataset. 180 & 54 false positive (FP) cases in each respective dataset. In both datasets the model seemed to have more FP (False Positive) predictions than FN (False Negative) predictions. Indicating it had more difficulty with identifying class 1, which is to be expected considering the imbalanced dataset. While SMOTE was applied to deal with class imbalance, the dataset retains a higher proportion of genuine non-churn samples. This provides the model with more robust, real-world data to learn from. Potentially affecting its ability to generalize to new churn instances when using synthetic samples.

5.2 Other models

In order to thoroughly asses the performance of the Siamese network, a comparative analysis was done against established models. These models serve as baseline models to compare its performance up against. The models selected include both linear and non-linear algorithms. SVM [7] with both linear and RBF kernels, Random forest [14], Logistic Regression [8] and Bernoulli Naive Bayes [4], providing a robust line up for evaluating the Siamese network.

Table 13: Baseline models, with their parameters, accuracy and F1 score on the test datasets for both the Orange and IBM dataset, having trained on the training sets.

Model	IBM Accuracy	IBM F1	Orange Accuracy	Orange F1
SVM Linear $C = 1$ $kernel = "linear"$ $degree = 3$ $gamma = "scale"$ $coef0 = 0.0$ $shrinking = True$ $probability = False$	79.08%	79.07%	75.09%	75.09%
SVM RBF $C = 1$ $kernel = "RBF"$ $degree = 3$ $gamma = "scale"$ $coef0 = 0.0$ $shrinking = True$ $probability = False$	80.76%	80.76%	90.76%	90.74%
Random Forest $n\_estimators = 100$ $criterion = "gini"$ $max\_depth = None$ $min\_samples\_split = 2$ $min\_samples\_leaf = 1$ $min\_weight\_fraction = 0.0$ $max\_features = "sqrt"$	85.02%	85.02%	95.91%	95.91%
Logistic Regression $max\_iter = 1000$ $penalty = "l2"$ $dual = False$ $tol = 1e^{-4}$ $C = 1.0$ $fit\_intercept = True$ $intercept\_scaling = 1$	78.83%	78.82%	74.27%	74.25%
Bernoulli Naive Bayes $alpha = 1.0$ $binarize = 0.0$ $fit\_prior = True$ $class\_prior = None$	76.50%	76.48%	61.64%	61.62%

On the IBM dataset, the models performed relatively similar within 10 percentage point range of accuracy. Only the Random Forest model outperformed the Siamese Network. In the Orange dataset there was a bigger discrepancy between the baseline models, presenting a greater challenge for certain models. Specifically, SVM linear, Logistic Regression & Bernoulli Naive Bayes had trouble identifying predictive patterns, while Random Forest again surpassed the Siamese networks accuracy. SVM RBF did not quite reach the performance of the Siamese network, as seen in table 10, or the Random Forest model,

but still performed well on the Orange dataset. The parameters described in the "Model" column of the table 13 are used further in the sections following.

5.3 Few-shot Learning

Siamese Networks are usually employed on image data. In these scenarios Siamese Networks perform exceptionally well in few shot learning scenarios [32] [21] [38]. To investigate if this also is the case for tabular data, the same formula has been applied. The datasets were divided into varying class ratios (Churn, Non-churn), ranging from (5, 25) to (3100, 3100) for the IBM dataset & (5, 25) to (1900, 1900) for the Orange dataset. This will determine whether the Siamese Networks success translates to the domain of tabular data. The Models were put up against each other in these different Few-shot learning scenarios.

5.3.1 Few-shot Learning for IBM dataset

In figure 13 and figure 15 the performance of the different models for the different sample sizes are shown through a line graph, the same performance are also shown in numerical format in the tables 14 & 15 with the respective sample size on the left. The comparison will give an insight on the effectiveness of Siamese networks in improving the accuracy and efficiency of few shot learning models when applied to customer churn prediction.

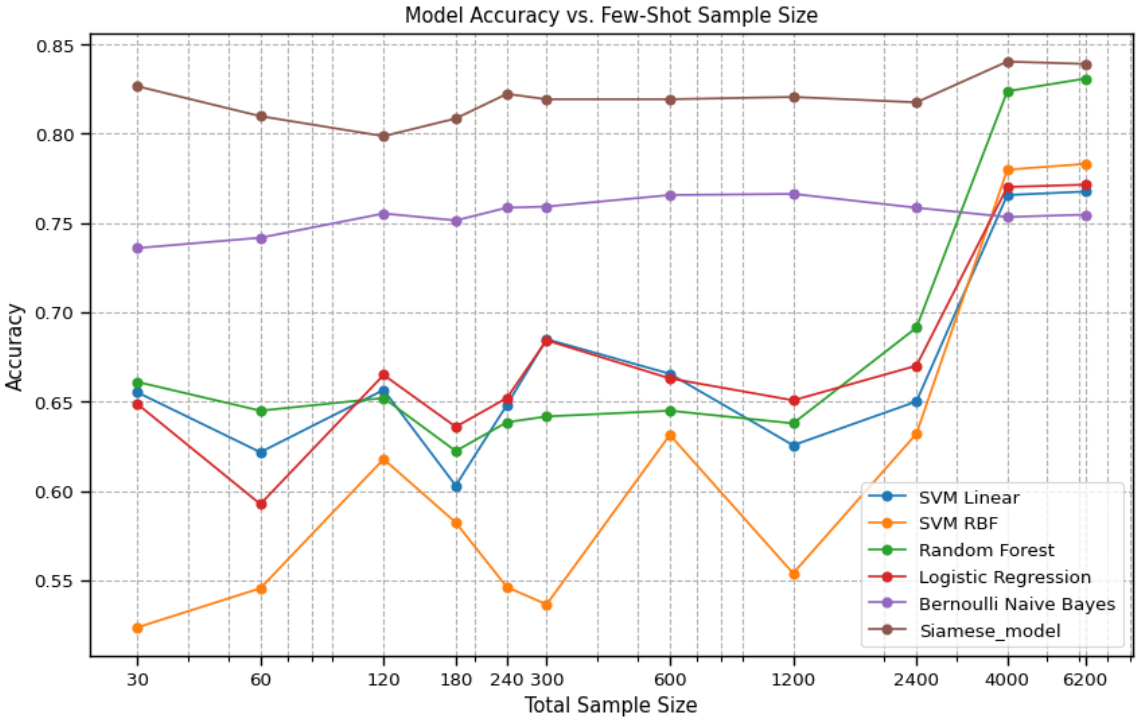


Figure 13: Prediction accuracy, y-axis, of all models on the IBM dataset up against different sample sizes on the x-axis, the divide between churn and nonchurn cases for the sample sizes can be seen in the subsequent table.

Table 14: Model accuracies across different sample sizes on the dataset, sample sizes on the left, with the best result for each sample size across the models in green and the worst in red.

Sample Size	SVM Linear	SVM RBF	RF	LR	Bernoulli NB	Siamese model
(5, 25)	0.567	0.500	0.572	0.546	0.719	0.824
(10, 50)	0.591	0.500	0.542	0.571	0.732	0.797
(20, 100)	0.692	0.582	0.662	0.666	0.755	0.788
(30, 150)	0.602	0.562	0.602	0.633	0.755	0.802
(40, 200)	0.668	0.581	0.638	0.660	0.760	0.822
(50, 250)	0.648	0.631	0.644	0.660	0.759	0.804
(100, 500)	0.600	0.554	0.648	0.640	0.766	0.809
(200, 1000)	0.661	0.643	0.660	0.661	0.759	0.810
(1000, 1000)	0.672	0.662	0.702	0.676	0.763	0.814
(1500, 1500)	0.764	0.779	0.817	0.771	0.755	0.840
(1900, 1900)	0.765	0.783	0.828	0.772	0.754	0.840

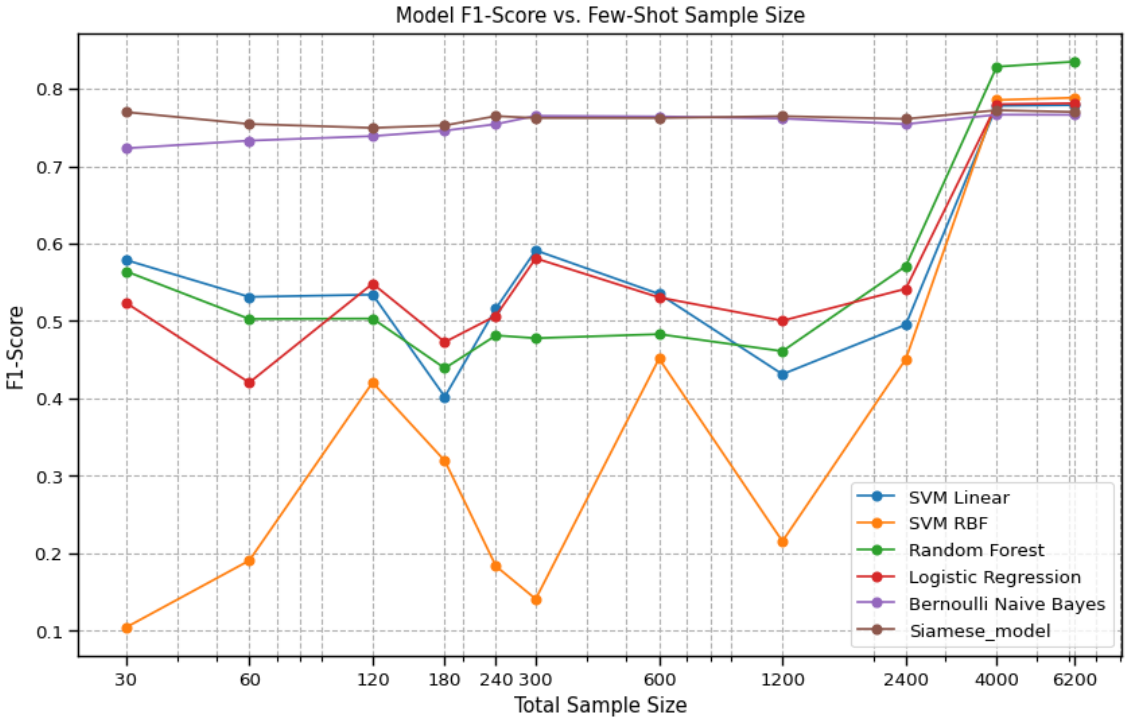


Figure 14: F1 score, y-axis, of all models on the IBM dataset up against different sample sizes on the x-axis, the divide between churn and nonchurn cases for the sample sizes can be seen in the subsequent table.

Table 15: Model f1 scores across different sample sizes on the IBM dataset, sample sizes on the left, with the best result for each sample size across the models in green and the worst in red.

Sample Size	SVM Linear	SVM RBF	RF	LR	Bernoulli NB	Siamese model
(5, 25)	0.579	0.104	0.564	0.524	0.723	0.770
(10, 50)	0.531	0.191	0.503	0.421	0.733	0.754
(20, 100)	0.534	0.421	0.503	0.548	0.739	0.749
(30, 150)	0.402	0.320	0.439	0.473	0.746	0.753
(40, 200)	0.516	0.184	0.481	0.507	0.754	0.765
(50, 250)	0.591	0.141	0.478	0.581	0.765	0.762
(100, 500)	0.535	0.451	0.483	0.531	0.764	0.762
(200, 1000)	0.431	0.216	0.461	0.500	0.762	0.765
(1000, 1000)	0.495	0.451	0.571	0.542	0.754	0.761
(1500, 1500)	0.778	0.786	0.828	0.780	0.767	0.772
(1900, 1900)	0.779	0.788	0.835	0.781	0.766	0.770

Table 14 and figure 13 presents the performance of the Siamese model alongside the baseline models and their parameters described in table 13.

Examining the figure 13, showcases the Siamese network significantly outperforms the

other models in few-shot learning scenarios. Notably, even at a dataset size of 2400 (400, 2000) samples, the baseline models do not reach the efficacy shown by the Siamese network. However, when the dataset is balanced, with an equal amount of churn and non-churn cases, the baseline models experience a performance improvement. The Siamese network and Bernoulli Naive Bayes remains unaffected by this change in class balance, exhibiting their consistency across varying sample sizes.
Further, figure 13 also shows that the performance of Bernoulli Naive Bayes model closely mirrors the Siamese network. Suggesting similarities in their handling of data sparsity and class distribution, only on a lower performance level.
The Siamese network maintains the best prediction accuracy across all of the different sample sizes as presented in table 14, shown by its accuracy's being filled in green. However in the last two sample sizes (2000, 2000) & (3100, 3100), random forest showcases similar efficacy only slightly falling short. This could indicate its potential advantage in scenarios with a lot of balanced data.

Table 15 and Figure 14 presents the F1 scores of the evaluated models, highlighting their ability to accurately identify churn cases. The Siamese model outperforms the other models across most few-shot learning scenarios, with the exception of the (50, 250) sample size where the Naive Bayes Bernoulli model displays a slightly better performance. This close competition between the Siamese network and Bernoulli Naive Bayes model aligns with their similar accuracy trend in Figure 13. However, the Siamese model showcases comparatively lower performance with the balanced sample sizes (2000, 2000) & (3100, 3100).

5.3.2 Few-shot Learning for Orange dataset

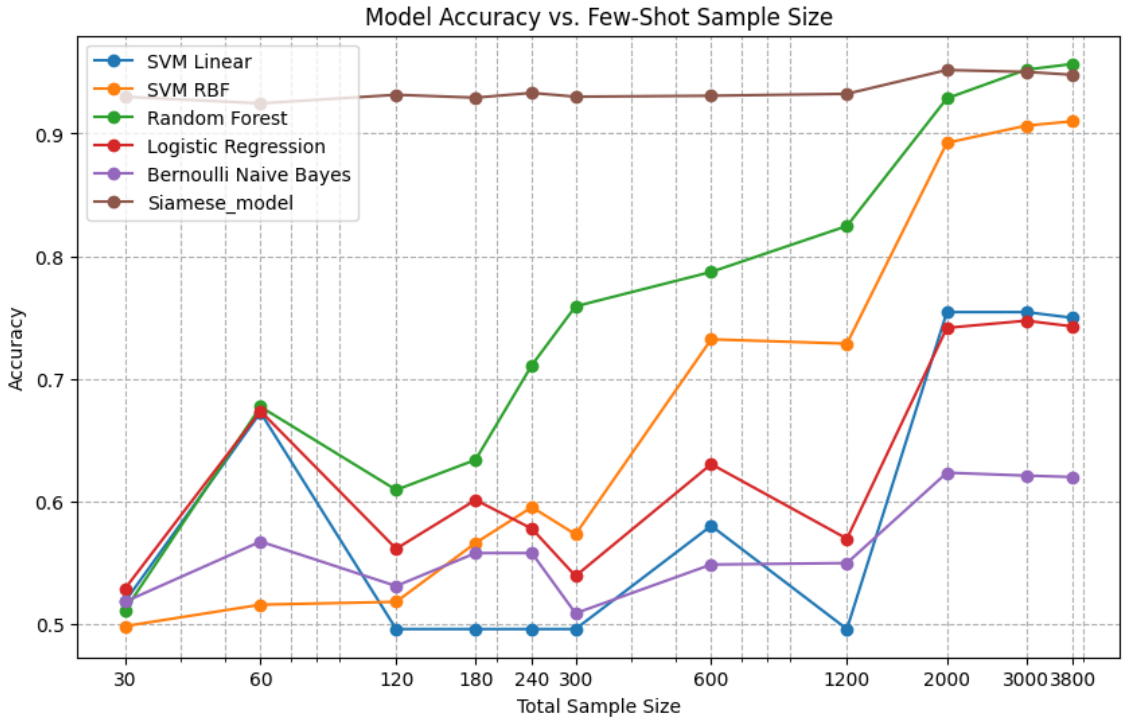


Figure 15: Prediction accuracy, y-axis, of all models on the Orange dataset up against different sample sizes in the x-axis, the divide between churn and nonchurn cases for the sample sizes can be seen in the subsequent table.

Table 16: Model accuracy's across different sample sizes on the Orange dataset, sample sizes on the left, with the best result for each sample size across the models in green and the worst in red.

Sample Size	SVM Linear	SVM RBF	RF	LR	Bernoulli NB	Siamese model
(5, 25)	0.518	0.498	0.511	0.529	0.518	0.930
(10, 50)	0.673	0.516	0.677	0.674	0.567	0.925
(20, 100)	0.496	0.518	0.609	0.561	0.531	0.932
(30, 150)	0.496	0.566	0.634	0.601	0.558	0.929
(40, 200)	0.496	0.595	0.711	0.578	0.558	0.933
(50, 250)	0.496	0.573	0.759	0.539	0.509	0.930
(100, 500)	0.580	0.732	0.787	0.630	0.549	0.931
(200, 1000)	0.496	0.729	0.825	0.570	0.550	0.932
(1000, 1000)	0.754	0.892	0.929	0.742	0.623	0.952
(1500, 1500)	0.754	0.906	0.952	0.747	0.621	0.950
(1900, 1900)	0.750	0.910	0.957	0.743	0.620	0.948

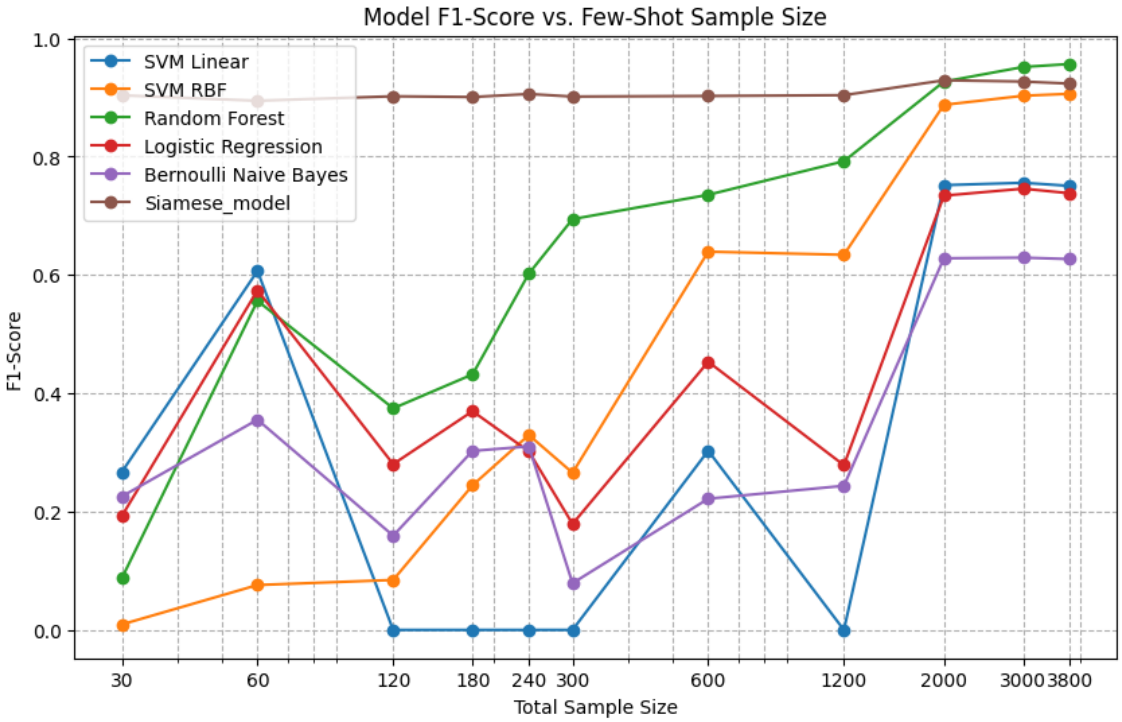


Figure 16: Prediction F1 score, y-axis, of all models on the Orange dataset up against different sample sizes in the x-axis, the divide between churn and nonchurn cases for the sample sizes can be seen in the subsequent table.

Table 17: Model f1 scores across different sample sizes on the Orange dataset, sample sizes on the left, with the best result for each sample size across the models in green and the worst in red.

Sample Size	SVM Linear	SVM RBF	RF	LR	Bernoulli NB	Siamese model
(5, 25)	0.267	0.009	0.087	0.192	0.226	0.904
(10, 50)	0.607	0.076	0.556	0.574	0.355	0.894
(20, 100)	0.000	0.084	0.375	0.280	0.159	0.902
(30, 150)	0.000	0.244	0.432	0.370	0.303	0.901
(40, 200)	0.000	0.329	0.602	0.302	0.310	0.906
(50, 250)	0.000	0.266	0.694	0.179	0.079	0.902
(100, 500)	0.303	0.639	0.735	0.453	0.222	0.903
(200, 1000)	0.000	0.634	0.792	0.278	0.244	0.904
(1000, 1000)	0.752	0.888	0.927	0.734	0.628	0.929
(1500, 1500)	0.756	0.903	0.952	0.746	0.629	0.927
(1900, 1900)	0.751	0.907	0.957	0.738	0.627	0.923

Table 16 and figure 15 presents the performance of the Siamese model alongside the baseline models, but on the Orange dataset. The models in the comparison not altered and have the same parameters presented in section 13. Analysis on the Orange dataset

reaffirms the Siamese networks strength in few-shot learning scenarios. It outperforms all the baseline models significantly in the few-shot scenarios, echoing the observations from the IBM dataset. Baseline models struggle once again with class imbalance, experiencing a distinct performance boost in the balanced scenario settings. Contrary to the Siamese network which demonstrates stability, reaching a performance plateau at approximately 93.5% with a low standard deviation (1.5%) across different sample sizes. Interestingly, Random Forest this time surpasses the Siamese network with larger balanced datasets, in the final two scenarios (1500, 1500) and (1900, 1900), as seen in table 14. The analysis of the Orange dataset reinforces the insights gained from the model comparison on the IBM dataset, demonstrating the robustness of the findings.

The results demonstrate the effectiveness of Siamese network for predicting churn in telecommunication. The L1 distance was the ideal distance metric in this case. The Siamese network outperformed several baseline models such as SVM, Random Forest, Logistic Regression and Bernoulli Naive Bayes, on both the IBM and Orange dataset. The Siamese network also performed exceptionally in a Few-shot learning environment. The results will be discussed in the following chapter providing further insight on its significance and implications.

6 Discussion

The findings in the study demonstrate the efficacy of Siamese networks for churn prediction, particularly within Few-shot learning scenarios. The results are analyzed in the context of the research question, with other interesting insights discovered also included. Prior to the discussion, the limitations of the present study and their potential impact on the validity of our findings.

The datasets used in the study may not encompass churn behaviours seen in various industries. The study is about customer churn prediction within telecommunication. The same results might not translate to churn prediction problems within other industries
Another potential limitation are the baseline models, which were employed using their default parameters without any specific tuning to optimize performance for this task. Possibly restricting their performance.

6.1 Interpretation of Results

The Siamese networks demonstrated robust performance in few-shot learning scenarios, consistently achieving higher accuracy compared to other models. This is significant as it suggests that Siamese networks can effectively learn complex patterns from a limited number of tabular examples, reducing the need for large datasets.
The Siamese networks performance with limited data in our study aligns with its positive results in image classification with few samples, seen in Du et al. [10] and Koch et al. [18]. In this study its application has been extended to the realm of tabular churn data, proving its functionality with tabular data, extending the utility of these networks beyond their traditional domains. This finding represents a notable contribution to the field, as machine learning models often struggle extracting meaningful information from limited labeled data. Such scenarios are often experienced in practical application.

This builds on the consensus that similarity learning works well in situations where labeled data is scarce, as it focuses on extracting discriminative features within the data itself rather than relying solely on explicit labels, further supporting the consensus.
The Siamese network had better performance when working with an imbalanced input than the other baseline models. The other models experienced a jump in both their accuracy and F1 score, showcasing how models might struggle on generalizability when working with imbalanced data. Data imbalance is very common when working with churn data, insinuating Siamese networks would perform better when working in real-world scenarios.

The ability of Siamese networks to operate with smaller datasets can be useful in practical applications, particularly beneficial for startup and small to medium enterprises, which may not have large amounts of customer data available.

In Few-shot learning scenarios, the small amount of data one has is crucial to the model. By using different feature pre processing techniques the Siamese network was able to perform excellent. The feature pre processing techniques had a great impact on the other baseline models as well, achieving better performance than previous studies done using the Orange dataset, as presented in section 1.
The impact of different feature pre processing techniques was huge on the performance of the Siamese network and Few-shot learning in this churn prediction task.

While the Siamese network demonstrated its effectiveness, its important to acknowledge that its performance, similar to other machine learning models, depends on the quality of the training data. This observation underscores the importance proper data preparation prior to model training, as emphasized by prior research in this field. The notable performance of both the Siamese network and baseline models on the entire Orange dataset, surpassing results seen in related studies, suggests the positive impact of the feature pre processing techniques employed in this thesis (outlined in section 4.2.1). This likely contributed to the Siamese networks success in Few-shot learning scenarios, highlighting the value of techniques such as noise removal, handling missing values (NaN), feature scaling, feature encoding and SMOTE.

6.2 Analysis of Research Questions

1. How does the performance of Siamese networks compare to traditional machine learning models in churn prediction tasks using tabular data?
2. How effective are Siamese networks in improving the accuracy and efficiency of few-shot learning models when applied to customer churn prediction?

These findings suggest that Siamese Networks present an adaptable solution for customer churn prediction when using tabular data, further expanding its field of use.
It has presented itself as an effective tool in improving the accuracy and efficiency in Few-shot learning when applied to customer churn prediction.

6.3 Comparison with Previous Research

Previous studies in customer churn prediction frequently highlights the challenge of class imbalance [31] [33]. This study demonstrates the robustness of the Siamese network when working in such environments, as evidenced by its results in the results chapter 5. This would suggest that Siamese networks offer a possible solution to bypass this common obstacle in the field.
Additionally, the ability of Synthetic Minority Oversampling Technique (SMOTE) in handling class imbalance is well presented in this study. The models strong accuracy when trained on the balanced dataset underscores the value of SMOTE as a tool for churn prediction.

The Siamese network also achieved 94.40% accuracy on the Orange dataset which has been used in previous studies on churn prediction. Outperforming many of the results in previous studies on the same dataset seen in Table 1, such as, 85.24% achieved by Jain et al. [16], 92.6% attained by Hassonah et al. [13] as well as 88.63% gotten by Ullah et al. [30].
Its important to acknowledge the results was not obtained in a Few-shot learning environment. Nonetheless this performance suggests the models potential for effective similarity learning in general, providing a basis or further exploration.
Some of the baseline models also achieved reslults exceeding those reported in previous research 13, despite using default model parameters. Suggesting the data pre processing applied in this study could have enhanced the quality and informativeness of the datasets, enabling even simple models to perform well. While this is not part of the research question, it is still an interesting insight on the importance of data pre processing in machine learning.

The Siamese networks good results in the study aligns with more recent studies leaning towards more deep learning techniques when working with customer churn prediction tasks. Corroborating deep learning techniques are the right way to go.

The findings highlight the networks strong performance across the datasets and its ability to overcome class imbalance, even outperforming established baseline models. Even though certain limitations are present, such as potential dataset bias and the use of default parameters for the baseline models, the results showcase the networks success in extracting informative patterns from limited labeled data. Thus, along with its adaptability to real-world scenarios, positions Siamese networks as a promising tool for advancing churn prediction methodologies.

7 Conclusion

7.1 Summary of Findings

The thesis evaluated the potential of Siamese networks for tabular churn prediction, also using Few-shot learning attempting to recreate its success with Image classification. Some of the key findings included:

The Siamese networks documented good performance on image data classification translates onto tabular data as well for both the whole dataset and Few-shot learning instances. The Siamese network outperformed traditional machine learning models in Few-shot learning, showcasing its strength in extracting meaningful information from limited labeled data.
Its similarity learning, assesses its similarity or dissimilarity between pairs, being able to generalize better with less training data. The findings are evident from both the Orange and IBM telecommunication datasets on train, test, and validation sets, suggesting generalizability has been achieved.
The Siamese performance was not affected when working with imbalanced data as the other baseline models, highlighting its potential for real-world churn data, which is commonly imbalanced.
The network outperformed traditional baseline models such as SVM, Random Forest, Bernoulli Naive Bayes and Logistic Regression across different evaluations in Few-shot learning. Suggesting its ability to learn meaningful data from limited labeled data. The network had the best performance using L1 distance for this churn prediction task. Indicating its suitability for this specific task of churn prediction.

7.2 Implications

The demonstrated performance of Siamese networks in a new application area, tabular churn data, contributes to extending the potential of Few-shot learning models. It expands the application of Siamese networks beyond traditional image classification.
The networks ability to learn meaningful information from limited examples addresses a significant challenge in machine learning. Siamese networks should be a considered machine learning model when working with limited labeled data. Its characteristics makes it especially useful for smaller firms or startups who does not have access to large datasets.

7.3 Recommendations for Future Research

This thesis provides insights into the efficacy for Siamese networks within churn prediction. To build upon these findings, future research could explore techniques identified in previous research, seen in Table 1. Evaluate the potential of a hybrid model, combining a Siamese network with another machine learning model on tabular data. Such a model could yield a performance gain.
Additionally, investigating customer segmentation using Siamese networks could offer insight into the models decision making. This would illuminate how the model differentiates churner subgroups, providing valuable information for customer retention.

To asses the generalizability of the Siamese network across different types of tabular data, future research should apply it to churn prediction within other industries or other data entirely. Assessing the Siamese networks capabilities within a broader context.

8 Appendix

AI as a helping tool.
ChatGPT an AI developed by OpenAI was taken in use when working with the master thesis. It provided assistance in different areas, helping to simplify some challenges along the way. It provided assistance in translating the abstract to Norwegian, debugging when working with the code and helping with formal language fitting for a master thesis.

References

[1] Richeldi M & Perucci A. Churn analysis case study. *Deliverable D17*, 2002.

[2] Saran A. and Chandrakala D. A survey on customer churn prediction using machine learning techniques. *International Journal of Computer Applications*, 154:13–16, 2016.

[3] Farah Alkhalid. The effect of optimizers on siamese neural network performance. 2022.

[4] Thomas Bayes and Mr. Price. An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, communicated by mr. price, in a letter to john canton, a.m., f.r.s. *Philosophical Transactions*, 53:370–418, 1763.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[6] Zhen-Yu Chen, Zhi-Ping Fan, and Minghe Sun. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, 223(2):461–472, 2012.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[9] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors,

Computer Vision – ECCV 2018, pages 472–488, Cham, 2018. Springer International Publishing.

[10] William Du, Michael Fang, and Margaret Shen. Siamese convolutional neural networks for authorship verification. 2017.

[11] Abhishek Gaur and Ratnesh Dubey. Predicting customer churn prediction in telecom sector using various machine learning techniques. In *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, pages 1–5, 2018.

[12] John Hadden, Ashutosh Tiwari, Rajkumar Roy, and Dymitr Ruta. Computer assisted customer churn management: State-of-the-art and future trends. *Computers Operations Research*, 34:2902–2917, 10 2007.

[13] Mohammad A. Hassonah, Ali Rodan, Abdel-Karim Al-Tamimi, and Jamal Alsakran. Churn prediction: A comparative study using knn and decision trees. In *2019 Sixth HCT Information Technology Trends (ITT)*, pages 182–186, 2019.

[14] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[15] D.C. Howell. *Statistical Methods for Psychology*. Thomson Wadsworth, 2007.

[16] Hemlata Jain, Ajay Khunteta, and Sumit Srivastava. Churn prediction in telecommunication using logistic regression and logit boost. *Procedia Computer Science*, 167:101–112, 2020. International Conference on Computational Intelligence and Data Science.

[17] Muhammad Joolfoo, Rameshwar Jugurnauth, and Khalid Joolfoo. Customer churn prediction in telecom using machine learning in big data platform. *Journal of Critical Reviews*, 7:1991, 07 2020.

[18] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[19] Praveen Lalwani, Manas Mishra, Jasroop Chadha, and Pratyush Sethi. Customer churn prediction system: a machine learning approach. *Computing*, 104:1–24, 02 2022.

[20] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[21] Iulia-Alexandra Lungu, Alessandro Aimar, Yuhuang Hu, T. Delbruck, and Shih-Chii Liu. Siamese networks for few-shot learning on edge embedded devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10:488–497, 2020.

[22] Jesper Ødegård Marcus Andre Glover Meek. Explainable machine learning for customer churn prediction. *NTNU*, 2023.

[23] Loris Nanni, Sheryl Brahnam, Alessandra Lumini, and Gianluca Maguolo. Animal sound classification using dissimilarity spaces. *Applied Sciences*, 10(23), 2020.

[24] Loris Nanni, Giovanni Minchio, Sheryl Brahnam, Gianluca Maguolo, and Alessandra Lumini. Experiments of image classification using dissimilarity spaces built with siamese networks. *Sensors*, 21(5), 2021.

[25] Loris Nanni, Giovanni Minchio, Sheryl Brahnam, Davide Sarraggiotto, and Alessandra Lumini. Closing the performance gap between siamese networks for dissimilarity image classification and convolutional neural networks. *Sensors*, 21(17), 2021.

[26] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. https://github.com/keras-team/keras-tuner, 2019.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] Frederick F. Reichheld and David W. Kenny. The hidden advantages of customer retention. 12(4):19–24, 1990.

[29] Lewlisa Saha, Hrudaya Kumar Tripathy, Tarek Gaber, Hatem El-Gohary, and El-Sayed M. El-kenawy. Deep churn prediction method for telecommunication industry. *Sustainability*, 15(5), 2023.

[30] Irfan Ullah, Basit Raza, Ahmad Kamran Malik, Muhammad Imran, Saif Ul Islam, and Sung Won Kim. A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE Access*, 7:60134–60149, 2019.

[31] Sharmila K. Wagh, Aishwarya A. Andhale, Kishor S. Wagh, Jayshree R. Pansare, Sarita P. Ambadekar, and S.H. Gawande. Customer churn prediction in telecom sector using machine learning techniques. *Results in Control and Optimization*, 14:100342, 2024.

[32] Junhua Wang and Yongping Zhai. Prototypical siamese networks for few-shot learning. *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 178–181, 2020.

[33] Shuli Wu, Wei-Chuen Yau, Thian-Song Ong, and Siew-Chin Chong. Integrated churn prediction and customer segmentation framework for telco business. *IEEE Access*, 9:62118–62136, 2021.

[34] Xiancheng Xiahou and Yoshio Harada. B2c e-commerce customer churn prediction based on k-means and svm. *Journal of Theoretical and Applied Electronic Commerce Research*, 17:458–475, 04 2022.

[35] 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, 2009.

[36] Tianpei Xu, Ying Ma, and Kangchul Kim. Telecom churn prediction system based on ensemble learning using feature grouping. *Applied Sciences*, 11:4742, 05 2021.

[37] Liu Xuxing, Weize Gao, Rankang Li, Yu Xiong, Xiaoqin Tang, and Shanxiong Chen. One shot ancient character recognition with siamese similarity network. *Scientific Reports*, 12:14820, 2022.

[38] Ansi Zhang, Shaobo Li, Yuxin Cui, Wanli Yang, Rongzhi Dong, and Jianjun Hu. Limited data rolling bearing fault diagnosis with few-shot learning. *IEEE Access*, 7:110895–110904, 2019.

[39] Bing Zhu, Bart Baesens, and Seppe vanden Broucke. An empirical comparison of techniques for the class imbalance problem in churn prediction. *Inf. Sci.*, 408:84–99, 2017.

[40] Özden Gür Ali and Umut Arıtürk. Dynamic churn prediction framework with more effective use of rare event data: The case of private banking. *Expert Systems with Applications*, 41(17):7889–7903, 2014.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway