KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First cycle 15 credits

Detecting Chargebacks in Transaction Data with Artificial Neural Networks

THEODOR GÜNTHER & OTTO PAGELS-FICK

Stockholm, Sweden 2022

Detecting Chargebacks in Transaction Data with Artificial Neural Networks

Upptäcka återbetalningar i transaktionsdata med artificiella neurala nätverk

Otto Fick, *Student*, Theodor Günther, *Student*
EECS - Royal Institute of Technology, May 2022
[ottopf | theodorg] @kth.se

***Abstract*—The chargeback process is costly for the merchant. Not only does the merchant lose the revenue from the purchase, but it must also pay handling fees to the bank and risks never getting paid for provided service. The purpose of this study is to examine and investigate how to prognosticate future chargebacks by using machine learning in form of *Artificial Neural Network* on transaction data. Doing so can be used to minimize and decrease financial costs for the merchant. The study indicates that it's complex to prognosticate chargebacks, but illuminates that it's possible under certain circumstances. The created model has been concluded to be more suitable as a compliment, rather than a substitute for the current *Rule-Based classification* system. The model should be implemented based on economic analysis since it can be used to reduce costs and contribute to profitability over time. Furthermore, the study highlight lessons learned and complementary research areas for future studies.**

***Abstract*—Återbetalningar medför stora kostnader för handlare i form av förlorade intäkter vid återbetalning av transaktionssumman, samt tillkommande handläggningsavgifter i processen. Syftet med rapporten är att utvärdera och undersöka möjligheten att prognostisera framtida återbetalningar, genom att applicera maskininlärning i form av *Artificiellt Neuralt Nätverk* på transaktionsdata. På så sätt kan återbetalningar minimeras och reducera finansiella kostnader hos handlaren. Studien påvisar att det är komplext att predicera återbetalningar, men att det är möjligt under särskilda omständigheter. Modellen som skapats har konstaterats mer lämpad som ett komplement till det aktuella regelbaserade klassificeringssystemet än ett substitut. Utifrån en ekonomisk analys klargörs att algoritmen bör implementeras för att reducera kostnader och på sikt bidra till lönsamhet. Studien belyser även lärdomar, samt kompletterande forskningsområden för framtida studier.**

***Index Terms*—Artificial Neural network, Chargeback, Fraud, Machine learning**

I. Introduction

HANDLING credit card transactions in an online-based, service-providing, multinational business is a complex task. On the one hand, the business must ensure that its services are easy to access and can be bought from anywhere, at any time, to remain competitive in the market. On the other, the business doesn't want to make it too easy to buy services since this can make the company vulnerable to fraudulent transactions and unsatisfied customers. Having a low threshold when accepting credit card payments will attract persons with fraudulent intentions to exploit the service in different ways. For example, if a business only requires a user to be logged in to perform transactions with a credit card attached to the user's account, one would only need a user's login credentials to exploit the user's credit card on the platform. Another way to make transactions easy is not requiring multi-factor authentication while registering a credit card. But this paves way for using stolen credit card information to make purchases on the platform. Another risk of making transactions easily accessible is that customers will make transactions that they are not completely agreeing with.

When a credit card holder discovers a transaction not caused by the holder himself or a transaction that the holder believes should be refunded due to other reasons, the cardholder can appeal directly to their bank. It then becomes a *customer initiated chargeback* and the bank usually returns the money to the customer and consequently demands a return of the money transferred to the merchant[1].

A. The Merchant's Perspective

NinjaCall (fictional name for the real company) is a tech company focusing on communication and international calling to connect migrants with people around the world. Calls are connected globally through a combination of using the internet and local phone lines. The service is application-based and requires a lot of small credit card transactions. NinjaCall finds this project interesting and highly relevant since they daily suffer from large amounts of chargebacks with associated costs and fees. Both provider costs and Payment Service Provider (PSP) fees.

NinjaCall currently has a rule-based classifier based on if-statements, which demands continuous updates and maintenance to adapt to new trends of fraudulent behaviour. This is costly and NinjaCall would like to develop a more independent and self-going model, which automatically adapts to new fraudulent trends.

Apart from the financial incentives, NinjaCall believes that a solution to the problem would be value-adding from a customer service perspective. Chargebacks can, irrespective of underlying causes, result in dissatisfied customers and be connected to customer complaints. This also has a negative financial effect.

Regardless of the reasons causing chargebacks, NinjaCall wants to know if a transaction will become a potential chargeback before it happens. Thereby, NinjaCall would have the

[1] Chargeback, Wikipedia, https://en.wikipedia.org/wiki/Chargeback

possibility to act and avoid negative financial effects. This paper will examine how to do this with the help of supervised machine learning, more specifically with an artificial neural network.

B. Purpose

The value of this thesis is grounded in absence of research where machine learning has been combined with the detection of potential chargebacks. The previous thesis has investigated either the process of chargebacks and their effect on involved parties or the use of machine learning to detect incorrect transactions by focusing on the fraudulent behaviour of the user. Since this thesis concerns an unexplored area of machine learning about chargeback, our work will contribute to the multidisciplinary research. Our goal is to answer if it's possible to predict whether a transaction will become subject to a chargeback with the help of supervised machine learning in form of a virtual neural network. The desired outcome is that the model will give reliable results that reflect reality. Therefore, this thesis will be interesting for all parties involved in the process of chargebacks, people interested in machine learning and legal actors working with the detection of frauds connected to transactions.

C. Ethical Aspects

Since this thesis is performed in collaboration with NinjaCall, our main goal is to forecast whether or not a transaction will become a chargeback, in the interest of NinjaCall. Worth mentioning is that the chargeback process brings additional work for all parts involved. From this perspective, minimizing chargebacks will benefit all actors in the chargeback process. By preventing chargebacks, this thesis can contribute to ensuring legit transactions, which will directly help NinjaCall and indirectly contribute to the overall work of eliminating unlawful transactions. By this, the paper will contribute toward the UN's sustainable development goal number sixteen, Peace, Justice and Strong Institutions. Furthermore, sustainable development goal number nine, Industry, Innovation and Infrastructure, is touched upon when enabling innovative services to safely be sold in numerous markets over the world.

When developing a machine learning model, different kinds of biases can affect the outcome of the model. For example, can a model judge card holders from a specific country be more likely involved in fraud compared to other countries? This increase the risk of cementing cultural preconceptions and limit legit users to use the service because of, for example, country of residence. The ethical issue here is complex and there is a risk of it ending up as a trade-off between reducing unlawful transactions and reducing the possibility to use the service for people of a certain country. Whether a machine learning model bases its outcome on a discriminatory basis is not always easy to detect, especially when using artificial neural networks. Further studies on which features are significant for the model should be carried out in future studies within the field and definitely before it is implemented. Such studies are, however, left outside the scope of this paper.

D. Research Question

This thesis will examine the following research question:

> *To what extent can an artificial neural network model be used in order to predict whether a transaction will become subject to a chargeback?*

and this consequent business research question:

> *To what degree can such an artificial neural network model be used to reduce costs for a merchant?*

To answer the research question an artificial neural network model is built. The model is then trained with data gathered from NinjaCall to classify transactions as being chargebacks or not. The features of the data are selected to correspond to the information available at the moment of the transaction. This is done so that the model theory can be used to predict chargeback and prevent them even before they become transactions. The model is evaluated as a substitute and as a complement separately. In combination with the model, an economic evaluation of implementation is conducted. The economic evaluation is aimed to answer whether or not an implementation of a machine learning model could reduce costs for a merchant.

II. Previous Research

The previous research made on the prediction of chargebacks on credit card transactions is, to our knowledge, nonexistent. There has, however, been an extensive amount of research on the detection of fraudulent credit card transactions. Bourgne et al state, in their handbook, that there have been significant advancements in the research area of fraud detection with the help of machine learning and that this has resulted in a decrease in financial losses due to fraud starting in 2016.[9] If the same is true for chargebacks is still to be investigated. Overall, it is difficult to determine if the classification of transactions as future chargebacks is at all used in the industry. The classification of transactions, as fraudulent or not, is very similar to the classification of transactions as potential chargebacks with the main difference that a chargeback can have other causes than fraud. It can be that some studies actually use chargebacks as a measure of fraud and simply ignore the fact that they are slightly different.

When it comes to machine learning methods Jóhansson emphasizes that neural network models have been more successful in fraud detection compared to techniques such as SVM, and binary logistic regression.[8] Bourgne et al however do not compare neural networks to these methods but state neural networks cannot be assumed to outperform other methods within the field of deep learning as XGBoost and random forest. At the same time, Bourgne et al mean neural networks could be preferred for other reasons like the possibility to use for incremental learning.[9] This possibility is highly relevant in the application of chargebacks since new data is constantly received by the merchant and can be used to improve the model. Common in the literature mentioned above is a warning that the neural network method requires a lot of computing power and may not be suitable for applications where the amount of data is immense. We estimate that the data we are handling is not considered to be too large for this method and therefore that the possible advantages with neural networks outweigh the drawbacks.

Another great contribution to the research area is done by Correa Bahnsen et al. In their study, they underline the importance of evaluating a fraud detection model in its financial context and not solemnly on its intrinsic performance.[3] This has been taken into account in this study by formulating potential costs and benefits with a chargeback classifying model. Correa Bahnsen et al also emphasize the importance of creating aggregated features to capture the individual's behaviours. They show inclusion of such variables can lead to a 200% increase in model performance. While we acknowledge the importance of such features we believe Correa Bahnsen et al overlook the fact that some transactions are connected to a first-time customer and in such cases building aggregated features is impossible.[3] Since the first-time payment make up a considerable share of our data, we have chosen not to build aggregated features in this study.

III. Theory

A. The Chargeback Process

The customer-initiated chargeback process involves five different key players: the cardholder, the merchant, the card issuer, the acquirer, which has the task to acquire the payment on behalf of the merchant and the card network that oversees the whole transaction process.

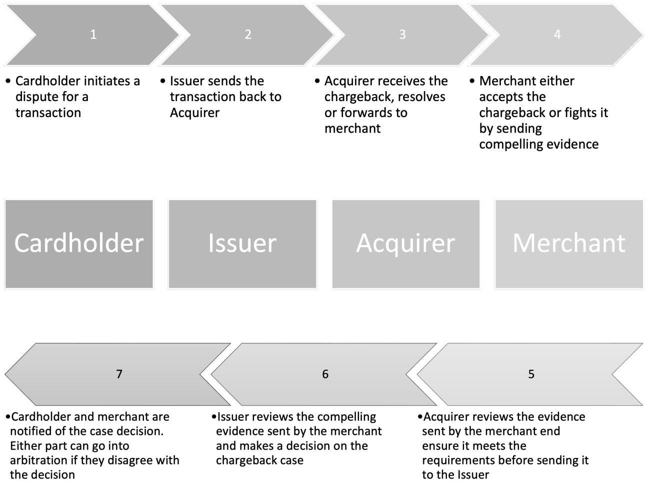
1
2
3
4
• Cardholder initiates a dispute for a transaction
• Issuer sends the transaction back to Acquirer
• Acquirer receives the chargeback, resolves or forwards to merchant
• Merchant either accepts the chargeback or fights it by sending compelling evidence
Cardholder
Issuer
Acquirer
Merchant
7
6
5
•Cardholder and merchant are notified of the case decision. Either part can go into arbitration if they disagree with the decision
•Issuer reviews the compelling evidence sent by the merchant and makes a decision on the chargeback case
•Acquirer reviews the evidence sent by the merchant end ensure it meets the requirements before sending it to the Issuer

Figure 1. The Chargeback Process

The chargeback process is complex and requires a lot of bureaucracy. Simply put, if a customer declares to their issuing bank a valid reason why they shouldn't have been billed for a particular charge, the bank can do a conditional refund to the cardholder. The issuer then requires a recoup of money from the acquirer. The acquirer follows by returning the amount of the original transaction, in combination with fees connected to the handling of the process, from the merchant's account. In other words, the merchant loses the merchandise, and the revenue generated by it and suffers additionally fees. The merchant can dispute a chargeback by providing clear evidence to the card association, which then passes it to the issuing bank, but few cases result in the favor of the merchant. This is since it's hard to show evidence that the transaction wasn't made by fraudsters, in combination with the card association's protection to preserve individual costumer's rights.[2]

B. Machine Learning

Most machine learning algorithms can be divided into either supervised or unsupervised learning. This thesis will only include supervised learning. Supervised machine learning is used by training a model on a data set which is categorized with the desired output feature. The categorized data is used to train a binary logistic regression model to be able to classify uncategorized data. In this study, the data points labelled as a *chargeback* are the positive case whereas *not chargeback* is the negative case. To train the model, input data is fed into the model and weights are adjusted to yield the corresponding output. The correct output is compared with the model's output which gives a measure of how well the model is fitted to the data. The training is ended when the measure reaches a satisfactory low level.[p. 95-106 7] By doing this, the model can recognise patterns and can be used to predict labels for unlabelled data. In this study, it will be used to label transactions and predict whether they will become subject to chargeback.

C. Neural Networks

To capture nonlinear patterns regarding an output, a model with a single layer of weights is not enough. This is when Neural Networks come into play. With the use of a neural network, different combinations of feature values can give the same classification as compared when only one layer is used. When using a single-layer model internal relationships between different features can not be taken into consideration. An example of such non-linear relationships is if a low value of a feature x_1 in combination with a high value of a feature x_2 should lead to a positive classification and that the opposite, high x_1 value in combination with a low x_2 value also should lead to a positive classification. These relationships occur in real life but cannot be captured in single layer model.[p. 166-171 7]

1) Layers and Neurons: One aspect of neural networks worth mentioning is the number of layers and neurons. It doesn't exist any strict rules on how to determine the quantity choice of neurons and layers. It does, however, have existing guidelines to help the developer in the trial and error process.[3] In this case these guidelines do not give much help since

[2] Chargeback, Wikipedia, https://en.wikipedia.org/wiki/Chargeback

[3] An example given by Ahmed Gad in his article *Beginners Ask: How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?*, Towards Data Science (2018).

the used data set has a too large amount of data points and features. We have thus used a trial and error process in combination with advice from proficient programmers in the area, to examine the most suitable quantities for neurons and layers.

D. Categorical Data

Categorical variables consist of a limited number of values. Examples are countries, blood type, address, service type and mobile type. Categorical data differs from numeric variables, by consisting of distinct values, not directly translatable to numbers. Numeric operations can't, therefore, be used on Categorical data. If simply translated to numbers the order is based on the categories' order and not the values' lexical meaning. One way of treating categorical variables is by giving the categories embedding which are trainable in a neural network. This way the model can learn a representation of every category which suits the purpose of the model.[6]

E. Imbalanced Data

A classic problem when training a machine learning model is skewed training data, meaning that the proportion between positives and negatives is far from one-to-one. This can result in poor classification because the model doesn't learn the characteristics of the minority class since it has such a small representation in the data set.

1) Over- and Undersampling: An equally classic antidote to imbalanced data is over-and undersampling. In our case, the positive class, a chargeback, has much fewer occurrences than the negative case. Training the model with such data can lead to an inefficient model with poor performance since it overrates the number of negative cases. Oversampling is achieved by increasing the number of cases in a category, simply by duplicating them, to even out the proportions. Undersampling is the opposite of oversampling and is done by removing data points in the predominant category.

2) SMOTE and Tomek Links: Over- and undersampling, in their simplest forms, are known to have drawbacks. Namely, oversampling risks overtraining the model and the undersampling takes away too much important information about the majority class. To counter these drawbacks two techniques can be used. SMOTE helps create new synthetic data points in the minority class by interpolating between existent data points. When undersampling removing Tomek links has been found successful to retain relevant information about the majority class. Simplified, a Tomek link consists of two data points from different classes, which are closer to each other than any other data point. One of the data points can be considered noise and is therefore removed.[1]

F. Over- and Underfitting

When training neural networks, over-and underfitting can occur if the hyperparameters are incorrectly chosen. Overfitting is a statistical modelling error which arises when a function is too adapted to the training data. The implication of overfitting is a bad generalization of unseen data.[p. 107 7] Underfitting is the opposite and occurs either if the model is trained too little or if the model cannot learn patterns in the training data.

When training the model a validation set is used to keep an eye on whether the model is over-or under fitted and helps to determine the hyperparameters. By using the method *Early Stopping* you can stop the training directly when the model's performance stops improving its loss in the validation data. The loss is a measure of how wrong the model predicts a certain value and is discussed more in the method. Depending on the purpose of the algorithm, different measures can be used to monitor the training.

G. Evaluation of the Machine Learning Model

1) Intrinsic: There are several evaluation metrics regarding classification. Accuracy is the most general and is simply computed by taking the *Correctly Classified Data Points* over *All Data Points*. False negatives and false positives are samples that were incorrectly classified and true negatives and true positives are samples that were correctly classified. Denoting the different classes as *Positive* and *Negative* and a correct classification as *True* and false as *False*, the accuracy is commonly presented as following:

$$\frac{TP + TN}{TP + TN + FP + TN}$$

Accuracy is, simply put, a metric showing the share of data points that were labeled correctly. Precision gives the share of the data points with a certain given label that was classified correctly. It will therefore be showed as (regarding positives):

$$\frac{TP}{TP + FP}$$

Recall is the complementary metric to precision and displays the share of data points with a known certain label that were labeled correctly. More precisely the percentage of actual positives that were correctly classified:

$$\frac{TP}{TP + FN}$$

To consider precision and recall at the same time, F-score can be used where F-score gives an geometric mean of the two metrics. Evaluating with F-score as an basis punishes a low value on either precision or recall compared to an arithmetic mean.

$$\frac{2PR}{P + R}$$

When evaluating a machine learning model it is important to use a distinct data set, which has not been seen by the model during the training. The sets have to be distinct to ensure the model is useful on real-world data that the model has never seen before.

2) Extrinsic: Comparison with other available prediction methods allows for a more practical *real-world* evaluation and can be performed in various ways. One approach is to compare with a baseline. This can for example be a method that is currently used in the company or a method that is widely for similar tasks. Compared with what an expert could achieve, the *golden standard* is also a widespread extrinsic evaluation method.

H. Economic evaluation

1) SWOT: SWOT is a framework used for evaluating a company's competitive position and stands for Strengths, Weaknesses, Opportunities, and Threats. It's therefore used for strategic planning, by focusing on both external and internal factors in combination with future and current potentials. The framework is used to look at the strengths and weaknesses of an organization, initiatives, or within its industry. It can thus be used to evaluate initiatives as new technical solutions. SWOT should be used as an assessment tool, not necessarily as a guide or seen as a regulation. [p. 345 4]

2) Payback Method: To be able to evaluate the implementation of the model at NinjaCall from a financial perspective, the payback method will be used. The payback method helps evaluate how long it takes for an investment to recover from its initial investment, the payback period. This is also called the break-even point for an investment.
cite[p. 230][]introindek

3) Cost-Benefit Analysis: This method can be used to analyse the reduction or increase of the cost base. The existing costs regarding the current system are balanced against the potential cost reduction when implementing the new model. The net benefit of implementing an alternative model is calculated by subtracting the cost reduction from the cost of implementing the new system. The advantage of this method is the ability to compare different projects with each other and in comparison with the existing system. The disadvantage is the fact that the calculations are based on approximate values, which can contribute to discarding the project solely on incorrect approximations.[5]

IV. Method

The majority of the initial work has been considered from previous research studies on the area. The work process has been iterative regarding the development of the model. Methods we initially believed would work, turned out not to support the process, which we on the other hand have gathered learning's from. Continuous discussion regarding our approach has been combined with further research. That research has been applied to our implementation and contributed to further development and adjustments.

A. Gathering of Data

The data was gathered from NinjaCall's database on transaction data. For the data to be relevant in this study it must contain a representative amount of chargebacks. We wanted to choose as new data as possible to enable our model to catch current chargeback patterns. However, since, the banks report chargebacks with a few months' delay we had to choose data at the earliest from the beginning of this year. Furthermore, NinjaCall handles an enormous amount of transactions. Extracting data from a whole year would result in almost a hundred million transactions. With these considerations in mind, data from the last three months of 2021 was gathered, which meant a total of more than 10 million data points. NinjaCall has an existing cloud-based API called Google Looker, which is connected to their database. Looker offered various tools and filters, which were used when extracting the data sets.[4]

Two types of data sets that are used, consist of both numeric and categorical features. The data sets contain both general transaction data such as amount, currency and product group, but also more user-specific features such as payment country, IP address and fraud score. The initial data set, with *8 331 938* data points, had 2.15 data points classified as chargebacks. The data set is randomly divided into train-, validation and test sets with the distribution 70 | 15 | 15, where the amount of chargeback has corresponding distribution.

B. Choice of Features in Data

As mentioned above, 23 features are used in the training of the model. Out of these 23, 4 are derived from 2 features with a cyclic pattern. The features have been chosen together with employees at NinjaCall. When choosing features our goal was to gather as many different features as possible that are available for NinjaCall at the moment of approval of the transaction. This ensures the model can be used in a real-life situation. By choosing as many features as possible, we eliminated subjectivity, since we do not consider the relevance of the feature. Our approach is that we do not know what features are going to be valuable in the classification of chargebacks. Furthermore, by only considering features available at the moment of purchase we do not choose features that are an effect of a chargeback happening. Such features risk ruining the model by essentially training on the feature we are trying to predict.

Categorical Features	**Numeric Features**
Payment platform	Payment amount
Currency	Days since last payment
Product ID	IP fraud score
Product group	Sinus repr. weekday
Most called country	Cosine repr. weekday
User's last bought service	Sinus repr. hour
Payment method	Cosine repr. hour
Product subcategory	
Area for use of product	
Product country	
Response type	
IP connection type	
User's country	
Payment country	

Table I
Features info

C. Choice of Model

NinjaCall's existing rule-based classifier is based on an algorithm where if-statements are combined to detect potential fraudsters. This way of working implies continuously maintenance and adaptation to new fraudulent behaviour. It also contains elements of bias, as a result of the qualitative design of the if-statements. Our scope doesn't focus strictly on frauds, which is one of several causes of chargebacks, but rather on the classification of chargebacks irrespective

[4] Google Looker, https://cloud.google.com/looker

of causing attributes. We, therefore, wanted to research if it's possible to implement an algorithm that decreases the demand for frequent maintenance to adapt to new types of factors causing chargebacks. As mentioned before, fraudulent behaviour is often complex to discover for several reasons. Fraudulent behaviour is seldom differentiable from non-fraud when looking at data points. The same conclusion can be drawn with chargebacks. Illuminated in the section regarding previous research, the lack of research in this area entails the fact that we are unaware if it's possible to classify chargebacks at all.

To implement a model, which minimizes the risk of bias and maximizes the possibility to classify chargebacks, we argue that a neural network would be the best fit. Neural Network enables complex different combinations of feature values and gives the same classification as when only one layer is used. We, therefore, decided to create our own Neural Network.

1) Python Libraries: Key frameworks we use are Pandas, Numpy, Scikit-learn, Tensorflow, Imblearn and Matplotlib. Tensorflow is used to create the neural network[10], whereas SMOTE is applied to resample the data. Pandas and Numpy are used when processing tabular data versus numerical data. Pandas generate useful series and data frames. Scikit-learn is used to divide the data set into training-, evaluate- and test data.

D. Data Processing

We initially removed around 2 million data points due to them being incorrectly created or not necessary for this task, for example, transactions connected to subscriptions. This resulted in a higher percentage of chargebacks among the data points. Followed, rows with null values were either removed or the value was replaced with the string "no_value", were found appropriate. Then a dummy feature was created with the help of dates of chargebacks and Nov (notification of fraud), where 0 represent no chargeback and 1 chargeback. If a date of chargeback existed, then we knew that it had become a chargeback. Likewise, a notification of Fraud (NoF) is fraudulent activity reported by the cardholder's bank. NoF has the purpose to inform the merchant (NinjaCall in this case) that a payment isn't done by the legitimate cardholder. The NoF is not a dispute and therefore incapacitates the possibility to reply with defence.

Two of the features have a cyclic pattern, namely *weekday* and *hour of the day*. These were converted to numeric values and then normalized between zero and 2π. Then, for each feature, a cosine and a sinus feature were created from the normalized value. This ensures that for example Monday and Sunday were interpreted as two neighbouring values and not each other's opposites.[5]

By using Pandas Series.cat.codes [6] the categorical feature were given numerical integer representations. This enables us to use the Keras embedding layer to represent different categories, embedding which was later trained to represent these categories.[2] Regarding how many weights were used, the emb_sz_rule works as a rule of thumb[7]. The features "pay_country", "user_country" and "issuer_country" use the same embedding layer whereas, the other categorical features have their own embedding layer.

We divided the data into training, validation and test sets with the following proportions 70—15—15. The training and validation set are separately resampled using Batista et al's method SMOTETomek[1], implemented in scikit-learn.[11] The resampling method turned out to need very high computer performance and hence a bottleneck for how much data we could include. When reducing the input data set to 3 million points the computing time became manageable. When resampled according to the algorithm only about 350 000 data points remained, of which 20% chargebacks, real and artificial.

E. Model Optimization

The training of the model was continuously evaluated by considering the validation loss. As the model is trained both the validation and training loss decrease. After a while, though the validation loss started to increase or at least stopped to decrease while the training loss continued. When this occurs the model can be considered optimally trained since it is well trained on the data but still can generalize on other than training data. To save the model at this stage *Early Stopping* was used, implemented with help of Keras library[8].

The number of neurons was optimized through repetitive testing. With a too simple model, with few neurons, the model was not able to learn from the training data manifested by a training loss which wouldn't decrease. The opposite occurred when the model had too many neurons. Then did the model learn too much, including noise data points with poor information? This was noticeable by validation loss which almost instantly increased a lot.

F. Evaluation of Classification Model

The evaluation of the model is dependent on what is supposed to be used. Today NinjaCall has a program which every transaction goes through before it is approved. The program consists of a variety of conditions the transaction has to conform to, to be approved by the system. The conditions are specified *by hand* and are supposed to reflect traits that have historically been connected to fraud. The program is not "chargeback proof", hence the need to develop new methods to predict chargeback. The program is explicitly designed to rather accept suspicious transactions than deny when there is any uncertainty regarding if it is a chargeback or not. This is because NinjaCall prefers letting small amounts of chargebacks through rather than upsetting honest customers. Today's system does, hence, prefer a higher precision over recall in the classification of chargebacks.

Since we are designing a new model to classify chargebacks, comparing it to the old method as a baseline would be a good way to evaluate our model's performance. This has regrettably

[5] Pierre-Louis Bescond, *Cyclical features encoding, it's about time!*, by Towards Data Science (2020).

[6] Pandas python library for Data Analysis, https://pandas.pydata.org

[7] See Keras documentation on Embedding Layers.

[8] See Keras documentation on Model Training.

not been possible in this thesis due to that the orders which are denied and don't become transactions are not accessible for NinjaCall. In other words, the data that is used in the training of our model has already gone through NinjaCall's current rule-based classification tool for chargebacks. NinjaCall's current program and the model we develop are in a sense not performing the same task. None of the potential chargebacks that are inhibited by the current program is seen by our model. Our model is only trained on the chargebacks the current program fails to classify. Using the current model is both not feasible and appropriate.

Let us assume our model would perform just as good or better on the raw orders NinjaCall receives. This is a reasonable assumption since the chargebacks, which the current model filters out, are of a simpler and more obvious character than the chargebacks which are left in the data we use. The logic is that if our model succeeds in classifying the chargebacks in our data set it will also do so on the simpler data points which are the original orders. With this aim, we could evaluate our model as a replacement to the current model and hence mainly consider the precision it achieves on test data since this is the main priority for the current model.

In discussions with NinjaCall, an idea arose that our model could be used to classify suspicious transactions that would, when detected, be subject to an extra *challenge* before it would be accepted. The challenge could be any extra authentication, for example confirming one's e-mail or answering a security question. If the model were to be used in such a context the most important metric would rather be recalled (which it normally is when considering fraud detection). With a high recall, the merchant could be sure that almost all potential chargebacks are being considered and thereby prevented. The cost of having low precision would be low since it in practice only mean that more persons have to authenticate their account.

In summary, the model's performance is evaluated in two ways:

- as a substitute for today's program, high precision needed
- as a complement to today's program, used to *challenge* certain orders, high recall needed

G. Economic Evaluation

As mentioned in the introduction, NinjaCall suffers from both the refund connected to the amount of the original transaction and fees associated with the handling of the process. NinjaCall's desire to classify potential chargeback is also due to non-monetary aspects, such as the customer service perspective. One can argue that these are inseparable since unsatisfactory customer service has negative financial effects. If NinjaCall would be able to classify chargebacks before the transaction, they could decrease their financial cost and increase value for the customer. Partly by decreasing fees connected to the chargeback and by maintaining customer satisfaction.

Moreover, the existing system does only consider one of several factors which contribute to chargeback cases. It thus only focuses on preventing fraud, not reducing chargebacks independent of cause. Additionally, NinjaCall's existing rule-based classifier demands a lot of maintenance adaption to new types of fraud. This implies related costs and expenditures which could be saved. We, therefore, evaluate the financial aspects of adapting our model, by using SWOT analysis, the Payback method and Cost-Benefit analysis. By doing so, we estimate both the financial and non-monetary benefits and profitability by implementing our model.

V. Results and Analysis

A. Learning Curve

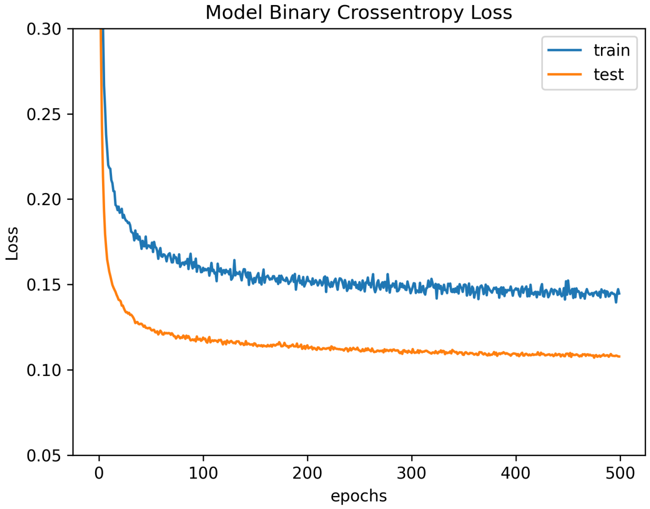

Figure 2. Learning curve when training the model

In the figure, we can see how the training and validation loss decreases as we train the model but after about 100 epochs (data set passes through the training data) no large improvement is seen. There is a clear rift between the two graphs which can have various explanations. It could indicate that the model is simple to learn the training data. This hypothesis was challenged by making the neural network more complex but without making the rift smaller. Since every layer in the model applies a 30% dropout when training we can expect the model to perform better on the validation data which do not experience any dropout. This is believed to be the main reason behind the rift between the two graphs.

	Predicted: Not Chargeback	**Predicted: Chargeback**
Actual: Chargeback	339198	2922
Actual: Not Chargeback	292	435

Table II
Confusion matrix

Class	Precision	Recall	F1-score
Not Chargeback	1.00	0.99	1.00
Chargeback	0.13	0.60	0.21
Accuracy	0.99		

Table III
RESULT OF CLASSIFICATION OF TEST DATA

The confusion matrix shows that the model manages to classify 435 chargebacks right which is 60% of the total chargebacks in the data set. The model classifies another 2922 data points wrongly as chargebacks. 13% of the classified data point is chargebacks.

When examining the results of the model's test the first thing we notice is that the accuracy of the model is 99%. This tells us that the model has captured the true proportions of the data, which means that only 2 of the data is chargebacks. Furthermore, we can see that our model manages to find about 60% of all the chargeback and predict right on 13% of its chargeback predictions. We can from these results quickly rule out using our model as a substitute for NinjaCall's current classifier because of the low precision. In a real-world scenario denying around 1% of the order requests that the company receives, because they have a 13% chance of being a chargeback would be too costly for the business. The cost is associated with lost revenue and unsatisfied customers.

What the results indicate, though, is that the model could be useful as a complement to NinjaCall's current rule-based classifier. In such a case, the model would be used to challenge incoming orders, which are classified as chargebacks. The low precision does, in that sort of case, mean that only 13% of the orders that are challenged are prone to become a chargeback. In other words, a majority of the customers would be unnecessarily challenged. An exact evaluation of the potential consequences of this is left outside the scope of this study, but based on discussions with NinjaCall we conclude that the consequences do not rule out such a solution. The recall of 60% indicates that a majority of the chargeback could be avoided with the help of this model. The potential economic upside of this is covered in the financial part of this study.

B. *Business Case for Chargeback Detection*

Description	Quantity	Unit
Monthly maintenance & development	3	employees
Distribution of working hours	≤ 5	%
Monthly average employee cost	60 000	SEK
Fees to PSP	10	$
Provider cost	20	$
Average basket	12 – 15	$
Average win of chargeback cases	200 – 300	per month
Average cases of chargebacks	1300 – 1400	per month

Table IV
ECONOMIC INFO – CURRENT SYSTEM AT NINJACALL

For NinjaCall, the most important in terms of economic perspective is the savings that can be done in terms of provider cost and PSP fees. From a payback perspective, monthly cash inflows can be regarded as cost savings in this case. The calculations below don't involve the maintenance cost.

Description	Quantity	Unit
Monthly average employee cost	80 000	SEK
Resources	4	man-month
Monthly maintenance & development	0.2	employees / service

Table V
APPROXIMATED IMPLEMENTATION COST BY NINJACALL

$$\text{Payback} = \frac{\text{Cost of project/investment}}{\text{Annual Cash savings}} = \text{months}$$

If we look at the possibility of eliminating all chargebacks in the future, it will take, as calculated below, 5 and a half months to generate enough cost savings from investment to cover the outflow of cash for the investment.

$$\frac{\text{Approx. cost of investment}}{\text{Chargeback fees}} = \frac{80000 * 4}{1350 * (10 + 20 + 13,5)}$$
$$\approx 5,4 \text{ months}$$

If we instead account for the current average amounts of chargeback cases won, it will take around 6 months to reach the break-even of the investment. When a merchant wins a legal dispute regarding a chargeback, they only get back the transaction amount, not the added fees.

$$\text{Payback} = \frac{80000 * 4}{((1350 * (10 + 20 + 13,5)) - (250 * 13,5))}$$
$$\approx 5,8 \text{ months}$$

One more realistic output is the model's capability to eliminate 50% of all chargebacks, which is in line with our model's performance. We assume that NinjaCall will be able to win the same percentage, as before, of the remaining chargebacks. The last mentioned is therefore not included in the calculation. Why this is realistic is discussed in the *Discussion* part below.

$$\text{Payback} = \frac{80000 * 4}{((1350/2) * (10 + 20 + 13,5))} \approx 10,9 \text{ months}$$

As mentioned, a realistic view is that the break-even point for the implementation of our model will be after 11 months. If analysed from a cost-benefit analysis (CBA) over 5 years period, by measuring the benefits of an implementation minus the costs associated with it and the current system.

$$\text{CBA} = ((((1350/2) * (10 + 20 + 13,5)) + (3 * 60000 * 0,05))$$
$$*56) - ((80000 * 4) + (0,2 * 80000 * 56)) \approx 932000 \text{ SEK}$$

NinjaCall will therefore save approximately 930 000 SEK in 5 years if they implement our model and all assumptions and circumstances are fulfilled.

VI. Discussion

A. Model Performance

The results indicate machine learning is a relevant method to consider to tackle the problem that chargeback merchants have to deal with. There are patterns in the transaction data which indicate whether it will turn into a chargeback or not. On the other hand, the results do not indicate that machine learning would be successful. The possibility to achieve better model performance than this study has shown can not be excluded. However, even with the better model performance, there are some factors to take into consideration which could question the usability of the model.

1) Imbalanced Data: The result shows that the model overrates the number of chargebacks in the test set. The true proportion is about 2‰, but the model classifies almost 1% as chargebacks. This is not much taking into account that the training data is resampled to almost 20% chargebacks, but it is relatively large compared to the true proportion of chargebacks. The implication of the model's overestimation is a suffering precision. The resampling technique that has been used in this study is regarded as the most successful, but to truly tackle the problem with imbalanced data future studies should explore other prominent techniques.

2) Implications of Data Being Processed in Existing Rule-Based Classifier: All the data points used in this study have already been ruled as non-chargebacks by NinjaCall's existing classifier. One implication of this is that the remaining de facto chargebacks are particularly difficult to classify. This can of course is a reason why the model is not performing better. The remaining chargebacks in our data set probably have a broader set of reasons. For example, irritated customers, which directly go and complain to the bank, have a more difficult pattern to recognize and are therefore more prominent in our data set. To truly explore the possibility to use a machine learning model as a substitute for NinjaCall's rule-based classifier, it would be necessary to use the raw transaction orders as training data.

3) Either Recall or Precision: In the discussion regarding the evaluation method, it was concluded that the model would either have to show high precision or high recall to meet the needs of NinjaCall. This study has not fully considered this in the choice of the training method of the model. In this study, the model was trained by considering the binary cross-entropy loss. This is a good way to maximize the overall accuracy of the model. Further research on the subject should however explore other loss functions better suited for the imagined area of use. A loss function that penalizes false negatives and false positives differently could result in a model which is more useful for the merchant.

Regarding classification, one could explore the range of recall and precision results that can be achieved with different classification thresholds. This study has only regarded the case with a threshold of 0.5. Future studies should, hence, take, for example, ROC curves into consideration. Furthermore, this study has not explored the possibility to rather use the probabilities of the model outputs. Such a study would however also have to consider how much output could be used regarding chargebacks.

4) Difference in Time, Between Training and Usage of the Model: Another factor is the fact that chargebacks are noticed by the merchant about 3 months after the transaction took place. This means the model can only be trained on transactions which are a few months old but are supposed to be used on today's transactions. It could be that the patterns in chargeback transactions don't change much over time, but if it does there is a risk that even a high-performance model would become obsolete by the time it's supposed to be used. Using a neural network with the possibility to retrain the model on new data alleviates this problem but it doesn't completely remove the time difference problem. This should therefore be subject to further research.

5) Feature Aggregation: As discussed in the theory part of this thesis, other studies have found it successful to use aggregate features in fraud detection applications. In this study, we chose not to use aggregate features to be able to classify new customers with no transaction history as equally good as recurring customers. Future work could focus on chargebacks among recurring customers to be able to take advantage of feature aggregation.

B. Ethical

One question to regard is if this automatic classification algorithm is fair and ethical from the customer perspective. As mentioned in the clarification of chosen features, some information in the data is personal such as users *Country*, *Days since last payment* and *Users most called country*. Some argue that ML algorithms must only use *neutral* data. On the other hand, the definition of *neutral* information is vague. This is something we have discussed and considered when choosing which type of model to use.

When we decided to create a neural network, we argued that if we succeed in creating a deep model which concludes complex and numerous patterns, the model won't be non-ethical in form of just categorising a user's country for example. It will consider that specific feature value, in combination with a lot of other features. By making that combination, the model can use all kinds of data when analysing and creating patterns, without being unethical. The aim when using a virtual neural network is to use as many different features as possible since it might find connections and patterns unthought-of by the programmer. The unethical part isn't the data itself, it's how the model uses it.

C. Economical

As mentioned before, a realistic view of the model's capability is to eliminate 50% of all chargebacks and that NinjaCall will be able to win the same percentage, as before, of the remaining chargebacks. This is realistic since NinjaCall, due to being a merchant, has the burden of proof in legal disputes concerning a chargeback. It's thereby easier to prove that a customer has been using the service than to show that a fraudster hasn't been involved. More easily explained, it's easier for NinjaCall to win the chargeback case if it's caused by unhappy customers (friendly fraud) compared to fraudsters (true fraud).

Furthermore, according to NinjaCall, the share of true fraud versus friendly fraud causing the chargeback is around 50/50. This is both in line with our result and economic evaluation. Below, a SWOT analysis is made to illuminate further analysis concerning financial and non-monetary aspects of the implementation of our model.

centring

STRENGTHS	WEAKNESSES
1) Discover complex /deeper patterns. 2) Eliminates elements of bias compared to existing Rule Based Classifier.	1) The model's parts are complex to understand and work with. 2) Might to simple.
OPPORTUNITIES	THREATS
1) Minimize chargebacks. 2) Less adjustment to new fraudulent behaviour. 3) Reduce costs. 4) Increase costumer satisfaction.	1) Perhaps not best fitted for the problem. 2) Risk for increased costumer dissatisfaction depending on area of implementation. 3) Might not be god fit for future data points.

Figure 3. SWOT analysis

VII. Conclusion

This study has examined how an artificial neural network can be trained to classify transactions as chargebacks and how it can be used in a business context. The resulting model was able to find a majority of the transactions that would become chargebacks, but at the same time, it overrated the number of chargebacks. It was concluded that the model could be used to complement the merchant's existing Rule Based classification system. By doing a *Payback* analysis, this study found that such implementation could reduce costs significantly and if the introduction runs swiftly, profitable in less than a year. Furthermore, a cost-benefit analysis shows that the potential gains over five years amount to almost 1 million SEK.

The classification of probable chargebacks is a complex task. Partly because of the uncertainty of what triggers a transaction to become a chargeback and the arbitrariness which makes it possible for two identical transactions to independently become a chargeback or not. Furthermore, partly due to systems which already are in place to alleviate at least a part of the problem with chargebacks. As the first study of its kind, this paper has given a foundation for further research on the subject and discussed ways to improve the prescribed approach to classify chargebacks, but also other possible ways to use the artificial neural network when dealing with the chargeback problem. We hope this study can be a meaningful contribution to the research area and encourage more studies on how machine learning can be applied to counter the occurrence of chargebacks in a business environment.

References

[1] Gustavo E. A. P. A. Batista, Ana L. C. Bazzan, and Maria Carolina Monard. "Balancing Training Data for Automated Annotation of Keywords: a Case Study". In: *II Brazilian Workshop on Bioinformatics, December 3-5, 2003, Macaé, RJ, Brazil*. Ed. by Sérgio Lifschitz et al. 2003, pp. 10–18.

[2] François Chollet et al. *Keras*. https://keras.io. 2015.

[3] Alejandro Correa Bahnsen et al. "Feature engineering strategies for credit card fraud detection". In: *Expert Systems with Applications* 51 (2016), pp. 134–142. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.12.030. URL: https://www.sciencedirect.com/science/article/pii/S0957417415008386.

[4] M. Engwall et al. *Modern industriell ekonomi*. Studentlitteratur., 2017. ISBN: 9789144116914.

[5] Farbey, B., Land, F. Targett. "D.Evaluating investments in IT. J Inf Technol". In: *Springer link* 7 (1992), pp. 109–122. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[6] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition*. O'Reilly Media, Inc., 2022.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[8] Jökull Jóhannsson. "Detecting fraudulent users using behaviour analysis". MA thesis. KTH, School of Computer Science and Communication, 2017.

[9] Yann-Aël Le Borgne et al. *Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook*. Université Libre de Bruxelles, 2022. URL: https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook.

[10] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

Author presentation

Theodor Günther is a third-year Industrial Engineering and Management student, majoring in Computer Science at KTH Royal Institute of Technology. Theodor has contributed to all parts of this work. With his experience of running his own business, he has especially contributed to the work with a financial perspective and has taken a leading role in the economic analyses and also the writing of the report.

Otto Pagels-Fick is a third-year Industrial Engineering and Management student, majoring in Computer Science at KTH Royal Institute of Technology. Otto has contributed to all parts of this work. Otto has especially contributed by eagerly optimizing the model training and making sure the coding has progressed.

TRITA-EECS-EX-2022:423