Bachelor Degree Project

Time to Strike: Intelligent Detection of Receptive Clients

Predicting a Contractual Expiration using Time Series Forecasting

Author: Jonathan Alklid
Supervisor: Johan Hagelbäck
Semester: VT 2020
Subject: Computer Science

Abstract

In recent years with the advances in Machine Learning and Artificial Intelligence, the demand for ever smarter automation solutions could seem insatiable. One such demand was identified by Fortnox AB, but undoubtedly shared by many other industries dealing with contractual services, who were looking for an intelligent solution capable of predicting the expiration date of a contractual period. As there was no clear evidence suggesting that Machine Learning models were capable of learning the patterns necessary to predict a contract's expiration, it was deemed desirable to determine subject feasibility while also investigating whether it would perform better than a commonplace rule-based solution, something that Fortnox had already investigated in the past. To do this, two different solutions capable of predicting a contractual expiration were implemented. The first one was a rule-based solution that was used as a measuring device, and the second was a Machine Learning-based solution that featured Tree Decision classifier as well as Neural Network models. The results suggest that Machine Learning models are indeed capable of learning and recognizing patterns relevant to the problem, and with an average accuracy generally being on the high end. Unfortunately, due to a lack of available data to use for testing and training, the results were too inconclusive to make a reliable assessment of overall accuracy beyond the learning capability. The conclusion of the study is that Machine Learning-based solutions show promising results, but with the caveat that the results should likely be seen as indicative of overall viability rather than representative of actual performance.

Keywords: Machine Learning, Artificial Intelligence, Time Series, Time Series Forecasting, Controlled Experiment, sklearn, Contractual

Preface

I would like to extend my most sincere gratitude to both Fortnox AB and my Supervisor for giving me this incredible opportunity to work with something that I love. I also wish to thank them for taking the time out of their busy schedules to answer my questions and put up with my endless nagging for more data. Without you, this study would have achieved nothing.

Contents

1 Introduction **1**
1.1 Background . . . 2
1.1.1 Machine Learning . . . 2
1.1.2 Time Series . . . 4
1.2 Related Work . . . 4
1.3 Problem Formulation . . . 6
1.4 Motivation . . . 6
1.5 Objectives . . . 7
1.6 Scope/Limitation . . . 8
1.7 Target Group . . . 8
1.8 Outline . . . 8

2 Method **9**
2.1 Reliability and Validity . . . 10
2.2 Ethical Considerations . . . 11

3 Implementation **12**
3.1 Rule-based Data Extraction & Re-purposing . . . 12
3.1.1 Time Series Construction . . . 13
3.1.2 Time Series Scoring . . . 14
3.1.3 Training Set Creation . . . 15
3.2 Time Series Forecasting . . . 17

4 Results **18**
4.1 Rule-based Series Discovery & Creation . . . 18
4.2 Rule-based Expiration Prediction . . . 18
4.3 Machine Learning-based Expiration Prediction . . . 19

5 Analysis **21**
5.1 Series Discovery & Creation . . . 21
5.2 Rule-based Predictions . . . 22
5.3 Machine Learning-based Predictions . . . 23

6 Discussion **25**
6.1 Similarities of Findings . . . 26
6.2 Effects of Study . . . 26

7 Conclusion & Future Work **27**

References **29**

A Appendix 1 **A**

1 Introduction

In the modern world of today, products and services come in a variety of shapes and sizes that are often adjusted to accommodate its intended audience. Some are one-time purchases that give a buyer access to a product (or less often a service) indefinitely for a specific fee. Common examples of this would include products such as cars and perishables. Others are contractual by nature and represent agreements that run for a predetermined period of time (usually annually, bi-annually, or monthly) during which each new period provides access to a service (or less often a product) and prevents new service agreements (such as those of other providers) from being made until its expiration. Common examples would include services such as insurance and cellular subscriptions. Naturally, this makes it so that businesses that offer these contractual services are interested in when a period ends and a potential customer becomes available: as in, becomes receptive to a sales pitch.

Timing is Everything

This ties in with the issue of timing things right in business, something that has been likened to "swimming with the tide" where something that "may have been a good deal yesterday may not fit today's circumstances" [1]. As could likely be expected, this has perplexed the industry for potentially as long as there has been an industry - be it contractual restrictions, identifying a client's state of mind, or something else [1].

In the case of contracts, a small window of opportunity opens shortly before a client's current contractual period runs out and lasts until a new contract has been signed. It is during this window that competitors to the current provider could approach the potential client with their own offers in the hope of providing a better deal. In doing so, there is a chance that the client would sign a new contract rather than renewing with their current provider.

Naturally, in order for this to be possible, "it is needed to accurately forecast the future [expiration] in order to make right decisions [of when to market]" [2]. Would this window be correctly identified it could be a significant advantage over the competition and allow the opportunity to be seized without giving the opposition much of a chance.

Intelligent Detection

With the advent of, and the rapidly increasing interest in, Artificial Intelligence (AI) and especially Machine Learning (ML), many quickly turn to those for potential solutions to a vast variety of problems [2, 3, 4]. Needless to say, ever since its inception, Machine Learning has redefined several society processes and enabled things that were previously not deemed feasible: such as handling big data or effectively searching the World Wide Web [4].

Despite seemingly underdeveloped, Intelligent Detection of contract expiration appears to be no different in this regard seeing how contractual periods are essentially time-ordered sequences of payment events. As such, they constitute time series whose observations (previous payments and renewals) depend on time and can therefore be used to forecast the future (period expiration) [2]. After all, time series forecasting is relatively well-established and has been an active area of study for over a century in a variety of sciences [2, 5]. Being an event sequence, it could also qualify as an *event prediction* problem as described by [5] where the focus is on predicting specific, potentially rare events within a certain window of time, much like predicting an expiration in a myriad of payments.

Nevertheless, it stands to reason that, given a sufficient amount of good data, an estimation - a prediction - of when a period could end should be plausible to be produced using a Machine Learning-based approach. This is, in essence, the intent of this paper which aims to investigate the feasibility of estimating contract expiration dates using Machine Learning methods and answer the following research questions:

1. Can a contract expiration be predicted with acceptable accuracy using Machine Learning methods (in this case, Time Series Forecasting)?
2. Is the accuracy better than, or comparable to, a rule-based prediction approach?

While this is almost certainly a topic far too large for a single paper - or even 100 papers for that matter - the primary goal is to address the feasibility of this approach overall, perhaps in the most basic sense, and compare it to a purely rule-based approach. Should the results be sufficiently satisfactory, the implementations could serve as a primitive proof-of-concept that could be used as a basis for future work.

1.1 Background

In order to get the most out of this paper, certain prevalent topics have to be put on common ground. These are *Time Series* as in what they are and for what they can be used and *Machine Learning* as in what it is, what it can do in a general sense, and how it can be measured. This is, however, a brief overview as a more detailed view constitutes several books in its own right.

1.1.1 Machine Learning

Machine Learning is the part of the Artificial Intelligence-suite that deals with learning from past experiences. In order to be intelligent, a "system that is in a changing environment should have the ability to learn" [4]. If a system can learn and adapt to changes in its environment, "the system designer need not foresee and provide solutions for all possible situations" [4]. More specifically, it builds on the theory of statistics in building mathematical models that allow for making inferences from samples [4]. Those samples may be example data (such as *ground truths*) or alternatively past experiences. This allows an intelligent machine to recognize patterns, make somewhat informed decisions, and finally learn from their past decisions.

Supervised Learning

One type of learning, and indeed the relevant one for this paper, is Supervised Learning. This is done through the classification of unknown entities into known categories. Using a data set of ground truths (essentially classifications that are known to be true) - known as training - machine learning algorithms can “predict” the appropriate category of an unknown input, known as *generalization*. While the result is not always accurate, and with a multitude of different algorithms, or classifiers, available that all come with their own sets of strengths and weaknesses in terms of accuracy and complexity, it can still be considered to be a good and useful approximation [4].

Two of the more common algorithms, which are also used for this study, are *Neural Network* and *Tree Decision Classifier*. The former is a very powerful algorithm and is considered to be a central part of what is known as *Deep Learning*[1]. It is created in an attempt

[1] Deep essentially refers to having more than three layers, Neural Network or not. See [6].

to mimic the human brain by organizing nodes (artificial neurons) within a layered context where each successive node places a specific requirement (so called "weights" with accompanying thresholds which are selected during training) on the input data. Should the input data satisfy the constraints on any given node it will be passed along by the current node to the node whose conditions were met, allowing the input data to travel to any node on the successive layer, in order to, eventually, reach the final node and thus a prediction (output) [6]. The latter is similar to a binary tree where there are any number of nodes in a chain where each node represents a question which is then connected to sub-nodes using *edges* that represent answers, either "yes" or "no" (both nodes and edges are created during training). The input, then, starts at the root node and then travels down the chain, successively answering questions, until it reaches the leaf node and in doing so a prediction (output). Both algorithms revolve around finding the most optimal path in order to reach the most accurate prediction.

The prediction accuracy of Supervised Machine Learning can be measured in several different ways with the use of an aforementioned training set and a *validation set* (which is a data set of unknown entities that need to be classified). One way is to validate the accuracy using the same data set as was used for training. While this is a fast and easy way of getting an estimate, it comes with a variety of caveats, such as over-training if the data set is too small [4]. An arguably better way is that of *cross-validation* where the training set is split into two smaller sets, where one part is used for validation and the other for training [4]. This is then repeated for a number of iterations, re-splitting the data each time. This allows for a greater level of confidence since the validation set is not a part of the training set, making them guaranteed unknowns while also allowing for gauging the best training set composition.

Supervised Learning also allows for *Reinforcement Learning*, which revolves around trial and error where an agent (in this case likely a machine) makes decisions in some environment. The agent is then subsequently given a reward or penalty based on the decision taken (as in, the outcome of said decision). After a number of executions, it should learn the preferable policy which constitutes the sequence of actions that maximizes the total reward [4]. This is different to, for instance, *unsupervised learning* which has no ground truths or reinforcement learning.

Regardless of any chosen accuracy measurements, Machine Learning is always subjected to *Noise*. Noise is "any unwanted anomaly in the data" which may make it "more difficult to learn and zero error may be infeasible" [4, 7]. It could arise from imprecision or omissions, accidental or otherwise, in the recording of the data attributes which could end up shifting *data points* (a data point is one instance in a set of data) or negatively affecting the weights during training. Another point of concern is that of categorization errors where one or more instances in the training set are assigned to incorrect categories which may cause, for instance, false positives and vice versa [4].

Machine Learning Today

Today, Machine Learning is a widely-used, popular solution to a vast variety of problems. For instance, it can solve many problems in machine vision, speech recognition, and robotics, as well as being an important part of database management, security systems, and network security [4]. It can also be used to track and predict stock market developments [8, 9], technical equipment failures [5], road safety developments [7], meteorology, and air pollution [10]. Some even consider committing to complete workplace replacements with intelligent, robotic coworkers - although not everyone seems to be quite as enthusiastic at this prospect [3].

1.1.2 Time Series

A Time Series is a "time-oriented or chronological sequence of observations on a variable of interest" [7, 11]. This essentially means that it is, simply, a "series of data points ordered in time" where the time itself (usually a timestamp of some description) is an independent variable and each data point is accompanied by zero or more associated variables of interest [2, 7, 8, 11]. Data points are collected at a certain rate, known as the rate variable, which is generally at equally spaced time periods. Those time periods are typically daily, weekly, monthly, quarterly, or annually, but "any reporting interval may be used" [11]. The collected data points can be in any format that is deemed desirable, such as instantaneous (such as an individual purchase), cumulative (such as the total number of sales over a month), or statistic (such as something that in some way reflects activity during the specified time period), as long as the same format is used consistently throughout the data set [11]. They are very similar to *event sequences* which are essentially the same, being sequences of timestamped observations, but have support for a wider range of associated variables and can predict specific events within a given time frame [5].

While Time Series can serve a worthy purpose on their own, such as through something known as a *Time Series Plot* that can be used to, as an example, track changes over time in an area of interest, the primary purpose of Time Series, in Machine Learning, is its ability to *forecast* the future [11].

Forecasting the Future

A forecast is "a prediction of some future event or events" based on patterns found in past observations and developments and can be applied to a large selection of different topics [2, 8, 11]. This is, naturally, an important aspect in a variety of fields as the "prediction of future events is a critical input into many types of planning and decision-making processes" [11] and in many areas it is necessary to "accurately forecast the future in order to make [the] right decisions" [2, 9]. This includes, but is not limited to, finance, management, environment, medicine, social science, politics, and many others [2, 8, 10, 11].

As noted by [11], forecast-related problems are generally classified as short-term, medium-term, and long-term. Short-term problems "involve predicting events only a few time periods (days, weeks, and months) into the future", whereas medium-term extend between a year or two, and long-term can extend far beyond that. Short- and medium-term forecasts are generally used for "activities that range from operations management to budgeting and selecting new research and development projects" while long-term forecasts are more concerned with more overarching strategic planning.

1.2 Related Work

While this particular issue does not appear to have been actively researched, it is related to time series and forecasting. Since those have both been actively researched and used for over a century, there is naturally a large body of research on the matter [2]. Because of this, this section will list a couple of examples of research that has been done on those topics and how they are used around the world today.

In [5], the author presents a forecasting problem suffered by AT&T in relation to predicting telecommunication equipment failures from a large pool of varying alarm messages. It is stated that being able to predict rare events in sequences of varying events with potentially different features is an important problem that normal learning methods are not sufficiently evolved or adapted to solve (the difficulty of predicting rare events is also noted by [10]). The paper discusses event sequences, which are essentially time series but instead of predicting the next n events, it "predicts rare events by identifying predictive temporal and sequential patterns in the data" which constitutes the rare event of an equipment failure. The proposed solution is a "genetic algorithm based machine learning system", known as Timeweaver, that would be able to predict equipment failures from 110,000 alarm messages dispatched by AT&T's 4ESS switches. It is then compared to several other solutions, such as C4.5rules, RIPPER, and FOIL, but is shown to outperform all existing methods at this particular prediction task.

In [9], a framework based on tempered stable innovations for evaluating stock market risk exposure during distressed market periods, as well as extensive empirical performance testing of said framework compared to industry standard models, is presented. The authors begin by exploring the limitations of standard time series models which rely on normal innovation and are widely used in the industry when it comes to forecasting financial market meltdowns. They then offer a solution in the form of a framework that is capable of forecasting both extreme events and highly volatile markets with higher predicative accuracy than that of any standard models. They also provide a relatively extensive empirical study of the performance of their framework compared to other models by applying them to the "analysis of the S&P 500 index during highly volatile markets". They find that standard models either fail completely and are empirically rejected, or do not "provide a reliable forecast of the future distribution of returns, even if they account for volatility clustering". The results from their own framework, however, indicate that it has better predicative power in measuring market risk. The authors conclude that this provides empirical evidence of their framework performing better than the alternatives in a volatile market environment.

In [10], time series prediction models are used to forecast volcanic air pollution in Hawaii where volcanic smog, known as vog, has become a major issue after the continuing eruption of Mount Kilauea. By looking at data sets consisting of time series for SO_2 and SO_4 levels from various coastal locations spanning Hawaii, as well as the city of Hilo, the author measured and compared the forecasting results using two different models: frequency domain and neural network. Overall, the results were promising and showed that the concentrations of SO_2 and SO_4 could be forecasted reasonably well, at least on average. They were, however, unevenly distributed across the different models with the frequency domain algorithm yielding the best overall accuracy over short horizons. While competitive and more favourably presented in literature, the author notes, the neural network could not quite compete outside of certain time frames. The author concludes that the models "capture the central tendency of the data, but are less effective in predicting the extreme events", a conclusion that is in line with [5].

In [7], the authors discuss the complications and risks of making erroneous inferences when trying to create road safety observatories using time series of sequential observations of various types of safety performance indicators (such as speeding, alcohol, etc.) or general road statistics (such as traffic accidents, kilometers driven, etc.). They state that most traditional techniques make incorrect assumptions about the data points and ignore important properties, namely the serial dependency between the observations. This noise, they note, can result in under- or over-estimations and could consequently produce erroneous inferences. They then proceed to suggest various rigorous statistical techniques used to overcome serial dependency issues. Those being time series analysis techniques of varying complexity that are employed to "describe the development over time, relating the accident-occurrences to explanatory factors such as exposure measures or safety performance indicators, and forecasting the development into the near future". They find that traditional regression models, while the easiest to apply and likely often sufficient, are shown not to "properly capture the time dependencies between consecutive observations". Structural time series models turn out to "reduce to classical linear regression when the unobserved components are treated deterministically" which results in no obvious benefit. Dedicated time series analysis techniques are shown to have the best overall performance and accuracy, despite needing access to excessively large data sets - a requirement that is currently not possible to satisfy properly according to the authors. Finally, they note that all approaches have their pros and cons and conclude with giving some general recommendations to using time series models in road safety research.

1.3 Problem Formulation

While, as previously noted, Machine Learning has been used to solve a variety of other Time Series-related problems, its feasibility in determining whether the “time is right” when predicting specifically contractual periods remains to be investigated as available data is suggestive of its viability, but inconclusive. Naturally, the ability to predict contractual expiration is something that is of interest to many in the industry, perhaps every entity that offers some form of contractually-based services, and just so happened to coincide with an initiative taken by Fortnox AB, who is one of many with a vested interest in the field, to do just that. Fortnox proposed a collaboration by offering access to its vast amounts of data and expertise to use for a project investing the applicability of ML in this context. The project entails the complex task of detecting when a window of opportunity appears in a given client’s currently ongoing contractual period, irregardless to what contract that may be, which in turn could indicate receptiveness to a sales pitch, all by using Machine Learning-methods and Time Series. As such, the goal of this project is to be able to answer, or at least attempt to answer, the question: "when does the client’s current contractual period end?" which equates to answering the first of the aforementioned research questions. Fortnox has, in the past, attempted to solve this particular problem using rule-based solutions, but they have never been deemed sufficiently accurate. This, in turn, resulted in the second of the two research questions which entails a comparison between the two solutions that this project also aims to cover.

1.4 Motivation

First and foremost, and perhaps most importantly, this problem was brought to light by a business involved in the relevant industry, that of offering contractually-based services. In fact, it is a problem that they, and likely many others, have been looking into for a while to no avail. However, as mentioned in earlier sections, this particular problem that this project aims to investigate is not restricted to Fortnox and their use case, or even necessarily other entities within their immediate contractually-based services industry, but could rather be applied to all entities within a vast array of industries. While the notion of "Timing it Right" would likely be the most relevant to industries dealing with critical timing, such as stock market fluctuation prediction, as noted by [9] or, indeed, time-based contracts, as is the topic of this project, those all fall in the realm of Time Series Forecasting. As such, any industry that deals with data streams that feature some form of timestamped variable or variables of interest could benefit from the findings of this study, as it is, in essence, an extension of an already well-established system of time-based analysis. Such industries include, but are not limited to, the examples provided in the Related Work-section as well as fraud detection in finance, diagnoses in medicine, and optimizations in manufacturing and telecommunications [4].

While it is clear that Machine Learning and Time Series are used for many other purposes, the apparent lack of a viable solution to this specific problem, however, suggests that there is a need for extensions to the Time Series-methodology by applying it to new areas and measuring its viability. As such, it should be noted that it is currently not known whether Machine learning can solve this problem, as it is different in that it does not predict a variable of interest, but rather the independent variable that is the timestamp of the Data Point itself[2], which makes this project into a somewhat scientific venture. However, judging by its viability in similar Time Series-based problems, it stands to reason that there is a relatively high chance of success. As such, overall, this problem seems to apply - to the relevant industries - on both an industry- and, to a lesser degree, a scientific level.

1.5 Objectives

The objectives of this project primarily revolve around the aforementioned goals of evaluating potential solutions and determining which would best satisfy the requirements. This roughly breaks down into the following objectives:

O1	Investigate available data and identify key characteristics.
O2	Implement rule-based solutions for series construction and expiration predictions.
O3	Implement the Machine learning solution and a solution for training set creation.
O4	Produce controlled testing datasets consisting of fabricated series and real-world examples.
O5	Fit the models and run experiments to evaluate the accuracy of both solutions.

The first task was to investigate the data that Fortnox had provided and identify any key characteristics that would be needed to do further processing. As there were no premade or off-the-shelf datasets available, all raw data that was available had to be processed and evaluated for relevance and usefulness (Objective 1).

Then, using the insights gained from the first objective, the necessary rule-based solutions had to be implemented. This entailed implementing a solution that could take raw and unordered timestamped events and producing sufficiently accurate and usable time series consisting of time-ordered event sequences where each sequence represented a contract. These time series were then used to produce rule-based forecasts of the assumed expiration of each contract (Objective 2). Once that was done, the Machine learning solution had to be implemented as well as a minor, secondary solution that could be used to convert processed time series, such as the outputs of Objective 2 and Objective 4, into a training set that could be used by the ML models. Essentially, this entailed setting up the ML models and creating a short script that converts the time series into a usable format with appropriate categories (Objective 3).

In order to achieve reliable accuracy measurements, the next step would be to produce two datasets containing known series. These will be divided into fabricated (series that were created explicitly for this study) and real-world examples (verified series acquired from the raw data provided by Fortnox) and used as truths when performing a controlled accuracy measurement (Objective 4). Afterwards, the Machine learning solution has to be fit with the training sets (using the script from Objective 4) created from the output of Objective 4. The accuracy of the rule-based and the Machine learning predictions alike

[2] See section 1.1.2 for more information.

are then measured and compared to one another in order to evaluate the applicability and performance of both. The accuracy of the series identification and creation of the former is also measured as a complimentary metric (Objective 5).

Finally, and as stated prior, this is uncharted territory and therefore the success of this project cannot be guaranteed. However, as has been mentioned, Machine learning has solved similar issues in the past and therefore it is expected that this problem can be solved and the objectives completed.

1.6 Scope/Limitation

As mentioned in the introduction, it was not clear whether Machine Learning could offer a solution to this problem in the first place. This naturally made it significantly more complex in nature but also caused the evaluation of employed approaches to be of increased importance. To accommodate this, the scope had to be narrowed - primarily in the form of focusing on interests of Fortnox and less so on other concerns. In practise, this limited the chosen approach to be based on the promising prospect of Time Series forecasting, albeit with some project-specific alterations.

Furthermore, due to time constraints, data selection methods had to be adjusted as it is a costly endeavour. As such, the overall selected data amount had to be reduced and selection criteria had to be altered to accommodate. It also limited any potential output implementations. Granted, the different implementations are supposed to work as one solution, with rule-based training set creation followed by Machine learning-based predictions. However, due to the aforementioned restrictions, this is not planned to be a finished software solution at the end of the study but rather a proof-of-concept that could be further developed.

1.7 Target Group

The primary audience, or target group, is, naturally, Fortnox as they desire a solution that could be used to accomplish current and future business goals. That said, and as hinted in the Motivations-section, the problem and its specifics are currently seemingly underdeveloped in the industry and could prove interesting, not only to similar businesses but also researchers within the area of Machine Learning.

1.8 Outline

First of all, the method will be laid out, including reliability, validity, and ethical concerns associated with the given approach. Then the implementation will be described, its functionality and limitations. That said, in this particular project, the implementation would entail either a primitive proof-of-concept, or nothing at all if all approaches failed to deliver satisfactory results. The results-section will lay out the actual results of the chosen approaches and their reliability ratings in regards to provided data sets. Analysis will take the results to a somewhat more thorough level and investigate what they mean and why the results are relevant for this paper. Discussion will provide an overview of why (or why not) the results were satisfactory in relation to the study goals and research questions. Conclusion will wrap up the findings of the paper and what could be reasonable future work.

2 Method

First of all, there were certain constraints on this project that had to be respected. As previously mentioned, the goal of this project was to investigate whether Machine Learning could be used to predict the expiration of contractual periods on behalf of Fortnox. This was in response to previous unsuccessful attempts by Fortnox at finding satisfactory rule-based solutions to this same problem. This was in part due to its exponential growth in complexity as more parameters were added in order to increase its accuracy and area of applicability. Because of this, and the ever-increasing popularity, understanding of, and usage of Machine Learning, a solution of this type was proposed.

However, it was not known whether this particular issue could be resolved using Machine Learning methods as it did not appear to have solved sufficiently similar issues in the past. Despite this problem being a Time Series issue with sequences of payments, the inherent complexities of the subject domain, such as varying contractual periods, payment plans, customer-instigated changes, and so on, the viability of an ML-based design could not be confidently assumed.

To add to this complication, there was no premade dataset of contractual payment time series available and instead datasets had to be manually selected and verified from whatever raw data was available as well as fabricated for the explicit purpose of this study. As such, this project had major inherent uncertainties and there were no tried-and-tested reference solutions available. Arguably being somewhat of a new venture, a method capable of accounting for a relatively large amount of unknown variables was necessary.

Additionally, some form of frame of reference would be required in order to explore the feasibility of Machine Learning methodology adequately and that an ML-based approach would be sufficiently beneficial to warrant its creation or subsequent development. Should it not perform better than a rule-based solution, it would be unlikely that Fortnox would be interested in investigating this further. To accomplish this, a complementary rule-based solution, tailored to the domain data, was created and used as a point of comparison to see whether the ML outputs were preferable. Should the results prove satisfactory, the implementations could serve as a proof-of-concept that could be used by Fortnox for future developments. This same rule-based solution was also intended to produce training sets, although that was not a feasible task in this particular study due to aforementioned data problems.

Finally, due to having access to *ground truths* in the form of real-world data that could be used for validation, the study was essentially broken down into two parts, each employing a specific type of method: *design science* for the former and *controlled experiment* for the latter. The first part involved developing the previously established solutions which were both seemingly new ventures while also requiring a great deal of customization to be compatible with the provided ground truths while also accommodating any and all insights gained from investigating said data. The second part involved validating the accuracy and performance of the solutions, both the rule-based and the machine learning-based solutions, using the aforementioned ground truths as well as fabricated data derived from them. Due to those truths, this validation could be performed in a controlled and measured way using a selection of real-world examples instead of exclusively theoretical scenarios. The ultimate goal being to investigate which solution would provide a more reliable prediction.

2.1 Reliability and Validity

There were some inevitable concerns in regards to this study, many of which likely stemmed from the uncertainties inherent in this type of study. Being a purely exploratory venture, these concerns were not deemed significant by any involved party as they were considered a natural part of the experiment. The points of concern were the following: data sources, the rule-based solution, and Machine Learning applicability and subsequent forecasting accuracy.

The only data source available, at this stage, was internal log files of bookkeeping records within Fortnox. As such, they did not contain any actual transactions and had shown, in some cases, to have large amounts of noise. This included, but was not limited to, incomplete sequences, missing segments, and odd fluctuations - in both the dates and values of the time series. This data inconsistency could likely be because of time discrepancies between an actual transaction taking place and said transaction being book-kept in addition to unidentifiable contractual changes and other abnormalities. As such, data could not be directly extracted and used by the rule-based solution to create a dataset at this stage. Instead datasets had to be created for the explicit purpose of being used for this study. This included, as mentioned, a dataset containing samples that had been extracted and verified from the raw data source and another dataset of fabricated series that had been created in the former's image. This was done in order to incorporate ground truths into the study while at the same time promoting more coverage. However, due to the inconsistencies in the raw data, the correctness of either dataset could not be entirely guaranteed - despite domain expert involvement. That said, this was the only data source available to use for this project: effectively creating a constraint that could not be resolved at the time being. In response to this, the solutions were fitted to have a degree of leniency when encountering potential inconsistencies as to minimize the fallout of this particular complication.

While effectively an independent solution for the purpose of this study, the rule-based solution came with a number of caveats. It was, and still is, essentially, a rule-based approach to building time series consisting of sequences of contractual period payments, including any and all changes that occur during a contractual period (such as something being added or removed which causes a change to the owed amount), and then producing estimations of when those series could expire. Those series and expiration assumptions could then, at a later stage, be used to create training sets, if sufficiently accurate. The rules themselves correspond to a set of regulations that contractual series are supposed to follow, provided by domain experts within Fortnox. Abstractions of those same rules are then used to score all potential series that are discovered in order to select the most likely sequences for further processing.

Naturally, a rule-based approach is not guaranteed to be accurate, nor was this solution intended to be perfect at this stage (especially considering the problematic domain and associated time restrictions). Indeed, it was prone to produce errors and other incorrect time series, either on its own or because of noise in the raw input data. That said, should the results be sufficiently satisfactory, the rule-based logic could either be further developed, or left as is and be re-evaluated at a later stage. Either way, this would affect the results of the rule-based solution and in doing so the results of the study overall as the point of comparison may be at least partially compromised.

The applicability of Time Series was, primarily, assumed based on its ability to solve relatively similar issues and that the problem data consisted of time-based sequences, albeit unprocessed and incomplete, and its subsequent forecasts. While not conclusive, it proved sufficient to warrant further investigation. The used forecasting models were chosen primarily based on internal assumptions, but also from overall accuracy ratings (see, for instance, [4]). That said, the selected models were not guaranteed to provide the most optimal results which was why the models in question were to be tested and compared to the rule-based solution in order to determine the most optimal performance and, at a later stage, disqualify inappropriate or lacklustre models. Furthermore, by having access to ground truths, it was possible to perform probability calculations which could be used to measure the prediction accuracy (dependent variable) of the models' outputs given domain-relevant data (independent variables), and by doing so, potentially prove the viability of the models and the design overall. This is despite the arguably roundabout way of doing so in this particular study due to available data restrictions.

2.2 Ethical Considerations

By default, there were no explicit ethical considerations in an experiment of this nature. However, there were some potential complications that had to be considered due to circumstances inherent in this particular project.

Firstly, the project was performed on behalf of an external company. As such, certain data and/or practises may be intentionally or otherwise infused in one or more areas of the design and/or analytical process which should not become publicly available. To account for this, company representatives should be consulted and any parts which are deemed to be potentially integrity-violating would promptly be excluded from this report. This means that the experiment contained herein may or may not be entirely reproducible as data used or assumptions based on company-exclusive insights incorporated into the implementation may not be available.

Secondly, the used data may or may not include information which can, in some way, be traced back to one or more individuals that are in some way connected to Fortnox or its subsidiaries. In such a case, the discriminating information will be promptly excluded so that the privacy of the individuals in question would not be compromised.

Lastly, while not directly relevant to this study per se, there are countless ethical aspects inherent in the use of Machine Learning and AI as a whole. Much like the never-ending discussion of who is at fault in an accident involving self-driving cars, there are inherent and unavoidable complications in letting machines make decisions. Regardless of safety measures and priorities, the outcome does not seem to be completely predicable and as such not verifiable. This type of issues applies to the use of Machine Learning in predicting a contractual expiration as it could be inconvenient, both for the service provider and the customer, if provided predictions cause unnecessary contact. Furthermore, if the same solution is used for more significant and critical operations where, as an example, automatic signing of contracts is involved or even physical harm is a plausible outcome, this problem increases exponentially. As for this study, however, this question is not directly relevant as it is merely a proof-of-concept investigating feasibility, and not a finished solution that is ready for production (or even intended to be used in production as of now). Regardless, the study does employ a variety of accuracy measurements in order to produce a, perhaps somewhat basic, understanding of how this solution could behave if employed in a real-world environment.

3 Implementation

The implementation in this project consisted of multiple different stages. This was because the implementation consisted, in essence, of two different solutions. First, the rule-based solution, with series discovery and construction along with its accompanying expiration predictions, had to be developed. Once that was done, a method of converting the output of the rule-based solution to a Machine Learning-friendly format was necessary. Lastly, the ML-models themselves had to be implemented and trained on the produced datasets (which were created with the help of the conversion solution).

Because of these stages, this section will be divided into a number of subsection, each dealing with its own particular stage. An example will be included throughout to show how each stage affects and transforms a particular input. Note that the example is simple in nature for the sake of clarity and it is entirely random and is not linked to Fortnox in any way. Additionally, the example has been truncated (indicated by "...") which denotes that similar instances (increasing date of payment but with the same amount) have been omitted. The example input can be found in table 1 below.

Date	**Amount**
2012-01-03	295
2012-02-01	295
2012-03-02	304
2012-04-02	304
2012-05-01	341
...	341
2013-03-02	364
...	364
2013-07-03	253
2013-08-02	364
...	364
2014-03-02	370
...	370
2015-03-04	379

Table 3.1: Example of input payment series.

3.1 Rule-based Data Extraction & Re-purposing

In this section, the discovery and construction of time series (in this case: payment series) from raw data that is in no particular order will be described. This includes all steps of each of those processes. The first step is producing all potential series that could be derived from the available payments (a simplified overview of this process can be seen in Figure 3.1), after which the produced series are scored based on a number of criteria. The highest scored series are then selected to be transformed into a training set. This entails producing rule-based expiration predictions for each selected series and then converting the entire dataset into a format that is usable for a standard Machine Learning algorithm.

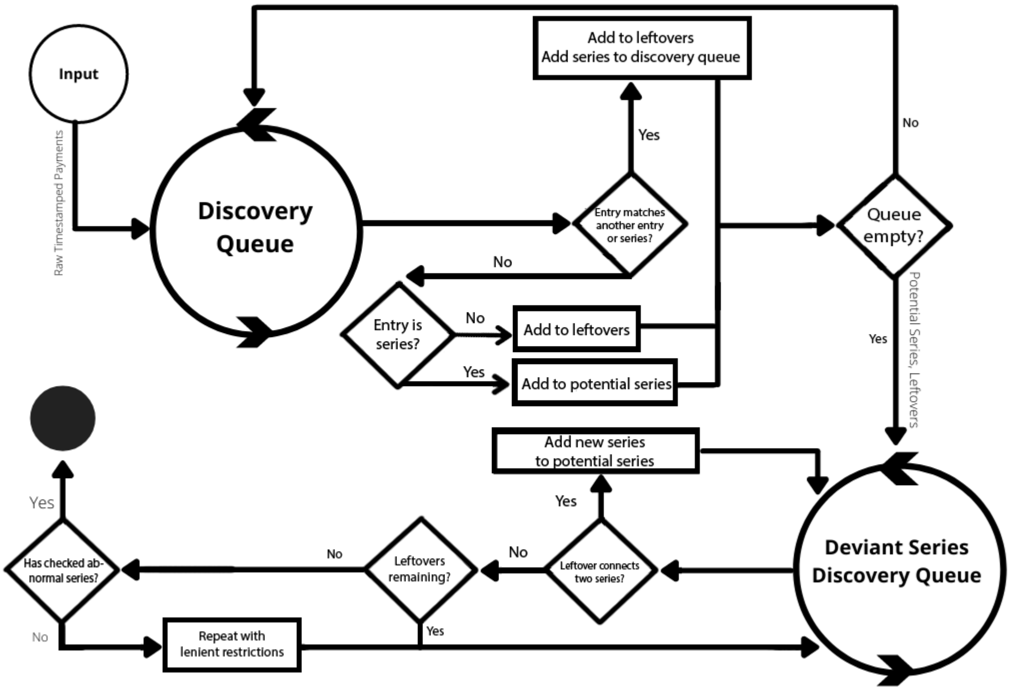
Input
Raw Timestamped Payments
Discovery Queue
Add to leftovers
Add series to discovery queue
Yes
No
Entry matches another entry or series?
No
Queue empty?
Yes
Potential Series, Leftovers
Entry is series?
No
Add to leftovers
Yes
Add to potential series
Add new series to potential series
Yes
Deviant Series Discovery Queue
Yes
Has checked abnormal series?
No
Leftovers remaining?
No
Leftover connects two series?
Yes
No
Repeat with lenient restrictions

Figure 3.1: Time Series Construction Process

3.1.1 Time Series Construction

The first step is to match identical, or as good as identical, payment entries with each other to produce very basic series. Those series follow very basic rules, such as follow a certain interval denominator and have roughly the same payment amount, give or take a few percent to accommodate minor fluctuations. However, this produces very simplistic series and does not account for the often turbulent fluctuations found in many real-world payment series, but does serve as a basis for further processing. In terms of the example series, this stage would connect the series: 295, 304, 341, 364, 364, and 370, with the leftovers being 253 and 379.

As such, the next step entails attempting to discover potential series by connecting those series to each other and to any other leftovers from the previous step. This is done semi-recursively using a Queue-like data structure where each newly discovered series is added to the back of the queue. This is because each discovered series could in turn offer one or more additional potential series that require processing. For example, a leftover entry could act as a connector between two different series. At this stage, the entries go through stricter selection criteria than the previous step. As all selections are based on the earlier given series, the intervals have to match so that yearly payments are not connected to monthly ones, and so on. The fluctuations in payment amounts, while more lenient than before, are still restrictive based on advice given by Fortnox. This is, in general, due to payment series generally having little, if any, fluctuation outside of anomalies.

This is done until the pool of potential series has been exhausted and there are no more series left to discover. As for the example series, this would connect the 295s to 304s, then to the 341s and 364s. The second set of 364s would then be connected to the 370s. Finally, 379 would be appended as a fitting leftover, leaving 253 as the final leftover. As could be observed in the previous step, the entry with 253 is still unassigned after both steps whereas it would appear to be a fitting connector between the two 364 series which in turn would connect the independent series fragments into one whole, complete, series.

Additionally, while not present in the example series, there is a variety of different abnormalities that can appear in otherwise perfectly fluent series. Those include, as demonstrated above, volatile fluctuations as well as overlapping or missed payments altogether. To account for those special cases, an additional step had to be introduced which would attempt to connect and complete potentially abnormal series. This was determined after domain experts at Fortnox concluded that various contractual factors, human error, software faults, amongst many other, could cause more inconsistencies than what had originally been assumed. This issue is of extra significance due to, as previously mentioned, the raw data containing bookkeeping entries and not actual payments. This makes it further susceptible to date inconsistencies as there is not necessarily any direct time connection between the two events. For the example series, the output would be a complete series, using the unassigned 253 to connect the two series into one whole.

3.1.2 Time Series Scoring

The final step in the time series creation process is the scoring and selecting of the most likely, and indeed promising, series. This is done by calculating a score for each series based on a variety of different factors, all deemed important by consulted domain experts. This is intended to weed out undesirable, unlikely (or even impossible), or otherwise poorly constructed series and provide only the best series for further processing, even if this would, in rare cases, result in no series at all. The used criteria and method of score calculation are based on simple foundation and designed to be easily modifiable so that criteria can be added or removed as new information becomes available in further developments. The used criteria are listed below:

- Series Length
- Series Span
- Span Accuracy modifier
- Span Type modifier
- Amount Consistency modifier

Series length (L) is an integer that, simply, represents the length of the series and correlates to the *n* data points, or entries, found in the series. This measurement is used to promote series that use more data points than those that use few.

Series span (S) is an integer that represents the total span of the series, as in the *n* months covered by the series, starting at the first data point and spanning the entire series up to the last data point. This measurement is used to promote series that span a longer period of time as those are not only preferable but indirectly balances out the low score gained from series length for, for instance, yearly payment series.

Span Accuracy modifier (A) is a floating point number between .75 and 1 that represents how well each entry abides by the target interval window. As an example for a monthly series, how close to a month there is between each entry. This is achieved by retrieving the aforementioned difference between each set of entries, calculating an average, and then inverting it by dividing 1 by the result. This measurement is used to promote series where the payment intervals are consistent with little fluctuation.

Span Type modifier (T) is an integer representing a unique modifier for the relevant interval window. This measurement is used to promote series that are of either a preferable or more likely interval. An example would be a monthly series being preferred over a quarterly series as those are, according to domain experts, far more common. Due to the dynamic nature of this particular modifier as different use cases prioritize different spans and differences in the structure and consistency of available data, there is no universal truth to what those modifiers should be. However, in this case where monthly payment intervals are significantly more common, a good starting point would be something like: monthly: 2, yearly: .5, quarterly: .25.

Amount Consistency modifier (C) is a floating point number between .75 and 1 that represents how faithful each payment amount is to the overall series. For a series of identical, or very similar payment amounts, this modifier would return a higher number than a series where payments fluctuate wildly. This is achieved by retrieving the difference in payment amount between each entry and that of the average for the whole series, calculating an average, and then - just like before - inverting it by diving 1 by the result. However, due to inconsistent, yet legal, series this modifier had to be capped as not to invalidate otherwise valid series. This measurement is used to promote series that have consistent payments as those are, generally, more common.

Finally, the score for any given series is calculated as follows: *(L + S) * A * T * C*. The highest scored series are then extracted from the pool of available entries (corresponding to the initial pool of unconnected entries). This is intended to accommodate overlapping or even duplicate series, allowing for a series to be extracted multiple times if a sufficient amount of entries are present to construct more than one. This extraction process lasts until either the pool runs out of entries or there are no more series to attempt extraction. An example of desirable - and very consistent - series, being the running example input, can be seen in Figure 3.2 where scored attributes (such as abnormalities and span and amount consistency) have been highlighted.

3.1.3 Training Set Creation

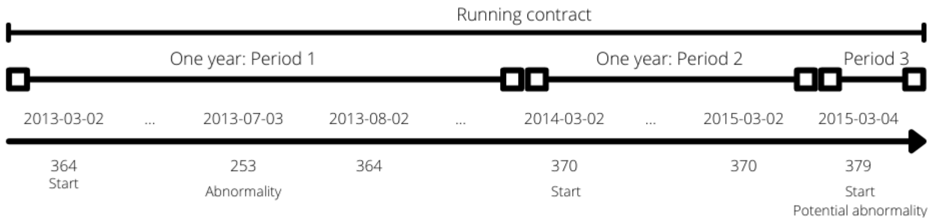

Figure 3.2: Periods Making up a Contract

The previous two sections have dealt with the actual creation of payment series out of raw, unordered data. The next step is to use the created series, assign a reasonable rule-based expiration assumption to each series, reformat the series into a Machine Learning-friendly format, and finally produce a training file that can be used in further processing. Those assumptions are based on the notion that each series is split into one or more subsets of contractual periods, where each period, in accordance to standard domain practises, runs for one year regardless of payment intervals. Because of this, the point of interest in each series is the start of the latest period which can then be used to calculate the month when the ongoing period ends and thus when the contract itself expires. Refer to the example input in Figure 3.2 for how periods construct a complete contract.

However, since some series are known to be incomplete, the first data point of a series cannot be assumed to be the start which requires a more complex approach to discovering the latest period. As such, the points of change in each series need to be investigated in order to discover any contained periods. However, due to fluctuations and other abnormalities, this task is often not as clear cut as simply identifying fluctuations. Furthermore, contractual payments are always paid in advance which has to be taken into consideration.

By looking at the last known data point and working its way backwards as long as the payment amounts remain consistent, identifying the latest known period comes down to determining, with adequate accuracy, when the consistency ends. This point could, if significant enough, represent a period start (which are generally slightly divergent from the period that is started), an unusually bold fluctuation, or alternatively a data point belonging to a different period entirely. Which alternative it is should be possible to determine by looking at its surrounding data points.

If the data points opposite to the divergent data point are consistent with the initial data points, then it is likely a fluctuation. If the opposite data points are consistent with the divergent data point it is likely a new period entirely. If they fulfil neither of those two conditions, it is likely a divergent start for the current period. There are exceptions to this and they will be discussed in later sections, but examples of all of those cases can be seen in Figure 3.2. Once the latest period start has been determined, the margin between the latest start and the last know data point is calculated. Since all periods are one year in length, removing the result from a complete year results in an assumption of how far into the future the series ends from the last data point, and from this an expiration date can be derived. For the example series, the latest payment of 379 would be interpreted as the start of an entirely new period (judging by how the previous segment, 370, stretches an entire year and thus creates a complete period) with an expiration assumed to be 12 months into the future as payments are always made in advance.

Once the series have been assigned their respective expiration, the format of the series has to be adjusted to allow for learning. As has been previously established, Machine Learning is based on learning (or detecting) patterns and then making predictions based on previously observed patterns. As such, the payment series have to be in a format that highlights observable differences in order to form intelligible patterns. To do this, the latest 12 months of each series was selected to highlight payment developments during the most recent period. This selection is specifically used due to its ability of showing a vast variety of period appearances and how they interact with other periods in any given series. Starting at the latest data point and going back 11 months in time, all fluctuations, inconsistencies, and otherwise spanning an entire contractual period are captured and used for learning. In practise, this results in an array of payment amounts ranging from month 0 to -11. For the example series, this would produce an array like this: [370, 370, 370, 370, 370, 370, 370, 370, 370, 370, 370, 379].

The series converted into this format, each together with its associated expiration, is then saved as a training set in .csv format where the headers denote how far back in the past that each value is situated as well as one column for the expiration. An execution could look like the sample below, and a sample training set can be found in Appendix 1.

```
> py parser.py "raw.csv" "training.csv"
Parsing file at /raw.csv into a training set.
[X] Success! Initial series have been produced.
[X] Success! All potential series have been discovered.
[X] Success! Series have been scored and extracted. Total: 82.
[X] Success! Series have been assigned points of expiration.
[X] Success! Series have been transformed.
[X] Success! File has been created.
```

```
Done! Results have been saved to /training.csv.
> py parser.py "raw.csv" "series.csv" --seriesonly
Parsing file at /raw.csv and printing resulting series.
[X] Success! Initial series have been produced.
[X] Success! All potential series have been discovered.
[X] Success! Series have been scored and extracted. Total: 82.
[X] Success! File has been created.
Done! Results have been saved to /series.csv.
>
```

3.2 Time Series Forecasting

With a training set available, the next step is to fit a Machine Learning model with it and produce predictions that can be compared to the rule-based ones in order to determine which is the most accurate. To do this, the library *sklearn* is used to implement and train a machine learning model, primarily that of a Neural Network but also a Tree Decision classifier for testing purposes. The input training set is normalized to avoid large fluctuations negatively affecting the results. Model-specific accuracy is then measured by predicting the training set itself and then through 5-fold cross validation. A sample of two standard executions (single and multiple) using a training set to produce ML-based predictions can be found in the example below (the input has been truncated as indicated by "..."). The output format is identical to that of the parser, *parser.py*.

```
> py predict.py [..., 370, 370, 379] "training.csv"
Predicting single expiration.
Model: TreeDecision classifier, fitted with /training.csv.
Done! Contract is predicted to end in approx. 12 months.
> py predict.py "series.csv" "training.csv" --out="result.csv"
Predicting expiration of series in /series.csv.
Model: TreeDecision classifier, fitted with /training.csv.
[X] Success! Predictions have been produced.
[X] Success! File has been created.
Done! Results have been saved to /result.csv.
>
```

4 Results

In this section, the results of this study will be presented. This includes the results of each step as described throughout this paper. First, the accuracy and performance of the series-parser and creator will be presented. Then the results of the expiration predictions, both rule- and ML-based, will be presented. All measurements are performed using a mixture of manually fabricated verification data that has been approved in coordination with domain experts, as well as real-world examples (ground truths) provided by Fortnox. Due to containing those incriminating ground truths, the used datasets have not been included in this paper.

4.1 Rule-based Series Discovery & Creation

In this section, the accuracy of the series discovery and subsequent creation will be presented. The verification of the rule-based series creation was performed using a dataset containing 50 manually constructed (fabricated series for the explicit purpose of verification) and 50 verified (ground truths provided by Fortnox) series of varying length and complexity. Each entry was created or selected with the explicit purpose of providing an as diverse basis of verification as possible, as well as a means of measuring theoretical as opposed to real-world series. The verification series were then identified using *parser.py* and the output series were verified against a key. The results can be seen in Table 4.2.

Type	Count	Correct	Percent
Fabricated	50	48	96%
Real-world	50	39	78%
Total	100	88	88%

Table 4.2: Accuracy of Rule-based Series Creation

As can be seen in Table 4.2, fabricated series were easier for the algorithm to discover with an accuracy of 96%. Real-world series were slightly more difficult to discover with an accuracy of 78%. Overall, for all of the 100 verification series, 88% were accurate.

4.2 Rule-based Expiration Prediction

In this section, the accuracy of predicting series' expiration using rule-based methodology will be presented. Like for the series discovery and creation, the same dataset of 50 fabricated series and 50 real-world examples was used, this time with each series being assigned an expiration. Those points of expiration have been assigned to the actual periods contained within the series as to increase coverage and promote higher quality measurements. As for the real-world examples, 40 were the same as used previously with an addition of 10 new ones. This discrepancy was because some of the initial series had missing expiration data and were therefore not viable to use at at this stage. Like before, the goal was to be able to measure success with fabricated series as opposed to real-world series. Furthermore, the series were constructed in advance as to keep the prediction accuracy measurement independent of the series creation process and accuracy. The respective expiration of each verification series was identified using *parser.py* and the output training sets were verified against a key. The results can be seen in Table 4.3.

Type	**Count**	**Correct**	**Percent**
Fabricated	50	50	100%
Real-world	50	36	72%
Total	100	86	86%

Table 4.3: Accuracy of Rule-based Expiration Predictions

As can be seen in Table 4.3 and similarly to the creation accuracy, the expiration of fabricated series was easier for the algorithm to predict with an accuracy of 100%. Real-world series were more difficult to predict with a total of 72% of the series correctly predicted. Overall, for the entire sample of 100 series, a total of 86% of series were correctly predicted.

4.3 Machine Learning-based Expiration Prediction

In this section, the accuracy of predicting series' expiration using machine learning will be presented. The prediction accuracy was measured using two different models, Neural Network[3] and Tree Decision Classifier. First, the accuracy of Neural Network and Tree Decision classifier was measured individually by predicting their respective training sets as well as through 5-fold cross validation[4]. The library *sklearn* was used to perform these measurements. The training sets will consist of the same fabricated and real-world example datasets, with the same pre-created series (this time pre-processed into an appropriate format), as was used for the rule-based predictions. This was done in order to produce results comparable to that of the previous step. However, to complement this, the same measurements will be performed for both models using a combination of both datasets. This was done in order to produce more meaningful results during 5-fold cross validation due to the low data count. The results of the accuracy measurements of the two models on their own individual datasets can be found in table 4.4, and the results of the accuracy measurements of the two models on the combined dataset can be found in table 4.5.

Datatset	**Model**	**Count**	**Accuracy**	**5-fold Accuracy**
Fabricated	Neural Network	50	92%	58%
	Tree Decision	50	94%	56%
Real-world	Neural Network	50	100%	24%
	Tree Decision	50	100%	22%

Table 4.4: Accuracy of Machine-learning Models on Individual Datasets

As can be seen in the table above, fabricated series were easier to predict through 5-fold cross validation with an accuracy of 56-8% compared to 22-4% whereas the opposite is true for the real-world examples which had an accuracy of 100% as opposed to 92-4%.

[3] Neural Network was set to 2,000 maximum iterations with a random state of 42.

[4] 10-fold could not be utilized due to available data limitations.

Model	**Count**	**Accuracy**	**5-fold Accuracy**
Neural Network	100	91%	49%
Tree Decision	100	97%	37%

Table 4.5: Accuracy of Machine-learning Models on Combined Datasets

As for the combined datasets, Neural Network performed better at 5-fold cross validation with an accuracy of 49% as opposed to 37% whereas Tree Decision classifier otherwise performed better with an accuracy of 97% compared to Neural Network's 91%. Across all tests, on average, Neural Network had a slightly higher accuracy of 69% as opposed to Tree Decision's 67.7%.

5 Analysis

In this section, the results laid out in the previous section will be analysed and given meaning. However, it is important to note the relatively low sample size of the utilized datasets, both on their own but also in their combined form. The reason for this lack of data items is twofold: first, the amount of effort and time required to create and validate series vastly exceeded that of what was available for the project; second, there was a scarcity of available real-world samples, most of which were invalid or otherwise unfit for this study, which could have caused an imbalance in the results should the fabricated series have been used to pad the datasets extensively.

Furthermore, there is an assumption that fabricated series were to perform better than real-world examples. The reason for this is due to the nature of series and how the algorithm was developed. According to domain experts, relevant contractual periods followed a certain set of basic rules and would be composed of specific periods that operate within a given frame of date and payment fluctuations (as discussed in the Implementation-section). As such, the algorithm, and the fabricated series for that matter, was designed accordingly and emphasized discovering series that followed the given rules, while still having a certain level of leniency for the abnormalities that had been observed in the real-world examples. Naturally, as a consequence, series that do not follow those rules, oftentimes being excessively abnormal real-world examples that may or may not have incorrectly book-kept entries (as discussed earlier), are more difficult, or sometimes even impossible, to discover.

Despite this, a few conclusions can still be drawn - although they should, just like the results themselves, not be considered to be representative but rather suggestive of overall implementation performance and machine learning validity. For ease of reference, the analysis has been split into subsections which each corresponds to their respective result.

5.1 Series Discovery & Creation

In terms of series discovery and creation, the algorithm performed quite well: correctly identifying and constructing a total of 88% of all tested series. As could perhaps have been expected, fabricated series were easier to identify with an overall correctness increase of 18% compared to the real-world examples. While the difference in identification correctness may not have been as significant as could have been assumed, the reason for the increase in correctness is likely because of the fabricated nature of the series and the difficulty of perfectly mimicking the often chaotic and unpredictable state of the real-world examples. As mentioned above, fabricated series were created to adhere to the given rules while at the same time attempt to mimic as many abnormalities (such as gaps and significant fluctuations) as possible. Oftentimes, this entailed producing functional series and then deliberately tearing them apart or injecting various series-breaking flaws, which can be seen in the failing fabricated series which all contained multiple gaps that prevented proper series construction. However, as can be seen in the results, the ability to mimic and successfully sidestep a variety of artificial abnormalities falls short of appropriately navigating the chaos that is real-world series.

That said, this could also indicate that real-world examples may not necessarily adhere to the given rules as reliably as was originally anticipated. As an example, it is possible that changes during the contractual period or other types of adjustments are more common than expected. It could also, as discussed in previous sections, be an issue of the available data being derived from book-kept records rather than actual transactions which may cause, among other things, unintended date fluctuations which in turn create unnecessarily complex series.

Overall, this does suggest that using a rule-based solution to create series from raw data appears to be viable. However, the difficulties that could arise from the lower accuracy of constructing real-world series should not be understated. While the performance using mixed datasets that include fabricated series is indeed quite promising, should the algorithm be left to operate in an exclusively real-world environment, reliability could, and probably would, become an issue. Naturally, with further work and more reliable data sources, those issues could be minimized.

5.2 Rule-based Predictions

As established at the start of the study, Fortnox has attempted to solve this problem using rule-based solutions in the past. However, they were never successful in finding a wholly satisfactory solution. While a correctness requirement was never provided, a cut-off point of 50% was established as a basis towards which to strive. As such, the algorithm performed far better than required: correctly predicting the expiration of 86% of all tested series which is well above the cut-off point. Like before, the points of expiration of the fabricated series were notably easier to predict than those of the real-world examples, with a flawless accuracy of 100% as opposed to 72%. Now, the reasons for this discrepancy could be many - and it is worthwhile to mention that the predicted series were provided as is, without any discovery and creation phase (as explained in the Results-section), which is why the prediction accuracy could be higher than the creation accuracy.

At this stage, problems generally stem from having a less-than-ideal data source which in turn produces complications such as violent or inconsistent date fluctuations as well as series with periods that contain payments that have consistent fluctuations throughout. Other problems relate to that of maintenance and, as [4] notes, complicated development. Being a rule-based solution, the designer needs to "foresee and provide solutions for all possible situations" [4] and should one or more rules change, potentially extensive changes to the algorithm may have to be made.

The first complication becomes apparent during the scoring of potential series. While series are given a higher score if their respective payment occurrence dates come at consistent intervals that correspond to their intended span (monthly, yearly, and so on), this may completely invalidate an otherwise valid series because payments were book-kept either before or after the actual payments took place. This can be mitigated by allowing dates to fluctuate slightly and by imposing a cap on the score given based on span accuracy. This would allow for some degree of inconsistency in book-keeping while also not completely invalidating the series' score. At the same time, however, it may produce invalid series if data points with similar dates and payment amounts are available which may then be given a similar or higher score than an abnormal valid series due to the score cap.

The second complication arises during the actual prediction phase. This is due to problems with period identification because of payment fluctuations. As previously described, periods represent a 12-month span, and a series can be made up of several periods. They can normally be told apart by looking at notable fluctuations in the series and their surroundings as a new period usually starts with a significant fluctuation. That said, real-world examples often have semi-consistent fluctuations throughout, such as [350, 350, 352, 350, 352, ..., 380], which requires a work-around such as allowing payments belonging to a given period to fluctuate by, as an example, 5%, which provides much-needed leniency in processing this type of series (and, correctly identifying 380 as a new period). While this generally works well, should those fluctuations be too consistent while the fluctuation amount is minor, this type of solution would cause the periods to coalesce; making them indistinguishable from one another and effectively making period identification impossible.

That said, if there are truly no defining fluctuations, such as 380 in this case, this would likely be an issue regardless of whether this solution is employed. This is due to there being no reliable way of determining to which period each payment belongs. For instance either 350 or 352 could indicate a new period or they could all belong to the same. It could be assumed that the first data point is the start of a series (and as such the start of the first period), but this is not accurate as raw series were generally not provided in their entirety, as in including their definite starting dates, but rather at an arbitrary point during a period. Because of this, solutions such as piecing together complete years was not a viable work-around for this particular issue.

Those two complications both, eventually, result in difficulties in determining the starting point of the most recent period of a series. This, in turn, leads to inaccurate expiration predictions as those are based on detecting the start of the latest period and calculating the offset to reach a full contractual period: a complete year.

Overall, the results suggest that a rule-based approach could serve as a solid starting point, but troubles in regards to maintenance, data sources, and potentially underdeveloped period identification would likely invalidate it as a more permanent solution. This becomes evident when looking at the lacklustre accuracy for real-world examples, whereas the flawless predictions of the fabricated series suggest that those did perhaps not adequately mimic the relevant abnormalities and complications found in real-world examples. That said, the accuracy performance could likely be greatly increased with further work on the period identification module and a more reliable data source.

5.3 Machine Learning-based Predictions

As for the Machine Learning-based predictions, they generally surpass the cut-off point with generous margins, although with some 5-fold accuracy results slacking behind significantly. Here, there was no significant difference in general accuracy performance between the datasets or the models, all landing at 91% or above. 5-fold accuracy was expected to be lacklustre due to data unavailability and performed significantly worse across the board. Despite this, the 5-fold accuracy for the combined dataset is only slightly below the cut-off point, at 49%, using Neural Network, which, incidentally, performed slightly better overall whereas Tree Decision had a higher overall general accuracy. However, due to the low amount of data and the similarities in general accuracy, it is important to note that no significant conclusions can be drawn at this point.

That said, the results suggest that Machine Learning-models would indeed appear capable of learning the relevant patterns needed for the task at hand which can be derived from general accuracy rates of 91% and above, where Tree Decision performed slightly better. Neural Network, on the other hand, had a higher 5-fold accuracy which suggests that it is slightly better suited at recognizing the relevant patterns at lower data amounts. However, the potential impact of complications arising from data unavailability cannot be overstated as it likely reduced not only accuracy but also reliability of the results. This is because cross validation loses reliability the less data there is while the models themselves depend on a healthy and varied training set. As such, it would seem feasible to assume that the results would have been better should there have been a larger pool of reliable data to use as a training set.

6 Discussion

In this section, the research questions that were presented in the Introduction-section will be answered based on the conclusions drawn in the previous section. The two questions are as follows:

1. Can a contract expiration be predicted with acceptable accuracy using Machine Learning methods (in this case, Time Series Forecasting)?

2. Is the accuracy better than, or comparable to, a rule-based prediction approach?

For the first question, the results indicate a high probability of Machine Learning-models being able to learn the relevant patterns in payment series and fluctuations in order to forecast a series's (contract) expiration successfully. As noted, however, the data unavailability does make the result unreliable, despite the ostensibly high accuracy rate outside of 5-fold cross validation. As such, the results are suggestive of positive overall Machine Learning viability, but acceptable accuracy cannot be guaranteed. It could be argued that the average accuracy for both models was 67.7% compared to 69% which is well above the cut-off point at 50%, but the data issues - as well as the lacklustre 5-fold performance - makes it unfeasible to assure, or even assume, satisfactory accuracy should these predictions be used in production.

That is not even considering the increase in uncertainty should the solution be employed in its entirety, as in the complete cycle of series discovery and creation, rule-based predictions, creation of a training set, and finally Machine Learning-based predictions. Despite the arguably promising results of each step, the accuracy deficit of each step stack on exponentially through the process and could result in a completely unreliable training set primarily consisting of lies rather than truths. Furthermore, the presented results were, as previously stated, produced using isolated datasets that had been either created specifically for this study or extracted from real-world samples and verified for applicability. As such, the uncertainty increases further should any and all raw data be used to create a training set in a real-world scenario.

That said, with more and better data, such as actual transactions rather than bookkeeping entries, and further work on the algorithm, it is likely that the results would be much better and as such increasing the accuracy of the predictions. Similarly, could the rule-based steps be removed altogether, such as procuring an extensive dataset of confirmed payment sequences that could used as a training set, it stands to reason that the performance of the predictions would be far more satisfactory. With all of this in mind, the conclusion would have to be a tentative yes. If and when more reliable results become available, this could change.

For the second question, as was already touched upon, the low sample size prevents any overly meaningful conclusions to be drawn. However, the results do suggest that a ML-based solution is comparable to, and in some instances better than, a rule-based solution. That said, ML-based predictions had an overall better performance than the rule-based ones, as seen by, for example, the higher accuracy of real-world examples. In the end, the conclusion would be another tentative yes, but with the caveat that this would, as discussed earlier, likely change depending on the amount and quality of available data as a ML model is highly dependent on a healthy training set.

6.1 Similarities of Findings

While no strictly similar studies were found, other studies, such as [5] and [10], found Time Series Forecasting to be beneficial to their Time Series-related problem. Like in those studies, Fortnox was confronted by sequences of consecutive date:value-pairs. In this case, they consisted of book-kept entries containing the date of payment (in theory) and the amount that was paid. By converting these pairs into Time Series, it became possible to Forecast (predict) future values, in this case the final payment and in doing so the expiration, with promising results.

Similarly, in [10], the dates represented points of observation whereas the values represented recorded SO_2 and SO_4 levels in the air. Time Series Forecasting was then used to predict volcanic air pollution in Hawaii that could then be used for a variety of purposes. While the results were promising, the author concluded that the solution was capable of learning the general problem, it was less effective at predicting extreme events - similarly to how this study's solution was less effective at predicting the expiration of series with extreme abnormalities (something that is likely to apply to the ML predictions as well). Unlike this study, however, Neural Network was found to have inadequate performance compared to other models.

Likewise in [7], the dates represented points of observation whereas the values represented road safety observations such as traffic accidents or kilometers driven. A variety of Time Series-related analytical methods were employed in an attempt to reduce the risk of making incorrect inferences (predictions) using said data. The authors found that while all models had their pros and cons, dedicated time series analysis provided the best overall performance and accuracy, with the caveat that an excessively large dataset would be needed to sustain the desired performance - similarly to how this study was negatively affected by a lack of data.

6.2 Effects of Study

As for whether the outcome of this study will positively affect the relevant industries remains to be witnessed in potential future endeavours into predicting the date of future events using Machine Learning. As mentioned in previous sections, this study did not, nor was it intended to, necessarily create something new. Using Machine Learning for Time Series-based analysis and prediction of various types of variables is nothing new, and instead this study applied the same basic methodology, with some changes, to a previously untested use case: predicting the independent variable, the timestamp, of a Time Series rather than some other variable of interest. The results of this study are very suggestive, but naturally not entirely conclusive, of that Time Series analysis can indeed be used to predict even the timestamp should some alterations be made to the prediction process and the format of the data used - depending on the level of accuracy that is desired. This knowledge could be useful in a variety of industries and sciences, as touched upon in the Introduction but indeed for contractual purposes, where the actual date of a future prediction, instead of or in addition to the actual predicted value, is of interest. Future additions or developments in this field could naturally improve the overall applicability of using Time Series this way, but this study could serve as a step, albeit small, for a more accurately predicted tomorrow.

7 Conclusion & Future Work

The aim of this study was to investigate the viability of using Machine Learning-based methodology to Forecast (predict) the expiration of contracts consisting of sequences of date:payment-pairs, and whether it would perform better than a rule-based solution with the same purpose. It did not place any expectations on how extensive the solution would be, how accurate it should be, or how generally applicable the results would be, and was merely checking for feasibility. To do this, a variety of algorithms were implemented to solve the following problems: rule-based identification and construction of Time Series from raw data, produce and assign rule-based expiration predictions to each series (contract), and finally convert the series and associated points of expiration into a Machine Learning-friendly format (a training set). This training set was then used to train two Machine Learning models that could then Forecast a contract's expiration.

While the results were not conclusive, they do suggest that Machine Learning models are indeed capable of learning the problem at hand which would indicate overall viability of Machine Learning methodology as a solution to this type of problem. That said, the study had a number of limitations, primarily in regards to data availability, which makes it difficult to draw any conclusions in regards to the accuracy of the implemented solutions. While accuracy ratings were generally on the high side and the Machine Learning-based predictions were, on average, better than the rule-based ones, the lack of datasets available for training and testing makes the results unreliable. Despite this, while the results could likely not be seen as representative of actual performance, they could serve as an indication of overall viability, both for Fortnox and the relevant industries, but also to to promote future studies of Machine Learning applicability and performance for this type of task. Despite this study being focused on a very particular domain, it stands to reason that similar solutions can be applicable for other, similar problems. As seen in other studies such as the ones mentioned in this study, it is likely that Machine Learning models can learn similar issues that consist of Time Series, such as date:payment-pairs. Naturally, this is largely dependent on the model chosen and how the training sets are created.

As the most significant problems of this study were in regards to an insufficient amount of good data, the most apparent improvement would have been an attempt at procuring more data to use for training and testing. As mentioned earlier, there was a real-world example scarcity and the processing was costly which means that data acquisition was a difficult and time-consuming task. That said, this study was primarily intended to investigate the viability of Machine Learning to solve a specific problem and not to develop an extensive rule-based solution. As such, the results could likely have been improved if more time would have been invested in producing testing datasets rather than developing the rule-based algorithms. Naturally, this would have resulted in lower performance of the rule-based solution which could have negatively affected the comparison factor, but would likely have increased the reliability of the results overall - especially since the actual training datasets were independent of the rule-based algorithms anyway.

There is arguably an abundance of other approaches to the topic of Machine Learning, or Artificial Intelligence in general, as there appears to an insatiable demand for intelligent solutions to all kinds of problems - big or small. However, as mentioned in previous sections, the applicability of Machine Learning in the prediction of a contractual expiration appears to be an underdeveloped area with seemingly no work readily available outside of studies that deal with Time Series Forecasting in general. While the results of this study could be indicative of overall Machine Learning viability and performance, it is far too inconclusive and limited to have much validity or applicability outside of its scope, and perhaps not even there. More work could be done in the examination of Machine Learning applicability for contractual predictions as this could prove crucial for future developments and ML utilization in related areas and industries. This also applies to using the same or similar methodology to solve other problems that can be broken down and produce a similar training set in order to investigate applicability in other areas.

In a similar vein, and has been mentioned throughout this study, procuring more and better data for this type of study is very likely to produce much better results, especially if it includes more examples of each category. Several examples of improvements like this have been provided already, such as producing more fabricated series, finding a data source where extensive processing is not needed, or having an implementation that can reliably convert raw data into a usable training set. Although, as [7] notes, satisfying the dataset requirements to obtain optimal Machine Learning model performance is not always feasible. Additionally, more work dedicated to implementing a more extensive rule-based solution, together with more and better data for testing purposes, could provide a completely different take on this issue. While this study is suggestive of the performance of ML-based solutions surpasses that of rule-based solutions, more work confirming its adequacy could prove useful. Those are but a few suggestions and the list of potential future work remains long.

References

[1] G. Leder, "Timing is Everything: Clients' financial behavior can be quite volatile. Finding the right moment to touch base is essential." *On Wall Street*, p. 1, 2005.

[2] C. Hakan Aladag and E. Egrioglu, *Advances in Time Series Forecasting*, 1st ed. Bentham Science Publishers, 2012. [Online]. Available: https://ebookcentral-proquest-com.proxy.lnu.se/lib/linne-ebooks/detail.action?docID=1041544

[3] K. Lazanyi, "Readiness for Artificial Intelligence," in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, sep 2018, pp. 000 235–000 238. [Online]. Available: https://ieeexplore.ieee.org/document/8524740/

[4] E. Alpaydin, *Introduction to machine learning*, third edition.. ed., ser. Adaptive computation and machine learning. Cambridge, Massachusetts: MIT Press, 2014.

[5] G. M. Weiss and H. Hirsh, "Learning to Predict Rare Events in Event Sequences," Tech. Rep., 1998. [Online]. Available: www.aaai.org

[6] IBM, "What are neural networks?" 2020. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks

[7] J. J. F. Commandeur, F. D. Bijleveld, R. Bergel-Hayat, C. Antoniou, G. Yannis, and E. Papadimitriou, "On statistical inference in time series analysis of the evolution of road safety," vol. 60, pp. 424–434, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.aap.2012.11.006

[8] M. Peixeiro, "The Complete Guide to Time Series Analysis and Forecasting," 2019. [Online]. Available: https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775

[9] Y. S. Kim, S. T. Rachev, M. L. Bianchi, I. Mitov, and F. J. Fabozzi, "Time series analysis for financial market meltdowns," *Journal of Banking & Finance*, vol. 35, pp. 1879–1891, 2011.

[10] G. Reikard, "Forecasting volcanic air pollution in Hawaii: Tests of time series models," *Atmospheric Environment*, vol. 60, pp. 593–600, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.atmosenv.2012.06.040

[11] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*. Somerset, UNITED STATES: John Wiley & Sons, Incorporated, 2015. [Online]. Available: https://ebookcentral-proquest-com.proxy.lnu.se/lib/linne-ebooks/reader.action?docID=1895570

A Appendix 1

This is a sample of the fabricated dataset that was used to train and test the ML models.

```
m-11,m-10,m-9,m-8,m-7,m-6,m-5,m-4,m-3,m-2,m-1,m-0,months_until_expirati
224,224,224,224,224,224,224,224,224,224,224,224,1
3511,3511,3511,3511,3511,3511,3511,3511,3511,3511,3511,3511,1
672,663,663,663,663,663,663,663,663,663,663,663,1
1789,1789,1789,1789,1789,1789,1789,1789,1789,1789,1789,1789,1
0,250,250,250,250,250,250,250,250,250,250,250,2
0,793,793,793,793,793,793,793,793,793,793,793,2
0,388,388,388,388,388,388,388,388,388,388,388,2
1358,1358,1399,1399,1399,1399,1399,1399,1399,1399,1399,1399,3
963,963,963,972,972,972,972,972,972,972,972,972,4
0,0,0,252,252,252,252,252,252,252,252,252,4
2385,2385,2385,2435,2435,2435,2435,2435,2435,2435,2435,2435,4
527,527,527,0,233,232,233,232,233,232,233,232,5
0,0,0,0,0,437,485,485,485,485,485,485,6
0,0,0,0,0,202,202,404,202,202,202,202,6
2520,2520,2520,2520,2520,2600,2600,2600,2600,2600,2600,2600,6
0,0,0,0,0,0,300,416,416,416,416,416,7
0,0,0,0,0,0,234,292,234,234,234,234,7
0,0,0,0,0,0,644,644,644,644,644,644,7
0,0,0,0,0,0,1289,1289,1289,1289,1289,1289,7
988,988,988,988,988,988,1020,1018,1018,1018,1018,1018,7
0,0,0,0,0,0,0,376,370,370,370,370,8
0,0,0,0,0,0,0,796,796,796,796,796,8
0,0,0,0,0,373,373,341,341,341,341,340,8
0,0,0,0,0,0,0,247,259,259,259,259,8
0,0,0,0,0,0,0,0,3333,3333,3333,3333,9
0,0,0,0,0,0,0,0,0,2886,2886,2886,10
1236,1236,1236,1236,1236,1236,1236,1236,1236,1266,1266,1158,10
852,852,852,852,852,852,852,852,852,852,1859,1859,11
11066,11066,11066,11066,11066,11066,11066,11066,11066,11066,13279,11211
859,859,859,859,859,859,859,859,859,859,859,878,12
597,597,597,597,597,597,597,597,597,597,597,591,12
552,552,552,552,552,552,552,552,552,552,552,667,12
1044,1044,1044,1044,1044,1044,1044,1044,1044,1044,1044,1069,12
```