Machine learning approach to support ticket forecasting from software logs

Matti Haukilintu

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 31.07.2022

Supervisor

Prof. Arto Visala

Advisor

MSc Petri Pyöriä

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Matti Haukilintu

Title Machine learning approach to support ticket forecasting from software logs

Degree programme Automation and electrical engineering

Major Control, Robotics and Autonomous Systems **Code of major** ELEC3025

Supervisor Prof. Arto Visala

Advisor MSc Petri Pyöriä

Date 31.07.2022 **Number of pages** 21+2 **Language** English

Abstract

Your abstract in English. Keep the abstract short. The abstract explains your research topic, the methods you have used, and the results you obtained. In the PDF/A format of this thesis, in addition to the abstract page, the abstract text is written into the pdf file's metadata. Write here the text that goes into the metadata. The metadata cannot contain special characters, linebreak or paragraph break characters, so these must not be used here. If your abstract does not contain special characters and it does not require paragraphs, you may take advantage of the abstracttext macro (see the comment below). Otherwise, the metadata abstract text must be identical to the text on the abstract page.

Keywords Machine learning algorithms, Robotic process automation, log parsing, random delay, Microsoft Azure ML Studio

Aalto University
School of Electrical Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Matti Haukilintu

Työn nimi Sovelluslokien ja vikatikettien yhteyden löytäminen koneoppimista hyödyntäen

Koulutusohjelma Automaatio- ja sähkötekniikka

Pääaine Ohjaus, robotiikka ja autonomiset järjestelmät **Pääaineen koodi** ELEC3025

Työn valvoja Prof. Arto Visala

Työn ohjaaja FM Petri Pyöriä

Päivämäärä 31.07.2022 **Sivumäärä** 21+2 **Kieli** Englanti

Tiivistelmä

Tiivistelmässä on lyhyt selvitys kirjoituksen tärkeimmästä sisällöstä: mitä ja miten on tutkittu, sekä mitä tuloksia on saatu.

Avainsanat Koneoppiminen, koneoppimisalgoritmit, ohjelmistorobotiikka, loki, lokin parsiminen, satunnainen viive, satunnaisviive, Microsoft Azure ML Studio

Preface

Pitkäksi venynyt työ

Kiitokset

Otaniemi, May 7, 2022

Matti Haukilintu

Contents

Symbols and abbreviations

Symbols

$\mathbf{B}$	magnetic flux density
$\mathbf{B}$	magnetic flux density

Operators

$\nabla \times \mathbf{A}$	curl of vectoring $\mathbf{A}$
$+$	yep, a plus

Abbreviations

RPA	Robotic Process Automation
ML	Machine Learning
SQL	Structured Query Language
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
GDPR	General Data Protection Regulation
ADA	Anomaly Detection Algorithm

1 Introduction

Artificial intelligence has found its way to more and more fields of business. In banking business it is already used in credit card fraud detection,

Samlink Oy (Samlink from now on) was founded in 1994 and is now owned by Kyndryl. From the early years and going by the name of Samcom the company was owned by several Finnish banks and offered all sorts IT solutions to them. Most notable product is Codeapp that makes mobile authentication possible for client bank end customers.

Besides banking, Samlink develops multiple other IT solutions to wide range of customers, for example, entertainment platform solutions for DNA. Even though Samlink can be considered a modern technology company the most modern technologies has not yet been adapted in tool variety used in development. However, robotic process automation has been adopted and is actively used in some banking solutions to reduce the amount of manual labor required from banking clerks.

Like all other IT companies, Samlink has a lot of data that is produced and moved through its systems daily. And as a company offering continuous development as well as product maintenance services, Samlink also serves as a technical help desk regarding the software solutions produced. As no IT solutions comes without bugs or misbehaviour, Samlink service desk has to use considerable amount of time to resolve whether current tech help request is due to the problems in programming. In this process, machine learning is able to help.

<RPA in use, but not intelligent processes>

<ML and Azure>

1.1 Background and motivation

In the field of information technology logging is one of the most important methods in problem-solving, be it software or operating system related.[1]

Typically, at least in Samlink processes, logging is a bit more verbose than it needs to be. This is usually because when the problem occurs it is easier to already have the verbose logs available than trying to replicate the issue after setting logging to more verbose mode. Too verbose logging, however, leads into two problematic issues. First of, the size of log is huge and finding the critical leads for solving the problem in hand takes more time. Of course, with more strict logging pinpointing the issue from within the logs would be faster, but then again, solving the problem with only critical error messages could be more time-consuming if crucial context is missing. Secondly, as software tends to see some issues it encounters worth of several dozens of log rows, some rows are not critical information if the software is able to continue, especially if it is able to retry the process after failure and get the job done at the second time. These issues make log parsing considerably laborious.

<importance of logs>

<current state of samlink AI processes>

<Log format>

<Intelligent automation>

1.2 Research objectives

This research aims to kickstart the machine learning application usage in Samlink operations. Multiple obstacles need to be tackled as most of the phases in this study has not yet been encountered inside the company.

First and foremost, it is crucial to construct some basic form for log data so it is usable by machine learning algorithms. Log data forming is on of the key elements in automatic log parsing applications as it is not for just machines but also for people to read.

As today more and more concern is set on anonymization not only due to GDPR, the data used for machine learning must be sanitized. Because of this, one major objective is to create a clean dataset that is safe to use in cloud environment without creating concern around security and privacy issues. In addition to this, data must also be clean enough so that machine learning algorithms are able to process it.

Finally, the main objective this research is aiming to answer is whether it is possible to use machine learning algorithms in such ways that combine support ticket timestamps and software run log so that it can predict if new support ticket might be coming.

Few things to consider:

```
  * Counting anomaly "probability" for individual log rows
  * Random delay! Solving issue by grouping
  * Hybrid ML with anomaly detection and regression
  => Can this give any results for forecasting?
  ?? Assumption to lean on:
    !! Ticket causing log rows are ANOMALIES in log data.
    TODO: explain why this assumption is necessary!!!
    - Log amount is not necessarily proportional to ticket creating error amount, e
    - Considerable amount of rows may be related to one, or several different, tick
```

What really happens in this research:

```
<Anonymization scope>
<Azure ML studio setup and prerequisites>
<Log data and timestamp combining>
<Connection for data, ML estimates>
```

1.3 Scope

In order to limit the study to feasible length and content it is necessary to define the scope for the thesis. As major part of the time consumed for the study was used for

data acquiring and anonymizing yet the most value comes from content considering machine learning, the length of the content per section does not reflect the amount of time used for each section.

Data anonymization

Anonymization in the context of this thesis refers to data sanitization process purposed to edit the data into more secure form in the privacy point of view. In this study we aim to create a dataset usable in ML training. In this respect anonymization is not in the main focus of the study but only treated as a sub-phase of the data preprocessing in whole. Nevertheless, anonymization is from the privacy point of view the most important phase of data preprocessing.

Keeping this in mind, anonymization is covered rather superficially, only enough to explain the reasons behind actions taken during anonymization process.

Azure setup

The ML training and result scoring is done in Azure ML environment. As part of the study aims to create an initial guideline for ML process commissioning the Azure setup phase is documented in such detail reflecting the importance of this information for future developers starting Azure ML projects.

Data requirements

Data purity in a sense of how easy it is to be used by ML algorithms creates challenges at the beginning of ML training. If data is not consistent, has lots of missing values or is formed in unanticipated way, it requires considerable amount of preprocessing slowing the training process and causing errors in pipeline runs.

In order to create a baseline for Samlink ML projects this study aims to give basic criterion what is required from the data, so it is easily parsable by ML algorithm.

Machine learning methods

Several different machine learning algorithms exist that are aimed for different applications in mind. For example, to make an algorithm that can predict the price of an apartment listed[2] we could be using linear regression and in order to detect possible cyber threats from network traffic[3] a two-class support vector machine could be utilized. These two methods are very different in usage and has their pros and cons in different applications.

As different methods can be used in creative ways in very different applications depending on how the data is presented and how the ML problem is formed, this study focuses on just a few easily approachable training methods.

When it comes to anomaly detection, only principal component analysis (PCA) is considered as PCA-based Anomaly Detection component is usable from existing anomaly detection algorithms in Azure ML Studio. As purely Azure ML Studio is used during this study, no other anomaly detection algorithms are debated.

1.4 Structure

At the beginning of this thesis we opened the key concepts discussed in the study. We also defined the

2 Background

Machine learning is part of the artificial intelligence field. Typically, machine learning refers to

Machine learning usage has become more common and is nowadays widely used in many fields, not just general information technology and computer science. This is because data can be gathered everywhere, and where there is data to be processed, machine learning can be there to process it.

Machine learning divides commonly in several categories, which are explored briefly in the next section.

2.1 Machine learning field

ML training

Machine learning algorithms can be trained in two basic ways: supervised and unsupervised learning.

In supervised learning, algorithm is given data with ready answers on how the data needs to be interpreted. Algorithm then tries to figure out how given data and the correct answers are related.

In unsupervised learning, on the other hand, algorithm does not get model data from which to train itself, but instead it tries to find clusters or groups inside the data that are linked together more closely than to other parts of data.

Typically, when training an algorithm, some predefined portion of the data is used as training data. The rest is used to validate the results so that validation data and training data do not overlap. Instead, trained algorithm is given data it has not seen before and the result it produces with it is then validated. For example, in supervised learning the key values the algorithm is trained to find out are hidden from the validation data. The resulting values produced by the algorithm are compared to those hidden values and the difference between the estimate and the real value can be used to determine how well the current algorithm compares to others.

However, in this study, we are going to break that rule about non-overlapping training and validation data. The reason for this is explained further in section 3.5)

ML algorithms

Algorithms are the main "magic" inside machine learning. They can be generally divided into <four> categories

Regression algorithms predict values and are typically used with supervised learning.

Classification algorithms predict categories. Depending on the algorithm, they can predict between two or several categories.

Clustering algorithms use unsupervised learning to find structures inside data.

Anomaly detection algorithms work also unsupervised and try to find unusual or rare data points from data.

2.2 Cloud ML platforms

Machine learning algorithms are not light to operate. Depending on the amount of data, it can be a serious

Especially with online applications where real time analysis of new input data is required, cloud computing resources can make a huge difference in terms of processing speed.

Online market offers several solutions for ML computing in cloud.

Google

Amazon AWS(?)

Microsoft Azure[4]

2.3 Log data parsing with ML

2.4 Random delay in time series

2.5 Hybrid ML approach with anomaly detection

3 Research material and methods

In the next section we explain in more detail what the data used in the study consists of and what methods were used in attempt to answer the research goals.

The data in the research consists mainly of two parts. The most important part is, obviously, the log data produced by the numerous RPA processes. The second part complementing the study is the support ticket data written by clerks of customer banks. In order to use the data safely in the cloud environment it was necessary to sanitize the data from any sensitive information. This was done by anonymizing the log data and using only timestamps from the support tickets.

3.1 Support ticket data

Like all other software, RPA components fail from time to time. As described above, RPA logs are verbose making possible error identification from among it hard. Due to that, it is not feasible to create log parsers that would be able to identify critical errors from within thousands of lines of log. When critical error happens and it causes the RPA process to fail, the banking clerks need to fix manually the job left by the RPA process. Every time this happens, these clerks then send a support request ticket to Samlink technical helpdesk and ask to fix the issue.

When clerks send the ticket to technical support a verbose description of the situation is written to help developers to identify the problem. This description often contains sensitive end customer information like bank account details and social security numbers. To avoid privacy issues when processing this data, it was decided to use only timestamps of the tickets. The resulting data was practically a list of date and time values. More about the issue from privacy point of view is described in section 3.3

3.2 RPA log data

Robotic process automations used in Samlink are engineered to ease the workload of bank clerks. RPA components work mostly with loan applications among other routine tasks that require mostly manual labor.

Like other software, RPA also produces log data during runtime. As several RPA processes are running the amount of log produced daily is also significant. This log data is not in consistent form forming out of typical CSV data injected with even more inconsistent JSON data that varies in contents vastly.

RPA log data is stored in SQL database. The database is split in live production log that is gathered for few months and then moved to archive that has several years worth of log.

In this study we used archived data as it was easier to acquire in one run without the need to merge different parts together. Archive also had data that was seen as sufficient amount for machine learning algorithm training.

Data formatting

At the beginning of the research, the log data from RPA was in SQL database. However, the database used was not »pure» in a way that typical relational databases are, but some columns included JSON-formatted data in them. For ML algorithms to be able to read the given data with ease this sort of »impurities» needed to be cleared from the data.

When feeding the log data to anomaly detection algorithm it was necessary that all the rows were as minimally unique as possible. Thus, all unique data was stripped from the data, especially the fingerprint value that was unique for each row. Also, job ID information and timestamps of the rows were removed momentarily.

3.3 Data anonymization

Support ticket data privacy

Samlink handles highly sensitive banking customer data in its processes, such as personal identification numbers, home addresses, email addresses and bank account numbers. All possibly sensitive data must be removed before data can be transferred out from production environment to cloud. Due to bureaucratic reasons, technical support tickets were under more strict policies. Because of this, they were allowed to be used in the research on condition that no business critical nor customer sensitive information was processed in the first place. Only way to assure this was to select solely timestamp fields from ticket data. Thus, no sanitation for ticket data was needed.

RPA log data sanitization

Information privacy is one of the key values in Samlink business promise as company develops high security banking applications and processes sensitive customer data. Thus, several aspects were needed to take into consideration before log data could be authorized for thesis study usage. To improve privacy, it was decided to assume that personal customer details are not critical information for training the ML algorithm if goal is to find possible problems in RPA runtime and not detect individual customer related problems. This way it was not necessary to achieve adequate security by less secure and more effort consuming ways such as pseudonymization or k-anonymization, which would have also required strict inspections before data could have been approved for cloud processing.

As production environment is built on Microsoft Server based solution, and because it was highly unrecommended to install additional software to the production server, data acquiring and anonymization tools were chosen based on what was already usable in the RPA production environment. Microsoft Powershell offers sufficient tools for database SQL querying and stream editing. The amount of data was significant which made straight file editing impossible due to the memory limitations. Thus, stream editing was necessary for finding and replacing sensitive information from the data

Anonymization took good proportion of the time in workdays as processes were slow, amount of data was big and multiple re-runs were needed before the results was seemed adequate.

3.4 Azure environment

Several different ML methods are usable with Azure ML Studio. <???>

Usually <with usual ml methods> the estimates created using ML algorithms are formed based on the certain features presented on a one element of the data, or on one row. This means that in typical case, there is one column in the data given to the ML algorithm that is removed from the training data and this column value is what algorithm aims to predict.

In this study case, however, data does not contain clear values that are being estimated and that can be used as comparison.

LOG_DATA				EFECTE_DATA
a=date	b=msg	c=etc.		A YYYY.MM.DD hh:mm:ss
a	b	c	n1	B YYYY.MM.DD hh:mm:ss
a	b	c	n2	C YYYY.MM.DD hh:mm:ss
a	b	c	n3	D YYYY.MM.DD hh:mm:ss
a	b	c	n4	E YYYY.MM.DD hh:mm:ss

n1 = SUM(AB)
n2 = SUM(C)
n3 = SUM(DE)
=>
We could try to predict nx but usually this is done by making estimate based on a, b and c. Instead, we aim to estimate the sum of events in timeframe. We should also skip event instances that are close to each other to avoid counting multiple values linked to same error as different possible ticket creators.

3.5 Machine learning methods and pipelines

As stated in section 2.1, the approach in this thesis is, if expression is allowed, unorthodox Usually, it is not a good idea to use same data points in ML algorithm training and validating. Acting otherwise leads to algorithm processing with same data it was trained with thus creating a situation where algorithm already knows what to do with the current datapoint. If the results were validated after this the algorithm would get unreliable score as it had the validation data already in the training phase. This could be compared to giving some right answers to students during test and scoring test results as if no help wasn't given.

Initial plan when starting the ML pipeline testing was to feed the log data to anomaly detection algorithm and try to get some sort of estimate of possible anomaly count. This plan had several problems. First, as stated, logging is very abundant and

several thousands of rows was logged during a single day. This meant that among the

Two different algorithms are needed. In phase 1, algorithm defines how likely one datapoint is to be considered an anomaly. In phase 2, another algorithm aims to predict how many tickets are to be expected to receive within a time frame. Phase 1 is purely anomaly detection while phase 2 could use classification within time frame

Possible algorithms to consider in phase 2:

```
Artificial Neural Network, reinforced learning
```

Anomaly detection algorithms

Here we discuss only about the PCA-based anomaly detection as it is the only anomaly detection algorithm available in Azure ML Studio.

Move this section to Section 2 (Background) if it fits there better.

Even though only one ADA is usable, the data it processes can be given in two formats. Most simple way is to give the data as-is letting the PCA-based anomaly detection component work without further modifying the log rows. This way, the component tries to recognize the anomalies based on all the information included in the row. Practically, this means that the component processes data in textual format. Another way is to create numerical values out of the textual data. Each word or N-Gram is now a number of said instances found on the row being processed, and each row can be presented as a sequence of numbers indicating the number of those words.

N-Grams can in addition have a weight based on the frequency they appear in the whole data. Different weights usable in Azure ML component are listed below

1. Binary Weight
2. TF Weight
3. IDF Weight
4. TF-IDF Weight

4 Results

4.1 Azure and Azure ML Studio

<general about ml inside azure>

Azure resources

<what resources was needed inside Azure?>
<virtual machines etc.>

Azure ML Studio components

<clusters and data>
<Memory problems>
During the initial pipeline runs the execution came to an abrupt stop and Azure notified about memory issues. These problems were linked to the data amount which had to be reduced to 600 megabytes before any pipeline could be finished using the data. This reduction was against the initial goal where preferably all the data could have been used.

Considerable amount of time was used to fix or avoid this issue but nothing clear was found that would explain the error received. While working with with the issue it was also noted that data needed more cleaning in order to ease the preprocessing phase as described with more detail in section 3.3 Thus, the data had to be imported from log archive and anonymized once more.

Two choices was possible to take:

1. Continue working with full data and attempting to fix the memory issue by consulting Azure experts

2. Trim the data to reduce the data size by declaring info-type log messages as unnecessary and working with vastly diminished data until the memory issue would be solved one way or another

To advance the study more efficiently it was decided to trim info-type log messages from data hence reducing the data amount considerably. Meanwhile, <fixing the memory issue>

4.2 ML training and validation

<anomaly detection>
<N-Gram Feature extracting>
<Some regression algorithm to predict event count. Poisson only for poisson distributed data.>
<two-class classification> <support vector machine etc>
preformatting data!

1. remove fingerprint etc unique values from raw message
2. calculate anomaly probability per line

! CAN WE COMBINE THIS PER JOB-ID?

3. create new table consisting:
   (a) amount of rows per timeframe
   (b) amount of unique job ID's per said timeframe
   (c) anomaly probability value (median, mean etc)
   (d) efecte tickets received in said timeframe

<Integrating with timestamps>

To avoid the problem with random delays between log rows and technical ticket timestamps, log rows were grouped by time stamp certain time frame groups.

```
<todo:

***A***
count sum of incidents in timeframe x
set x to each row in data by timestamp
predict amount of incidents based on data

***B***
Use efecte data as reference values
(regression, predict amount in timeframe -> compare)
(classification, count TRUE in timeframe -> compare amount)

>
```

5 Summary

<Sum up here what we did and why>

5.1 Discussion

<Here some thinking what should have been improved>

Data formatting

The most time-consuming tasks in the study was without a doubt the anonymization and preformatting of the data. Although sensitive information may sometimes be crucial in error fixing as problems may consider just one client, it is necessary that the data sanitation is possible to do in order to use the data in less secure environment. By preformatting the data in such way that all different personal information types do not differ between use cases.

Possible ML methods

Some sort of time delay forecasting?[5] Could we estimate the number of tickets based on some log metrics in time frame?

References

[1] A. DeLaRosa, "Log monitoring: not the ugly sister," *Pandora FMS*, 2018. [Online]. Available: https://web.archive.org/web/20210901031146/https://pandorafms.com/blog/log-monitoring/

[2] W. K. Ho, B.-S. Tang, and S. W. Wong, "Predicting property prices with machine learning algorithms," *Journal of Property Research*, vol. 38, no. 1, pp. 48–70, 2021. [Online]. Available: https://doi.org/10.1080/09599916.2020.1832558

[3] K. Ghanem, F. J. Aparicio-Navarro, K. G. Kyriakopoulos, S. Lambotharan, and J. A. Chambers, "Support vector machine for network intrusion and cyber-attack detection," in *2017 Sensor Signal Processing for Defence Conference (SSPD)*, 2017, pp. 1–5.

[4] "Comparing machine learning as a service: Amazon, microsoft azure, google cloud ai, ibm watson," https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai-ibm-watson/, accessed: 2022-04-04.

[5] G. H. Erharter and T. Marcher, "On the pointlessness of machine learning based time delayed prediction of tbm operational data," *Automation in Construction*, vol. 121, p. 103443, 2021.

A Esimerkki liitteestä