UNIVERSITY OF TARTU

Institute of Computer Science

Software Engineering Curriculum

Henri Trees

Predictive Monitoring of Multi-level Processes

Master's Thesis (30 ECTS)

Supervisor:	Anna Leontjeva, MSc
Supervisor:	Prof. Marlon Dumas

Tartu 2016

Predictive Monitoring of Multi-level Processes

Abstract:

The ever increasing use of Information Systems causes ever more information to be stored. As organizations and businesses become more efficient due to competition they need to gain competitive advantage over others. More and more companies and institutions have turned to Information Technology to find business value in a data-driven world. Modern Information Systems maintain records of process events, which correspond to real-life activities. As processes evolve and become more complex, so does the information that reflects them. In this thesis, we propose an approach to predictive monitoring of complex multi-level processes. In this context, a multi-level process consists of a high-level parent process which spawns multiple low-level subprocesses, which have their own life cycle and run independently of one another. The author proposes constructs called milestones, which include both parent- and subprocesses and are used for the predictive monitoring classification task. This approach has been validated on a real-life event log of the business-to-business change management process in place at Baltic's largest telecommunications company Telia Estonia.

Keywords: Process Mining, Predictive Monitoring, Machine Learning

CERCS: P170 - Computer science, numerical analysis, systems, control

Mitmetasandiliste protsesside ennustav seire

Lühikokkuvõte:

Infosüsteemide laialdane kasutamine järjest rohkemates valdkondades tekitab aina suuremaid salvestatavaid andmemahte. Organisatsioonide ja äride efektiivsuse kasvuga tekib suurem vajadus leida alternatiivseid viise konkurentsieelisteks. Järjest rohkem hakatakse antud infoajastul otsima ärilist väärtust andmetest. Protsessikaeve meetodeid kasutades üritatakse justnimelt seda teha, kuid äriprotsesside arenedes muutuvad keerukamaks ka andmed, mis neid protsesse kirjeldavad. Hetkel keskendutakse protsessikaeve uurimustes protsessidele, mida on võimalik väljendada järjestikkuste sündmuste jadana. Käesolevas magistritöös esitatakse uudne lähenemine äriprotsesside ennustava seire rakendamiseks mitmetasandilistele äriprotsessidele, mis sisaldavad paralleelseid alamprotsesse, ning mida pole võimalik sündmuste järjendina väljendada. Väljapakutud meetodi suutlikkuse hindamiseks rakendatakse antud meetodit elulisel andmestikul telekommunikatsiooni tegevusalalt. Tulemusi võrreldakse lähenemisega, mida kasutatakse ühetasandiliste äriprotsesside ennustavaks seireks.

Võtmesõnad: protsessikaeve, äriprotsesside ennustav seire, masinõpe

CERCS: P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Contents

1 Introduction

In this thesis an approach using milestones is presented for predictive monitoring of complex business processes and is validated on a real-life dataset from Telia Estonia. Predictive monitoring is used to predict deviant processes as early as possible during the execution of a single ongoing process. This section introduces the motivation behind the topic as well as the contribution and outcome of the thesis.

1.1 Motivation

The shift to computerization throughout industries causes ever more information to be stored. This leads to gigabytes of stored data that is mostly unstructured, noisy or saved with the only purpose of being a backup. However, organizations are starting to acknowledge the power of data and particularly, making sense of the information that is captured by their Information Systems, trying to gain business value from it. Process Mining is a research area which aims to extract useful and meaningful information from event logs. It combines methods for extracting a structured process description from a set of real executions, for most cases some type of a log or event history. Process Mining involves techniques like new model discovery from the recorded events in an event log. Conformance checking compares an existing process model against the historical recorded events and performance analysis deals with analyzing the quantitative measures of a process. Predictive monitoring, a part of process mining, aims at predicting the outcome of a process on runtime.

Telia Estonia is one of the largest telecommunications companies in the Baltic states, operating in Estonia. For now, the main subsidiaries, mobile operator EMT and broadband operator Elion, have merged together behind the name Telia, both holding the market leader position in their fields [mar], providing mobile, broadband, TV, IT solutions and content to the whole nation. Their incentive is to provide customers with quality services with high usability and the best customer service. To analyze and improve the business-to-business change management process, the organization is collaborating with science and educational institutions for the purpose of process mining and data analysis of the captured event logs extracted from their Information System. The process consists of two separate life-cycles of processes where one can include multiple child tasks which are carried out in parallel. This creates the need to find and apply effective methods to extract knowledge from two seemingly different logs, which are also both noisy and with a lot of missing information. The main research question is how to efficiently extract valuable business information from a complex multi-level event log.

To our knowledge, there have not been academic studies implementing predictive monitoring on multi-level processes, on real-life data. The author is motivated to present a method how to abstract the information from top-level processes and from the spawned subprocesses to combine the logs of parent and child processes in order to conduct meaningful predictive monitoring study.

1.2 Research problem

This thesis addresses the following predictive business process monitoring problem: given the execution trace of an ongoing processes case, which has subprocesses, and a set of traces of completed cases from the past, contained in the event log of a process, predict the most likely outcome of the ongoing case, outcome being a particular classification of a case. That said, this thesis concentrates on predictive monitoring of multi-level processes, which means that the underlying process is not flat, as opposed to the majority of research [VDLR$^+$15, MDFDG14, DFDMT15, LCDF$^+$15] which is currently carried out on flat processes. A multi-level process starts with a parent process and at some time spawns zero to multiple subprocesses which run in parallel but are not dependent on each other. To be able to extract real knowledge and value from these types of process logs, we need to find a way of dealing with the leveling, as analyzing logs of both parent- and subprocesses separately does not entail enough information to make conclusions about the whole process. In Figure 1 we present a simple example of a multi-level process. After the parent process has been registered, multiple parallel subprocesses can be created, which are a part of the overall process, as opposed to a flat sequence of events.

1.3 Contribution

Using process mining and predictive monitoring techniques, we propose an approach to predictive business process monitoring task of multi-level processes through constructs named milestones, which are used to aggregate the parent- and subprocesses. The ap-

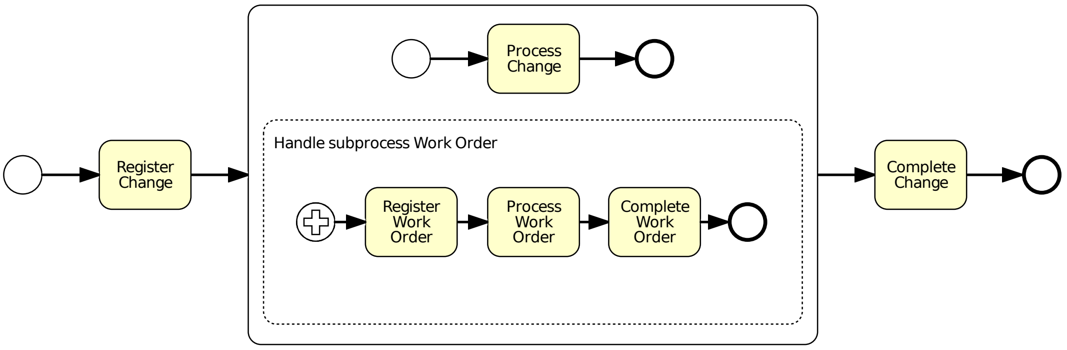

Figure 1: Simple example of a multi-level process in Business Process Modeling Notation [OMG].

proach is then evaluated by the application on a real-life dataset, which includes logs from parent process and subprocesses. This encompasses preprocessing the logs and applying feature transformations. As the logs have multiple free-text features about the events, different statistical representational models for free-text are used to enrich the initial dataset. Both textual and categorical features are then combined to create classifiers with different machine learning algorithms and hyper-parameter optimization. The performance of the milestone based model is also compared to a straightforward adoption of a method used for flat processes.

In the context of Telia Estonia, historical information about past executions of the business-to-business change management process could be used in order to predict when the target date of completion of a process is going to be postponed. This information is particularly useful for supporting the decisions of the teams and employees involved in the process. Consequently, this leads to improved lead times and customer satisfaction. The approach presented in this thesis aims at supporting the Telia Estonia process managers in their decisions through classification of ongoing cases of processes, so that they can evaluate the need for better governance or influence the decision-making to better achieve their business goals.

To evaluate the efficiency of the proposed approach of deviant process prediction, qualitative and quantitative evaluation is made along with the baseline comparison. The comparison of two different text mining methods is also presented.

1.4 Structure description

The thesis is divided into five parts. Chapter two gives an overview of the underlying rationale of methods and techniques used in this work, particularly about Process Mining as a whole and details about the evaluation dataset. The section also includes a review of the state of the art in Predictive Monitoring and in which setting has it been applied thus far. Chapter 3 outlines the approach that is being proposed in this thesis for predictive process monitoring, acting as the contribution of the master's thesis. Chapter 4 includes the information about the dataset used as the real-life example to which the proposed method of predictive monitoring is applied. It also includes the reasoning about the evaluation metrics used and also overall results. The thesis finishes with the chapter about the conclusions and future work.

2 Background and related work

In this chapter, we describe the technical background and knowledge supporting the thesis and also related work carried out in the field of Process Mining and Predictive Monitoring in particular. We aim to give explanations of the concepts and methods used for Predictive Business Process Monitoring in this thesis.

2.1 Process Mining

2.1.1 Overview

Process mining [VDA11] is a body of methods designed towards analysis of business processes based on event logs. These methods provide analysts with an in-depth understanding of the execution of a given process based on information obtained by Information Systems. The analysis also includes detection of deviations of the processes based on the performance or other business objectives. The primary input to a process mining method is an event log. The event log is recorded during the execution of the processes and represents a collection of event records related to the process being analyzed.

A single execution of a business process is called a process instance or *a case*. The event logs consist of a set of *traces*. Each trace is a sequence of *events*, each referring to an execution of an activity. An event denotes the state change of the process, like start, end or cancellation. Events in a trace may have payload, consisting of attributes bound to an event. These can, for example, be the resources involved in the execution of an activity or other recorded data. Usually, the payload data contain information about the point in time when the event is executed (a timestamp) and the actor who performed the task. The model of a single process instance is called a process instance model. A process model defines the behavior of all process instances that correspond to the same process. The event log contains all sets of traces, or instances, of a given process and is, therefore, used for process mining.

2.1.2 Predictive Monitoring

Predictive business process monitoring [MDFDG14] is a subcategory of process mining methods which aims at predicting a possible outcome of a given currently running (incomplete) case at runtime and as early as possible in the process. The prediction is typically based on some previous historical data about similar cases of the same process. The outcome, which is being predicted, may be the fulfillment of a constraint or a business rule. For example the overall cycle time of a given case, the validity of a temporal logic constraint, or any predicate over a completed case. More of a real-life example would be predicting if an ongoing case will finish in a specified time frame. Or in a medical treatment process, if a patient will be recovered after a specific treatment is foregone. The outcome refers to a label associated with completed cases, usually being binary.

2.2 Machine Learning

Supervised learning is a family of machine learning algorithms which entails learning a mapping between an input variable x and an output variable y so that the mapping could be then used to predict the output variable for unseen data [CCD08]. The input is typically a vector of features and the output is a label describing a property (called classification) or a continuous value (called regression). Supervised learning produces an inferred function, which as said, can be used on unseen inputs.

2.2.1 Support Vector Machines

Support Vector Machines (SVM) is a blend of linear modeling and instance-based learning. The fundamental principle is that the input variables are transformed into a high-dimensional feature space, by using a nonlinear mapping. This can be achieved using various nonlinear mappings as polynomial, sigmoid, etc. Then the best linear separator, a hyperplane, is found [WF05]. The hyperplane(s) can be used for classification, regression or other machine learning tasks. The underlying probability and density functions do not need to be known prior to the modeling, which means that SVM can easily be used in practical settings. The best performing separation is achieved when the hyperplane has the largest distance to the nearest training points from any class. Essentially, more significant margin leads to lower generalization error.

As machine learning problems are usually defined in a finite dimensional space and from which the separation of different sets of classes is not trivial, the SVM method proposes to map the original finite space into a much higher-dimensional space, making the separation of classes easier. To maintain a tolerable computational load, the mappings are designed to ensure an easy computation of dot products, with variables in the original space. This is done by defining them with a kernel function, which is selected based on the initial problem. There is no straightforward way of determining which kernel to use for the best performance, in practice, kernel choice does not yield much better results [CM98].

2.2.2 Logistic Regression with Gradient Descent

Logistic regression is a type of generalized linear model. It models the probability of an event being in a particular class as a linear function of a set of input predictor variables [HKP11]. To further explain logistic regression, we present the derivation of logistic regression with gradient descent presented in [Ren03]. Consider a binary classification where examples are labeled. The examples have l features of which each one can take values zero or one. We denote the example by a vector $\vec{x}$ and also the value of the k^{th} feature as x_k. We define also an additional bias feature $x_0 \equiv 1$. The probability of a positive class example being drawn is

$$p(y = +1|\vec{x}) = g\left(\sum_{k=0}^{l} w_k x_k\right), \tag{1}$$

where $g(z) = \frac{1}{1+e^{-z}}$. The w_k, $k \in \{0, ..., l\}$, is used to denote the weight of the k^{th} feature, and w_0 as bias weight. The weights are learned to maximize the likelihood of data. Let $(y_1, ..., y_n)$ be the corresponding output labels of training data $(\vec{x_1}, ..., \vec{x_n})$, while x_{ik} being the value of the k^{th} feature of example i. Logistic regression maximizes the log-likelihood of the data,

$$L(\vec{w}) = \sum_{i=1}^{n} \log g(y_i z_i), \tag{2}$$

where $z_i = \sum_k w_k x_{ik}$.

The weight-learning process with gradient descent: the gradient of the log-likelihood with respect to the k^{th} weight is:

$$\frac{\partial L}{\partial \vec{w}} \text{ where } \frac{\partial L}{\partial w_k} = \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i). \tag{3}$$

Recall that $z_i = \sum_k w_k x_{ik}$, $k \in \{0, ..., k\}$, and $x_{i0} \equiv 1$. Increasing the weight vector in the direction of the gradient increases L. Now for each round the new weights are calculated by adding a fraction of the gradient,

$$w_k^{t+1} = w_k^{(t)} + \epsilon \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i). \tag{4}$$

ϵ denoting the learning rate. While iteratively updating the weights, we increase the likelihood, and as it is convex, we eventually reach a maximum. Near the maximum, the changes become smaller and thus the iterations are stopped when the sum of the absolute values of difference in the weights is less than a predefined small number.

For regularized logistic regression, a regularization term is added, enforcing a trade-off between training data matching and future data generalization. For regularized logistic regression objective, the sign is changed and square L2 norm added.

$$L = -\sum_{i=1}^{n} \log g(y_i z_i) + \frac{C}{2} \sum_{k=1}^{l} w_k^2 \tag{5}$$

C is used for balancing between two terms and bias weight is not regularized. The derivatives are

$$\frac{\partial L}{\partial w_k} = -\sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i) + C w_k, \quad k \neq 0, \tag{6}$$

$$\frac{\partial^2 L}{\partial w_k \partial w_k} = \sum_{i=1}^{n} x_{ik}^2 g(-y_i z_i) + C, \quad k \neq 0. \tag{7}$$

And using the gradient for gradient descent we change the sign for minimal search,

$$w_k^{(t+1)} = w_k^{(t)} + \epsilon \sum_{i=1}^{n} y_i x_{ik} g(-y_i z_i) - \epsilon C w_k^{(t)}, \quad k \neq 0. \tag{8}$$

The bias weight updating is identical to the previous non-regularized method.

2.2.3 Cross-validation

Cross-validation is a method for performance estimation of a classifier. In this work, *stratified k-fold cross-validation* is used. In k-fold cross-validation, the initial dataset D is randomly split into k mutually exclusive subsets of elements, or *folds* $D_1, D_2, ..., D_k$ of equal size. The classifier which is cross-validated, is trained and tested k times. In every iteration of i of k, the subset D_i is reserved as the testing set, and the classifier is trained on the remaining $k - i$ datasets. For example, at the first iteration, subset D_1 is reserved for testing and classifier is trained on combined subsets of $D_2, D_3, ..., D_k$. The cross-validation estimate of a performance metric is calculated based on the combined k iterations. We present the formal definition of cross-validated accuracy presented in [K$^+$95]. Let $D = \{x_1, x_2, ..., x_n\}$ be a dataset containing n instances, where $x_i = \langle v_i \in V, y_i \in Y \rangle$, V being a set of unlabeled, and Y, a set of labeled instances. A classifier C maps an instance from $v \in V$ to a label $y \in Y$ and the inducer $\mathcal{I}$ maps a given whole dataset D into a classifier C. Let the notation $\mathcal{I}(D, v)$ denote the label assigned to an unlabeled instance v. Let D_i consist of $x_i = \langle v_i, y_i \rangle$ test sets, then the cross-validated estimation of accuracy would be

$$\mathtt{acc}_{CV} = \frac{1}{n} \sum_{\langle v_i, y_i \rangle \in D_h} \delta(\mathcal{I}(D \setminus D_{(i)}, v_i), y_i), \tag{9}$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise.

As the average of choosing m/k instances out of m is usually too expensive computationally, the estimate of k folds is used. In *stratified cross-validation* the elements in the subsets are chosen so that the folds would contain approximately the same proportion of label classes as the entire initial dataset [K$^+$95].

2.2.4 Hyper-parameter Optimization

Learning algorithms (classifiers), like support vector machines and logistic regression, which are used in this work, ultimately try to learn a function, which would minimize some expected loss over samples from a certain dataset. A learning algorithm itself produces the final mapping function through the optimization of a training criterion with respect to some set of parameters. In addition, the learning algorithms themselves have parameters called *hyper-parameters*. For example, the support vector machine classifier described in Section 2.2.1 can be trained with different penalties (regularization term), like ’l1’ or ’l2’. The purpose of hyper-parameter optimization is to find the best set of hyper-parameters to use, given a certain dataset and thus minimize the generalization error [BB12]. As the dataset for evaluation of milestone approach is rather small and the classifiers that are used are linear, thus are not computationally expensive to train, we use simple *grid search* for hyper-parameter optimization.

Grid Search is a hyper-parameter optimization method which executes an exhaustive search over a predefined subset of hyper-parameters of a learning algorithm. The performance metric used for optimization is the area under the ROC-curve, as it is the metric used for our approach evaluation (see Section 4.5.1 for details). The input to a grid search is the parameter space of parameters and their respective values. For example for SVM classifier, the ’penalty’ parameter can accept values ’l1’, ’l2’ and ’elasticnet’. The combination of parameters which yields the best cross-validated performance metric is the output of grid search.

2.3 Telia Estonia

2.3.1 Overview

Telia Estonia is the largest telecommunications company in the Baltic states. It is a subsidiary of Telia Company, which is the fifth largest telecom operator in Europe today and continues building on its pioneering spirit and high technology expertise within fixed and mobile communications. It was founded in 1853 with now over 27 million subscribers and about 21,000 employees [tela]. Their strategy is to create value through superior network connectivity, increase customer loyalty by creating, across technologies, services and channels, a seamless customer experience and also to ensure competitive operations, through creating agility and cost efficiency [telb].

One of their responsibilities is to provide business clients with telecommunications services. As the company is the largest in Estonia, many high profile clients have opted for the services provided by Telia, in hopes of great service and seamless customer experience. As for many large industry leaders, to be able to provide comprehensive range of services, other smaller companies that provide a certain service, are acquired. The process in question in this thesis is, in fact, subject to this kind of phenomenon. The B2B change management process is still in place as the service was previously provided by another firm, and is now incorporated into Telia. This means that the process is largely different from other processes in place at Telia and includes many shortcomings, for example, the information system that they use is still the same which was used when the same service was provided by another company, fraction of the size of Telia.

At the moment, Telia is in the state of transforming this process and is looking to find ways to improve the process, and as a result, in the near future, develop a process for the change management, which has been tailored to satisfy the needs of all their stakeholders, for the best possible performance. Inge Laas, the Head of Process Management Development Group in Telia, is conducting a Master's thesis on possible ways to model, find shortcomings and improve the overall process. As opposed to Inge's thesis, which focuses on overall process improvement, this thesis aims to apply predictive monitoring to a small part of the overall process, excluding the legal, financial and support domains of the overall process. The justification for this is that the specific technical change management part of the process had feasible event logs and was subject to a large proportion of the combined cycle time of the overall process.

2.3.2 Business to Business Change Management process

The real-life event log, which is used for evaluation of the proposed approach to multilevel process predictive monitoring, is the business-to-business change management process in place at Telia Estonia. The process is a traditional task handling and management problem, as in basically all industries nowadays. The overall process includes financial, legal and support departments, but the particular part that is investigated in this thesis is the technical implementation of a Change. The overall process starts with a customer's wish, based upon which, a more detailed budget and analysis is made, which is then passed to project management and legal departments, from where it arrives at the service delivery segment (handled in this thesis). The last activities are quality control and financing.

The service delivery portion of the process is being analyzed in this work. It starts with an input from the project management and legal department about the legal state of the documents and the confirmation of the change. *The Change* is created by a person responsible for a particular service group through the information system that is used for task management. The input data is then entered and different subprocesses (called *Work Orders*) are created at the same moment in time. The subprocesses, or Work Orders, are created for the various groups or teams which handle some particular logical part of the overall work that is maintained in the parent process, or in terms of this process - a *Change*. The subprocesses run in parallel and usually do not depend on one another, which means that the *Work Orders* can be registered, transition their state or complete, reject, cancel, at any time of the whole process. The parent process, or Change, is managed manually, so that when the project manager knows that all the subprocesses are carried out, and the service change is finished, then the status for the Change is set to *Completed*, for example. This implies that in the logs, there is noise due to human intervention. Another source of data noisiness is due to the fact that project managers forget to change the statuses or edit other data about the Change.

The Changes and *Work Orders* both have textual descriptions of the work that needs to be carried out. It is usually a quite short sentence of technical description. The Work Orders also sometimes receive a textual feature called Result, which describes the details of the work that was carried out and, if needed, then what kind of work needs to be implemented after the Work Order is completed. One important attribute that a Change has, is *the Target Date*. This is a strict target set for the parent process, that the change needs to be carried out due that specified time. This is the label based upon which, in this thesis, the negative impact cases are identified, and which is going to be our label what we classify, as soon as possible since the case execution starts.

The process models mined from the separate event logs are presented in Figures 2 and 3. Figure 2 shows the model for the parent process, or Change, from which then other subprocesses, or Work Orders, are spawned. The process model mined from the Work Order event logs is presented in Figure 3.

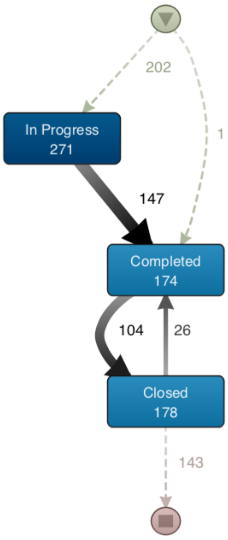

Figure 2: The mined process model of Changes event log with Disco [Roz]. The numbers correspond to absolute frequency, darker color and thicker lines indicate larger values.

2.4 State of the Art

Time-related approaches try to predict the timeliness of the processes to find cases which tend to run overtime. Van der Aalst et al. [VdASS11] proposes an approach that uses a process discovery algorithm presented in [VdARV+10] and builds an annotated transition system. The proposed transition system is annotated with information about elapsed times, sojourn and remaining times. This annotated system is then used for predictions. The label, or prediction outcome, in this case, is remaining process time based on the earlier cases which have visited the same state as the one that's being predicted. Van Dongen et al. [vDCvdA08] proposes an approach that also predicts the remaining cycle time of a case based on historical data by using non-parametric regression which is based on case-related data. The paper explicitly uses information about the durations of all the activities, the occurrence of all activities and any other case-related data. Their approach was proven to be more accurate than the straightforward approach of calculating average durations and subtracting the elapsed time of a case. Jallow et al. [JMV+07] approach defines a framework for identification and analysis of the operational risks which are associated with single business process activities and also

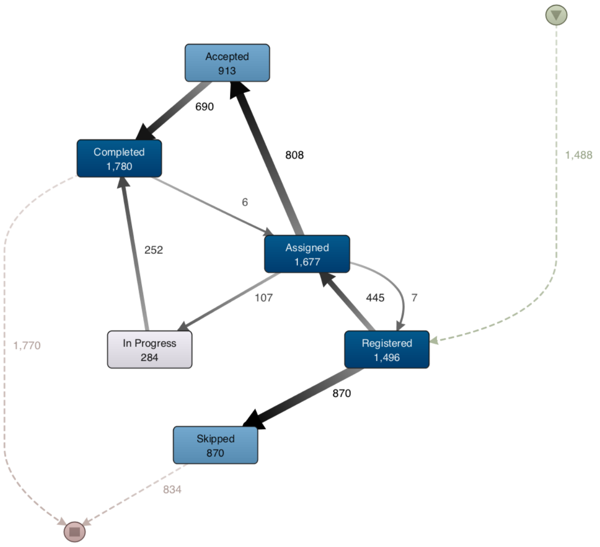

Figure 3: The mined process model of Work Orders event log with Disco [Roz]. The numbers correspond to absolute frequency, darker color and thicker lines indicate larger values.

the whole process, but their research is more focused on risks and does not specifically work with event logs. The research made by Wickboldt et al. [WBL^{+}11] does, in fact, make use of process models and historical event data, and even in the same context of IT Change management, for risk management, but relies on generated process models.

A more specific predictive monitoring framework was proposed by Maggi et al. [MDFDG14]. The paper presents the Predictive Business Process Monitoring framework, that based on the continuous generation of predictions and recommendations, minimizes the likelihood of violations of business constraints, based on the activities to perform and what input data values to provide. It specifically uses Linear Temporal Logic to define business goals, then analyses the execution traces to give likelihoods of achieving defined goals. The predictions depend on two aspects of the historical traces, as opposed to the ones spoken before, which rely on just control-flow or data perspective. Maggi's framework takes into account the sequence of activities executed in a trace (control-flow) and values of data attributes after each activity (data perspective). It continues the prediction as a two-phased approach. Given a case, the similar traces are selected, which have the same completed execution prefixes (events), know as control-flow matching. Next, for each selected trace, a data snapshot is generated, consisting of data about each attribute of a completed prefix. The gathered historical traces are then labeled, as a positive or negative outcome, based on the business rule. That means that the prediction task is mapped to a classification task, where the goal is to determine if a given data snapshot leads to a set business goal fulfillment and with what probability. Finally, a decision tree is used to estimate the likelihood that the business goal will be achieved, for every input attribute value. This framework can be applied for prediction and recommendation.

The approach presented by Maggi et al. [MDFDG14] is furthered in the papers by Leontjeva et al. [LCDF$^+$15] and Verenich et al. [VDLR$^+$15]. Leontjeva et al. [LCDF$^+$15] state that the previous existing approaches are essentially mapping the problem of predictive monitoring to early sequence classification [XPK10] which trains a classifier over the set of prefixes of historical traces in an event log. For predicting, the classifier is used on runtime based on the case's current prefix of the trace. The features for classification are obtained as a symbolic vector, a sequence of characters representing events in a trace. In Maggi et al.'s [MDFDG14] work, only the attributes from the payload of the last event are included. The authors propose an alternative approach, which builds on top of the usual sequence classification, by adding data payload of each event consisting of attribute-value pairs to the symbolic sequence, together making it complex. For the critical part of feature encoding for these complex symbolic sequences, the authors propose two encodings. First based on indexes, which specifies for each event, the event and its additional data payload in that position when the event occurs. The second encoding combines the first with an encoding based on Hidden Markov Models, which is a well-known generative probabilistic technique.

Verenich et al. [VDLR$^+$15] identifies the Maggi et al. [MDFDG14] approach to be relatively accurate, but at the expense of high runtime overhead, as the classifiers which are used are generated at runtime. To make this approach feasible for high throughput and instantaneous response times, which could help users make rapid decisions, the framework has been extended by Di Francescomarino et al. [DFDMT15] by introducing a clustering preprocessing phase. This means that the classification models are precomputed, which decreases the prediction time and allows for high throughput. Similarly for the motivation in [LCDF$^+$15], Verenich et al. extend both of these methods because only the payload data of the last event is taken into account in the paper by Maggi et al. [MDFDG14]. They extend it through the use of a multiple classifier method coupled with the combination of the clustered-based approach. Lastly incorporating the approach based on complex symbolic sequences proposed by Leontjeva et al. [LCDF$^+$15]. Similarly, being two-phased, first, the log prefixes of historical cases are extracted, encoding them with index based encoding thus obtaining feature vectors to be clustered. The second phase trains an intra-cluster classifier. Random forest is used to predict the outcome of an ongoing case.

The key difference between the methods for predictive monitoring, outlined in the presented overview of the state of the art, and the approach proposed in this thesis, is that the previous methods are designed for flat processes. Whereas in our case, the process has a parent and multiple subprocesses, and combining them to a linear process, which could be used for classification methods presented above, is not trivial. When a flat process can be described with a linear prefix of events, a multi-level process would lose the underlying information and would not represent the events as they occur in reality. To overcome this problem, in the next section, we present the notion of milestones, through which one can apply the previously mentioned methods.

This subsection provided the review of the state of the art in predictive monitoring of business processes. The described literature focuses primarily on the prediction of underlying flat processes, however to our knowledge, none of them actually proposes the solution to the prediction of complex multi-level processes. Previous research has been centered around time-related, risk-related and business constraint related predictive monitoring, although they all encompass the effective risk management side of process monitoring. Therefore, we conclude that the contribution of our work is novel.

3 Approach

In order to address the problem of combining the two datasets of a process that has a parent process which includes multiple parallel subprocesses, we propose an approach which relies on so-called *milestones.* Through the use of milestones, the multi-level process is treated as such, as opposed to the straightforward method of combining all events for both event logs by timestamps. In the case of the straightforward method, the events of subprocesses are not in a logical order, and parallel executions of Work Orders are combined into a flat sequence of events. This means that the classifier and model created would not take into account the dimensionality of the process and combine the subprocesses and parent process into a flat model where events are disorganized.

The assumptions needed to be made before presenting the approach are as follows (including basic assumptions used by process mining methodologies). First, it is assumed that there is a historical event log available, containing executions traces from the past regarding the process in hand, from which the information about how the process was executed can be extracted. Based on the historical data, classifiers can be trained for classification. Second, it is assumed, that the business process in question is in some way non-deterministic. The persons involved should not know the mechanisms that guide the decisions taken during the executions of an ongoing trace. This being the reason that any prediction made by the approach would be useless if the process participants already know what the process outcome would be, given a set of input data.

3.1 Milestones

The core parts of the approach proposed in this thesis are *milestones.* Instead of using the straightforward approach for multi-level processes including the control-flow (meaning the sequence of events in a trace) as the state of the art research suggest for flat models at the moment, we propose to use milestones which could better maintain the dimensionality of the parent-subprocesses and increase the prediction performance. The purpose of this approach and the use of milestones is to aggregate the data, structure it, without losing much data during the aggregation.

In our context a milestone can be interpreted as a function, that given the combined traces of parent and subprocesses, including the prefix of events that have occurred, the function returns 'True' if a particular milestone has been reached, or 'False' is a milestone has not yet been reached.

$$\text{Milestone: } \mathcal{I}(trace_i) = \{ \text{ True, if } m_k \in M; \text{ False, if } m_k \notin M \ \}, \qquad (10)$$

where $\mathcal{I}$ is indicator function, M is a set of reached milestones, $k = 1, ..., |M|$.

Milestones declaration is a manual process based on the prior knowledge of the process in question. For example, in the case of change management process in Telia, we propose milestones which include all the different events in the Change workflow (Change Created, Change In Progress, Change Completed, Change Closed) and also creation and completion events in Work Orders workflow according to the number of the Work Order in the overall sequence (First Work Order Created, Second Work Order Created, Second Work Order Completed, etc).

For clarity, one can imagine that before a case starts, all of the defined milestones are initiated to 'False'. Now every time that an event occurs in the parent process, or in one of the subprocesses, the case is evaluated against all the milestones. So for example, the criteria for evaluating milestone Second Work Order Created as 'True', the traces of the parent and subprocesses need to contain events that mark the registration of a Change (parent process), and also two Work Orders (subprocesses) have to be created. If any of these milestones which are defined evaluate to 'True', a prediction is made using the classifier for that specific milestone. Lastly, the classifications are combined. This is further specified in Section 4.2.

3.2 Feature Engineering

Feature engineering plays an important role in predictive analysis or any machine learning task. Feature engineering in concerned about how to generate features from an initial dataset so that the feature vector used by the data mining algorithm would be the most relevant and in the best suitable format. In this section, the rationale is presented, based on which the features of the evaluation data were engineered and transformed.

3.2.1 Categorical Feature Expansion

Categorical features are features which represent a value as a category rather than a magnitude. The dataset used for evaluation primarily contains only categorical values. The approach that is used in this thesis, to transform categorical values to create better predictors, is called expansion. The core principle of this method is that the categorical values are expanded into multiple boolean features. This means that each of the values of the categorical feature that appears in the dataset is transformed into a separate binary feature which indicates if it was that specific categorical value.

3.2.2 Text Modeling

To accommodate the fact that free-text features could be used to improve the predictions for our implementation of predictive monitoring, the textual descriptions need to be transformed into features. While implementing the proposed milestone approach, two different text mining methods were compared so that the best performing one could be used for this particular dataset. The two method which were compared, are *Latent Dirichlet Allocation* (LDA) and *Bag-of-n-grams* (BoNG). In our presented results (Section 4.6) besides the evaluation results of milestone based approach to predictive monitoring, we present the results of comparing textual feature creation methods LDA and BoNG, compared to not including the textual descriptions at all.

Before applying the textual feature extraction methods, the unstructured free-text data fields were tokenized with whitespace tokenization. All of the numbers and punctuation marks were removed, as the evaluation dataset periodically contains numerical codes, IP-addresses, and other seemingly unnecessary information, that can not provide information for a classifier, i.e., it does not include phone numbers, dates or times, that could be generalizable. The tokenized text is then lemmatized (in Estonian). Lemmatization is a computational linguistics term, describing a process of defining which word variants belong under a common lemma. Lemma represents a group or a cluster and does not have to be a basic word form [Alk01]. The different inflected forms of a word are grouped together to a base called lemma.

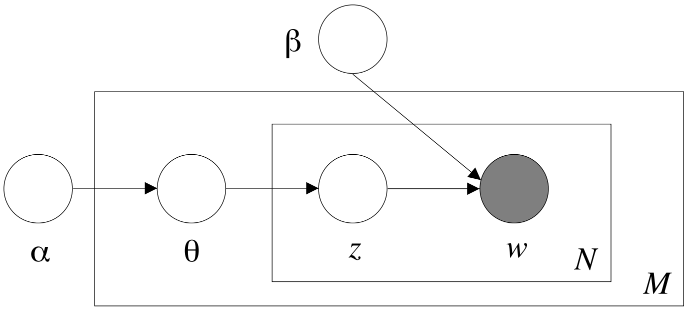

Figure 4: Graphical model representation of LDA.

An n-gram is a slice of N-characters which is part of a larger string [CT+94]. The slices can be characters of words or entire words altogether. In the current thesis, we use whole words. Unigram is the smallest n-gram, which corresponds to one character or word. The number of how many characters or words are included in the features is called n-gram. If unigrams are used, the resulting methodology is called 'bag-of-words'. If the number is increased, the bag also includes phrases and multi-word expressions, which could improve the performance of the classifier, as more features are added to the model. When enlarging the n-gram range - the number of words included when creating the resulting feature space - the overall performance of the system will increase. Although, adding higher n-gram limits, the performance will eventually flatten out or decrease [Ver14]. This limit of reasonable n-gram range largely depends on the type and quantity of the used data.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for a set of discrete data such as a corpus. The primary goal of LDA is to find fairly short descriptions of the members of a large collection so that the statistical relationships are preserved for tasks like classification and summarization, while being efficient. Documents (collections of discrete data) are represented as random mixtures over latent topics. Each of these topics is defined by a distribution over words [BNJ03].

The LDA generative process by [BNJ03] is described below. For each document w in a corpus D, LDA assumes the following generative process:

1. Choose $N \sim$ Poisson distribution(ξ)

2. Choose $\theta \sim$ Dirichlet distribution(α)

3. For each of the N words w_n:

    (a) Choose a topic $z_n \sim$ Multinomial(θ).

    (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n

The graphical model representation of Latent Dirichlet Allocation is shown in Figure 4. It describes the three levels of LDA, parameters α and β being corpus-level parameters, sampled once in the corpus generating process. θ_d sampled once per document, being document-level variable and z_{dn}, w_{dn} being word-level variables. The last two are sampled once for every word for each document.

4 Evaluation

The approach to multi-level process predictive monitoring is evaluated by conducting the experiment on a real-life dataset from Telia Estonia. The data is extracted from a task management software. The time frame spans from January 5th, 2015 to January 15th, 2016. The dataset corresponds to a B2B change management process discussed in detail in Section 2.3.

4.1 Dataset

The initial data is provided in two different batches. One is for Changes event log (the parent process), second for Work Orders event log (the subprocess). The Work Orders are associated with Changes by an identificator. In the initial dataset, there are 261 cases with a median case duration of a Change being 28 days. Both of the event logs are presented in the same format. The logs are provided as an Excel file, which has rows as events, each having a column for Change or Work Order identification number. The basic data payload of an event is provided as columns, but many data fields corresponding to an event attribute, are presented as an assignment event on a separate row. Also the statuses, or specifically, the status change events that we consider as control flow (order of events in a case), are not presented as in a standard event log, as for example 'Accepted', but are represented as a status change from another status, and are textual, e.g., 'Status set to "Registered"' or 'Status from "In Progress" to "Completed"'. An example of how an event attribute is represented in the similar way - 'Reason set to "Expansion of Infrastructure or Service"'. This implies that in preprocessing there is a need to extract these control-flow events and data payload attribute-value pairs and transform them to a more standard representation, to only include control-flow events as rows in the dataset and transform data payload to columns. There are no specific event attributes, only case attributes representing the whole one execution of a case. The two Tables 1 and 2 give an example of what was the format of the initial event logs and what information it contained. The more distinct attributes and specifics of these events logs are described in the next two subsections.

Change ID	Description	...	Subject	Timestamp
1	Example 1	...	Status set to Registered	2015/01/05 00:00:00
1	Example 1	...	Target date set to 12.01.15 10:33 EET	2015/01/05 00:00:01
..				...
261	Example 261	...	Status set to Closed	2016/01/05 00:00:00

Table 1: An example of the initial Change event log.

Change ID	Work Order ID	...	Subject	Timestamp
1	1	...	Status set to Registered	2015/01/05 00:00:01
1	1	...	Group set to Maintenance	2015/01/05 00:00:02
..				...
261	10000	...	Status set to Canceled	2016/01/04 23:59:59

Table 2: An example of the initial Work Orders event log.

4.1.1 Changes

To get a better insight into the data provided, we investigate the Changes event log. Changes represent the top-level process in hand in this thesis, which focuses on combining subprocesses with parent processes, so as stated in the previous introduction, there are 261 initial cases of Changes. As the purpose is to get insight into the whole process, the Changes are in fact the process cases which are the core of this predictive monitoring task. In order to do this, a label needs to be assigned to the cases based on the outcome of what we try to predict. In our case, according to input from Telia's Process Manager, and further inspection of the logs, the outcome, which is classified in this thesis, is *'Target date changed'*. The target date is a case attribute which is assigned to a Change at the moment it is created. The target date represents a timestamp which marks the time, on which a certain Change needs to be completed. This is not just an estimate to the Change, but furthermore, the target date is stated in the agreement signed with a customer business which orders the Change. This means that target date postponing is not tolerated and not only causes damage to the Telia brand but can also result in profit loss to the company. This is the rationale why the target date postponing label is chosen for the outcome that is going to be predicted. Also, if we look at Figure 5 we can see that the median duration of a case to transition from 'In progress' to 'Target date changed' is 14.4 days, a loop of 14 days where the target is again changed, and then the same 12 days to complete the case, as for just transition time from 'In progress' to 'Completed'. This may be intuitive, that the longer a case runs, the more likely it is that the target date will be changed, but we need to keep in mind that the target date is set in the initial stage of the change creation and should adequately correspond to the estimate of how long a Change should last. This means that the target date attribute should not be changed during the case life cycle.

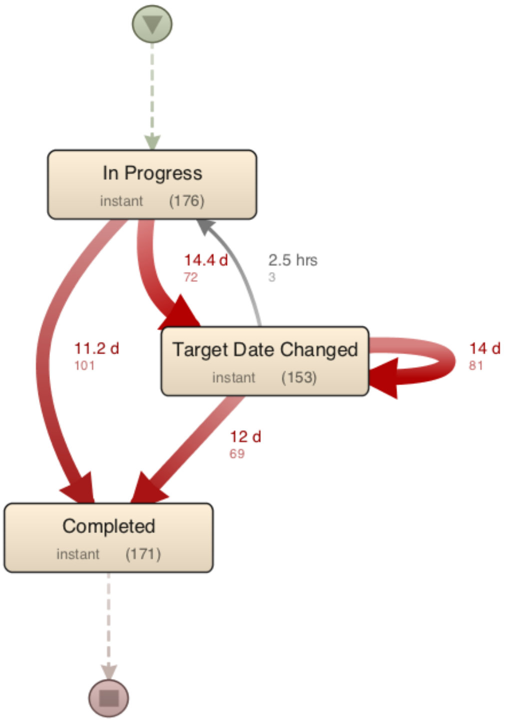

Figure 5: The mined process model with median duration (upper value) and absolute frequency (lower value) of transitions of the Changes event log (reduced to three main events).

Table 3 summarizes the characteristics of Change event log, after transforming the dataset to a more standard format, i.e., the events are represented only as Change statuses. Also, only the cases which have finished are included, so the total amount of cases is reduced.

As we can see there are very few event classes in the dataset and also the median trace length of 3 immediately shows that there might be little information in the Changes dataset alone. Applying predictive monitoring would most certainly yield bad results as

Characteristic	**Changes event log**
Normal cases	123
Deviant cases	89
Total cases	212
Median trace length	3.00
Number of events	613
Event classes	4
Median Work Orders per Change	12
Maximum Work Orders per Change	96

Table 3: Characteristics of the Changes event log.

there is little control-flow data. Regarding the payload data, the attributes for Change cases are Case_Description, Organization, Created_By, Group and Reason.

4.1.2 Work Orders

Next, the Work Orders event log is investigated. Work Orders represent subprocesses of the Changes process. Each Change can have multiple Work Orders which can be created in any part of the Change workflow. Work Orders encompass more detailed information than Changes, have more complex process workflow and are more specific.

Table 4 summarizes the characteristics of Work Order event log, after transforming the dataset to a more standard format, i.e., the events are represented only as Work Order statuses. Also, only the Work Orders which are subprocesses of the filtered Changes (Changes which have completed), are included. Regarding the payload data, the attributes for Work Order cases are Description, Classification, Group, Relation_CI and Result. Description is a free-text specification of the task in hand. Classification determines the overall classification of the Work Order. For example, it shows if the task involves changes to the firewall, operation system, servers, hardware, etc. There are 61 distinct classification types. Group specifies the workgroup, to which the Work Order is forwarded for implementation. There are 26 different groups to which these Work Orders are assigned in this event log. Relation_CI is an internal attribute with 261 values. Finally, the Result is also a field with the free-text description about the final state of the

Work Order. This is used if further action is needed regarding the specific Work Order or parent Change. This attribute is only included as a parameter if it has been set, as explained in the preprocessing subsection.

Characteristic	**Changes event log**
Total cases	2732
Median trace length	2.00
Maximum trace length	9.00
Number of events	7297
Event classes	10

Table 4: Characteristics of the Work Orders event log.

4.2 Preprocessing and classification

As the evaluation of the proposed method for predictive monitoring is conducted in Python [pyt], the two event logs in Excel spreadsheet were first converted to comma-separated value text files and then read into Python. Both of the logs contained rows corresponding to events, but as specified in dataset subsection 4.1, many other value assignments were represented as events also. So first, both of the event logs were iterated and necessary attributes were added to the case attributes using text processing, as the events were defined in such a matter. For the second step, only the events for statuses were filtered from the log, which resulted in a traditional event log format, where events (rows) represent statuses of a process and attributes hold the additional payload data. The filtering also included excluding the Change cases which were not finished before the end timestamp of the event log, including the Work Orders associated with the Changes.

After the datasets were formatted as event logs and included all valid cases, the two datasets were combined and sorted first by Change ID and second by Event Timestamp. Also, in order to evaluate the milestones for a given case, and to include their information in the training dataset, we annotate the first and last events of Work Orders.

As the thesis deals with multi-level processes and their corresponding event logs, the classification mechanism is based on a three-phased approach. First, a dataset is gener-

Change ID	Textual descriptions	...	Change Attribute 1	...	1st Work Order Attribute 1	...	1st Work Order Prefix	...
1	Free-text	...	Value	...	Value	...	Prefix	...
..								...
261	Free-text	...	Value	...	Value	...	Prefix	...

Table 5: An example output of the first milestone dataset creation step - a combined structured dataset of Changes and Work Orders.

ated for the particular milestone, which includes the appropriate information. Second, the data is preprocessed for a classification algorithm and then classified. To receive an overall classification for a specific execution of a trace, all of the milestones are evaluated, and if a specific milestone evaluates to true, the case is classified with that milestone classifier. The third phase incorporates the ensembling of all the milestones' classifiers, to create a final classification of a running case. Figure 6 presents the simplified three-phased approach in BPMN [OMG].

Step 1. Milestone dataset creation includes the iteration of all the Changes (top-level process) and creating a separate dataset consisting of cases which are combined together from Change and its subprocesses Work Orders. The process starts by assigning the Change case attributes to a combined case. As the Change is always created before the Work Orders, we can simply state that no matter what milestone has been reached, the Change has always been created and its attributes can be used for a particular prediction model. Then all the Work Orders which are created are iterated, and again, the attributes of specific Work Orders are added to the combined case as features. Also, for every Work Order that is added to the feature space, its control-flow data (execution order of events until a specific point in time) is added. This finally outputs a dataset which consists of rows corresponding to combined Changes with Work Orders, so the dataset contains the same amount of cases as Changes event log. The feature space consists of Change control-flow, it's data attributes, and also of each Work Order's control-flow and data payload. The example output of this step is presented in Table 5.

Step 2. Milestone classification. The previously generated dataset for a specific milestone (e.g. SecondWorkOrderCreated) is the input to this phase. As for all classification methods, the output feature is extracted from the dataset, in our case 'Target date changed'. The free-text features (Case_Description, Description, Result) are separated

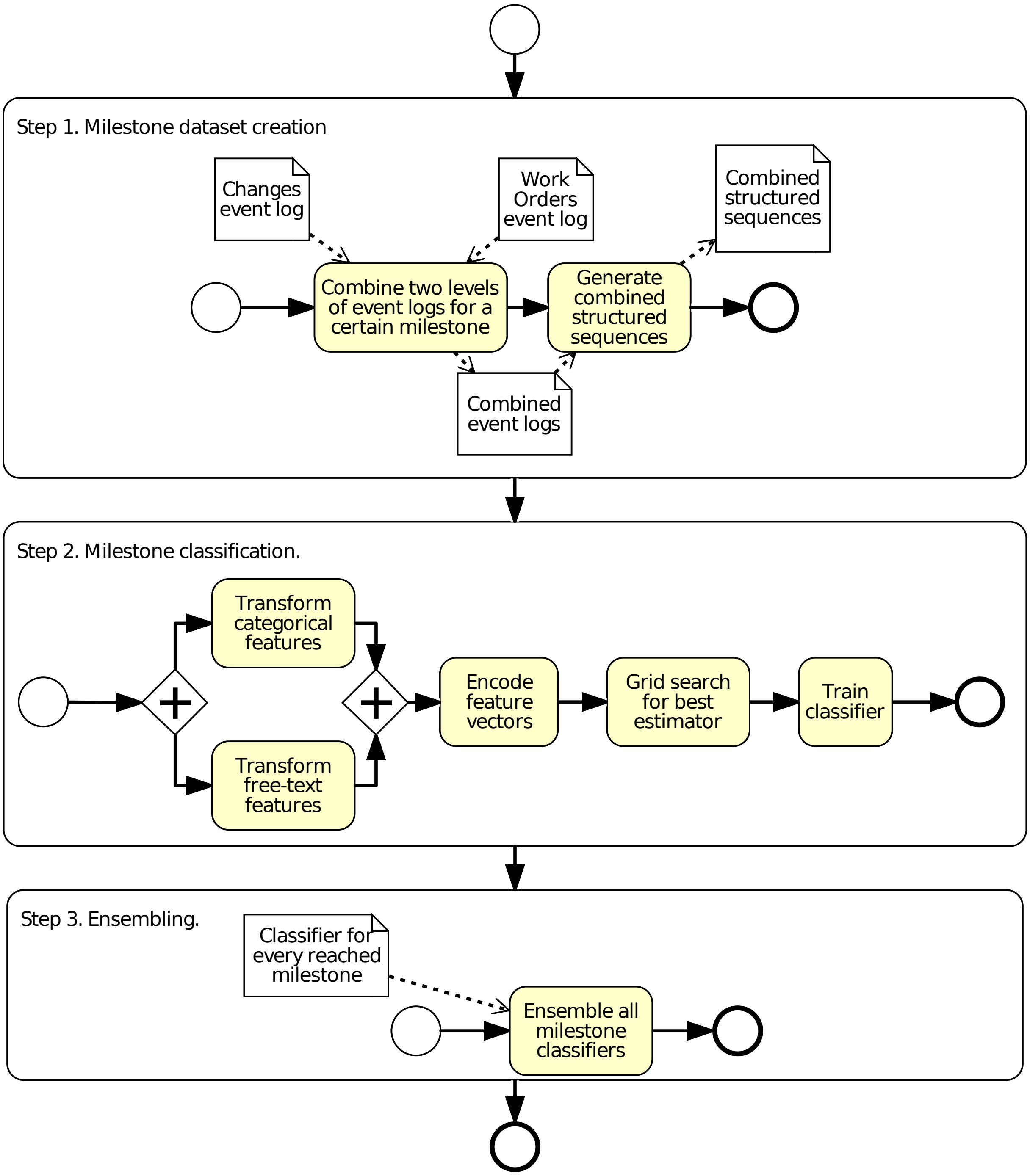
Step 1. Milestone dataset creation
Changes event log
Work Orders event log
Combined structured sequences
Combine two levels of event logs for a certain milestone
Generate combined structured sequences
Combined event logs
Step 2. Milestone classification.
Transform categorical features
Transform free-text features
Encode feature vectors
Grid search for best estimator
Train classifier
Step 3. Ensembling.
Classifier for every reached milestone
Ensemble all milestone classifiers

Figure 6: The preprocessing and classification process.

from the categorical features. Categorical features are then processed by a method called "one-hot" encoding which implements the categorical feature expansion explained in Section 3.2.1. This is the recommended method for encoding categorical features, which all of our features are, except free-text attributes. The basic idea of a "one-hot" coding is that one boolean-valued feature is constructed for every one of the possible categorical string values that the feature can take on. For instance, for the feature 'Group' all of the 26 different possible values will be transformed into a binary feature, signifying a specific value, i.e., 'Group=Maintenance'.

After the encoding of the categorical values, free-text attributes are processed. In the previous phase of dataset generation, all the free text features are concatenated over all the subprocesses of a Change. The Change itself has a textual description feature 'Case_Description', which by itself is an attribute. The 'Description' feature of Work Orders is specific for every Work Order. That means that for the combined dataset concatenate all 'Description' attributes of Work Orders associated with a Change. The same goes for 'Result' attribute, but it needs to be kept in mind, that only the textual descriptions are included, which have been assigned to a Work Order. So if the 'Result' attribute of a Work Order is not filled before a certain milestone, then it is not included in the dataset. After obtaining the lists of values for each of these features, numbers are deleted from each of them, as the descriptions largely contain specific codes used in Telia Estonia, and do not include values which could be of interest to the classification task, like telephone numbers or dates. For better prediction performance each text row is lemmatized (described in Section 3.2.2) and the output vector of lemmas is again concatenated into one case string attribute. The lists of text are then transformed to feature vectors, using the specific transformers, Bag-of-n-grams transformer in one case and also LDA transformer for the Latent Dirichlet Allocation method. The parameters for the two transformers were optimized manually and are described in the Results subsection.

The feature space from "one-hot" encoding of categorical features and features generated by textual value transformers are combined to create one big feature space which is used for the classification task. The combined feature space is standardized by removing the mean and scaling to unit variance. The dataset is then randomly separated to 70 percent training and 30 percent testing set. Centering and scaling are carried out separately by calculating the statistics on the training set and are after used to transform the testing set also. Standardization is a common requirement for a large number of machine learning algorithms.

For classification regularized linear classifiers with Stochastic Gradient Descent are used. Logistic regression and support vector machines are used as the linear classifiers. As detailed in the next paragraph, the grid search algorithm is used to find the best suitable linear classifier, meaning that each set of training data is trained with both of these classifiers and the best performing one is selected. The classifiers are implemented with stochastic gradient descent learning, meaning that the loss is estimated each sample and the model is updated with a decreasing strength learning rate. Other two parameters the grid search is evaluating are regularization term (penalty) and alpha (multiplies the regularization term).

Grid search is used for hyper-parameter optimization. Grid search is an exhaustive search over the input parameters for an estimator. This means that the parameters of the estimator, or classification algorithm, need to be specified. Based on this specified set, the Grid Search applies the parameters, so that they are optimized by cross-validated grid-search over a parameter grid. As the search also outputs the AUC-score for the estimator, we carry out a separate cross-validation with the same estimator and the same parameters. The stratified 10-fold cross validation AUC-score is produced for every classifier on the training dataset. Lastly, the testing set is classified and performance scores are calculated based on the output classifications.

Step 3. Ensembling of the different milestone predictions is used to generate a more accurate predictor. As said before, a running case is evaluated against a set of milestones. For each of the milestones that the execution evaluates to true, or in other words the milestones that the case has reached, a separate classifier is generated. These classify the running case, and finally, we use a simple majority vote method to combine these separate predictions to the final prediction of a running case.

4.3 Baseline

For evaluation of the performance of the proposed approach to predictive monitoring of multi-level processes presented in this thesis, we compare it to a straightforward approach. As there have not been other works, to our knowledge, related to the subject of how to handle processes which include subprocesses, in predictive monitoring context, the baseline method applies the generally used method for flat models. This means that the prediction is based on a preprocessed dataset which consists of control-flow and data payload. The specifics are detailed below.

As also described by Verenich et al. [VDLR^{+}15] and Leontjeva et al. [LCDF^{+}15], the baseline prediction is made based on previous known control-flow information and data attributes of all the events in the execution. Adopting this approach to multi-level processes means that although the control-flow data and data attributes are sequenced based on the occurrence of the events, they are all intertwined with events from Change and all the subprocesses (Work Orders). This, in turn, means that the output event execution sequence is not intuitive, they appear in a sequence, but it is a ”combined” trace where the Change and all of the associated Work Orders are combined and sorted based on the timestamps. In the preprocessing phase, the events are added to a combined dataset based on occurrence and do not actually contain meaningful information, as Work Orders' events and related data attributes are scattered across the whole feature space. The proposed milestone approach tries to intuitively conquer this problem by aggregating the Work Orders' data into a single instance of created or completed, while losing the least amount of data possible, and through the structuring of subprocess' data, hopefully, result in better prediction performance against this straightforward approach.

For preprocessing, first all the events of the Change and associated Work Orders are combined and sorted by the event timestamp at which it happened. The combined list of all events regarding a Change is then iterated and for each event which has not exceeded the number of events we want to include in the prediction, features are added to the dataset - event and it's corresponding Change or Work Order data attributes. This altogether created the dataset which is used for the prediction model. Other preprocessing activities are exactly the same as described in the previous subsection about the preprocessing of the proposed milestone based approach (Section 4.2).

4.4 Comparison methodology

One part of the evaluation of the proposed approach focuses on the comparison of the straightforward baseline approach to the proposed method using milestones. The way that these two seemingly different methods were compared was as follows. The baseline approach was the starting point. The dataset was randomly separated into 70 percent training set and 30 percent testing set. To classify an ongoing trace of a certain length n we build a classifier based on only similar cases that have a combined trace length of n. As the SVM and logistic regression with stochastic gradient descent generated predictions are slightly different on each model fitting, the model was generated 20 times and the performance evaluation metrics were then averaged to obtain an average result. To recap, the baseline prediction performance was obtained by averaging an individual trace length predictions 20 times with 30 percent training set.

To compare these different approaches, the same training and testing set, as described in the previous chapter, were used to classify the testing set traces based on the milestone approach. This meant that for each of the traces in the dataset, and for every number of n length traces computed for baseline - the milestones of a certain trace were evaluated. For example, if a trace of length three was predicted, first, all the milestones were assessed. If a certain milestone was reached within these three events from the combined events of Changes and Work Orders, then a classifier was built for that specific milestone and so on. After which the ensemble methods were used to combine these different milestone predictions.

4.5 Evaluation Metrics

For evaluation metrics for the comparison of the baseline and milestone approach to predictive monitoring, Area Under the Curve score and F-score are used because the simple classification accuracy is generally considered to be a poor metric for measuring classifier performance [LHZ03, PFK98, PF+97].

4.5.1 Receiver Operating Characteristic

For model evaluation the area under the Receiver Operating Characteristic (ROC) curve, known as the AUC is used. AUC is used as a standard method to assess the accuracy of predictive distribution models. Spackman [Spa89] is considered to be the first adopters of this signal detection theory tool as an evaluation and comparison method for machine learning algorithms. Fawcett [Faw06] states that the ROC graphs are conceptually simple, provide a richer measure of classification performance than other measures used, like accuracy and error rate. The ROC curve is created by plotting the true positive rate, also known as sensitivity against the false positive rate also known as recall.

We present a simplistic approach to AUC calculation as presented by Hand and Till [HT01]. The estimate of the probability that a randomly chosen example of class 1 has a lower estimated probability of that of belonging to class 0 than a randomly chosen class 0 point is

$$\hat{A} = \frac{S_0 - n_0(n_0 + 1)/2)}{n_0 n_1}. \tag{11}$$

where n_0 is the number of negative examples and n_1 the number of positive examples. $S_0 = \sum r_i$, in where r_i denotes the rank of i_{th} positive example in the ranked list.

4.5.2 F-score

In order to introduce our second metric f-score, we clarify the notions of confusion matrix, precision and recall. For a given number of m classes (in our case 2), the confusion matrix is at least an m times m matrix, where at each point $CM_{i,j}$ indicates the number of examples of class i that were labeled as class j by the classifier. Also, the points represent true positives, which indicate the number of positive examples correctly labeled by the classifier, false positives which indicate the negative examples incorrectly labeled. False positives are negative examples which are predicted incorrectly and false negatives being positive examples labeled incorrectly. An example of a confusion matrix is presented in Table 6. From this, we can define F-score ($F1$) as the harmonic mean of precision and sensitivity, or in other terms,

		Predicted label	
		C_1	C_2
Actual label	C_1	true positives	false negatives
	C_2	false positives	true negatives

Table 6: A confusion matrix

$$F1 = \frac{2TP}{2TP + FP + FN}. \tag{12}$$

4.6 Results

Before presenting the results of the comparison between baseline and the presented milestone approach to predictive monitoring, we demonstrate the comparison of the two text modeling methods (Latent Dirichlet Allocation and Bag-of-n-grams) to justify the reason why Bag-of-n-grams is used. LDA and BoNG comparison was performed on single milestone predictions, finishing with the 5WorkOrdersCreated milestone. The parameters used for the two methods (LDA and BoNG) are chosen manually by comparing different parameter sets. The final parameter sets used throughout the comparison, are for BoNG: 1 as the minimum number of n-grams, 2 as the maximum number of n-grams and 100 best features used. For LDA: 15 topics, 10 passes and 500 iterations.

The AUC-scores of the three different text mining method are shown on Figure 7. The BoNG method performs clearly better than LDA and the model with no textual features. Surprisingly, LDA seems to yield even worse AUC-scores than not using any textual features. The results can indicate the fact that the free-text descriptions used in this dataset can not be labeled to a topic and that the descriptions are more technical phrases. Based on this result, we use BoNG method for our evaluation of the proposed milestone approach.

Next, we evaluated the two different approaches to predictive monitoring, the straightforward baseline approach and proposed milestone based approach, for traces with length 1 to 19. The interval was chosen because as the median Work Order (subprocess) length

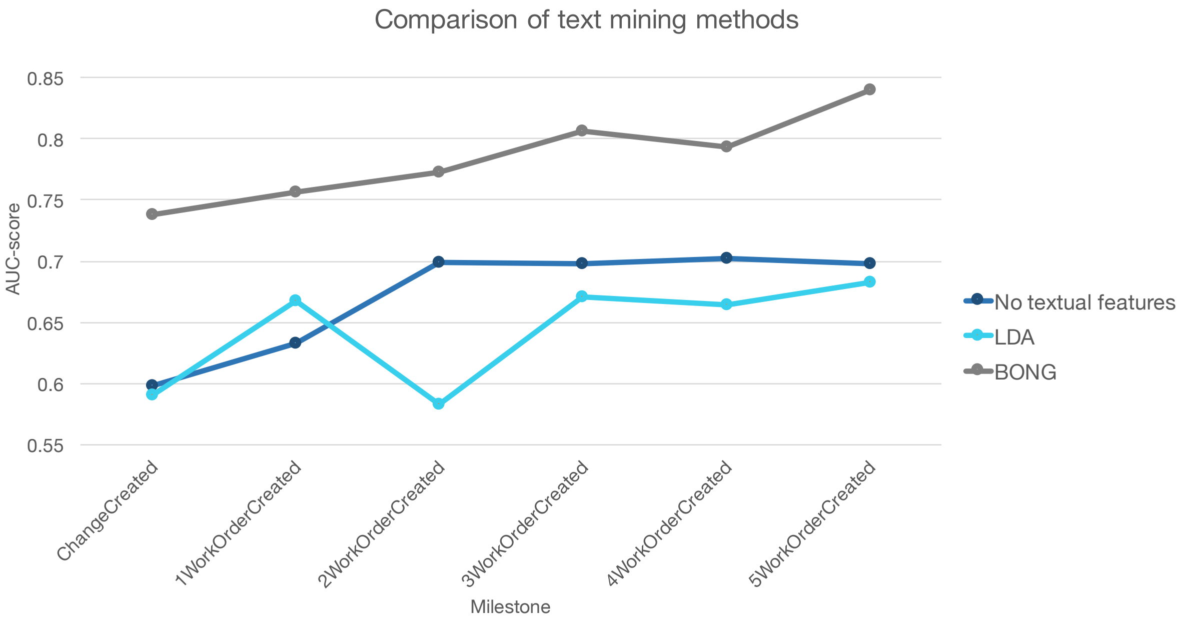

Figure 7

is 2 and the median number of Work Orders in a Change (parent process) is 12, we can quite clearly see the performance differences and majority of the cases included would not be finished. The AUC-scores of these methods is presented in Figure 8 and F-scores in Figure 9.

As can be seen from the Figures 8 and 9, the proposed milestone approach exceeds the performance of the baseline straightforward approach for every number of events included in the comparison, up to 19 events from the beginning of the Change, combined with Work Orders. The results also seem to correlate quite well, this being because both of the approaches use almost the exact amount of data, at each point. The first 8 events appear to be more correlated. From there on, the margin between the two approaches regarding the evaluation scores start to differ more than in the beginning. This can confirm our hypothesis, that using the information about the parent and subprocesses in a more structured way, than just using the events from all the associated processes combined, can yield better classification performance. The best performance of the milestone approach is reached when the number of events is $n = 15$, when the AUC-score and F-score both achieve values > 0.85. For prefix sizes > 15 the performance starts to decrease.

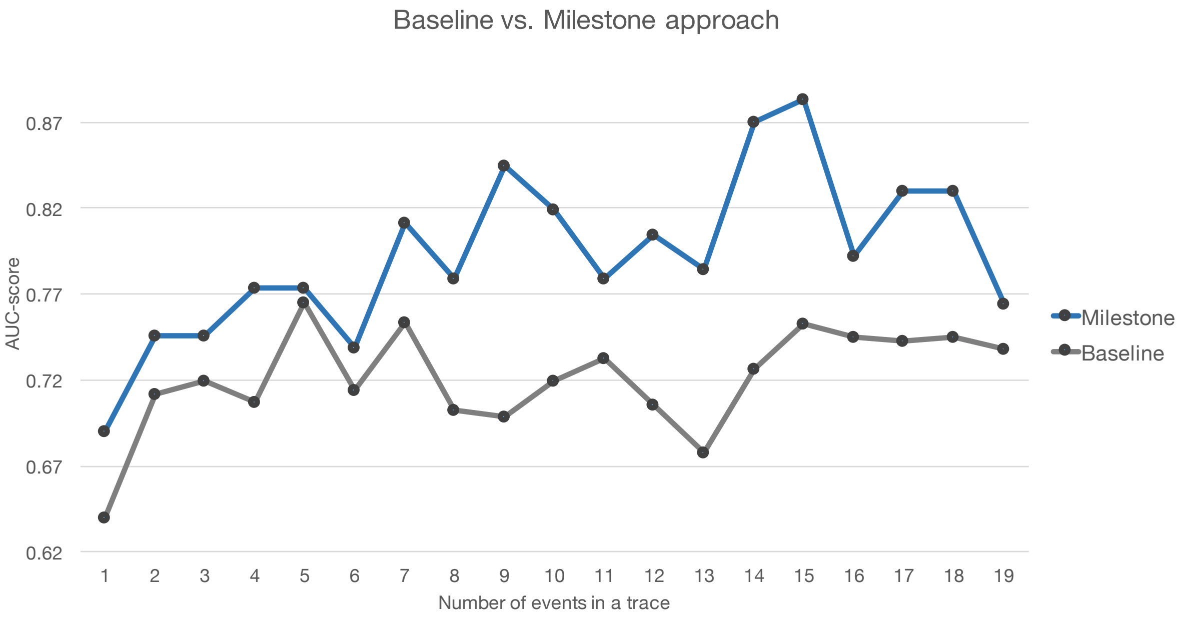

Figure 8

Also a graph showing the individual ROC curves for baseline and milestone approach is presented in Figure 10. The curves are from prediction models generated for the first $n = 9$ events in a trace. The prefix length $n = 9$ was chosen as it yielded close to the best prediction performance from the comparison Figures 8 and 9. As we believe that for the prefix length $n = 13$ and $n = 14$ the individual executions of a process have already progressed significantly, event threshold 9 is a good predictor in a real-life scenario. As can be seen from the Figure 10, milestone method exceeds the baseline throughout the curve. Both of the curves are fairly uniformly distributed, but it appears that both, milestone and baseline classifiers, perform better in the more conservative region of the graph (left-hand side), which means that the classifiers yield better results when identifying likely positives than likely negatives [Faw06].

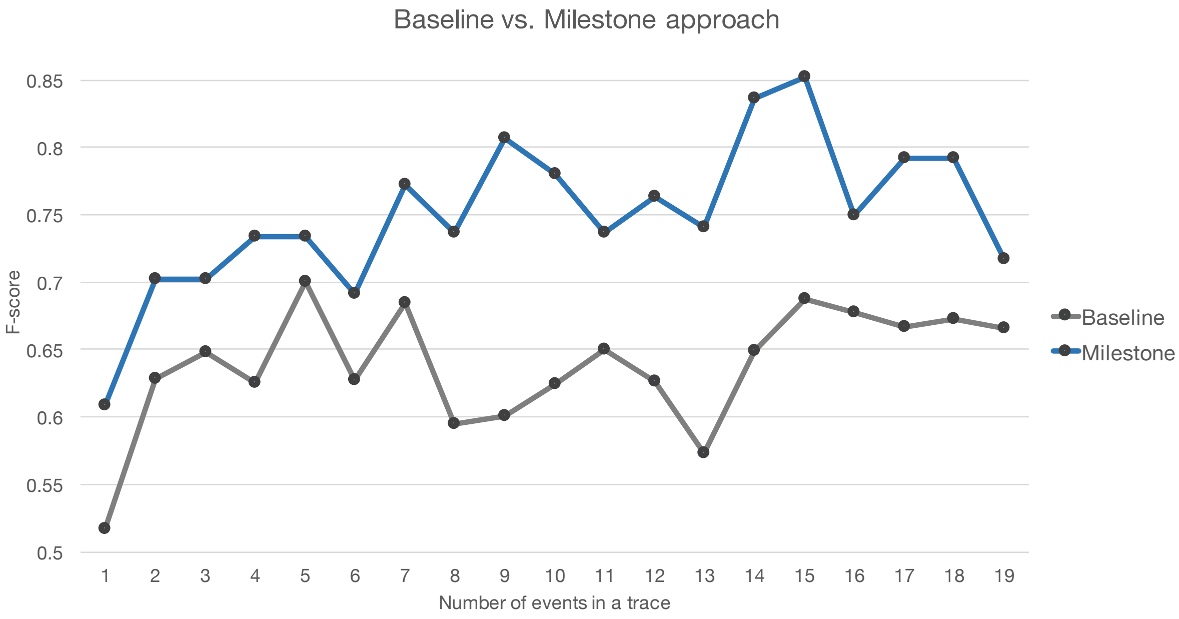

Figure 9

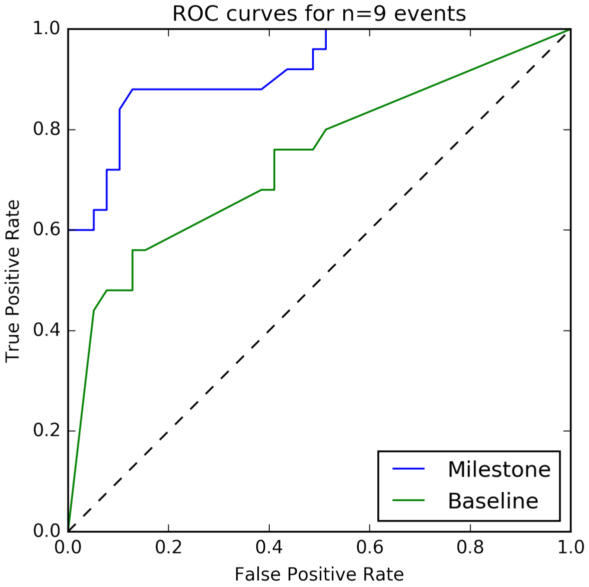

Figure 10

5 Conclusion

In this thesis, we have presented a milestone approach to address the problem of predictive business process monitoring of complex multi-level processes. The evaluation on a real-life dataset from telecommunications domain has shown that a milestone based prediction model achieves greater prediction performance figures throughout the compared trace lengths than the straightforward method of using index-based encoding that is designed for flat processes. While the results of evaluation on one dataset indicate that the proposed method exceeds baseline predictions, the margin of prediction performance boost is not completely systematic.

The threat to the validity of the presented approach is that the evaluation is only conducted on one event log, which although being a real-life log of a complex business process, the results may not be generalizable to other similar logs. This means that when used on different event logs and predicting different outcomes of a given trace, the method could yield a lower predictive power.

The most important direction for future work is to conduct the evaluation of the proposed milestone method on a wider set of logs to receive more indication of the predictive performance benefits over other methods. Also implementing further optimizations of the proposed method to increase predictive power is a possibility for future work.

References

[Alk01] Riitta Alkula. From plain character strings to meaningful words: Producing better full text databases for inflectional and compounding languages with morphological analysis software. *Information Retrieval*, 4(3-4):195–208, 2001.

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[CCD08] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia*, pages 21–49. Springer, 2008.

[CM98] Vladimir S. Cherkassky and Filip Mulier. *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1998.

[CT$^+$94] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[DFDMT15] Chiara Di Francescomarino, Marlon Dumas, Fabrizio Maria Maggi, and Irene Teinemaa. Clustering-based predictive process monitoring. *arXiv preprint arXiv:1506.01428*, 2015.

[Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition.

[HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[HT01] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186, 2001.

[JMV$^+$07] AK Jallow, B Majeed, K Vergidis, A Tiwari, and R Roy. Operational risk analysis in business processes. *BT Technology Journal*, 25(1):168–177, 2007.

[K$^+$95] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[LCDF$^+$15] Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. Complex symbolic sequence encodings for predictive monitoring of business processes. In *Business Process Management*, pages 297–313. Springer, 2015.

[LHZ03] Charles X Ling, Jin Huang, and Harry Zhang. Auc: a statistically consistent and more discriminating measure than accuracy. In *IJCAI*, volume 3, pages 519–524, 2003.

[mar] Telia Eesti juht: sailitasime oma turuosa. `http://w3.ee/openarticle.php?id=2320967&lang=est`. Accessed: 2016-04-28.

[MDFDG14] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, and Chiara Ghidini. Predictive monitoring of business processes. In *Advanced Information Systems Engineering*, pages 457–472. Springer, 2014.

[OMG] Business Process Model OMG. Notation (bpmn) v2. 0, 2011.

[PF$^+$97] Foster J Provost, Tom Fawcett, et al. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *KDD*, volume 97, pages 43–48, 1997.

[PFK98] Foster J Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *ICML*, volume 98, pages 445–453, 1998.

[pyt] Python software foundation. python language reference, version 2.7. `http://www.python.org`. Accessed: 2016-04-28.

[Ren03] Jason Rennie. Logistic regression. *online*, 23, 2003.

[Roz] Anne Rozinat. Fluxicon: Disco User's Guide. `https://fluxicon.com/disco/files/Disco-User-Guide.pdf`.

[Spa89] Kent A. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In Alberto Maria Segre, editor, *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160 – 163. Morgan Kaufmann, San Francisco (CA), 1989.

[tela] Telia Estonia company in brief. `http://www.teliacompany.com/en/about-the-company/telia-company-in-brief/telia-company-in-brief/`. Accessed: 2016-04-28.

[telb] Telia Estonia strategy. `http://www.teliacompany.com/en/about-the-company/strategy/strategy/`. Accessed: 2016-04-28.

[VDA11] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

[VdARV$^+$10] Wil MP Van der Aalst, Vladimir Rubin, HMW Verbeek, Boudewijn F van Dongen, Ekkart Kindler, and Christian W Günther. Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*, 9(1):87–111, 2010.

[VdASS11] Wil MP Van der Aalst, M Helen Schonenberg, and Minseok Song. Time prediction based on process mining. *Information Systems*, 36(2):450–475, 2011.

[vDCvdA08] Boudewijn F van Dongen, Ronald A Crooy, and Wil MP van der Aalst. Cycle time prediction: when will this case finally be finished? In *On the Move to Meaningful Internet Systems: OTM 2008*, pages 319–336. Springer, 2008.

[VDLR$^+$15] Ilya Verenich, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, and Chiara Di Francescomarino. Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring. 2015.

[Ver14] Lucas Vergeest. *Using N-grams and Word Embeddings for Twitter Hashtag Suggestion*. PhD thesis, 2014.

[WBL$^+$11] Juliano Araujo Wickboldt, Luís Armando Bianchin, Roben Castagna Lunardi, Lisandro Zambenedetti Granville, Luciano Paschoal Gaspary, and Claudio Bartolini. A framework for risk assessment based on analysis of historical information of workflow execution in it systems. *Computer Networks*, 55(13):2954–2975, 2011.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[XPK10] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48, 2010.

Non-exclusive licence to reproduce thesis and make thesis public

I, Henri Trees (date of birth: 10th of August 1990),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

Predictive Monitoring of Multi-level Processes

supervised by Anna Leontjeva and Marlon Dumas

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 19.05.2016