Software Requirements Specification

for

Event Log Sampling for Predictive Monitoring

Version 1.0

Prepared by Malek Alhelwany, Taekeun Jeong, Xiaoyan Jin

Chair of Process Mining and Data Science, RWTH

Created on: 01.11.2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Event Log Sampling for Predictive Monitoring (LSPM) is an instance selection procedure that allows sampling training process instances for prediction models. This method increases the training speed of the next activity prediction method significantly while maintaining reliable levels of prediction accuracy. In this project, this method will be implemented into a web application (web app) based Python implementation while the resulting accuracy is the same as the one from the original program.

1.2 Document Conventions

We will implement a web app with a front end part as an interface and a back end part that offers users LSPM.
The main functionality of this web app is basically sampling the event log for future analysis and training purposes.

1.3 Intended Audience and Reading Suggestions

This document is intended for the RWTH Process and Data Science Group (PADS) who developed and documented the original implementation and for people who are interested in sampling methods or in predictive monitoring, such as other researchers or students who are working on predictive process monitoring

1.4 Product Scope

The main purposes of this system are sampling event Log and allowing this functionality as a web app. Today's business analysts are using machine learning algorithms to predict future behaviour of processes because it is helpful to prepare or avoid risks from undesired results. However, an enormous event log takes a long time to train a model. For this reason, the reduced event log by this sampling method will reduce the time and workload for the training of models by accessing this web app.

1.5 References

Event Log Sampling for Predictive Monitoring M. Fani Sani et al.
https://www.researchgate.net/publication/355066922_Event_Log_Sampling_for_Predictive_Monitoring

The original implementation

https://svn.win.tue.nl/repos/prom/Packages/LogFiltering.

Other implementations that are referenced to develop this web app

https://github.com/MuhammadUsman05/Comparative-Process-Mining

2. Overall Description

2.1 Product Perspective

LSPM is a plugin built to be integrated in a bigger software or to work as a standalone as well. The pm4py software would be the main container to our tool. The whole system can analyze event log in order to discover models, check for conformance and compliance issues. pm4py also has many AI tools, like predictive monitoring for future events, which takes an event log as input, and produces new models, in order to monitor future activity and give hints to users about possible unwanted behavior or expected delay time in the process. Such event logs may be very big and could take too much time and space to train and produce the new models, and LSPM has the ability to simplify the event log to the minimum while maintaining good model accuracy. pm4py is built with the ability to be extended with various plugins and tools, and it requires the new tools to follow some criteria's to be integrated successfully.
LSPM is following the same criteria's of pm4py, it uses the same input file types (csv, xes) and produces (xes) files, which can be the input for the next step of training the new sampled event log in order to produce a predictive monitoring system.

2.2 Product Functions

LSPM takes an event log as an input and returns sampled event log as an output using three main sampling methods: unique selection, logarithmic distribution and division. The user can choose the method according to the characteristics of the event log, or use the default one (unique selection).

The major functions of LSPM is:

- Importing event log
- Setting sampling method
- Traversing event log
- Distribution Computation
- Sorting the cases of each variant
- Filtering event log using (unique selection, logarithmic distribution or division)
- Returning sample event logs
- Exporting sampled event log

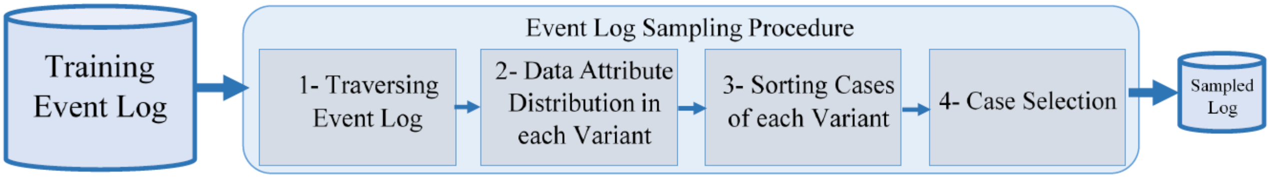

Training Event Log
Event Log Sampling Procedure
1- Traversing Event Log
2- Data Attribute Distribution in each Variant
3- Sorting Cases of each Variant
4- Case Selection
Sampled Log

Figure 1 A schematic view of the proposed sampling procedure

2.3 Behavior Models

2.3.1 Use case diagram

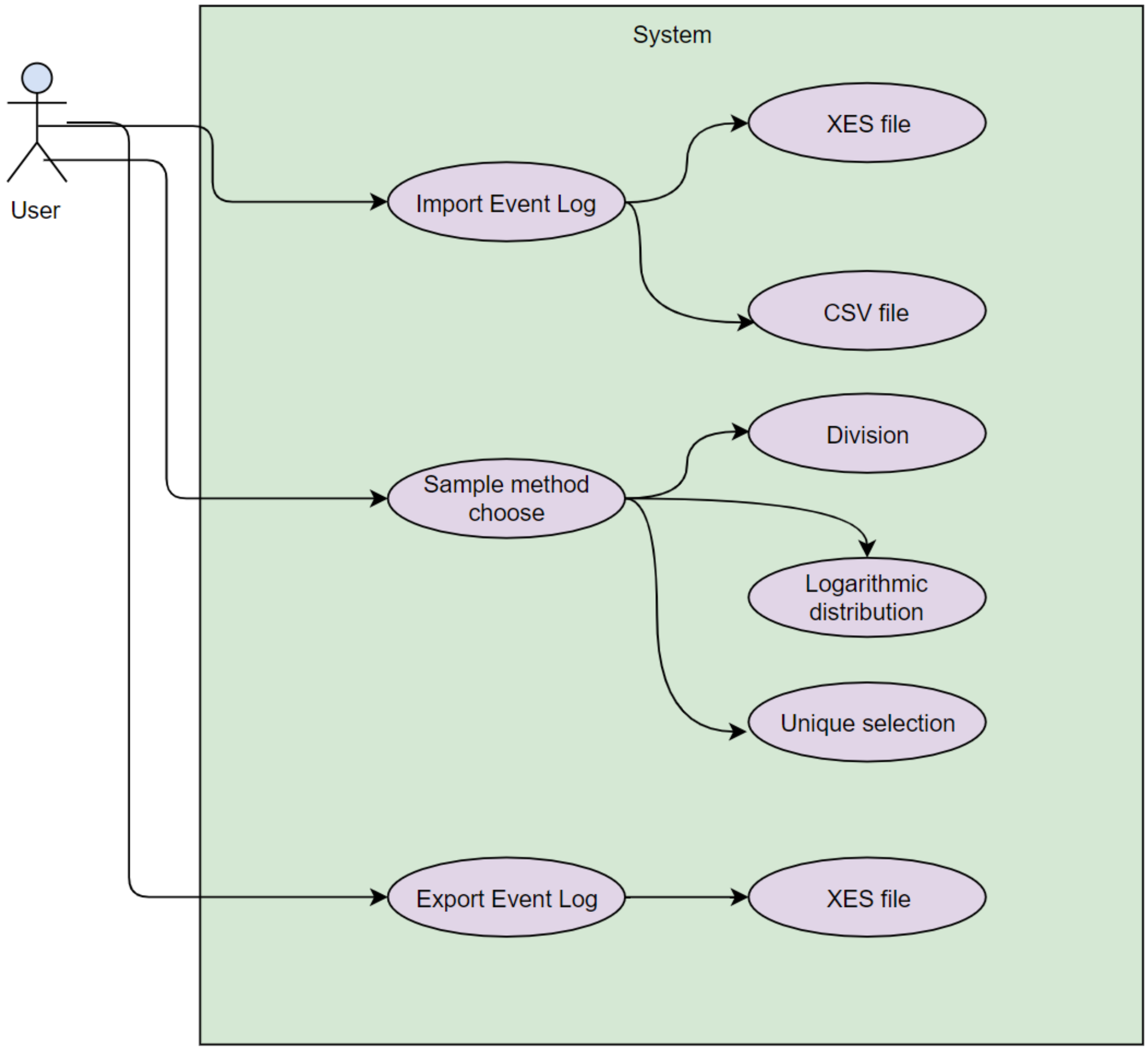


Figure 2 Use case diagram

2.3.2 Activity diagram

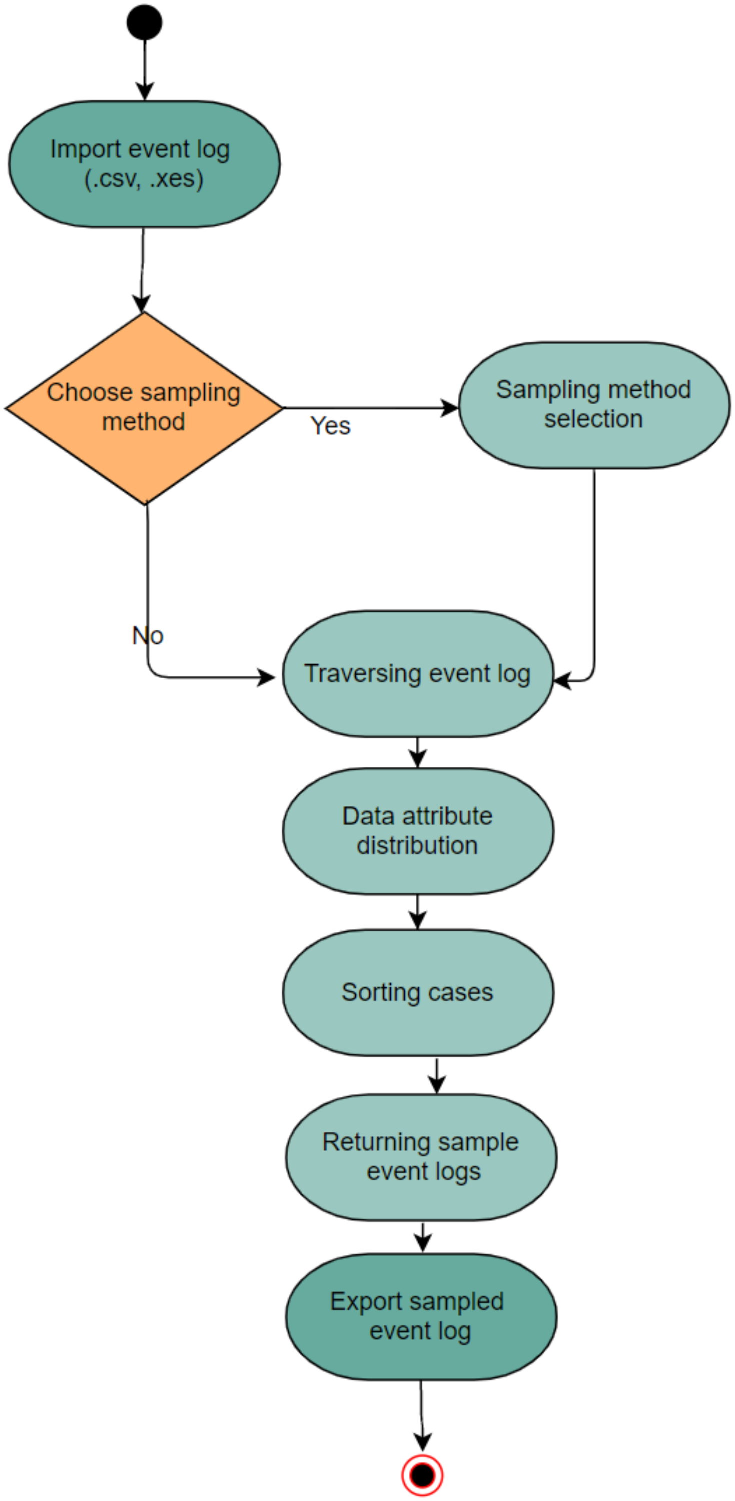

Import event log (.csv, .xes)
Choose sampling method
Yes
Sampling method selection
No
Traversing event log
Data attribute distribution
Sorting cases
Returning sample event logs
Export sampled event log

Figure 3 Activity diagram

2.4 Operating Environment

We will use many open source tools and technologies, according to efficiency and safety to ensure the best user experience and results. Tools and technologies including:

- pm4py package: a library built using python to serve process mining
- Django: web framework
- Visual studio code: coding
- Trello: project management tool
- GitHub: version control system
- Jupyter notebook: Python editor

2.5 Design and Implementation Constraints

Here is the list of the constraints of implementation and design:

- Limitation regarding input file type (xes, csv) only.
- Lack of a suitable event log for testing purposes
- The tool is built depending on pm4py, which is built using python, so we have to follow pm4py standards and developers should use python for compatibility to the library.
- For frontend design, we use Django, which runs on the web browser. This led us to use web development technologies for structuring and interactivity.
- Organizational limits, such as following Gantt chart, deadlines, and features described in the project initiation.

2.6 User Documentation

For ease of use, documentation files, as well as a user manual, including screenshots and instructions on how to use the tool, will be provided, and we will add hints on buttons, so the user can depend on himself when using the app.

2.7 Assumptions and Dependencies

The tool will be a web application, so it will be executable from any web browser which supports python and Django and the functionality of the app depends on the libraries and packages used in the source code:

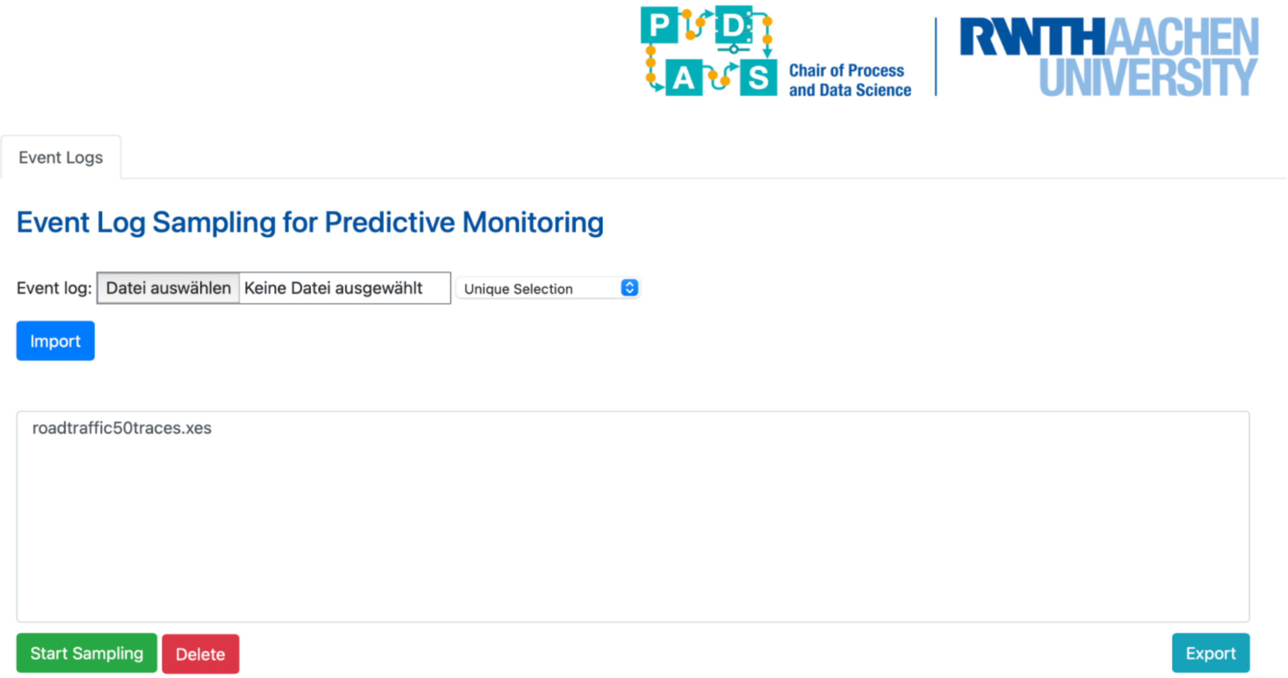
- Django 3.2.8
- pm4py 2.2.15
- Python 3.8

The tool will only accept (.xes / .csv) files, which should also have a valid syntax.

3. External Interface Requirements

3.1 User Interfaces

The purpose of this web app is clearly sampling the given event log, for this reason, this web app mainly consists of four buttons, two blank spaces for the file path and chosen file, and a dropdown menu. The four buttons are import(blue), start the sampling(green), delete the chosen file(red), and export the resulting XES file(turquoise). The import button enables the user to upload a CSV or XES file, and the blank space will show the file name. Before upload, other functions stay deactivated. The dropdown menu allows users to select methods for distribution , such as unique selection, logarithmic distribution, and division, of which default setting is a unique selection. Then, the user can start sampling or again import another event log file. If the user wants to start sampling, the user should click the green "start sampling" button. During the sampling, the system will show a progress bar with a statement such that the user knows the progression state.
If the sampling finishes, the progress bar will disappear, and the export button is activated such that the user can retrieve the sampled event log file.


P D A S Chair of Process and Data Science
RWTH AACHEN UNIVERSITY
Event Logs
Event Log Sampling for Predictive Monitoring
Import
Start Sampling
Delete
Export

Event log: Datei auswählen Keine Datei ausgewählt Unique Selection

roadtraffic50traces.xes

Figure 4 Frontend concept

3.2 Software Interfaces

This web app will utilize the library pm4py 2.2.15 to import, to transform, and to apply the LSPM.

This web app should be available on a local machine, in other words, the user should be able to download this file, launch the server, and execute LSPM. To facilitate this task, Docker will be used because it can generate a container that encompasses libraries, dependencies to run this including the source code file such that this web app is executable regardless of any operating system. Docker also simplifies the running of this web app by entering commands, thus, this is a useful choice for both user and developers.

4. System Features

We will discuss the system features from both user and system perspective as one block, because the user will only focus on the progress and result of the sampled file and will not look at the whole computation process.

4.1 Import Event Log

4.1.1 Description and Priority

Importing the event log is a high priority feature, where it’s the start point for our system. Importing event log including choosing the right event log form the (device/system) and upload the desired file to our systems database

4.1.2 Stimulus/Response Sequences

To perform Import event log, the user has to press the “Import” button, then a browsing window will be shown to the user, where he has to navigate to the right directory and choose the desired event log and to confirm the (import/upload) of that file.The user can as well delete or change the event log in case a wrong one have been selected.

4.1.3 Functional Requirements

REQ-1: User need to be able to import an event log
REQ-2: User can Delete/Modify the imported event log
REQ-2: User can navigate to the file location

4.2 Choose Sampling Method

4.2.1 Description and Priority

In general, we have three sampling methods: Division, logarithmic distribution and unique selection, the user has to decide which method to choose according to event log characteristics.

4.2.2 Stimulus/Response Sequences

To perform method selection, the user has to choose on the methods list and click on the desired method, then the system automatically changes the variable value to the selected sampling method.

4.2.3 Functional Requirements

REQ-1: User need to be able explore the available sampling method
REQ-2: User can click on the desired method to choose
REQ-2: System will change selection variable to selected method value

4.3 Compute and Sample the Event Log

4.3.1 Description and Priority

This stage consists of 4 steps: Traversing, distribution computation, sorting and filtering event log according to chosen method. It's the main activity and is also a high priority.

4.3.2 Stimulus/Response Sequences

To start sampling the event log, the user only has to click the start button and the system automatically takes the event log as an input and processes it to the tool and will do the necessary computation, while showing a progress bar to the user indicating the progress of the computation.

4.3.3 Functional Requirements

REQ-1: User need to be able click start button
REQ-2: An indicator bar will show the actual progress

4.4 Export Sampled Event Log

4.4.1 Description and Priority

After sampling the event log, user can export the sampled log and bypass it to the next step, which is training model

4.4.2 Stimulus/Response Sequences

To perform export, the user has to click on the “Export” button. Browser window will appear where the user can choose the desired location to save the sampled file to.

4.4.3 Functional Requirements

REQ-1: User need to be able click Export button
REQ-2: User can navigate to desired location

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The training process in machine learning algorithms suffer from the expensive computational costs and also its long training time limits the application of the techniques considering the limitations in time and hardware. Therefore, the LSPM devotes to present an increase of training speed at the same time with good prediction accuracy, and also less usage of the memory. It should purpose the selection procedure with a sampling training process for prediction model instantly while maintaining reliable levels of prediction accuracy.

5.2 Safety Requirements

In case of some possible error may occur to the procedure or event log due to any reason, the error notification will be sent to the client side, assisting both users and the developers identify the issues. Particularly, our tool should not modify the original event log due to any purpose.

5.3 Security Requirements

In LSPM we are using a web app for uploading datasets in order to continue with further steps, therefore our system should provide password protected access to web pages that are to be viewed only by the developers. Our tool should not store the uploaded event log or the generated prediction due to any reason, and also should not provide other users access to the data.
On the other hand, the user interface operations need to be well-defined to prevent our tool from malicious interaction with the system such as SQL injections.

5.4 Software Quality Attributes

Our tool should have an intuitive interface and be user-friendly, such as inputting the event log and choosing the suitable sample method. It's required to be flexible with the 2 common formats of the event data i.e. XES and CSV, and it's also significantly important to make sure the tool can perform well under any circumstances, especially when it comes to its maximum load.
Our tool should also perform the proper sampling training process for the user since different sample methods may lead to different results. Therefore, we should describe all the features and usages of the different sample methods on our tool to ensure that the users can choose the one that suits them most.

6. Other Requirements

Appendix A: Glossary

Key	Description
LSPM	Event Log Sampling for Predictive Monitoring : Instance selection procedure that allows sampling training process instances for prediction models, of which accuracy is similar to the result of the model using the whole event log.
Predictive Monitoring	Subfield of process mining that aims to estimate case or event features for running process instances such that stakeholders can, for example, predict undesired to prepare and minimize risks.
CSV	A Comma Separated Value(CSV) file is a text file that has a specific format which allows data to be saved in a table structured format. Every row contains values of different attributes(column), which are separated by comma.
XES	eXtensible Event Stream: standard is to standardize a language to transport, store, and exchange (possibly huge) event data (e.g., for process mining)
Web app	Web app is a computer program that utilizes web browsers to perform tasks over the Internet.
Activity / Event	An activity is an action that is performed by a resource in the scope of a process observed in an organization.
Log	A collection of a temporal-ordered sequence of activities. Generally assumed to be well-formed in a given standard for process mining, such as CSV and XES.