

User Manual

Master Thesis - Detecting Surprising Instances

Christian Kohlschmidt

06/09/2022

Table of Contents

1. Getting Started	2
a) Run the application inside a Docker Container.....	2
b) Development Setup:	3
2. How to import an event log	5
a) On the start page:	5
b) Error – Uploaded file type not supported:.....	5
3. Parameter Selection.....	6
Overview	6
1. Process Model.....	7
2. Parameter Selection.....	8
Similarity Graph.....	9
Classification	10
Clustering	11
3. Variant Information and filtering	12
4. Event Log Data	13
4. Surprising Instance Detection	14
5. Root-Cause analysis	16

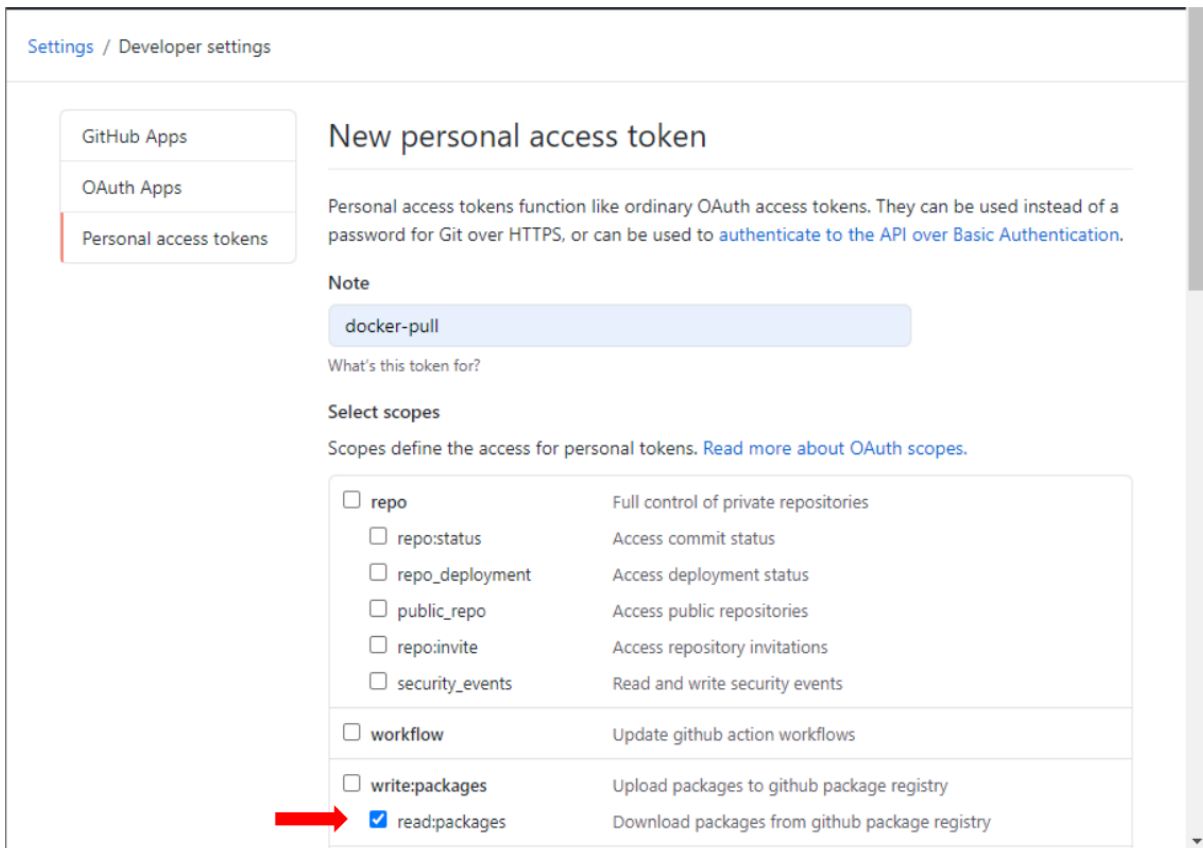
1. Getting Started

The first chapter will describe how to run the application on your machine:

a) Run the application inside a Docker Container

The Docker image for the application is available in the package registry of the [GitHub Repository](#).

To download the Docker image from the GitHub package registry, you first need to login. To login, generate a new personal access token in the GitHub [Developer Settings](#). The Scope needs to contain the “read:packages” attribute:



The screenshot shows the GitHub 'Developer settings' page for creating a new personal access token. On the left sidebar, 'Personal access tokens' is selected. The main heading is 'New personal access token'. Below it, a note states: 'Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.' A 'Note' box contains the text 'docker-pull'. Below this, a section titled 'Select scopes' explains that scopes define access for personal tokens. A list of scopes is shown with checkboxes: 'repo' (Full control of private repositories), 'repo:status' (Access commit status), 'repo_deployment' (Access deployment status), 'public_repo' (Access public repositories), 'repo:invite' (Access repository invitations), 'security_events' (Read and write security events), 'workflow' (Update github action workflows), 'write:packages' (Upload packages to github package registry), and 'read:packages' (Download packages from github package registry). A red arrow points to the 'read:packages' checkbox, which is checked.

Copy and paste the token in a Command Prompt as the password by using the “docker login” command:

```
docker login ghcr.io
```

```
C:\Users\user>docker login ghcr.io
Username: user
Password:
Login Succeeded
```

After that, use the „docker pull“ command in the Command prompt to download the image to your local machine (make sure to have Docker installed):

```
docker pull ghcr.io/ckohlschm/detecting-surprising-instances:1.1.0
```

```
C:\Users\user>docker pull ghcr.io/ckohlschm/detecting-surprising-instances:1.1.0
1.1.0: Pulling from ckohlschm/detecting-surprising-instances
1671565cc8df: Already exists
3e94d13e55e7: Already exists
fa9c7528c685: Already exists
53ad072f9cd1: Already exists
d6b983117533: Already exists
d8092d56ded5: Already exists
c71afc637d59: Already exists
864a10b3c704: Already exists
4334b2fe8293: Already exists
48b9ad049c69: Already exists
db2b7c0e8603: Already exists
66ca097097c1: Already exists
1bb19b96b2aa: Already exists
5ae2fc0e58f3: Already exists
d60868bc8357: Already exists
b53876d9bb85: Already exists
ae76fe42e955: Already exists
f1b7cfe4622a: Already exists
cf08f9360f21: Already exists
Digest: sha256:a0b4ea7ad300c554b7d0c717637a3df61d230d71b9d2dc843ea7eca195cbd78e
Status: Downloaded newer image for ghcr.io/ckohlschm/detecting-surprising-instances:1.1.0
ghcr.io/ckohlschm/detecting-surprising-instances:1.1.0
```

Now that the image is present on your local machine you can run the image with the “docker run” command. The frontend of the application inside the container is running on port 33333. If you want to access the application forward the port by providing the “-p” flag in the docker run command:

```
docker run -p 8080:33333 ghcr.io/ckohlschm/detecting-surprising-instances:1.1.0
```

Now you can access the application in your browser by using the URL: <http://localhost:8080/>

b) Development Setup:

To setup a development environment for the application clone the GitHub Repository:

```
git clone https://github.com/ckohlschm/detecting-surprising-instances.git
```

This will create the folder detecting-surprising-instances. Navigate into the folder and open a new Command Prompt. To install the required python modules either create a new virtual environment or directly install the modules by using the command:

```
py -m pip install -r requirements.txt
```

After installing the required python modules, you can now start the frontend of the application.

The frontend of the application is a Django app. Before running the application, you need to navigate to the frontend folder and migrate the application:

```
...\detecting-surprising-instances\frontend> py manage.py makemigrations  
...\detecting-surprising-instances\frontend> py manage.py migrate
```

Once the migration is complete you can run the development server with the command:

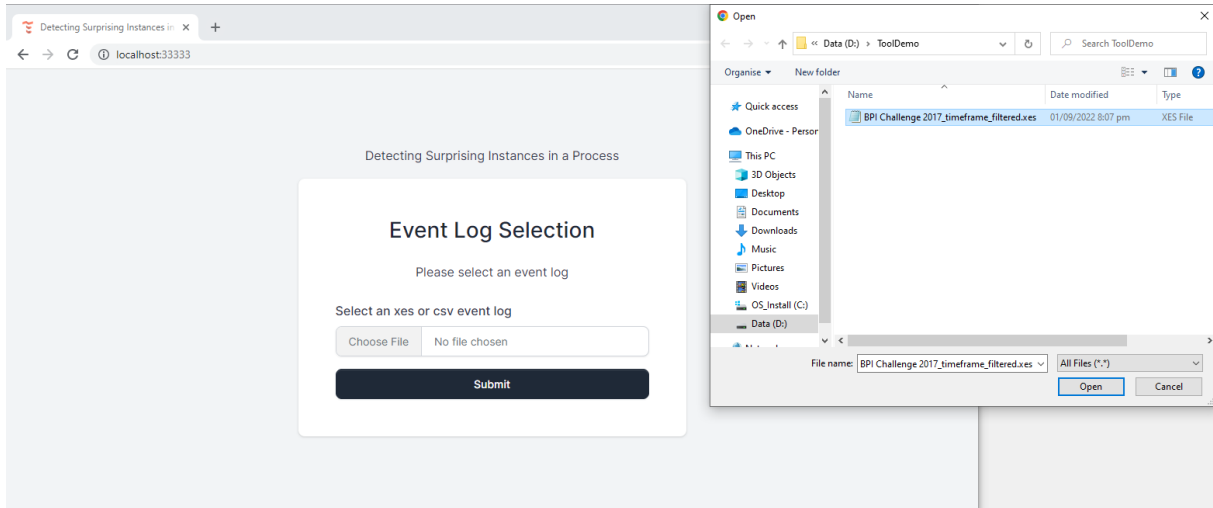
```
...\detecting-surprising-instances\frontend> py manage.py runserver
```

Now you can access the application in your browser by using the URL: <http://127.0.0.1:8000/>

2. How to import an event log

This chapter will show you how to import an event log from a XES or CSV file. This can be done in the first step on the starting page:

a) On the start page:



1 Click “Choose File”

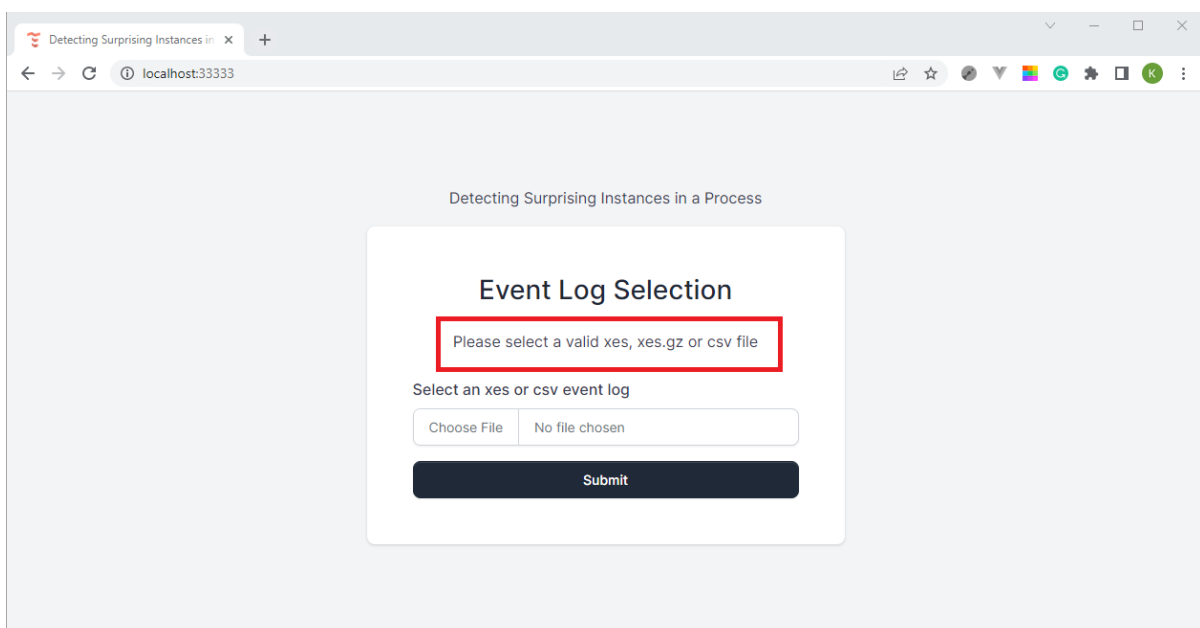
2 Select the file from the File Explorer

3 Click “Open”

4 Click “Submit”

b) Error – Uploaded file type not supported:

In case you try to upload a file that is not XES or CSV, you will be redirected to the start page and be asked to upload a valid file by receiving the error message “Please select a valid XES or CSV file”.

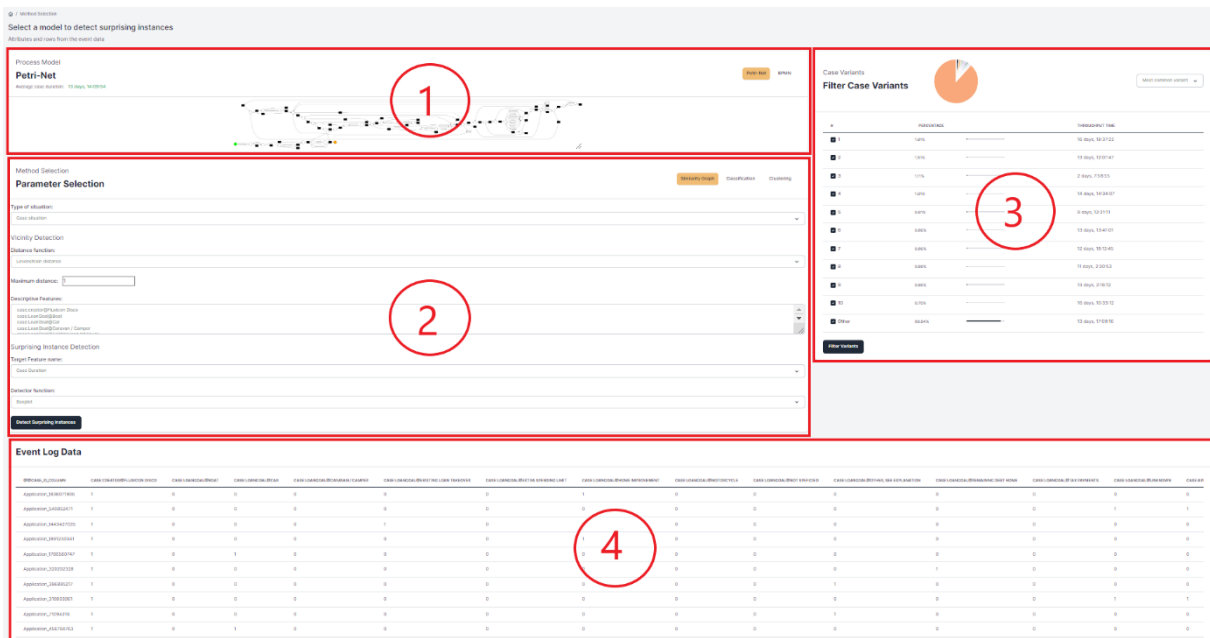


3. Parameter Selection

This chapter will show the methods and corresponding parameters that the application provides.

Overview

The parameter selection screen of the tool provides different information about the event log. Below, you can see a high-level overview about the different sections of the parameter selection screen:



The screenshot displays the 'Parameter Selection' interface. It is divided into four main sections, each highlighted with a red circle and a number:

- 1. Process Model:** Located at the top left, it shows a Petri-Net diagram representing the discovered process model for the event log.
- 2. Method Selection:** A large central form titled 'Parameter Selection' where users can configure various detection methods. It includes sections for 'Type of situation', 'Vicinity Detection' (with a 'Maximum distance' input), 'Outlying Instance Detection', and 'Surprising Instance Detection'.
- 3. Case Variants:** A table on the right titled 'Filter Case Variants' showing a list of cases with their IDs, names, and timestamps. A pie chart is visible above the table.
- 4. Event Log Data:** A large table at the bottom displaying the event log data in a tabular format, with each row representing a single case.

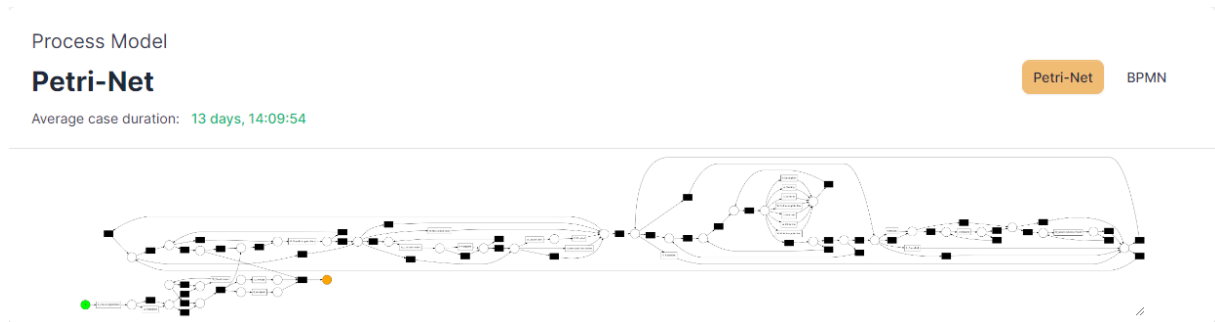
1. **Process Model:** The discovered process model for the event log
2. **Parameter Selection:** This section displays the parameters for the different methods. The tool supports three different types of vicinity detection methods that you can select in the top right corner of this section.
3. **Variant information and filtering:** Information about the variants in the process ordered by the criterion in the top right corner.
4. **Event Log Data:** Data extracted from the event log in tabular form. Each row represents one case.

Each of the sections and possible ways to interact with the tool are described in the following sections.

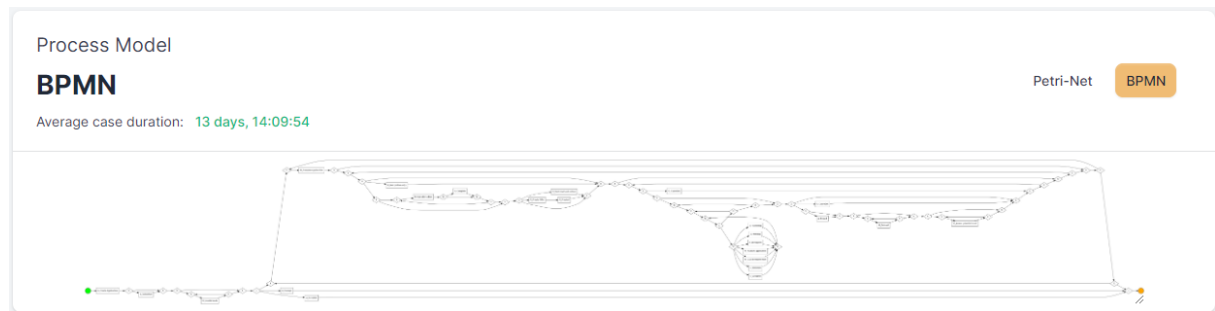
1. Process Model

The application automatically discovers a process model from the event log. You can choose between a Petri-Net or BPMN representation.

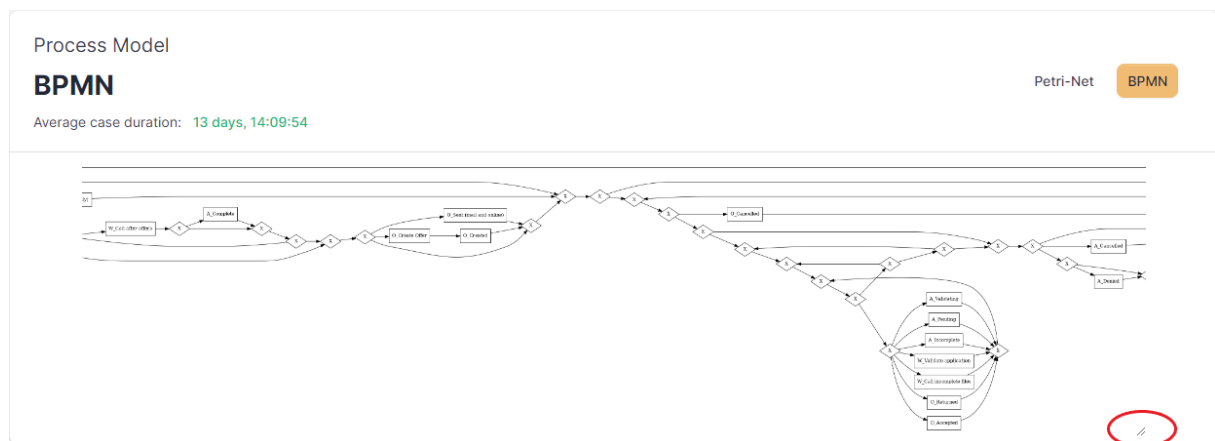
Click on “Petri-Net” to discover the Petri-Net representation:



Click on “BPMN” to discover the BPMN representation:



You can adjust the size of the window by scaling it on the bottom right corner. Use the mouse wheel to zoom in and out. Drag the image to show a specific part of the image.



2. Parameter Selection

You can select the parameters for the different methods in the Parameter selection window. The tool supports three different vicinity detection methods that you can choose in the top right corner.

Method Selection

Parameter Selection

Similarity Graph Classification Clustering

Type of situation:

Case situation

Vicinity Detection

Distance function:

Levenshtein distance

Maximum distance: 1

Descriptive Features:

case:creator@Fluxicon Disco
case:LoanGoal@Boat
case:LoanGoal@Car
case:LoanGoal@Caravan / Camper
case:LoanGoal@Fighting the television

Surprising Instance Detection

Target Feature name:

Case Duration

Detector function:

Boxplot

Detect Surprising Instances

1. **Similarity graph:** The similarity graph method creates a graph that represents the similarity of process situations. Similar situations are connected by an edge in the graph. This method requires a known distance function to identify the similarity of situations.
2. **Classification:** This method uses classification techniques such as a decision tree to identify the similar situations. Each leaf of the decision tree represents one set of similar situations.
3. **Clustering:** Cluster the situations into subsets of similar situations. Each cluster represents one set of similar situations.

Similarity Graph

The similarity graph method is the first of three methods. You need to specify the following parameters:

1. Type of situation: You can use case situations to get the whole case or a prefix of it by choosing the Event situation (Event situation also requires an activity)
2. Distance function: Select the distance function to use for the similarity graph. The application supports the Levenshtein Edit distance between the activity sequences or the Euclidean distance between data attributes.
3. Maximum Distance: Maximum distance for situations to be connected in the similarity graph
4. Descriptive features: List of descriptive features to use for the Euclidean distance and Root-Cause analysis. Click and hold the mouse to select multiple features. Ctrl-click to add a feature to the list.
5. Target feature name: Name of the target feature to predict
6. Detector function: Function to detect surprising instances. You can either choose boxplots or a specific threshold

Method Selection

Similarity Graph

Classification

Clustering

Parameter Selection

Type of situation:

Case situation

Vicinity Detection

Distance function:

Levenshtein distance

Maximum distance:

1

Descriptive Features:

event:concept:name@O_Returned
event:concept:name@O_Sent (mail and online)
event:concept:name@O_Sent (online only)
event:concept:name@W_Assess potential fraud
event:concept:name@W_Call after offers

Surprising Instance Detection

Target Feature name:

Case Duration

Detector function:

Boxplot

Detect Surprising Instances

Classification

The classification method uses machine learning models to identify sets of similar situations. You need to specify the following parameters:

1. Type of situation: You can use case situations to get the whole case or a prefix of it by choosing the Event situation (Event situation also requires an activity)
2. Vicinity Detection Method: Machine learning method to use
3. Decision Tree maximum depth: Maximum depth of the decision tree, used to avoid overfitting
4. Descriptive features: List of descriptive features to use for the Euclidean distance and Root-Cause analysis. Click and hold the mouse to select multiple features. Ctrl-click to add a feature to the list.
5. Target feature name: Name of the target feature to predict
6. Detector function: Function to detect surprising instances. You can either choose boxplots or a specific threshold

Method Selection

Similarity GraphClassificationClustering

Parameter Selection

Type of situation:

Case situation

Vicinity Detection

Vicinity Detection Method:

Decision Tree

Decision Tree maximum depth:

3

Descriptive Features:

case:LoanGoal@Boat
case:LoanGoal@Car
case:LoanGoal@Caravan / Camper
case:LoanGoal@Existing loan takeover
case:LoanGoal@Future prediction

Surprising Instance Detection

Target Feature name:

Case Duration

Detector function:

Threshold

Numerical Threshold:

11 day(s), 13:46:40

Detect Surprising Instances

Clustering

The clustering method uses clustering to identify sets of similar situations. You need to specify the following parameters:

1. Type of situation: You can use case situations to get the whole case or a prefix of it by choosing the Event situation (Event situation also requires an activity)
2. Vicinity Detection Method: Clustering method to use
3. K-Means number of clusters: Amount of clusters
4. Descriptive features: List of descriptive features to use for the clustering algorithm. Click and hold the mouse to select multiple features. Ctrl-click to add a feature to the list.
5. Target feature name: Name of the target feature to predict
6. Detector function: Function to detect surprising instances. You can either choose boxplots or a specific threshold

Method Selection

Similarity Graph

Classification

Clustering

Parameter Selection

Type of situation:

Case situation

Vicinity Detection

Vicinity Detection Method:

K-Means

Number of clusters:

3

Descriptive Features:

case:LoanGoal@Boat
case:LoanGoal@Car
case:LoanGoal@Caravan / Camper
case:LoanGoal@Existing loan takeover
case:LoanGoal@Extraneous items

Surprising Instance Detection

Target Feature name:

Case Duration

Detector function:

Boxplot

Submit Parameters

3. Variant Information and filtering

You can filter the process to only contain specific process variants. The tool shows the variants on the right side of the screen. The pie chart represents the frequency of the variants in the process. You can order them by

1. Most common variants
2. Least common variants
3. Longest Throughput Time
4. Shortest Throughput Time

Click on the checkboxes to select or deselect a variant. Confirm the selection by clicking the “Filter Variants” button.



4. Event Log Data

You can find the extracted data from the event log in a table on the bottom of the page. Each row represents a process instance. The columns represent the data attributes. Click on the buttons to navigate through the data. On the bottom right, you can find the total number of process instances.

Event Log Data							
@@CASE_ID@COLUMN	CASE@LOANGOAL@BOAT	CASE@LOANGOAL@CAR	CASE@LOANGOAL@CARAVAN / CAMPER	CASE@LOANGOAL@EXISTING LOAN TAKEOVER	CASE@LOANGOAL@EXTRA SPENDING LIMIT	CASE@LOANGOAL@HOME IMPROVEMENT	C.
Application_1836071906	0	0	0	0	0	1	0
Application_540852471	0	0	0	0	0	0	0
Application_1443427035	0	0	0	1	0	0	0
Application_1891230341	0	0	0	0	0	1	0
Application_1786560747	0	1	0	0	0	0	0
Application_320202328	0	0	0	0	0	0	0
Application_396885217	0	0	0	0	0	0	0
Application_319803961	0	0	0	0	0	0	0
Application_71094216	0	0	0	0	0	0	0
Application_456764763	0	1	0	0	0	0	0

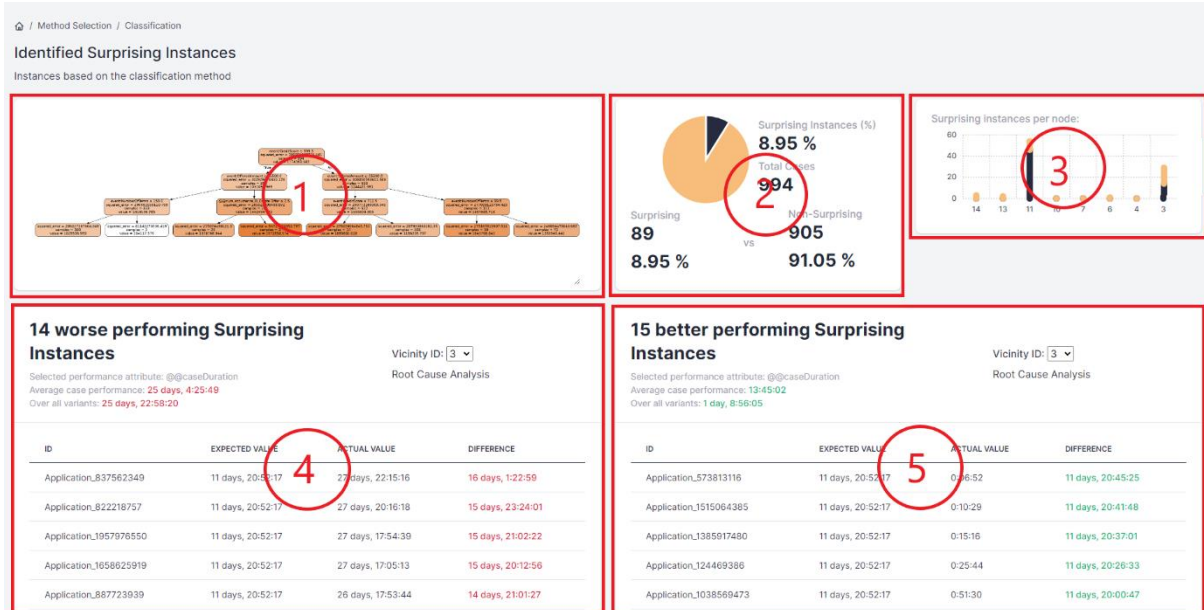
Previous12...9899Next

Showing 10 out of 994 entries

4. Surprising Instance Detection

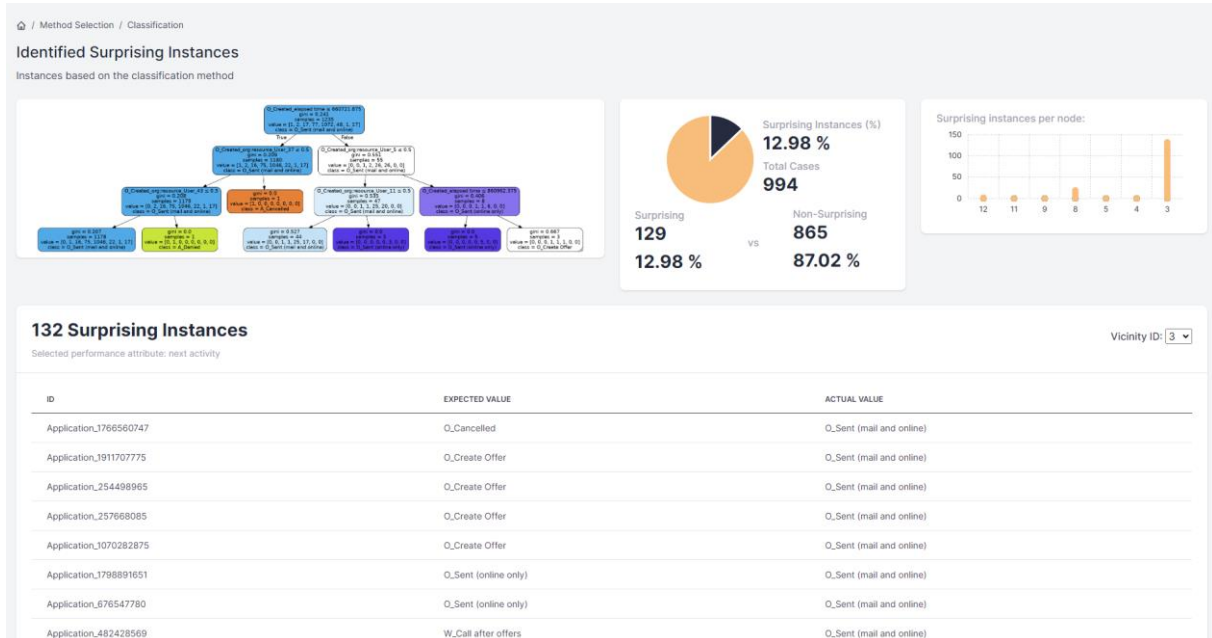
This chapter will show the results screen containing the detected surprising Instances.

The results screen shows information about the identified surprising situations in the event data. The data is grouped by the vicinity ID.



1. Decision Tree model that represents the data. You can resize the image in the bottom right corner. Zoom and drag to highlight specific parts of the image. (Note: This model is only available for the decision tree method)
2. Ratio of Surprising and Non-Surprising Instances. The number and percentage of surprising and non-surprising instances in the whole event log. The pie chart represents the fractions of surprising behavior.
3. The number of surprising instances in each vicinity. Each bar represents the number of surprising instances in a vicinity.
4. Worse performing instances. The number of worse performing instances for a specific vicinity. You can select the vicinity ID in the drop-down in the top-right corner. The instances are ordered by the difference. Press "Root Cause Analysis" to discover the causal effects of the variables in the selected vicinity
5. Better performing instances. The number of better performing instances for a specific vicinity. You can select the vicinity ID in the drop-down in the top-right corner. The instances are ordered by the difference. Press "Root Cause Analysis" to discover the causal effects of the variables in the selected vicinity.

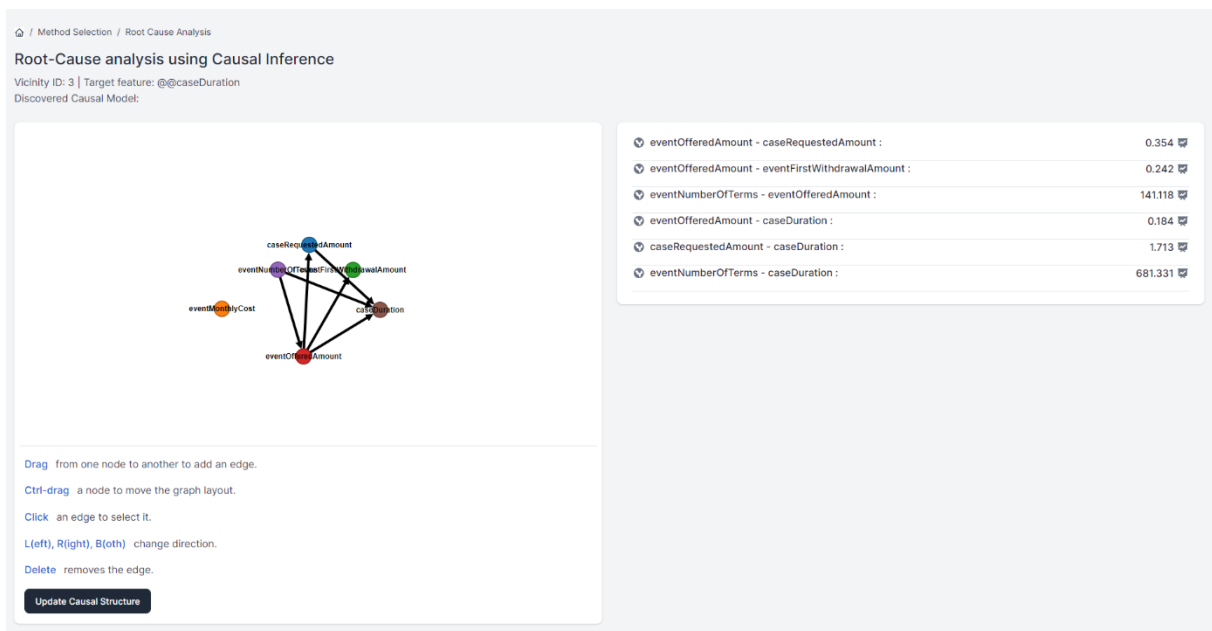
For categorical target features, the results screen shows only one list of surprising instances with the expected and actual attribute.



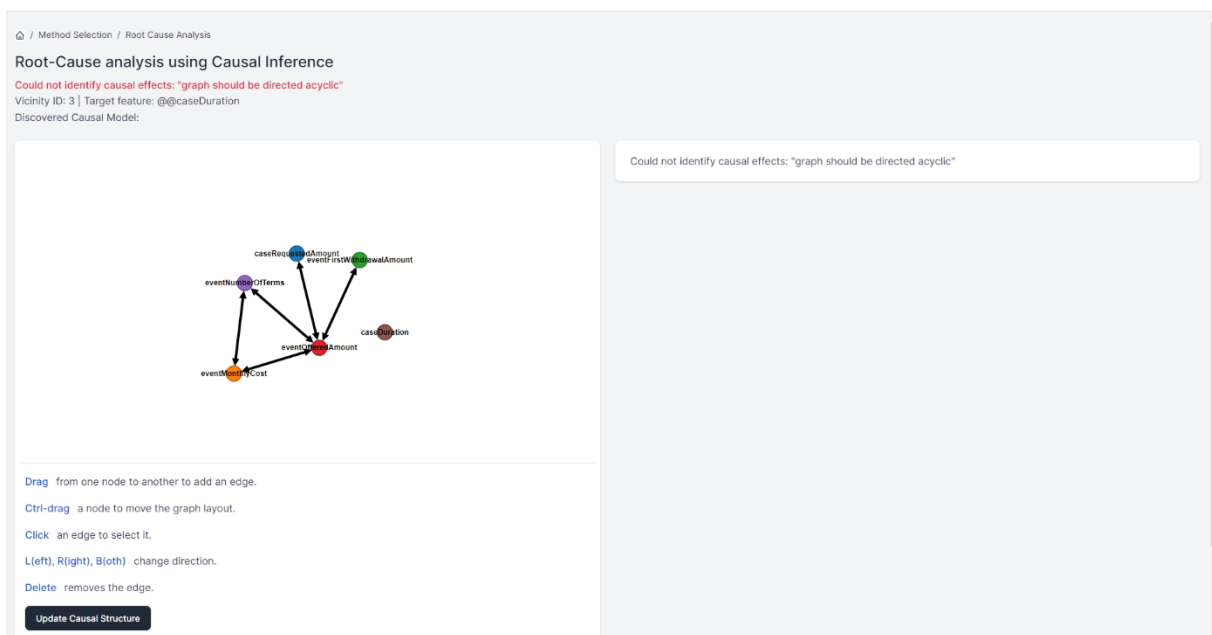
5. Root-Cause analysis

This chapter will show the implemented Root-Cause analysis techniques.

On the Root-Cause analysis page, you can find a causal structure on the left half of the screen and the estimated effects on the right side. The tool automatically discovers a causal structure based on the data in the specified vicinity. The Root-Cause analysis uses the attributes specified in the parameter selection screen.



If the discovered Causal Structure is not an acyclic directed graph, the tool displays an error.



You can edit the causal graph by:

1. Dragging from one node to another to create a directed edge
2. Ctrl-drag a node to move the graph layout
3. Click an edge to select it
 - a. Click L, R or B to change the direction (L=left, R=Right, B=Both directions)
 - b. Press delete to remove an edge