Specification draft

Goal

Adding support for anomaly detection to the IVIS framework. That includes default visualization and also anomaly detection.

SW requirements

Input

- Data requirements:
  - should be a valid signal set in IVIS.
  - signal set should countain the following to be valid for the anomaly detection framework of choice (Time-series Anomaly Detection Framework)
    - timestamp signal in `Date/Time` data type and with a constant frequency
    - one other signal representing the data with values valid in its data type and has enough values to make sense to detect anomalies (the dataset is large enough condition)
    - (further requirements for data might come up due to the anomaly detection framework)

Visualization

To add support for anomaly detection to IVIS framework two loosely connected components will be used:

- anomaly detection built-in task and IVIS jobs,
- anomaly report component.

Anomaly detection built-in task and IVIS jobs

IVIS already supports a pretty complex jobs and tasks framework - one part of it is a concept of built-in tasks.

The main piece of code will be such built-in task accessible via UI when creating a new job - that job/task will support the following parameters/options.

Parameters and options

It shall consist of at least two main parts - first for user input and second for vizualization on a subset of data from the selected valid signal set.

In the user input section there should be:

- a drop-down menu to select the signal set,
  - after that is done 2 more selectors appear to choose a signal representing the data and the timeseries,
    - if the selected signals are not big enough the user will be notified by a UI message,
- an option to have the frequency of the data derived or the user can input the value for it,
- a selector for the mode of the detection,
  - fully automatic - no further input needed,
  - automatic - user can input values into the fields described bellow,
  - iterative - user will be guided through an iterative process and will help decide what is or is not an anomaly on a series of examples.

The following fields are present for the automatic and iterative modes:

- a drop-down (checkbox) menu to select the seasonality with the following options:
  - `none`,
  - `automatic` - the framework will select the best one,
  - one or multiple options of the below - the framework will choose the best one of them:
    - `hourly`,
    - `daily`,
    - `weekly`,
    - `monthly`.
  - if a single value other than `none` or `automatic` is selected - a custom offset can be passed to specify when the season starts if it is not with the first data point in the signals,
- a tresholds field where if the option Custom is selected the user can choose the highest and the lowest values he doesn't deam to be anomalies.

User can then optionally select to preview the results of his choices and the visualisation section should appear. I thould consist of the following:

- a graph showing a subset of selected data and its state with respect to what has been selected
  - in the graph there should be lines representing the boundries,
  - the anomalies and non-anomalies should be visually different.

The last option for the user is to save the job so he can run it later.

Output

Once a run of the job is completed - new signals are created in the signal set where the anomalies were computed.

- boolean signal representing `isAnomaly` indicator.

- string note representing the `kind` of the anomaly
  - option to make it an enum of limited `kinds` will be explored
- int signal representing `score` in the anomaly detection framework sense
- two or four signals representing `boundaries` related to the calculation of the anomalies
  - both as values and scores

This effectively takes the pair `(time, value)` and turns it into a tuple of 8 values:

```
(
    time,
    value,
    isAnomaly,
    anomalyType,
    lowerBound,
    lowerBoundCcore,
    upperBound,
    upperBoundScore
)
```

which can be used by any visualization component of IVIS.

Anomaly report component

This component should be consumed mostly by the end users.

It will consist of multiple parts:

- scatterplot with the detected anomalies
- bar chart with the count of anomalies per (selected) interval of the timestamps,
- pie chart representing the count of the different types of anomalies,
- series related stats calculated in place

The scatterplot will show points from the original signal based on the `isAnomaly` signal.
The scatterplot will also show the boundaries in-between which the non-anomalies lie.
The scatterplot will use the existing tooltips present in IVIS - just expand on them with the score (s), delta (Δ) from the boundry and the kind (k) for anomalies.

The bar chart will show the number of anomalies in an interval selected by the default IVIS filter.

User stories

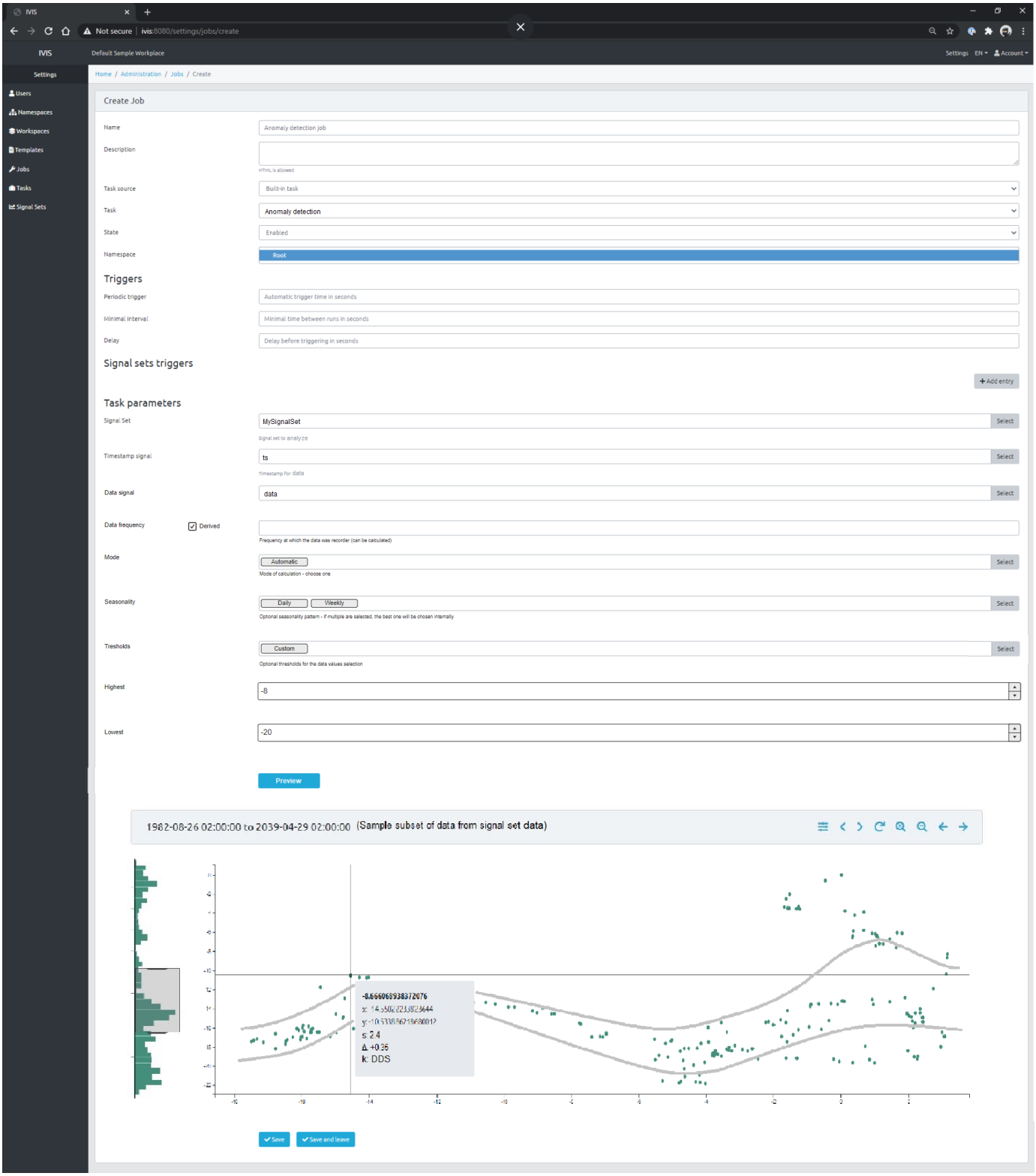
- As a user I can create a job with a built-in task type `anomaly detection`,
  - I can set it up as fully automatic, automatic or iterativ with an option to have a visible preview on a subset of the selected signalset/signals to see the detected anomalies
  - As a user I can choose to input seasonality.
  - As a user I can select lowest and highest boundry for anomalies.
  - As a user I can preview the results of the job and choose to save them.
- As a user I can create an anomaly report component (chart component) where
  - the anomalies of a data set are visible alongside their count per selected time interval. (count ~ separate linechart; anomalies ~ dots in the chart, avg ~ line and the limits as an area around it)

- types of anomalies are shown and their count displayed in a pie chart

Mockups

The following images represent the potential designs of the user interface that accomodates the features and functionalities discussed above. It is based on the existing UI features of the IVIS framework.

Job creation



Notes

- invalid data not shown
- custom offset for seasonality start not shown

Anomaly report