

# **Anomaly Detection in Industrial Networks**

PSE GRUPPE

KOMPETENZZENTRUM FÜR ANGEWANDTE SICHERHEITSTECHNOLOGIE

Advisor: M.Sc. Ankush Meshram

This Document outlines the requirements (both functional and non-functional), environment, target audience, and use cases of the software described below.

# Contents

<b>1</b>	<b>Purpose</b>	<b>1</b>
<b>2</b>	<b>Overall Description</b>	<b>1</b>
<b>3</b>	<b>Interfaces</b>	<b>1</b>
<b>4</b>	<b>Functional Requirements</b>	<b>1</b>
<b>5</b>	<b>Data Requirements</b>	<b>1</b>
<b>6</b>	<b>Non-Functional Requirements</b>	<b>1</b>
<b>7</b>	<b>Essential Testcases</b>	<b>2</b>
<b>8</b>	<b>Software Modelling</b>	<b>2</b>
8.0.1	GUI . . . . .	2
8.1	Scenario . . . . .	2
8.2	Use cases . . . . .	4
8.2.1	Interactivity . . . . .	4
8.3	Object Modelling . . . . .	5
8.4	Dynamic Modelling . . . . .	5
8.5	User Interface . . . . .	5
8.6	Glossary . . . . .	5

---

## 1 Purpose

The goal of this project is to create a software visualization tool for industrial network traffic to simplify the analysis of anomalous behaviour, both in realtime and from captured stored data.

This software is part of the ADIN framework and is referred to as the "ADIN Inspector". One component to achieve this goal is a web interface built with modularity in mind so as to make it easily extendable.

The Web view is able to display a series of diagrams and charts to easily identify the behaviour of the network. Within the Web view the user has the ability to zoom, select, highlight, and filter out data to better understand the aforementioned behaviour in different OSI layers, as well as visualize the flow rate between network nodes.

To support this Web view a back-end messaging solution is needed. This allows the user to easily switch between multiple streams of captured data.

## 2 Overall Description

## 3 Interfaces

## 4 Functional Requirements

## 5 Data Requirements

## 6 Non-Functional Requirements

- The web UI should be able to visualize the network topology and update the visualization based on real time data streams.
- The rendering latency should be no longer than 2 seconds.
- The framework should be able to handle data streamed from at least  $N$  sensors.
- The framework should be able to save data collected in a specified period of time.
- The web UI should be viewable on modern web browsers.

## 7 Essential Testcases

## 8 Software Modelling

### 8.0.1 GUI

The basic data structure needed for graphs are a given set of nodes and a given set of edges, as they are often drawn as node-link diagrams. In the postal data set, nodes could represent the origins and destinations of postal flows. Edges represent the flows between the respective origins and destinations. In intelligence analysis, investigators use semantic graphs to organise concepts and relationships as graph nodes and links in hopes of discovering key trends, patterns, and insights.”

A key issue in graph visualisation is the size of the graph, i.e. the size of the data to visualise. With a growing amount of data to display, graphs can become too complex and overburdening for the analyst’s cognitive capacity. It thus becomes difficult for the user to conduct significant analysis. Because of the issues described above, research often focuses on ways to solve the problems of visual clutter, e.g. by aggregation or clustering techniques, which is also one of the main topics of cartographic generalisation.

### 8.1 Scenario

- /S100/ : An operator wants to check manually/visually whether network nodes appeared or disappeared over the last day
  - the operator opens the web page
  - the operator selects the database as data source
  - the operator selects a timeline-based display
  - the operator selects node addresses as the data to be displayed
  - the operator moves to or selects the last 24 hours as the range of data to display
  - the operator closes the web page
- /S200/: A security analyst wants to look at the current flow rates between network nodes to see whether they change / there are trends
  - the analyst opens the web page
  - the analyst selects a source of live data
  - the analyst selects an appropriate visualization type
  - the analyst selects node addresses as the independent variable
  - the analyst selects flow rates as the data to be displayed
- /S300/: A security analyst wants to examine a specific point of data  
Precondition: the analyst has already selected the relevant dataset and visualization type
  - the analyst selects a data point by right-clicking

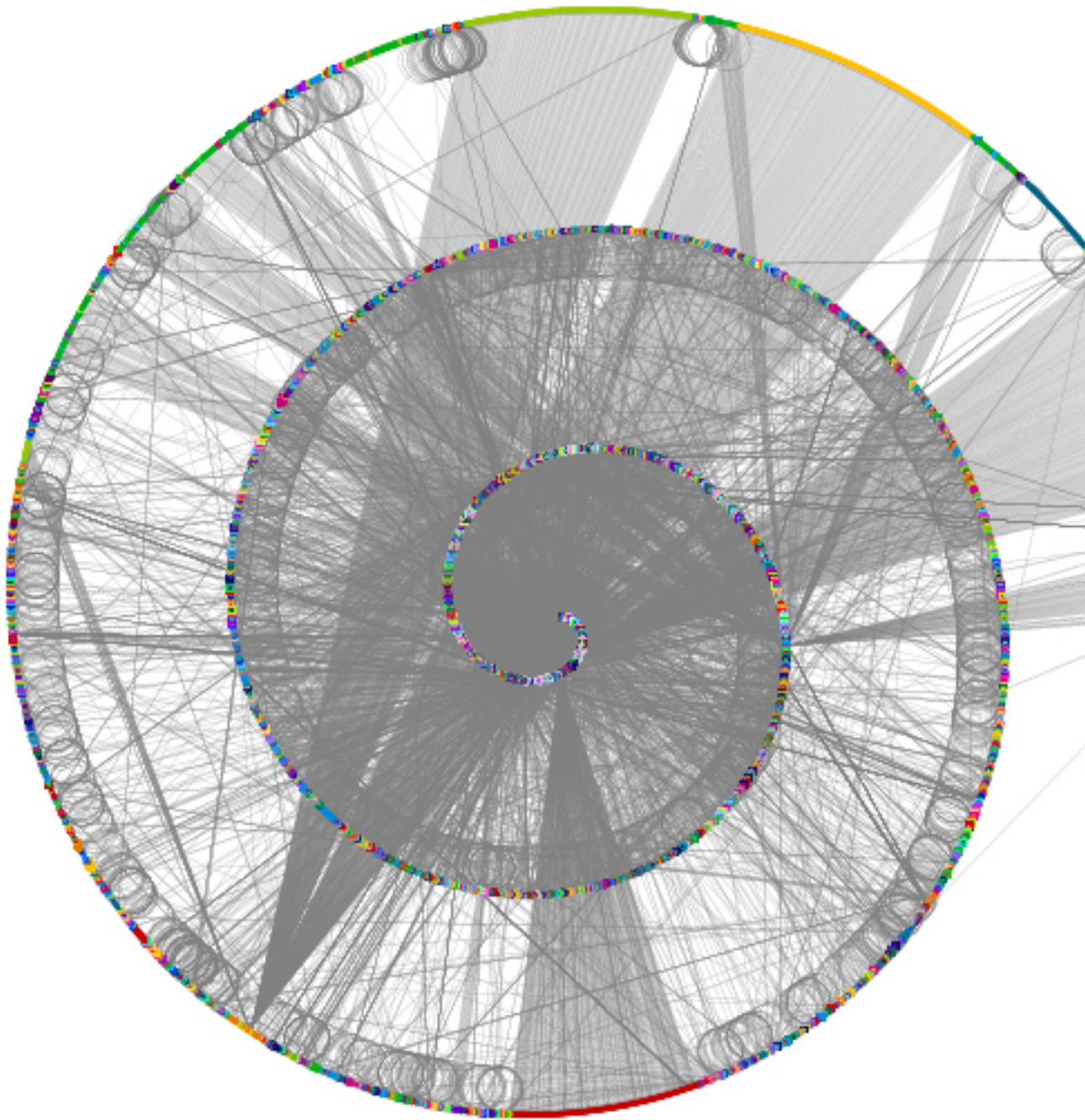


Figure 1: Node-Link-Graph

- the GUI displays a small pop-up window with all the data of this data point
  - the analyst right-clicks one of the attributes in the pop-up window and selects "Display all corresponding types"
  - the GUI marks all data points that have the same value in this attribute
- /S400/: The user wants to look at alarms/notifications
  - the user opens the web page
  - the user selects the database as data source
  - the user selects the data stream from the relevant dissector
  - the GUI displays the notifications along a timeline, according to the time of the event
  - the user right-clicks on the x-axis and selects "use record number"
  - the GUI displays the notifications along a timeline adjacently
- /S500/: The user wants to look at normal data together with alarms/notifications  
Precondition: Scenario /S100/ apart from closing the web page
  - the user selects menu "data", entry "sources"
  - the GUI displays a list of all known data sources with a checkbox in front of each
  - the user selects the checkboxes for the data sources they want to examine
  - the GUI displays data from all these data sources within the currently active visualization

## 8.2 Use cases

### 8.2.1 Interactivity

Visual analytics methods combine interactive visualisations with automated analysis techniques. This allows the user to decide e.g. which part of the data he or she wants to explore in more detail.

A basic principle for visual data exploration was introduced by Shneiderman (1997) by what he called the "The Visual Information Seeking Mantra:

Overview first, zoom and filter, then details-on-demand". This lets the data analyst define to a certain level what he or she wants to see and visualise.

Similar to this, Bertin (1983) specified three "levels of reading,: The elementary level (allowing the analyst to look at the information about a single data record), the intermediate level (showing summarised information about a group of data records), and the global level (providing an overview of all data elements).

**8.3 Object Modelling**

**8.4 Dynamic Modelling**

**8.5 User Interface**

**8.6 Glossary**