The DiSIEM project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 700692.
**Editor**
Michael Kamp, Fraunhofer
Cagatay Turkay, City

**Contributors**
Michael Kamp, Fraunhofer
Siming Chen, Fraunhofer
Linara Adilova, Fraunhofer
Phong Nguyen, City
Cagatay Turkay, City
Livin Natious, Fraunhofer

**Version History**

<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Michael Kamp</td>
<td>First version of the document for internal review</td>
</tr>
<tr>
<td>1.1</td>
<td>Michael Kamp</td>
<td>Revised after 1st review by ATOS</td>
</tr>
<tr>
<td>1.2</td>
<td>Cagatay Turkay</td>
<td>Merging and revision with FCIências.ID review</td>
</tr>
<tr>
<td>2.1</td>
<td>Cagatay Turkay</td>
<td>Document released to the coordinator for quality check and submission</td>
</tr>
<tr>
<td>2.2</td>
<td>FCIências.ID</td>
<td>Final minor improvements and submission to EC</td>
</tr>
</tbody>
</table>
Executive Summary

This deliverable provides the details and usage guidance for the prototypes developed to enable the interactive visual analysis and modelling using the different data sources considered within DiSIEM. The prototypes developed in WP5 showcase that visual analytics and machine learning facilitate a deeper understanding of SIEM data, provide novel means to help operators with time-consuming decision-making situations, and also support system developers to build effective models that can capture the complex, multi-faceted nature of a system’s status in a holistic manner. The prototypes documented in this report have been designed and built using a user-centred process by giving utmost priority to the needs of the users of the SIEM systems in the consortium. Several of the solutions we developed have also been evaluated by different stakeholders in the SIEM user partners in respective testing environment, and initial observations indicate promising potential for the utilisation of visual analytics as a key element in future SIEMs.

The prototypes presented in this deliverable offer powerful tools to analyse and understand SIEM data in depth, as well as to generate models to automatically detect potential attacks. The user behaviour analysis platform offers a range of integrated tools that allow to understand user behaviour, to investigate similarities in behaviour, and to predict behaviour with the goal to identify potential attackers. The diversity and forecasting analytics dashboard allow to analyse the employed SIEM system and its alerts in order to maximize the detection rate and minimize false alarms, and to evaluate and improve underlying statistical models of system status. All this functionality is provided through five distinct demonstrators where four of them are presented conceptually integrated under a single component to better reflect the consolidated design and development activity taking place within this work-package.

As described in this deliverable, the prototypes are developed in a modular, extendible way to ensure a tight integration with the SIEM systems, and we designed the components in a fashion that is easy to deploy and use. Despite being prototypes, all demonstrators are already in a usable state and gone through user testing with the respective application partners in a testing environment. This facilitates the subsequent evaluation of the components by the project partners and allows to focus on further improvements to the prototypes.
Table of Contents

1 Introduction .......................................................................................................................... 8
  1.1 Organization of the Document ....................................................................................... 8

2 User Behaviour Analysis Platform (UBAP) .................................................................... 9
  2.1 High-level Analytical Tasks .......................................................................................... 10
  2.2 Sub-components of UBAP ............................................................................................. 10
  2.3 Integrated Workflow for the Sub-components ............................................................... 11

3 UBAP-1: Topic modelling-based User Behaviour Analysis ............................................ 12
  3.1 Component description ................................................................................................. 12
  3.2 Demonstrator description ............................................................................................ 12
  3.2.1 LDA Modelling ....................................................................................................... 13
  3.2.2 Topic Projection View ............................................................................................ 14
  3.2.3 Topic-Action Matrix View .................................................................................... 14
  3.2.4 Venn Diagram ....................................................................................................... 15
  3.3 How to use the Demonstrator ..................................................................................... 15
  3.3.1 Interactive definition of topic groups ...................................................................... 16
  3.3.2 Editing groups and updating medoids ................................................................. 17
  3.3.3 Removing a group and its medoid ......................................................................... 17
  3.3.4 Multi-level exploration and Progressive categorization of behaviours .................... 17
  3.4 Further improvements ................................................................................................. 18

4 UBAP-2: Cluster-based Behavioural Model Building ..................................................... 19
  4.1 Component description ................................................................................................. 19
  4.2 Demonstrator description ............................................................................................ 20
  4.3 How to use the Demonstrator ..................................................................................... 22
  4.4 Further improvements ................................................................................................. 24

5 UBAP-3: U4 - Investigative Visual Analysis of Session Collections .............................. 26
  5.1 Component description ................................................................................................. 26
  5.1.1 The Problem .......................................................................................................... 26
  5.1.2 Our Proposed Solution ............................................................................................ 27
  5.2 Demonstrator description ............................................................................................ 27
  5.3 How to use the Demonstrator ..................................................................................... 28
  5.3.1 Goal 1 - Overall Understanding of Sessions ....................................................... 29
  5.3.2 Goal 2 - In-Depth Analysis of Unusual Sessions ................................................ 30
  5.4 Further improvements ................................................................................................. 33

6 UBAP-4: VASABI - Visual Analytics for Understanding User Behaviour ...................... 34
  6.1 Component description ................................................................................................. 34
  6.2 Demonstrator description ............................................................................................ 34
  6.3 How to use the Demonstrator ..................................................................................... 35
  6.3.1 Session Overview .................................................................................................. 36
  6.3.2 User Overview ....................................................................................................... 36
  6.3.3 User Profiles .......................................................................................................... 37
  6.3.4 Session Timeline .................................................................................................... 37
  6.3.5 Task Overview ....................................................................................................... 38
  6.4 Further improvements ................................................................................................. 38

7 Diversity and Forecasting Analytics Dashboard ............................................................. 39
  7.1 Component description ................................................................................................. 39
  7.2 Demonstrator description ............................................................................................ 40
7.3 How to use the Demonstrator ............................................................... 40
  7.3.1 Events View ...................................................................................... 41
  7.3.2 Modelling View ................................................................................ 41
7.4 Further improvements ............................................................................ 42
8 Summary and Conclusions ...................................................................... 43
References .................................................................................................. 44
List of Figures

Figure 1 Illustration of the interaction of all four components (with U4 and VASABI as separate sub-components) of the user analysis platform with a common data storage provided by the application partners................................. 9

Figure 2 Visual analytics system overview, including LDA modeling part (backend), Topic distribution view (a), topic-action matrix view (b) and Venn diagram view (c). The action class is shown in (d).................................12

Figure 3 An illustration for behavior analysis with LDA topic modeling..............13

Figure 4 Workflow for demonstrating how to use the visual analytics tool...........16

Figure 5 Illustrative user operation case. (a) Initial selection by brushing, (c) Updated results in topic-action matrix with shown patterns, (d) users can further drill down to details. And then (b) get the updated clusters of behaviour medoids.............................................................................................................17

Figure 6: Illustration of the interaction of session annotator and model training component with the common data storage.................................................................20

Figure 7. Different sources of signals and data utilised in the detection of anomalous behaviour within the SKEPTIC framework by Amadeus. ...............26

Figure 8. Linked visualisations in U4. The Session View (A) helps explore relationships between session attributes. Sessions in the selected office are displayed in the Activity View (B) and the Timeline View (C) for further investigation. Most common actions in the selected sessions are colour-coded according to their types (see the legend in D) or their clustering results.........................................................................................................................28

Figure 9. Workflow of U4. .................................................................................................28

Figure 10. Session View. Sessions are displayed as small rectangles with colour lightness showing anomaly score, height showing session length, and sessions are sorted decreasingly by length. Sessions are grouped by user. Mapping configuration can be changed using the options menu in the top-right corner of the view........................................................................................................29

Figure 11. Activity View. The view displays common activities (as sequences of actions) that are mined from our constraint-based pattern mining algorithm. Actions are colour-coded based on their types or their groups. The bars on the left show summary statistics of sessions having the activities.................................................................................................................................30

Figure 12. Different representations of a session. A black horizontal line in an action or activity indicates the size of an aggregate.........................................................31

Figure 13. Comparative analysis between each session and its past sessions performed by the same user. This shows the same sessions as in Figure 8C. Dark red rectangles indicate highly unexpected activities...........................................32

Figure 14. Linked visualisations in VASABI. The Session Overview (A) provides a temporal distribution of sessions and acts as a filter. The User Overview (B) provides a multi-perspective view on user attributes. The User Profiles (C) enables examination of user behaviour through multiple features concurrently. The Timeline View (D) allows detailed investigation of selected sessions. The Task Overview (E) summarises dominant tasks of selected sessions ..........................................................................................................................35

Figure 15. Workflow of VASABI.........................................................................................35
Figure 16. Session Overview. This histogram shows a temporal distribution of sessions.

Figure 17. User Overview. This view shows a summary of users through their data attributes. Each user is a bar and grouped by their organisation.

Figure 18. Each row is a visual profile for a user, consisting of visual summary of multiple facets.

Figure 19. Investigation with Profile view. Sessions of interest are shown as orange dots, allowing comparison with typical behavioural patterns drawn from the entire dataset, shown as histograms/barcharts. Red dots are sessions selected for further investigation (by clicking on a dot or a bar) in the Timeline view.

Figure 20. Task Overview. Each task is shown as a set of coloured squares, each representing a dominant action in the task. Actions are colour coded based on their group.

Figure 21. Linked visualisations. The Events view (left) provides a temporal distribution of events and acts as a filter. The Modelling view (right) visualises the model output and the prediction for selected events.

Figure 22. Events View shows a histogram of events, binned by time.

Figure 23. Modelling view shows the output of different models.

Figure 24. Modelling view updated with selected events.
1 Introduction

This report contains a description of the demonstrators developed in WP5. The first component, the user behaviour analysis platform, consists of multiple integrated and interacting sub-components that are each described separately, and each provided as distinct demonstrators. The goal of the platform is to analyse, understand, and predict the behaviour of users. The second component, the diversity and forecasting analysis dashboard, investigates the alerts of a SIEM system in order to maximize the detection rate of fraudulent behaviour, while at the same time minimizing false alarms and to help support the evaluation of statistical models of alerts as built in other work packages. In the following, the potential interaction methods and how the prototypes can be used are described in detail. The tools are available both as code collections through a web-page and as standalone web-based applications with links to each specific component provided in the descriptions.

1.1 Organization of the Document

Chapter 2 describes the user behaviour platform (UBAP), briefly introducing the design principles and motivation behind each sub-component's role and detailing how the components interact. UBAP comprise of four sub-components and each of these sub-components can currently be executed independent from the others and is thus described similar to an independent demonstrator in the document. Chapters 3 to 6 then introduce each sub-component in detail with:

- Chapter 3 describing the topic modelling-based user behaviour analysis,
- Chapter 4 describing the cluster-based user behaviour model building, and
- Chapter 5 and 6 describing two visual analytics solutions to analyse particular user behaviour and sessions, denoted U4 and Vasabi.

Chapter 7 then describes the second component, the diversity and forecasting analysis dashboard. The document concludes with a summary and concluding remarks in Chapter 8.

To summarise, the document provides details on five distinct demonstrators where four of them provided as sub-components of UBAP (Chapter 3, 4, 5 & 6 where each sub-component denoted with UBAP- to make the relationship to Chapter 2 clear) and the final demonstrator called diversity and forecasting analysis dashboard (Chapter 7)).
2 User Behaviour Analysis Platform (UBAP)

The user behaviour analysis platform is a combination of integrated tools that analyse individual aspects of user behaviour and combine the results to provide insights to SIEM operators, as well as automatically generated scores that indicate potential attacks (an overall illustration in Figure 1).

![Image](image.png)

Figure 1 Illustration of the interaction of all four components (with U4 and VASABI as separate sub-components) of the user analysis platform with a common data storage provided by the application partners.

The main intuition behind user behaviour analysis is that individual sessions tracked during user’s interaction within the application partner’s system can provide insights about possible attacks or malicious actions performed. Nevertheless, direct screening of all the sessions is typically impractical due to the high amount of activity happening within a given point in time. Moreover, trying to model all the possible diverse behaviour of users can easily be highly imprecise and not capable of identifying behaviour different from normal. These points motivate uniting together for analysing not only sessions of one and the same user, but all the users that have similar behaviour throughout all their interactions with the system. This allows to build a better-informed understanding of standard user activities and be more precise when identifying suspicious behaviour.

Overall, modelling user behaviour in order to automatically recognize suspicious activity is a challenging task. This is primarily because malicious behaviour is, by design, always aimed to be masked to be as close to standard behaviour as possible. Moreover, there is often high diversity within the actions of the users in a system – further complicating the precise modelling of common behaviour. Motivated by these challenges, modelling on clusters of users and/or user behaviour comes forward as a method that is in principle more beneficial and robust since such an approach would be less susceptible to the noise in the data. Despite the potential benefits of such a cluster-based approach, the manual investigation by operators on the identified inconsistencies in behaviour is still
desired, and often inevitable whilst making the eventual decisions on the anomalous sessions. Considering these factors, a visual analytics platform, where the involvement of human experts is supported along with robust computational modelling, stands out as an effective solution.

2.1 High-level Analytical Tasks

The platform considers three major tasks that are contributing to understanding users’ behaviour:

1) The analysis of similarities in users’ behaviour in order to learn patterns of activity;
2) The modelling of regular user behaviour based on the groups of common behavioural patterns;
3) The analysis of individual user behaviour from recorded activity as sessions of interaction with client’s system.

Each task is considered as a main problem by a specific sub-component of the platform, thus the three components together aim to analyse users’ behaviour through tight integration to provide a comprehensive solution. The following section introduces these SCs at a high level before they are being introduced in the following chapters of this document.

2.2 Sub-components of UBAP

The first sub-component – the *topic modelling-based user behaviour analysis tool (UBAP-1)* – is motivated by the first task above and responsible for visualizing the entirety of users’ behaviours and allowing to explore similarities between all the individual user’s sessions and to identify different behavioural patterns. The tool allows analysts to interact with automatically identified clusters based on the frequency of various actions that are performed by users. This interaction should be performed by specialists who are capable of adjusting the results of the clustering algorithm in order to better capture the similar behaving groups of users.

The integration of detected users’ clusters for considering them in the overall modelling is done via the second sub-component. This *cluster-based user modelling component (UBAP-2)* builds a model for approximating the behaviour of an average user in one specific cluster. This is performed for each of the clusters identified by the topic modelling component. After training on each of these clusters specifically, this model provides predictions on the likelihood of particular behavioural aspects. The resulting models are capable of pointing out the indications of attacks in case of large divergence from expectations for a particular cluster.

The final task (investigating users and their sessions closely through the involvement of an expert analyst) is addressed by the eventual user-behaviour focussed visual analytics sub-components named U4 (UBAP-3) and VASABI (UBAP-4). The output of the models from the other components (described
above) are also integrated into these tools and communicated to operators along with any other relevant data to support them in analysing potential malicious behaviour. The visual analytics tools U4 and VASABI allow to perform an in-depth investigation of a single user’s behaviour and to compare it to the behaviour of other users. We envision these data-driven interfaces as the operator-facing solutions where we surface the modelling work performed in the other components and where we provide interactive and accessible representations of individual sessions and users.

2.3 Integrated Workflow for the Sub-components

Each of the components of the platform can be used separately and provide useful insights about users’ activity during their interaction with monitored applications. So, the visual analytics tool displays various aspects of the current sessions, the topic modelling component helps to identify groups of users by the commonalities in their sessions, the user-modelling component can also build models for describing user’s behaviour in general. But the full potential of the platform is achieved through the interaction between them.

To achieve a productive exchange between the four sub-components, individual tools are integrated to work together on a common data storage (see Figure 1). This data storage is updated in real time and filled with logs of users’ sessions. The common storage serves as an exchange platform for the components (details are provided in the following sections) to read and write information into. At its core, our integration approach is designed to support an iterative flow of information within all the components. For instance, any knowledge obtained through the analysis of individual user behaviour in U4&VASABI can be utilised to guide the analysis of the similarities within different users’ sessions in the topic-modelling component to identify user clusters. After these clusters are stored in the data store, this information is available for all the components and can be used to split users into common clusters characterized by similar behaviour. To exemplify, the modelling component can utilise the clusters to generate more precise and powerful models compared to models that do not take user clusters into account. Eventually, the output of these models is produced in real time together with the recorded activity and stored in the same data storage which in turn are considered by the visual analytics tools U4 and VASABI – the visual representation includes these modelling prediction results to highlight individual sessions that should be investigated more thoroughly. It is, in essence, this idea of iterative analysis and modelling that fuels our user behaviour analytics approach. More information about the techniques used in these components can be found in deliverable D5.1 [D51].

In the following chapters, each sub-component is described in detail. To denote the links with the User Behaviour Analysis Platform clearer, we denote each of these sub-components with the acronym UBAP-. Also note that for each the four subcomponents, we are providing an individual demonstrator that currently can work in isolation but designed to work as a single component as part of the final phase of the integration.
3 UBAP-1: Topic modelling-based User Behaviour Analysis

3.1 Component description

We developed a visual analytics system with topic modelling ensembles for user behaviour analysis. The system enables analysts to interactively understand user behaviour clusters based on ensembles of topic modelling techniques. Taking the advantage of LDA (Latent Dirichlet Allocation) process in text mining, we map the sessions as documents and each action in a session as a word. Thus, we can generate the topic, i.e., the probability clustered results of user behaviours. We provide a visual interface, with the distribution of initial behaviour clusters, the distribution of actions in each cluster, and the overlap among selected clusters. Analysts can understand the behaviour and interactively select suitable clusters by observing how representative they are and how much overlapping exist between them. In such an iterative manner, the analysts can better understand the behaviour, generate the behaviour clusters, and then further improve the user behaviour modelling.

Note: An online version of the tool is available on: http://simingchen.me/demo/security_sequence/

3.2 Demonstrator description

The demonstrator has four parts, including the LDA modelling part, Topic distribution view, action matrix view, and Venn diagram view.

Cyber security experts are interested in detecting possible misuses or fraudulent activities that are carried out using the administrative interface of a login and
security server by the system managers. They manage user authentication, access control and more sophisticated user rights. Because of the severity of the application, it is crucial for the experts to understand how it has been used. The input data for the system is LSS data. The LSS data set consists of 14,360 sessions performed by system managers; each session is considered as one behaviour. A session consists of multiple actions, for example, "Search User", "Display One User", "Create User", "Create Login Area". There are 296 distinct actions. These actions can be grouped into classes by domain experts. For example, the actions for searching and displaying the user can be in "User View" class, while creating a user or a login area can be in "User Create" class.

The existence of action classes is not essential for the idea of behaviour modelling with LDA. We utilize action classes for creating simpler views. We support the use of several alternative classifications. For example, searching user and searching office can belong to "Searching Operation" class, while creating user or login area can belong to "Creating Operation" class. Alternative classifications can help to (a) vary the level of detail (bigger/smaller sets of action classes, different levels in a class hierarchy, if it exists) and (b) look at results from different perspectives (different themes).

### 3.2.1 LDA Modelling

LDA techniques were originally proposed by Blei [Blei 2012] to detect hidden topics from a collection of documents, each of which contains a set of words. The input to LDA consists of a document-word matrix $DW$, where $DW_{ij}$ represents the frequency of word $j$ in document $i$ where $N$ is the number of topics. LDA produces a bag of topics $T$, a topic-word matrix $TW$, and a document-topic matrix $DT$. $TW_{kj}$ describes the probability of word $j$ in topic $k$, while $DT_{ik}$ represents the probability of topic $k$ in document $i$.

![Figure 3 An illustration for behavior analysis with LDA topic modeling.](image)

In our approach, we treat the actions as words and each behaviour containing multiple actions as a document as illustrated in Figure 3. Thus, we can find hidden topics of behaviours, which can be seen as distinct behaviour types, or categories. A behaviour is affiliated to a topic only if it has the largest probability to this topic. A behaviour is associated to a topic if its probability to this topic is larger than a given threshold, which can be chosen according to the specifics of an analytical scenario. In our illustrations, we set the threshold to 0.3. One behaviour can be affiliated only to a single topic, but can be associated to zero,
one, or multiple topics. The LDA method has parameters $\alpha$ and $\beta$ controlling the
distribution of topics over documents and words over topics. In the default
setting, we set them as 0.1. Apart from $\alpha$ and $\beta$, the topic number $N$ is an
important parameter for LDA. An appropriate topic number is often not known
in advance. For our illustrations and in use cases, we used an ensemble of 18 LDA
runs with the topic number ranging from 2 to 19, producing 189 candidate topics.
Generally, the chosen upper limit should be slightly larger than the expected
maximal number of distinct action classes in a given domain. Then the LDA
results are loaded into the visual analytics system.

3.2.2 Topic Projection View

To reveal relationships among topics, we apply the commonly used projection
methods MDS (Multi-dimensional Scaling) [Cox 2001] and t-SNE [Maaten 2008]
for projecting all generated topics into a 2D space (Figure 2 (a)). The projection
is based on the inter-topic distances derived from the LDA output, i.e., the
probabilities of the actions for the topics. We use the accumulated Manhattan
Distance. In the visual display of the projection results, topics are represented by
dots or by pie chart glyphs showing the probability distributions of action classes.
The segments in a pie chart are drawn in the order of decreasing probabilities of
the action classes. The user can choose from different action classifications to
vary the level of detail and or the themes. Switching to an alternative
classification may also be used for checking the resulting topics from a different
perspective. From the glyph distribution over the projection space, users can
assess the similarities and differences among topics and on this basis
interactively select representative topics. The glyphs themselves provide
additional information for judging topic similarity, so that the user does not have
to fully rely on the distances in the projection. The projection display serves as
an overview, an interactive panel, and a presentation of results.

3.2.3 Topic-Action Matrix View

To gain a detailed understanding of the topic features in terms of action
probabilities, we visualize data from the topic-word matrices $TW$ received from
the LDA ensemble with a matrix visualization (Figure 2 (b)). The columns
correspond to the actions and the rows to all or selected topics. The colours in
the cells encode action classes. The vertical ordering is based on the similarity of
the topics, which is consistent with the topic projection view. The horizontal
ordering is done according to the descending action probabilities following the
vertical arrangement of the topics. This means that actions with high
probabilities for the first topic are put in the beginning, following by not yet
included actions with high probabilities for the second topic, and so on. Such
ordering method helps users to understand which topics are similar and how
they are similar in the feature space of action probabilities.

To support understanding of dominating actions in topics, we encode the action
probability by the colour opacity. The linear mapping of the probabilities to the
opacity levels may not work well for probability distributions with high peak
values. To solve this problem, we propose a mapping involving a threshold $t$. 
To solve the problem of losing details, we use the outlier definition from Tukey’s fences and set it as a threshold for multiple linear mapping. The threshold is calculated based on measures of the interquartile range.

\[ t = Q_3 + k \times (Q_3 - Q_1) \]

In our setup, k has been set to 3. One can observe both dominant actions and actions with lower probabilities.

### 3.2.4 Venn Diagram

The representativeness of a current topic selection with respect to the behaviour set can be judged based on the amount of intersection between topics, the numbers of associated documents, and the number of documents without topic association. To support these judgments, we propose using a Venn diagram (Figure 2 (c)), in which the circles represent topics and their sizes indicate counts of associated documents. The colour encodes the action class with the highest probability in the topic. Intersections between circles represent shared associated documents. We can clearly understand important intersections of topics and inclusion relationship among the topics.

### 3.3 How to use the Demonstrator

An online version of the tool is available on:

[http://simingchen.me/demo/security_sequence/](http://simingchen.me/demo/security_sequence/)

In addition to this, the code for the demonstrator is available through this link:

[https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-csm_lda.zip](https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-csm_lda.zip)

**How to install?**

- Unzip the code-csm_lda.zip file
- Start a web server inside the unzipped folder. In a Mac machine, simply go to a terminal, `cd` to the unzipped folder and run `python -m http.server 8080` to start a simple web server using Python 3 with port 8080.
- Assuming, a web server is running with port 8080, the prototype can be accessed at [http://localhost:8080](http://localhost:8080)

We propose the workflow for using such visual analytics tool (Figure 4). It enables interactive extraction and exploration of behaviour categories with the
use of topic modelling. Analysts can select representative topics and modify the selection after understanding the topic features, their capabilities to differentiate behaviours, and the coverage of the behaviour set.

The proposed approach to topic selection is based on interactive grouping of topics by similarity and picking a representative topic from each group. The user defines a topic group by interacting in the projection display. The group medoid, i.e., the topic with the minimal distances to all others, is automatically selected as the representative. Alternatively, one could create a representative topic by computing the average or median probability values among all candidates. The medoid approach has two advantages. First, the medoid is not affected by extreme values in outliers, in contrast to the average value. Second, a medoid is an LDA-produced value vector, rather than an artificially calculated average or median. Currently selected medoids are highlighted in all views by black strokes.

### 3.3.1 Interactive definition of topic groups

We support two alternative operations for creating groups. First, the user can define a topic group by outlining a visually perceived dense cluster of similar topics in the projection display using a free-form brush. In this operation, the user can consider not only the positions of the topics in the projection space but also the similarities of the topics reflected by the pie chart glyphs. Considering that the distances in the projection space may not be proportional to the distances in the high-dimensional feature space, we also enable the second interaction method. Users can start by clicking a topic of interest, and the system in response automatically selects all topics within a given user-controlled distance in the feature space. The medoid of the group created in this way is not necessarily the same as the originally selected topic.
3.3.2 Editing groups and updating medoids

After exploring the relationship between topics in a group, a user may want to expand or shrink the group (Figure 5). Multiple interaction methods support exploration of topic groups. The distribution of the actions over selected topics is visualized in the matrix. For example, we can see several subgroups with different dominating action distributions. Columns and rows can be filtered by brushing the axes either in select or de-select modes. Users can select regions of interest and drill down to a detailed view of the action-based features of selected topics. In this example (Figure 5), the user brushes the G1 group on the Y-Axis and de-selects the actions with low probabilities by brushing the X-Axis. Thus, subgroups can be detected. The results of interacting with the matrix are reflected in the projection view.

Users can edit the groups in the topic projection. Users can dynamically change the threshold for expanding a group or re-brush the updated regions for editing the groups. In the Venn diagram, users can observe whether two topics have large intersections, and, if needed, modify the group selection in the topic projection.

3.3.3 Removing a group and its medoid

After examining the topics in the feature space, users may see a need to remove a subgroup and its medoid. As the goal is to find representative topics, the user can further examine the topic-document distribution with the Venn diagram. User may decide to remove a group if the number of associated behaviours is too small. It may also happen that a topic has a high intersection with another one or even contained within another topic. The user can remove a topic by clicking on the topic circle in the Venn diagram.

3.3.4 Multi-level exploration and Progressive categorization of behaviours

We formulate criteria for evaluating a topic selection and then introduce interaction operations that support the process of exploration and progressive
definition of behaviour categories. The users follow the rules of finding distinctiveness and high coverage of behaviours for iterative exploration. For a good topic selection, it is expected that the action frequencies in the affiliated behaviours are consistent with the action probabilities in the topics. Detecting a mismatch can help users understand which features are not correctly captured in the current topic selection.

For topic groups, we allow users to drill down to the detailed action probability distributions. Action classes can be selected or de-selected for exploring the actions of specific classes in more detail. Other views are updated according to the operation. Thus, users can examine the details and be better informed concerning how to refine the current topic selection.

Once the user has selected some topics, behaviours that are not associated to these topics (the probability is less than a threshold) are put into an additional 'other' group. A good selection should make the 'other' groups as small as possible. It might not reach zero, since behaviour outliers may exist. In the process of progressive exploration, the user can click on the 'other' groups; in response, the system will highlight in the topic projection the topics to which the respective behaviours are associated, which allows the user to see from where to pick the next topic. We also apply a blurred shading to the already explored groups of topics, informing the user about the state of the process. Thus, the user can further add and explore the new groups with these hints. The user can be aware of the currently selected medoids, to see how representative and unique they are in the feature space.

After the user selects representative topics, the action frequency distribution of the affiliated behaviours can be visualized, which shows how representative each topic is and whether there exist unexplored behaviour patterns. By interacting with the topic-action matrix, we can find suitable topics with high probabilities of the not yet represented actions and add the representative of this topic group to the current topic selection. Finally, we can view the action frequencies of the updated set of affiliated behaviours and find many behaviours falling into this category.

In such a manner, analysts can select and understand the representative behaviour cluster representatives (Figure 2 (a) highlighted black stroke).

3.4 Further improvements

There are some to-dos for further improvement. The main task is to further study how the clustering affects the predictions of the clustering-based user behaviour modelling component. Here it might be interesting to incorporate a feedback from this component into the visualization to allow users gain additional knowledge about groups of users and to further refine the clustering.
4 UBAP-2: Cluster-based Behavioural Model Building

4.1 Component description

The cluster-based behavioural model building component allows to analyse, model, and assess the probability of observed user behaviour. Using the probabilities of actions of observed users’ sessions, an attack-score is derived which indicates the likelihood of certain behavioural patterns being harmless, or potential attacks. This technique allows to detect attacks that are not discovered by classical patterns or rules but are only revealed as deviation from benign user behaviour.

The component takes as input user actions organised into session (each session is a sequence of temporally coherent actions by a distinct user) and assigns a score to each action in the session (except for the very first 4-10 actions, since those are required to determine the behavioural pattern).

The action score is derived from one, or multiple Long Short-Term Memory models (LSTM) [Hochreiter 1997] which output for each action its probability given the previous actions in the session. The score is then calculated from the probabilities provided by multiple LSTM models: one general LSTM that models the general behaviour of all users in the system, and an LSTM model for each user cluster provided by the topic modelling-based user behaviour analysis component. The probabilities are weighted by the similarity of the observed user session to each cluster.

The advantage of using specific models for each cluster is that the models can specialize on a limited set of user behaviour patterns. A general model for all users can be too broad and thus miss smaller deviations from benign user behaviour that might indicate an attack or on the other hand it might learn too much about common behaviours and thus consider smallest deviation as bad.

The cluster similarity is measured by a one-class SVM trained per cluster, each assigning a cluster membership score to a session. These scores are normalized and used as weights when the final likelihood predictions are generated. Note that it is also possible to set the maximum of all scores to one and all other to zero, i.e., consider each session as belonging to one and only one specific cluster. This way, only the predictions of the LSTM model belonging to the most similar cluster are used and all others are ignored.
Figure 6: Illustration of the interaction of session annotator and model training component with the common data storage.

The LSTM models are trained on logged user sessions, given a specific clustering. That means that sessions are separated according to the cluster they belong to and each of the subsets of sessions is used as a training set for an LSTM model. If the clustering changes, all models are retrained to adhere to the new clustering. This allows to adapt the LSTM models to novel insights about the user behaviour clusters obtained from the topic modelling-based user behaviour analysis component as well as from U4 and VASABI.

The cluster-based behavioural model building component integrates with the SIEM, as well as with the other components, through a common data storage, e.g., elastic stack. The component reads sessions from the data storage and writes the action scores to a separate index in the storage. These action scores can then be used by the SIEM or read by U4 and VASABI. The clustering is stored into the same data storage by the topic modelling-based user behaviour analysis component. Changes in the stored clustering trigger a retraining of the models. Here, the model training and scoring are executed in different processes, so that the time-consuming retraining does not stall the scoring of novel user actions.

### 4.2 Demonstrator description

The demonstrator of the cluster-based behavioural model building component consists of two python programs, the session annotator and the model training component as illustrated in Figure 6.
The session annotator reads newly logged sessions from the common data storage, assigns action scores and writes the annotated sessions back to the data storage. The scores are based on the predictions of the LSTM models that indicate the likelihood of the action in session according to the learned common behaviour in the specific cluster of behaviours. Configuration file that sets up preferences for the session annotator allows to configure the source and destination indices identifying the sessions to read from the data storage and the place to store predictions, as well as to further specify filters allowing to narrow down the data that is taken from the storage. The session annotator is a streaming program that annotates each novel session using the current model stored for predictions. In order to annotate older sessions, it needs to be restarted with a configuration file specifying the particular sessions that need to be annotated (e.g., using a time-window as filter).

The model training part of the component is bound to the clustering saved in the data storage. For each cluster it identifies the respective sessions from the data storage and trains an LSTM model on these sessions. Moreover, a separate model is trained on the union of all the sessions. The resulting models are forwarded to the session annotator part which replaces its current models with the new ones for further annotation. Once the clustering is updated in the data storage by the topic modelling-based component the model training is triggered again leading to the updated predictions on the novel sessions when the training is finished.

The demonstrator constitutes a close-to-complete version of the envisioned component. Note that the component is intended as background application that provides scores for the SIEM and other components. The component should only be altered and improved by operators with a machine learning background. These operators should be familiar with the python scripts and configuration files and be able to interact with them directly. Thus, this component will not have a separate user interface.

The architecture of the two programs combined into the component, namely the session annotator and the model training script, follows Object Oriented Design (OOD). The model training\retraining component generates models per clusters and these models are used by the session annotator to generate the action scores, which are weighed by the similarity of the observed user session to each cluster.

The model training\retraining python program included in the demonstrator performs three main tasks:
- Reading data in the form of logged user sessions from the data storage. The sessions are already clustered according to the currently available clustering. The task is performed according to the required API, for example in one of the commonly possible storage types for this type of data it is Elasticsearch API.
- Training LSTM models based on the datasets that are formed from the clustered sessions read from the common storage. For each of the clusters one specific model is trained. Moreover, one general LSTM is trained on all the read sessions without separating to clusters.
• Saving the trained models. The models should be saved to the common storage in order to be accessible for the other parts of the component, namely session annotator, and for further inspecting and manipulation by operators in case of need.

Second python program, session annotator, completes the following tasks:

• Reading the data from the common storage (Analogously to the first program this task is included in session annotator). Here, the task is to read the novel sessions according to the timestamp of the storing. The starting point should be specified in the configuration file, allowing to reannotate older sessions or, on the other hand, skip some of the sessions.
• Annotating the likelihood of each of the actions in the session. The session actions get annotated after there are more than four actions performed by the user. This is to avoid the cases where the identification of the behavior pattern from smaller sequences of actions fail. In order to annotate the actions, the models stored by the model training program are loaded and output scores of them are used as likelihood of the actions. The scores are calculated according to all of the currently saved models.
• Identifying the corresponding cluster to the currently read session. This is done via applying cluster specific SVMs and calculating scores for each of them. According to these scores the likelihood obtained from the LSTM models are weighted and combined.
• Passing the session data along with the newly generated action scores and the global score into the data storage for further usage by visual analytics component.

4.3 How to use the Demonstrator

The code can be found on the following link:

http://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-UserBehaviourModelling.zip

The python code for these two applications are organised into two separate folders “modelTraining” and “sessionAnnotator”. In order to use the demonstrator, the two python applications (session annotator and model training script) need to be run on a server with the access to the customer’s data storage that has the ability to provide the data as json files. In the respective configuration files, the data storage, the desired input and output indices and further filtering parameters need to be specified. The scripts require an Elasticsearch data storage with possibility to load out sessions in the json format (this is the configuration that we will use as part of the integration/validation phase).

The component will require sessions to be represented in a specific JSON format. Nevertheless, if the data storage cannot provide this format, the data reader script within both components can be adapted easily.
After starting the scripts, the parts of the component will work independently. We demonstrate the main features of the component by visualizing the action scores using Kibana-based on Elasticsearch data storage. Moreover, we demonstrate the dynamic re-training of models by changing the clustering with the topic modelling-based user behaviour analysis component, storing this clustering, and showing a change in the models of the session annotator.

The model training program should be executed first in order to save models for following usage by the session annotator. After the models are trained and indexed, the cycle continues, thus the model annotator will use the most recently generated models by model training script to annotate new sessions’ action scores. The input to the model training program is \( n \) sessions to be trained, in the specific JSON format which holds \( actionQueue,\ time\ Stamp \) of actions and current \( clusterId \).

The produced models from this program are saved in the data storage in the specified JSON format in indices configured by the configuration file using Elasticsearch API. These indices are always watched by the session annotator component to predict action scores for the newly logged session.

The input to the session annotator program would be a new session which holds \( actionQueue \) and \( time\ Stamp \) of actions. The result of the run of the script after getting such input is again the session data in JSON format but including generated additional tags to hold action score of each individual action in the session and the global score. These are again stored in the specified indices using elastic search API.

Both the python programs are configured using separate configuration files. These configuration files should be provided by the data administrators and also operators with the background in machine learning in order to specify how exactly the predictions would be performed and what sessions would be used for describing common behaviour.

The model training script requires the following points specified in its configuration file:

1. Information regarding Elasticsearch connection establishment credentials and also IP address of the storage if it is not running on the same server,
2. The number of sessions \( n \) as well as the specific range of the records to read from the storage for training the models,
3. The index of the data storage where the clustered sessions are stored,
4. The target index of the data storage where all the new models will be stored.

The configuration file also contains other relevant machine learning parameters to tune the process of the models training when required.

Similarly, the configuration file of the session annotator contains:
1. Information regarding Elasticsearch connection establishment credentials,
2. The source index where the models in JSON format are stored,
3. The starting point in the storage from where a new session is to be read,
4. The target index where the resulting session data together with generated action score and the global score is to be stored.

It also contains other prediction related parameters and specifies the way to calculate global score, e.g., pdt of action score, mean of action score, etc.

4.4 Further improvements

In principle, the demonstrator constitutes a close-to-complete component from a customer’s perspective. However, the actual user behaviour modelling can be continuously improved. For that, we focus on three main tasks:

1. Integrating more information into the predictions, such as timing of actions;
2. Improving the model architecture and further adapting it to given data sets;
3. Refine the attack score derived from the model outputs using actual attacks discovered by the SIEM operators.

These three tasks will be performed in close collaboration with the partners in the consortium for whom the component is provided.

For now, only the limited information about the user session is used for forming a behavioural pattern. Nevertheless, amount of the information usually stored by logging programs is much more and might be very helpful when identifying attacks. So, for example, some specific range of IP addresses of users can give an insight of the common source of malicious behaviour.

Since the parts of the component are interacting through common storage, it is possible to integrate new improved ways of knowledge extraction from the stored user sessions and apply more elaborated machine learning techniques for scores predictions without changing the overall functionality from the customer side. The research concerned with the integration of various machine learning methods into the security community is growing nowadays and new promising methods can be put into the component in order to further upgrade the visual analytics results presented for SIEM operators. Moreover, based on the experience obtained while using the component some successful techniques might be combined, e.g., if it will be concluded that considering only one specific cluster modelling is better than taking weighted prediction it can be further developed into forming additional models trained together on one cluster for ensembles of predictions.

Currently trained models describe users’ sessions using the assumption that the model describing common pattern can be built. It is more common for Machine
Learning models to be trained on the objects that are already distinguished, so called labelled data. This allows then to distinguish newly coming data according the labels that the model was trained on. Following the work with SIEM operators a set of sessions with malicious actions as labelled data for attacks might be formed. This further will be a good source for training a model for direct distinguishing between attacks and harmless sessions.
5 UBAP-3: U4 - Investigative Visual Analysis of Session Collections

5.1 Component description

5.1.1 The Problem

Our industrial partners have a great interest in the detection of insider threats through analysis of the behaviour that application users have during their usage within the application. In particular, to support such detection, Amadeus developed the SKEPTIC II framework (in WP6) that applies unsupervised statistical learning techniques to the log data to look for abnormal behaviours. The log data is split into sessions, each including the start and end time of the session, other meta information (user name, IP address, browser, etc.) and activity focused information detailing what happened (see Figure 7). SKEPTIC computes an anomaly score for each feature and combine the scores in a weighted manner to produce the aggregated one. This final score is used to assess the anomaly status of a user session. When the score is high, an investigation into that session is required to validate the score and search for an explanation. However, currently such an investigation is highly manual and time consuming.

![Figure 7. Different sources of signals and data utilised in the detection of anomalous behaviour within the SKEPTIC framework by Amadeus.](image)

The analyst has to examine the actions that took place in a given session in a data table format. Several pie charts showing summary statistics of the action types, such as top 10 most common ones are also provided in the current system. To understand what was going on in the session, the analyst must go through all actions listed in the table, which could be time-consuming, error-prone and difficult to detect any patterns. More challenging, to make an informed decision, the analyst often needs to compare one session with other sessions
performed by the same user in the past. Both the large number of sessions required to analyse and the fact that the sessions are often long and complex with several activities make the investigation challenging to execute.

5.1.2 Our Proposed Solution

We propose a visual analytics approach, in which interactive visualisations are combined with automated data mining techniques to enable analysts gain deep understanding into the sessions of interest. Below are the main goals and tasks that our solution aims to support.

Goal 1: Overall Understanding
- *Task 1.1 – Multi-perspective exploration of sessions.* Currently, the analysts can explore the sessions based on only the anomaly score. It can be biased because the score can be imperfect. Therefore, it is necessary to complement the score with other information, such as duration and length, to increase accuracy in identification of potentially suspicious sessions.
- *Task 1.2 – Higher-level semantic summary of action sequences.* Besides exploring at the session level, it is useful to have an overall understanding of what actually happened in those sessions before diving into particular ones. Currently, analysts rely on the frequency of action types, which is a semantically poor summary of the content of sessions.

Goal 2: In-Depth Analysis
- *Task 2.1 – Multi-scale exploration.* Help analysts explore what happened in a single session at different levels of granularity: high-level abstraction for quick understanding and detailed actions for in-depth inspection. Also, the tool needs to support exploration of multiple sessions such as the ones performed by the same user or executed with a high rate.
- *Task 2.2 – Comparative analysis.* Help analysts evaluate a session by comparing it with the past behaviour; i.e., the sessions previously performed by the same user. This is currently the most challenging task in the analysis because of comparison of multiple action tables. The tool needs to quickly reveal both the similarity and difference between a given session and the past sessions.

In order to achieve these, this demonstrator considers all the data on the sessions as well as the scores from the SKEPTIC system and also the other components described in the earlier sections.

5.2 Demonstrator description

The demonstrator, an interactive visualisation tool named U4 (see Figure 8), consists of three linked visual components that are designed to support the tasks described earlier. The demonstrator currently uses static data files that are provided by Amadeus. In the later integration, the tool will support reading data dynamically such as through Elasticsearch API.

In addition to this, the code for the demonstrator is available through this link: [https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-u4-vasabi.zip](https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-u4-vasabi.zip)

**How to install?**
- Unzip the code-u4-vasabi.zip file
- Start a web server inside the unzipped folder. In a Mac machine, simply go to a terminal, `cd` to the unzipped folder and run `python -m http.server 8080` to start a simple web server using Python 3 with port 8080.
- Assuming, a web server is running with port 8080, the prototype can be accessed at [http://localhost:8080/u4/](http://localhost:8080/u4/).

**5.3 How to use the Demonstrator**

In this section, we explain how to use our tool to enable analysts accomplished the goals and tasks outlined earlier. Figure 9 shows how the views link together.

---

**Figure 8.** Linked visualisations in U4. The Session View (A) helps explore relationships between session attributes. Sessions in the selected office are displayed in the Activity View (B) and the Timeline View (C) for further investigation. Most common actions in the selected sessions are colour-coded according to their types (see the legend in D) or their clustering results.


In addition to this, the code for the demonstrator is available through this link: [https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-u4-vasabi.zip](https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-u4-vasabi.zip)

**How to install?**
- Unzip the code-u4-vasabi.zip file
- Start a web server inside the unzipped folder. In a Mac machine, simply go to a terminal, `cd` to the unzipped folder and run `python -m http.server 8080` to start a simple web server using Python 3 with port 8080.
- Assuming, a web server is running with port 8080, the prototype can be accessed at [http://localhost:8080/u4/](http://localhost:8080/u4/).

**5.3 How to use the Demonstrator**

In this section, we explain how to use our tool to enable analysts accomplished the goals and tasks outlined earlier. Figure 9 shows how the views link together.
5.3.1  Goal 1 – Overall Understanding of Sessions

We facilitate the high-level analysis of sessions at two different levels: session (i.e., summary of sessions through attributes such as score, length and duration) and action (i.e., the atomic events depicting what actually happened).

5.3.1.1  Multi-perspective Exploration of Sessions

The Session View (Figure 10) shows an overview of sessions through different data attributes. Each session is displayed as a small rectangle with colour lightness showing anomaly score (the darker, the more anomalous) and height showing a numerical attribute such as session length, duration, and action rate. Sessions can also be grouped by a categorical attribute such as user, office and IP address.

<table>
<thead>
<tr>
<th>Session Overview</th>
<th>Group</th>
<th>User</th>
<th>Sort Group</th>
<th>Length</th>
<th>Height</th>
<th>Length Avg</th>
<th>Sort Items</th>
<th>Length</th>
<th>Baseline</th>
<th>Bottom</th>
<th>Color Score</th>
<th>Quantized</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10. Session View. Sessions are displayed as small rectangles with colour lightness showing anomaly score, height showing session length, and sessions are sorted decreasingly by length. Sessions are grouped by user. Mapping configuration can be changed using the options menu in the top-right corner of the view.

**Interaction**

- **Colour encoding.** Grey colour is used to encode anomaly score: the darker the more anomalous. Two options are available: (1) continuous: different shades of grey for different score, and (2) quantized: only three shades of grey are used for rounded score: low/medium/high. The options can be changed using the Colour Score dropdown in the options menu.
- **Height encoding.** Use the Height dropdown in the options menu to select which data attribute to map to the rectangle's height.
- **Items sorting.** Use the Sort Items dropdown in the options menu to select which data attribute to sort the session rectangles.
- **Items grouping.** Use the Group dropdown in the options menu to select which data attribute to group users.
- **Groups sorting.** Use the Sort Groups dropdown in the options menu to select which method to sort groups such as the median score of all sessions within a group.
- **Session alignment.** By default, sessions are aligned at the bottom. Within a group, sessions can be aligned to the group average, showing the difference from the average. Use the baseline dropdown in the options menu to explore different options.
- **Sessions selection.** Sessions of interest can be selected to display in other views
  - Click on a session to select that session.
  - Click on a group to select all sessions in that group.
  - Hold Shift and make a rectangular selection to select all sections in the drawn region.
5.3.1.2 High-level Semantic Summary of Action Sequences

The techniques in this section aim to provide an overview of sessions at an action level, but with a richer semantic than only the frequency of action types. We apply the classic Generalized Sequential Patterns algorithm to extract frequent patterns. We call each pattern an activity. The mined activities are visualised as shown in Figure 11, split into two parts: the right part visualising the actions in an activity and the left part listing statistics on these actions. Each activity is represented as a contiguous sequence of colour-coded squares, where each square represents an action. To characterise the frequency of an activity, three statistics are visualised in nested bars: the number of times the activity appears (biggest bar), the number of sessions having that activity (medium bar), and the number of users performing it (smallest bar). For instance, compare activity (second top) and activity (second bottom). The former repeats many times more than the latter, but taking place in a few sessions by a few users, whereas the latter is spread more evenly across several sessions and users.

![Figure 11. Activity View. The view displays common activities (as sequences of actions) that are mined from our constraint-based pattern mining algorithm. Actions are colour-coded based on their types or their groups. The bars on the left show summary statistics of sessions having the activities.](image)

5.3.2 Goal 2 – In-Depth Analysis of Unusual Sessions

This section discusses support for in-depth analysis of unusual sessions that are discovered in the previous stage. When sessions are identified in the Session View (Figure 8A), they are made available in the Timeline View (Figure 8C) with the following features for further analysis.

5.3.2.1 Multi-scale Exploration

**Single Session.** To help analysts gain understanding of a single session, we provide a visual summary of its actions. Actions are represented as coloured rectangles (consistent with the visual representation of an activity) and are displayed sequentially along a horizontal time axis based on their chronological order. We then provide visual representations of actions at four levels of detail to
enable analysts to examine a session with different purposes: a high-level summary or a detailed examination. This also helps address the spatial scalability when a session contains a large number of actions.

Figure 12. Different representations of a session. A black horizontal line in an action or activity indicates the size of an aggregate.

- **Action.** At the highest level of detail, actions are shown separately (Figure 12(A)), providing a sense of the session length (in terms of the number of actions) and facilitating detailed examination of individual actions.
- **Action Aggregate.** At the second highest level, consecutive actions having the same type are combined into one with a superimposed subtle horizontal line indicating the size of the aggregate (Figure 12(B)).
- **Activity.** At the third level, actions are replaced by mined activities whenever possible (Figure 12(C)). An activity is represented as a contiguous block of colour-coded rectangles, leaving no padding in between them. The height of non-activity actions is reduced to half to distinguish them with activities.
- **Activity Aggregate.** At the lowest level of detail, consecutively repeated activities are combined into one with a superimposed subtle horizontal line indicating the size of the aggregate, similar to the *action aggregate* level (Figure 12(D)).

We also provide a layout option to position actions horizontally proportional to their temporal values. This may cause visual clutter when actions happen close together but could reveal interesting temporal patterns such as time gap between activities (Figure 12(E)).

**Interaction**

- **Level of detail.** Use the *Detail* dropdown in the options menu to select one of the four levels discussed earlier.
- **Colouring.** Actions can be coloured based on the action types or action groups. They can be changed at the *Colouring* dropdown in the options menu.
- **Time.** This option change between Relative and Absolute ordering of actions along the time axis.
- **Comparison.** Turn on/off the comparison mode that is described later in Section 5.3.2.2.
Multiple Sessions. Analysts often need to examine multiple sessions simultaneously such as those having high scores and coming from the same IP address. We support this exploration by providing small multiples of visual summaries of sessions, each shown as a separate row (Figure 8(C)). The representation of each session is the same as the representation of a single session discussed earlier.

Figure 8(C) shows sessions performed by users in the selected office and are grouped by user. To the left of the user name is a simplified box plot, showing the median and the interquartile range of the scores of all sessions performed by the user in the past. It provides a historical context of how a typical session score would look like. Having multiple compact visual summaries of sessions allows comparison of sessions performed by the same user and across users. User Agent Cheesecake seems to be a help-desk user doing a lot of “unlock” and “reset password” activities. Whereas, users Jolt and Jack of Hearts deal with “right management” activities. Finally, sessions performed by Hall, Franklin and Geirrodur perform “delete user” activity very frequently and receive quite high scores.

5.3.2.2 Comparative Analysis

One essential step in anomaly investigation is to compare what a user does in a given session against what he or she did in previous sessions. We compute an expectedness score for each action in a session based on the frequency of that action in the session and the frequency of that action in all sessions performed previously by the same user. The score is then used to colour code actions in the Timeline View (Figure 13). This feature is available through a toggle button Comparison On/Off in the top-right menu of the Timeline View.

---

Figure 13. Comparative analysis between each session and its past sessions performed by the same user. This shows the same sessions as in Figure 8C. Dark red rectangles indicate highly unexpected activities.
5.4 Further improvements

In this prototype, we provide many features supporting analysts to identify unusual sessions and explore them in more detail. So far, we only use a dataset provided by Amadeus. In the next phase, we have a plan to evaluate the interface with domain experts from all partners using their real datasets.

Our approach in identification of unusual sessions is through analysis of multiple session attributes concurrently. Alternatively, one can focus on users and derive behaviour that is deviated from their typical behaviour. That leads to the work that will be discussed next.
6 UBAP-4: VASABI - Visual Analytics for Understanding User Behaviour

6.1 Component description

This chapter tackles the same problem as described in Section 5.1 with a complementary approach: identification of unusual sessions through visual analysis of user behaviour. We identified the following goals and tasks that we want to support the analysts.

**Goal 1: Understanding users and their behaviour through exploratory analysis**

- **Task 1 – Identifying users of interest.** Analysts routinely explore large collections of sessions performed by the users of the system with the aim of building an understanding of the common and unusual characteristics of users and identify unusual users for further investigation.

- **Task 2 – Understanding the different facets of user behaviour.** Individuals have idiosyncratic ways in which they use a system and analysts need to build a multi-faceted understanding of users’ behaviour to be able to make decisions when they work on individual cases and when they are developing/improving the User Behaviour Analytics (UBA) models.

**Goal 2: Evaluating unusual users and sessions through Investigative analysis**

- **Task 3 – Investigating patterns and deviations in user behaviour.** Whilst investigating a user that is identified for investigation, analysts need to be able to compare the selected user to other users (in the same organisation/similar roles) to decide whether any anomaly can be spotted in their profile.

- **Task 4 – Putting sessions in the context of a user.** Whilst an analyst is making use of an UBA system’s alerts to decide on whether a session is unusual, one very important criterion is to be able to compare the session to how the user normally behaves and make a judgement based on the observations.

6.2 Demonstrator description

The demonstrator, an interactive visualisation tool named VASABI (see Figure 14), consists of five linked visual components that are designed to support the tasks described earlier. The demonstrator currently uses static data files that are provided by Amadeus. In the later integration, the tool will support reading data dynamically such as through Elasticsearch API.
Figure 14. Linked visualisations in VASABI. The Session Overview (A) provides a temporal distribution of sessions and acts as a filter. The User Overview (B) provides a multi-perspective view on user attributes. The User Profiles (C) enables examination of user behaviour through multiple features concurrently. The Timeline View (D) allows detailed investigation of selected sessions. The Task Overview (E) summarises dominant tasks of selected sessions.


In addition to this, the code for the demonstrator is available through this link: [https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-u4-vasabi.zip](https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-u4-vasabi.zip)

**How to install?**
- Unzip the code-u4-vasabi.zip file
- Start a web server inside the unzipped folder. In a Mac machine, simply go to a terminal, `cd` to the unzipped folder and run `python -m http.server 8080` to start a simple web server using Python 3 with port 8080.
- Assuming, a web server is running with port 8080, the prototype can be accessed at [http://localhost:8080/vasabi/](http://localhost:8080/vasabi/).

**6.3 How to use the Demonstrator**

This section explains how to use each view in detail and how the views can help fulfil the tasks identified above. Figure 15 shows how the views link together.
### 6.3.1 Session Overview

The Session Overview provides a *temporal* distribution of all sessions in the entire dataset (Figure 16) with a histogram (supporting **Task 1**). In this figure, each bar spans across 3 hours with the height showing the number of sessions occurring within that period. The colour of the bar indicates the average score of its sessions: darker grey means higher average score. Several patterns can be observed here. The first one is a typical working hour pattern: there is much less data over the weekends and the peaks are around the middle of working days. A more surprising pattern is the large number of sessions occurred at midnight in almost every single day. Notably, on average, those sessions have higher scores than others (darker bars). Sessions of interest can be selected for further exploration and they will be shown in the User overview that is described next.

![Session Overview](image)

**Figure 16. Session Overview.** This histogram shows a temporal distribution of sessions.

**Interaction**
- **Sessions filter.** Drag horizontally along the histogram to select sessions within the corresponding time frame. The selected sessions will be used as the input for the User Overview.

### 6.3.2 User Overview

This view (Figure 17) enables the analysts to explore sessions of interest (selected in the Session view) with user and organisation perspectives. Sessions are grouped by user. Each user is represented by a small rectangle, which is coloured coded by *anomaly score*. The height of the rectangle can be mapped to different metrics such as the number of sessions and the median score. Users can also be ordered by the same number of metrics. Users can be grouped by organisation, and each organisation is separated by a rectangular border. This view provides a distribution of sessions over users and their organisations through visual encoding of different statistical metrics, helping spot common and outlying characteristics (supporting **Task T1**).

![User Overview](image)

**Figure 17. User Overview.** This view shows a summary of users through their data attributes. Each user is a bar and grouped by their organisation.

**Interaction**
This view provides similar interaction as the Session Overview described in Section 6.3.1. Use the options menu in the top-right corner of the view to explore the interactive features.
6.3.3 User Profiles

The User Profiles view (Figure 18) consists of multiple facets reflecting different characteristics of user behaviour (supporting **Task 2**). For each facet, we use a standard chart to show a summary of user sessions and composite the charts to make a compact representation of a user profile. The visual profiles are stacked together to enable comparison.

- For a per-profile facet (number of sessions and number of unique actions performed): a single bar is used, which is unnecessary for one profile but useful for comparing multiple ones.
- For a per-session facet (such as session anomaly score and duration), a histogram (for continuous values) or a bar chart (for discrete values) is used.

![Figure 18. Each row is a visual profile for a user, consisting of visual summary of multiple facets.](image)

In investigation of anomalous sessions, it is crucial to analyse them within the context of the users performing the sessions. This helps identify the deviation of user behaviour from the norm (supporting **Task 3, 4**). We support this analytical task by superimposing the sessions of interest, as small orange circles, on top of the visual summary (Figure 19). A random noise is added to the vertical position of the dots to avoid overplotting.

![Figure 19. Investigation with Profile view. Sessions of interest are shown as orange dots, allowing comparison with typical behavioural patterns drawn from the entire dataset, shown as histograms/barcharts. Red dots are sessions selected for further investigation (by clicking on a dot or a bar) in the Timeline view.](image)

**Interaction**

- **Session highlighting.** Hover the mouse on a session (shown as red) highlights the same session in other facets, enabling to observe different facets concurrently.
- **Cluster of sessions highlighting.** Hover mouse on a bar highlights all corresponding sessions (shown as red).
- **Sessions selection.** Click on a dot or a bar to select sessions, shown in the Timeline view and the Task view.

6.3.4 Session Timeline

The Timeline view is the same as described in the U4 tool (Section 5.3.2.1) but actions are always shown individually for simplicity.
6.3.5 Task Overview

A task can be considered as a set of related actions, usually performed together. We applied a clustering analysis to find 10 clusters (i.e., tasks) of actions and assign each cluster to a session. This Task overview (Figure 20) aims to provide a closer look at what each task is about and facilitate task comparison. Each task can be represented by five dominant actions, shown as a set of five squares, each for an action, coloured based on the action’s group. The left of the view shows two statistics: the distribution of tasks in the entire dataset (lighter bars) and the distribution of tasks within a selection of interest, such as sessions performed by a particular user in a given time range (darker bars). This enables comparison across tasks within a user: task 7 (lighter bars) is the most popular task in the entire dataset; however, within the selected sessions, task 5 (darker bars) is the most dominant (supporting Task 3).

Figure 20. Task Overview. Each task is shown as a set of coloured squares, each representing a dominant action in the task. Actions are colour coded based on their group.

6.4 Further improvements

So far, we only use a dataset provided by Amadeus. In the next phase, we have a plan to evaluate the interface with domain experts from all partners using their real datasets. Currently, U4 and VASABI are two separate interfaces. In the integration phase, we will investigate the possibility to merge the views in the two tools to combine their strengths.
7 Diversity and Forecasting Analytics Dashboard

7.1 Component description

This dashboard and the Diversity and Forecasting Analytics Engine component described in D6.2 [D62] (which is in-submission parallel to this deliverable report) form a single component that aims to provide insight into the diverse configurations of the monitoring tools and their predictive capabilities. The ultimate goal is to build solutions for SOC operators and cyber security analysts in helping them make better informed decisions during both the design of a defence strategy and during the evaluation of ongoing attacks and anomalies.

In terms of pipeline, the dashboard component receives data from SIEMs, invokes the engine to apply the predefined models, visualises the model output, and offers interactive capability for the analysts to filter data and refine the models. More specifically, three main goals that we want to support the analysis are described below.

Goal 1 – Overview of alerts
The alert data comes from SIEMs and at this stage, no information on whether the alerts are real attacks or not is yet available. Due to this fact, the nature of the analysis is purely exploratory at this stage and has the goal to gain an overview of how the alerts are distributed over time and over different configurations, and to identify trends and outliers. The dashboard aims to:

- Display the overall distribution of alerts broken down by time and other informative attributes such as protocols and IP addresses
- Filter and focus on a particular time period or attribute value.

Goal 2 – Interactive exploration of diverse configurations
The dashboard invokes the engine to get labelled alerts, i.e., data on whether the alert is associated with a real attack or not. The dashboard enables analysts to:

- Manually adjust what the performance metrics are,
- Filter the alerts to focus on a particular subset (e.g., only False Positives),
- Filter time periods,
- Observe changes over time and/or performance during a particular instance such as an attack.

Goal 3 – Analysis and evaluation of model ensembles
This stage of the investigation involves a combination of the analysis modelling outputs with the aim of evaluating the forecasts for future potential vulnerabilities. The dashboard visualises the model output from the engine and enables analysts to

- Visually investigate several models with their forecasts in a synoptic way,
- Visualise the uncertainty in the predictions,
- Relate the predicted models to past raw data to provide context to the predictions.
7.2 Demonstrator description

The demonstrator, an interactive visualisation tool (see Figure 21), consists of two linked visual components that are designed to support the tasks described earlier. The demonstrator currently uses static data files. In the later integration, the tool will support reading data dynamically such as through Elasticsearch API.

Figure 21. Linked visualisations. The Events view (left) provides a temporal distribution of events and acts as a filter. The Modelling view (right) visualises the model output and the prediction for selected events.

An online version is available at:

http://www.staff.city.ac.uk/~sbkr014/divis/examples/events/

In addition to this, the code for the demonstrator is available through this link:

https://www.staff.city.ac.uk/cagatay.turkay.1/Resources/Projects/DiSIEM/D5_2/code-diversity.zip

How to install?
- Unzip the code-diversity.zip file
- Start a web server inside the unzipped folder. In a Mac machine, simply go to a terminal, `cd` to the unzipped folder and run `python -m http.server 8080` to start a simple web server using Python 3 with port 8080.
- Assuming, a web server is running with port 8080, the prototype can be accessed at http://localhost:8080/examples/events.

7.3 How to use the Demonstrator

This section explains how the views are designed to help fulfil the goals described earlier. So far, we have focused on goals 1 and 3.
7.3.1 Events View

The Events View shows a temporal distribution of events with a histogram (Figure 22). In this small example dataset, each bar spans 5 minutes. We can see an interesting pattern here: a gap in the middle of the view. Initially, all events are used for modelling. However, a subset of interest can be selected for modelling, such as only events before or after the gap.

Figure 22. Events View shows a histogram of events, binned by time.

**Interaction**
- **Events selection.** Drag horizontally along the histogram to select events within the corresponding time frame. The selected events will be used for modelling.

7.3.2 Modelling View

This view takes input as the events selected in the Events view. Intervals between events are computed and passed to the Forecasting engine [D62]. This view visualises the modelling output, containing the results from 8 different models. Initially, only intervals are shown (i.e., none of the models are visible) but models can be selected in the top-right menu as in Figure 23.

**Interaction**
- **Model selection.** Click to select/deselect the models on the top-right corner menu.
- **Uncertainty show/hide.** Click on the Uncertainty checkbox to show/hide the uncertainty of the models.
- **Model update.** Select the events in the Events view to see the models updated (Figure 24).
7.4 Further improvements

The visualisation tool has focused on providing an overview of alerts (supporting Goal 1) and visualising the model output (supporting Goal 3). Currently, the prototype is still progressively being improved. We continue providing additional solutions for the goals identified, including:

- Events View: provides other ways to filter data such as by protocols
- Modelling View: the current design is mainly for technical users who want to understand and explore which models are optimal and for which subset of data. A simpler view can be designed for practical users who only have interest in the computed best model and prediction value.
8 Summary and Conclusions

The deliverable described two components: the user behaviour analysis platform and the diversity and forecasting analysis dashboard. The user behaviour platform comprises of four sub-components which are all defined separately with an initial description that describes how these sub-components work together towards an effective analysis and modelling of user behaviour. For the diversity forecasting and analysis dashboard, we demonstrate how visual analytics can facilitate the interactive generation of statistical models for a more involved, better-informed user experience. For each demonstrator, a description of the components is provided followed by how the demonstrator can be used and how it can potentially be improved.

All demonstrators are in a usable state, so that the evaluation by the project partners can focus on improving their utility to the SIEM operators. A solution to integrate the various components under the user-behaviour platform is also presented. The emphasis will now be moved into the system integration and further iterative improvement of these components. The next deliverable for WP 5 (D5.3) will demonstrate the final components resulting from the evaluation by the project partners in real SIEM environments.
References


