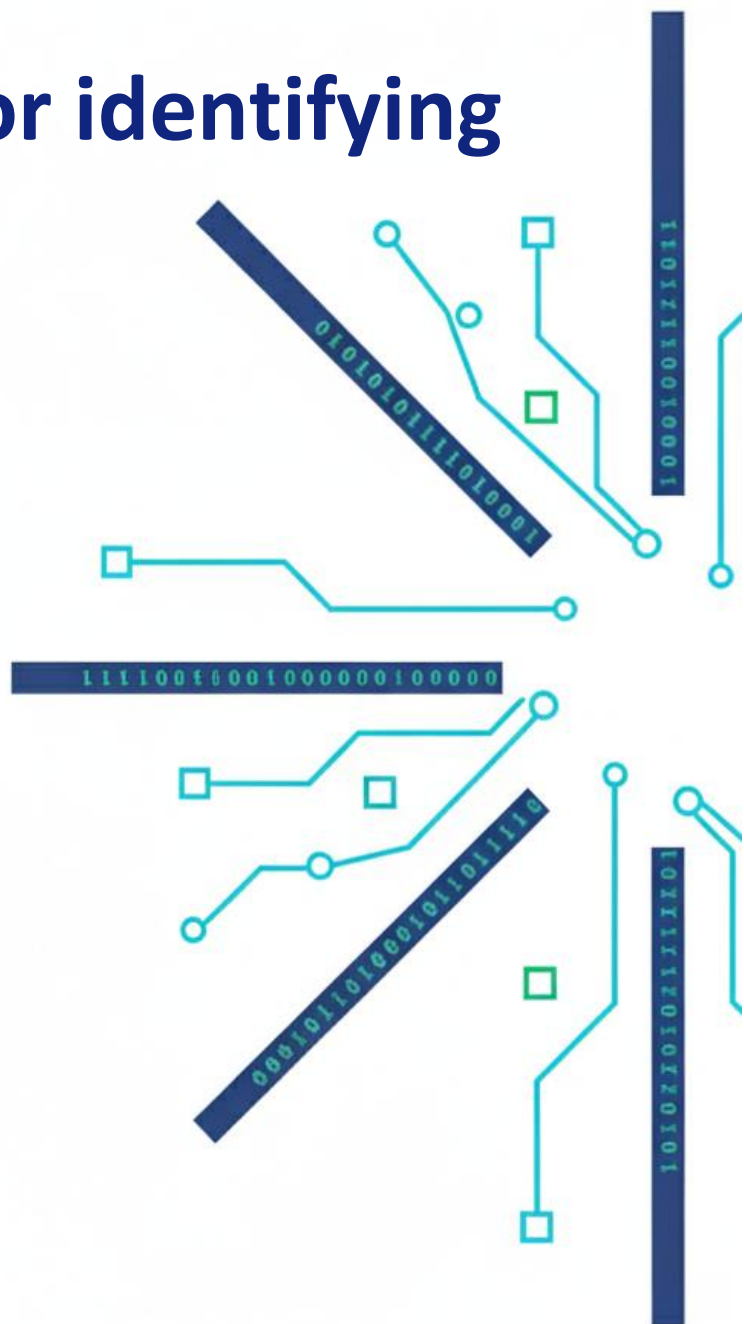


D19 – Method for identifying digital traces



The Automation of Digital Forensic Analysis and Incident Response (ADFIR) project, funded by the European Union – Next Generation EU through the Recovery and Resilience Plan of the Slovak Republic under project number 09I05-03-V02-00079.

CONTENTS

Project description	3
Introduction	4
1 An Introduction to Clustering.....	5
1.1 Clustering analysis in digital forensics.....	5
1.2 Hierarchical clustering and evaluation methodology	6
1.2.1 Forensic cluster metrics.....	6
1.2.2 Validation metrics for determining the number of clusters	7
1.2.3 Anomaly detection using clustering.....	8
2 Application of Selected Methods	9
2.1 Identification of digital traces using clustering	9
2.1.1 Data sets and their pre-processing.....	9
2.1.2 Hierarchical clustering	11
2.1.3 Interpretability of results	12
2.1.4 Experimental results	12
2.2 Identification of digital traces using large language models	15
2.2.1 Data sets and their pre-processing.....	15
2.2.2 Analysis using a large language model.....	16
2.2.3 Evaluation of success	17
2.2.4 Interpretability of results	17
Bibliography.....	20

Project description

The **Automation of Digital Forensic Analysis and Incident Response** (hereinafter 'ADFIR') project is funded by the **European Union – Next Generation EU – through the Recovery and Resilience Plan of the Slovak Republic** under project number 09I05-03-V02-00079. This project addresses one of the key challenges in the field of cyber security and information security – how to process the vast amount of digital evidence generated during cyber security incidents or forensic investigations. At present, this process is very demanding in terms of human resources and time. Automation using machine learning methods can therefore significantly **improve the quality of digital forensic analysis** and reduce the time required to carry it out. Overall, this enables security teams to respond more effectively to cyber threats. The main benefits of this project are:

- **Faster resolution of cyber security incidents.** The ADFIR project introduces automated approaches to the collection, processing and analysis of digital traces. This enables security teams to identify the causes of incidents more quickly and take effective measures to resolve them.
- **Reduced workload for forensic analysts.** Routine and time-consuming tasks associated with processing digital traces will be replaced by automated methods. This will allow analysts to focus on more complex cases and strategic decision-making.
- **Higher quality and consistency of outputs.** The use of standardised methodologies and tools ensures that processed digital evidence will be more accurate, consistent and easier to verify. This significantly reduces the risk of errors caused by human factors.
- **Potential use in criminal proceedings.** The project's outputs will be developed in accordance with legal requirements and standards, enabling digital traces to be accepted as relevant evidence for investigations and court proceedings.

Introduction

In the field of digital forensic analysis, the efficient processing of large volumes of heterogeneous data presents a major challenge, particularly when reconstructing events and identifying causal links between individual artefacts. Modern approaches therefore increasingly utilise machine learning and artificial intelligence methods, which enable the transformation of isolated forensic traces into a model that can be utilised by analysts.

This document focuses on the design and implementation of a method for identifying forensic traces within the ADFIR project (Task KPB4.1 – Identification of suitable methods for identifying digital evidence relevant to the case under analysis). The aim is to create a model that enables the systematic processing of artefacts, the reduction of the volume of records, their correlation across different sources, and their subsequent interpretation in the context of identified system or user behaviour scenarios.

A key part of the work involves data processing and the selection of suitable machine learning models for identifying digital traces in digital forensic analysis. This approach enables the search for digital traces using artefacts whilst reducing the volume of individual records that the analyst must process.

The proposed model thus provides a systematic way to move from fragmented data to an integrated view of digital traces, supporting not only the reconstruction of events but also the identification of behavioural patterns relevant to forensic analysis.

1 An Introduction to Clustering

In the context of digital forensic analysis, the identification of digital traces is a complex process aimed at uncovering relevant patterns, relationships and anomalies within large datasets. As the nature and structure of digital traces can vary significantly depending on the data source and the environment being analysed, there is no universal method suitable for all types of analysis. For this reason, this document presents several methods suitable for identifying digital traces from different perspectives.

The first method employed was hierarchical clustering, which enables the identification of naturally occurring groups within the data without the need to define in advance the exact number of resulting clusters. This feature is significant in the field of digital forensic analysis, as the number and nature of similar groups of digital traces cannot be reliably determined in advance. The method therefore enables the discovery of hidden structures within the data and the identification of potentially related activities or events.

The second approach examined was the identification of digital traces using large language models (LLMs). This approach differs from traditional machine learning methods primarily in its ability to work with complex textual representations of digital traces and to utilise knowledge acquired during the model's pre-training. In the field of digital forensic analysis, large language models enable the analysis of extensive and heterogeneous event timelines, the identification of potentially security-relevant activities, and the provision of a contextual interpretation of these activities. A significant advantage of this approach is the ability to work even with a limited amount of labelled data, which is important in the context of digital forensic analysis given the labour-intensive nature of manually labelling forensic records.

A key aspect of the use of machine learning methods in the field of digital forensic analysis is not only the ability to identify patterns, clusters or anomalies in the data, but also the ability to interpret the results obtained. In the field of digital forensic analysis, explainability is one of the key aspects, as the results of automated methods must be comprehensible and verifiable by forensic analysts. A model or method that identifies suspicious behaviour without providing information on the reasons behind its decision may have limited usefulness in the investigation process. For this reason, as part of the proposed approach, we also focus on identifying the attributes and characteristics that contribute most significantly to the resulting classification or the groups of digital traces formed.

1.1 Clustering analysis in digital forensics

In digital forensic analysis, clustering is used to automatically organise artefacts, logs or events into groups based on their similarity. This approach radically reduces the complexity of the data and enables more efficient analysis. Given that modern systems generate millions of low-level events during forensic timeline analysis[1] , clustering helps investigators reduce the volume of data to a manageable level and focus on representative groups rather than analysing individual records[2] .

The main challenge of digital forensic investigation is identifying relevant evidence amongst a vast amount of irrelevant data. Research shows that categorisation based on machine learning and clustering can effectively group files, prioritise evidence (e.g. in email correspondence) and significantly reduce the duration of investigations[3][4] .

Clustering is closely linked to **anomaly detection** and **event reconstruction**:

- **Anomaly detection:** Events (e.g. authentication logs) are clustered (e.g. using the MajorClust algorithm) and anomalies are identified based on specific properties of the clusters, such as their size, frequency or temporal characteristics[5][6][7] .
- **Event reconstruction:** This process is often fragmented. Clustering helps to organise events into meaningful sequences and reveal relationships between them, which, in combination with process mining, enables the extraction of real behavioural patterns even from incomplete or manipulated data[8][9][10] .

In modern systems, clustering is integrated with other techniques, such as **formal concept analysis (FCA)** and association rule mining, to gain a deeper understanding of the relationships between digital evidence[11] . Advanced frameworks also utilise large language models (LLMs) to interpret timelines[12] or combine dimensionality reduction with clustering to detect internal threats in real time[13] .

1.2 Hierarchical clustering and evaluation methodology

In the proposed approach, we apply **agglomerative hierarchical clustering**[14][15] to defined time windows. This is a bottom-up approach, where initially each time window forms a separate cluster. In each subsequent step, the two nearest clusters are merged until a single final cluster encompassing the entire dataset is formed.

The result is a hierarchy of clusters, such that for any two clusters A and B from different levels, exactly one of the following relationships holds:

$$A \cap B = \emptyset, A \subset B, \text{ or } B \subset A.$$

Individual agglomerative methods differ primarily in the way they update the dissimilarity measure between clusters after each merger. In our research, we do not assume that clusters must form strictly consecutive time windows – clusters may overlap and represent similar behaviour in different parts of the timeline.

1.2.1 Forensic cluster metrics

If similar time windows are found both before and after an incident, the cluster represents normal system behaviour. However, if the cluster is dominated by windows from the period following the onset of a cyber incident, we consider the cluster to be suspicious. To quantify this relevance, we introduce the following metrics:

- **Attack Rate:** The proportion of attack records within a given cluster.

$$attack\ rate = \frac{attack\ count}{cluster\ size}.$$

- **Attack Lift:** A comparison of the attack rate within the cluster against the global attack rate across the entire dataset.

$$attack\ lift = \frac{attack\ rate}{attack\ rate_{global}}.$$

- **Cluster Size Proportion (CSP):** Expresses the relative size of the i -th cluster in relation to the total number of time frames (N) in the dataset.

$$CSP_i = \frac{cluster\ size_i}{N}.$$

1.2.2 Validation metrics for determining the number of clusters

To determine the optimal cut in the hierarchical tree (and thus the optimal number of clusters), we use three internal validation metrics calculated in parallel, which we interpret independently:

1. **Silhouette score** [20]: Measures the balance between cohesion within a cluster and separation from other clusters. A higher value indicates better clustering.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where $a(i)$ is the average intra-cluster distance and $b(i)$ is the minimum average distance to objects in other clusters.

2. **Davies–Bouldin score** [21]: Evaluates the ratio of the variance within clusters to the distance between their centroids. A lower value indicates better separation.

$$DB = \frac{1}{k} \sum_{i=1}^k \max \frac{\sigma_i + \sigma_j}{d(c_i, c_j)},$$

where σ_i is the cluster variance and $d(c_i, c_j)$ is the distance between the centroids.

3. **Calinski–Harabasz score** [22]: Quantifies the ratio of dispersion between clusters to dispersion within clusters. A higher value indicates better-defined clusters.

$$CH = \frac{Tr(B_k)/(k-1)}{Tr(W_k)/(n-k)},$$

where B_k and W_k denote the matrices of dispersion between clusters and within clusters, respectively.

1.2.3 Anomaly detection using clustering

The results of hierarchical clustering can be used not only to identify groups of similar digital traces, but also as a basis for subsequent anomaly detection. Since individual events are first aggregated into time windows and then grouped according to similarity, anomalous behaviour can be identified by analysing the properties of the resulting clusters and their evolution over time.

The first approach involves evaluating the properties of individual clusters. Particular attention is paid to metrics characterising the size and compactness of clusters, such as the **Cluster Size Percentage** (CSP) indicator. Small or isolated clusters that differ significantly from the dominant behavioural groups may represent candidates for point or collective anomalies and indicate the occurrence of non-standard events in the analysed system.

The second approach involves monitoring the temporal evolution of individual clusters. If a particular type of behaviour only appears after the onset of a security incident, or if its occurrence increases significantly during a system compromise, this may constitute anomalous behaviour relative to the historical norm. For this purpose, the **Attack Rate** and **Attack Lift** metrics were utilised, which enable the quantification of the relationship between the occurrence of a specific cluster and the incident under analysis. Clusters with high values for these metrics represent suitable candidates for more detailed forensic analysis.

2 Application of Selected Methods

This chapter presents the practical steps involved in applying the selected analytical methods to prepared forensic datasets. The aim of the experimental section is to verify the usefulness of the methods described in the previous chapters in identifying significant patterns in digital traces, detecting anomalies and subsequently interpreting the results obtained. The individual methods are applied in successive steps – grouping similar records using clustering and identifying digital traces using large language models.

2.1 Identification of digital traces using clustering

This section investigated the possibility of using clustering methods to identify similar groups of digital traces in large forensic datasets. As it is often not known in advance in real-world investigations what types of activities or events are present in the analysed data, unsupervised learning methods were used, which enable the discovery of naturally occurring structures without the need for predefined classes.

The aim of this analysis was not to replace the forensic analyst's judgement, but to reduce the volume of data requiring manual review and to support the identification of potentially related events. The assumption is that digital traces belonging to similar activities or the same type of behaviour will exhibit similar characteristics in the represented data space and form clusters with a higher degree of internal similarity.

Hierarchical clustering was utilised in the experiments, which allows the creation of a hierarchical structure of relationships between objects without the need to specify the number of resulting clusters in advance. This approach is particularly suitable for forensic data, where the number and nature of individual groups of events cannot be reliably determined prior to the analysis itself.

2.1.1 Data sets and their pre-processing

For the experimental part, modified datasets originating from Capture The Flag (CTF) competitions were used; these competitions focus on the forensic analysis of Windows operating system disk images. These datasets contain realistic scenarios of user activity, system events and simulated security incidents.

The original disk images were processed to extract relevant forensic artefacts and transform them into a structured representation suitable for subsequent analysis. The extracted events, which included timestamps, were subsequently organised into time windows.

The following datasets were used as part of this approach:

- NIST Data Leakage Case[18] ,
- Magnet CTF 2019[16] ,
- Magnet CTF 2022[17] .

These datasets represent various types of operating system behaviour, including normal user activity, administrative operations and simulated security incidents, such as unauthorised access or data leaks. The use of multiple independent datasets allows us to verify the robustness of the proposed approach and reduces the risk of the method being tailored to just one specific environment.

Individual operating system events were divided into time windows of a fixed duration of 60 seconds. Each time window represents the state of the system during that interval and contains aggregated information from all digital traces recorded during that period.

The choice of a 60-second interval represents a compromise between the level of detail in the analysis and the number of resulting time windows. Whilst a shorter interval would provide a more detailed representation of the system's behaviour, it would significantly increase the number of units analysed. Conversely, longer time windows could lead to the loss of important details.

The number of time windows created for each dataset was as follows:

- NIST Data Leakage Case – 1,219 time windows,
- Magnet CTF 2019 – 4,745 time windows,
- Magnet CTF 2022 – 18,170 time windows.

Individual low-level events were transformed into a higher level of representation describing the system's behaviour over time. The Plaso framework was used to extract time-stamped forensic events; this framework enables the retrieval of information from various system sources, particularly the NTFS metadata system. This is described in more detail in the output (ADFIR-D13-KPB2-02 Method for creating a dataset).

Based on the extracted artefacts, approximately 200 attributes were created, representing various types of system activity, such as file operations, metadata changes, access to objects or system events.

The attributes were represented by discrete values:

- a value of 1 indicates the occurrence of a specific type of event,
- a value of 0 indicates the absence of an event,
- a value of NaN indicates unavailable or missing information.

Two aggregation functions were used to aggregate events within time windows:

- sum – represents the number of occurrences of a specific event within a given time window,
- max – indicates only whether the event occurred within the time window.

These two functions were chosen because they are suitable for forensic data containing missing values (NaN). Unlike statistical methods based, for example, on the mean or median, they are not affected by the absence of certain types of events.

The result of the pre-processing is a structured temporal representation of the system, which enables subsequent analysis using clustering methods and anomaly detection.

2.1.2 Hierarchical clustering

To identify similar patterns of system behaviour, a hierarchical clustering method was used, specifically agglomerative clustering.

This method employs a bottom-up approach, where initially each object represents a separate cluster. In successive steps, the most similar pairs of clusters are then merged until a single resulting cluster containing all objects is formed. The result is a hierarchical cluster structure that allows the analysis of relationships between groups at different levels of similarity.

The advantage of this method in the field of digital forensic analysis is that it does not require a predefined number of clusters. This is because the number and nature of individual groups of digital traces cannot be reliably determined prior to the analysis itself.

The analysis does not assume that similar time windows must follow one another directly on the timeline. The same type of behaviour may occur in different parts of the analysed period. Clusters containing time windows both before and after the start of the incident are likely to represent normal system behaviour. Conversely, clusters consisting predominantly of events occurring after the start of the incident may indicate potentially suspicious behaviour and are candidates for further forensic analysis.

To assess the significance of individual clusters, we used the metrics defined in the section **Forensic cluster metrics**:

- attack rate,
- attack lift,
- cluster size proportion (CSP).

To determine the appropriate number of clusters, we used the three validation metrics mentioned in the chapter **Validation metrics for determining the number of clusters** Validation metrics for determining the number of clusters:

- **Silhouette score**
- **Davies-Bouldin index**
- **Calinski-Harabasz index**

Individual clustering solutions were evaluated using all three metrics separately; no single combined scoring indicator was created.

The second step of the analysis utilises the hierarchical structure of the clusters to identify similar parent groups and to search for relationships between them. This procedure provides further options for the forensic analyst when examining digital traces.

2.1.3 Interpretability of results

The final part involves ensuring that the results are interpretable for the digital forensic analyst. The creation of clusters alone does not provide sufficient information about their nature; it is therefore necessary to identify the attributes that most clearly describe the individual groups of time windows.

The interpretation was carried out in the original attribute space, not directly in the transformed space used during clustering. Clustering therefore takes place on a modified representation of the data, whilst the significance of individual clusters is determined by a retrospective analysis of the original forensic attributes.

When assessing the significance of attributes, features that showed no activity either within a given cluster or across the entire dataset were removed. The attributes were then ranked according to the absolute value of the standardised score (score std), which allows the significance of attributes with different ranges of values to be compared.

This approach makes it possible to identify attributes with elevated or, conversely, suppressed occurrence within a specific cluster. The result is an interpretable description of the system's behaviour, which can serve as a basis for subsequent forensic investigation.

2.1.4 Experimental results

The proposed approach was experimentally validated on the NIST Data Leakage Case, Magnet CTF 2019 and Magnet CTF 2022 datasets. The aim of the experiments was to verify the ability of hierarchical clustering to identify groups of time windows representing

potentially suspicious behaviour whilst maintaining the interpretability of the results for digital forensic analysts.

During the experiments, various slices of the hierarchical tree, representing between 10 and 50 clusters, were analysed. A division into 50 clusters proved to be the most suitable solution, providing a good balance between the level of detail and the interpretability of the results. A higher number of clusters made it possible to better distinguish specific forms of system behaviour and to identify smaller groups of events related to a security incident.

Based on the analysis, four basic types of clusters were identified:

- dominant clusters representing normal system operation,
- clusters containing exclusively incident-related behaviour,
- clusters representing the period prior to the incident,
- suspicious clusters containing a combination of normal and incident-related behaviour.

Dominant clusters constituted a significant proportion of the analysed data. For example, in the Magnet CTF 2022 dataset, the largest cluster covered approximately 72.6% of time windows when aggregated using the sum function and approximately 49.9% of time windows when aggregated using the max function. These clusters were characterised by attributes associated with normal system activity of the Windows operating system.

Particular attention was paid to so-called 'incident clusters', for which the *Attack Rate* metric was equal to 1. Such clusters contained exclusively time windows occurring after the start of a security incident. In the Magnet CTF 2022 dataset, clusters were identified containing events related to the creation of a user account or the execution of a suspicious file utilising Windows persistence mechanisms. Analysis of the attributes within these clusters enabled the rapid identification of digital traces relevant to the investigation of the incident.

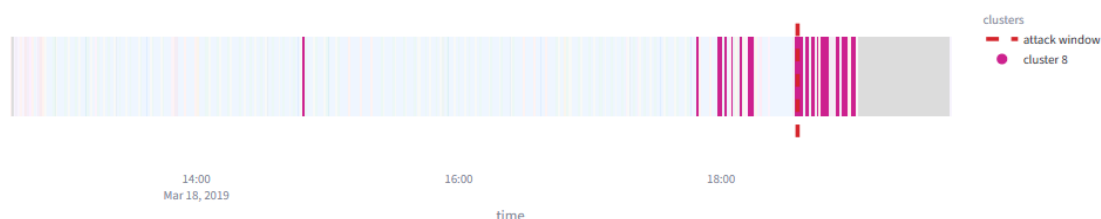


Figure1 : Clusters in Magnet 2019

A specific example is cluster 8 in the Magnet CTF 2019 dataset when using aggregation via the *max* function. Its temporal distribution fluctuates around the start of the incident, suggesting that the time windows assigned to this cluster occur both before and after the manually defined start of the incident. Such behaviour may point to events related to the

initial phases of the incident, or to inaccuracies in determining the exact time of its start. An example of this behaviour is illustrated at **Figure 1**.

The results also showed that a combination of *the Attack Rate*, *Attack Lift* and *Cluster Size Proportion* metrics enables the effective differentiation between normal system behaviour and behaviour associated with an incident. It was also possible to identify clusters located close to the start of the incident, which may indicate the initial stages of compromise or an inaccurate determination of the time the incident began.

The resulting structure of the time-window groupings is visualised at **Figure 2**. The figure shows the distribution of time windows for all three analysed datasets – NIST Data Leakage Case, Magnet CTF 2019 and Magnet CTF 2022 – using aggregation via the *sum* and *max* functions. The different colours indicate which cluster a time window belongs to; a total of 50 clusters were created. The red dotted line marks the manually defined start of the security incident and allows the temporal distribution of the identified clusters to be compared with the course of the incident. The visualisation shows that suspicious behaviour is concentrated in specific time intervals, whilst the dominant clusters cover the majority of the time windows outside the incident.

Experiments have confirmed that the proposed methodology significantly reduces the volume of data that a forensic analyst must examine manually, whilst maintaining a sufficient level of interpretability required for practical application in digital forensic analysis.

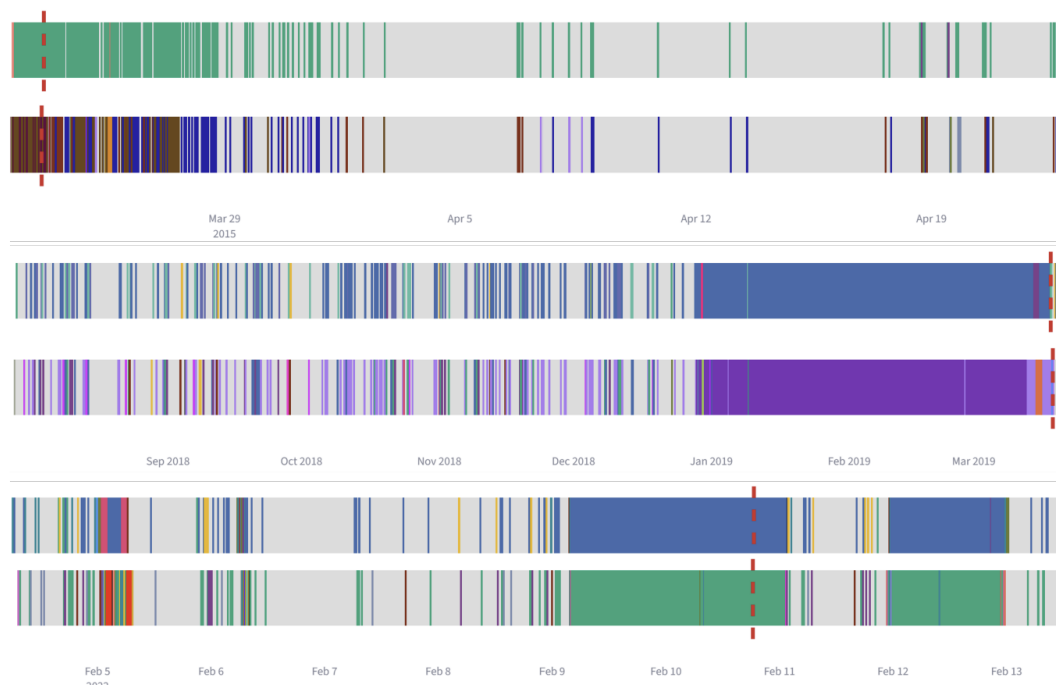


Figure 2 : Visualisation of time windows

2.2 Identification of digital traces using large language models

Another approach investigated was the identification of digital traces using large language models (LLMs). Unlike traditional machine learning methods, in this case no model specialised directly in forensic data was created; instead, an existing pre-trained language model was utilised, whose knowledge derives from extensive training on general text data. The aim was to verify the ability of such a model to interpret complex event timelines in the Windows operating system and to identify digital traces related to security incidents.

During the design of the solution, several architectures based on machine learning methods were experimentally validated. The first approach investigated was an LSTM autoencoder, which utilises unsupervised learning and detects anomalies based on the difference between the original and reconstructed event sequences. Subsequently, an approach based on the BERT-type Transformer architecture[19] was investigated; through its self-attention mechanism, this enables more effective capture of relationships between individual events, even within larger contexts.

However, both approaches encountered significant limitations arising from the specific nature of digital traces. Forensic timelines contain a high proportion of routine system events, whilst security incidents account for only a small fraction of the available records. In addition to this significant data imbalance, another problem was the lack of well-labelled and sufficiently diverse datasets needed to create a robust model capable of generalising to new types of incidents.

Based on the findings, a pre-trained large language model was selected as the final analytical approach. The open-source models Qwen[20] and Gemma[21] were tested as part of the experiments, with Qwen being selected as the final model due to its more stable and accurate results when interpreting forensic timelines. The model was utilised via inference without additional training on the analysed data. To process larger time windows, a context window of 65,536 tokens was used, which enabled the analysis of larger blocks of events at once.

2.2.1 Data sets and their pre-processing

To validate the proposed approach, the ‘The Case of the Stolen Szechuan Sauce’ (SSS) dataset[22] was used, which presents a realistic scenario of a Windows operating system environment being compromised. The dataset contains disk images of two devices:

- Domain Controller,
- Victim Desktop.

An advantage of this dataset is the availability of manually labelled events related to the security incident, which enable an objective assessment of the effectiveness of the proposed solution.

The first step in the processing was to create a unified chronological timeline using the **Plaso** tool[23] . This framework enables the extraction of digital traces from various sources within the Windows operating system, such as system registries, the file system, system events and other forensic artefacts.

The output of the Plaso tool contains a large number of attributes, many of which constitute technical metadata or redundant information that is not directly relevant to the interpretation of security events. For this reason, a selection of attributes was carried out with the aim of creating a compact representation suitable for processing by a language model.

The following steps were carried out during pre-processing:

- creation of a unified '**datetime**' attribute by merging the original date and time values,
- removal of irrelevant metadata,
- selection of significant attributes representing the nature of the event (sourcetype, type, MACB, user, host, desc),
- truncating the '**desc**' text attribute to a defined length in order to optimise the use of the model's context window,
- dividing the timeline into smaller blocks of events that can be processed within a single input for the language model.

The resulting representation consisted of a structured text input containing chronologically ordered Windows events.

2.2.2 Analysis using a large language model

The proposed approach utilises a large language model as an analytical tool for interpreting digital traces. Each time window is transformed into text form and, together with contextual information, forms the input prompt for the model.

In addition to the digital traces themselves, the input may also include supplementary context describing the analysed environment or organisation, thereby providing the model with the information necessary for the correct interpretation of events. The model's output is a structured set of identified security events, including their descriptions and other characteristics.

As the outputs of large language models do not always have a consistent structure, the inference phase was followed by a phase of validation and normalisation of the results. This involved checking the output format, removing invalid or duplicate records, and standardising the names of individual categories. The aim was to ensure a consistent and unambiguous output suitable for further analysis.

2.2.3 Evaluation of success

The success of the proposed approach was evaluated by comparing the identified events with manually labelled security incidents present in the dataset.

Standard binary classification metrics were used for the evaluation:

- **Precision**, which expresses the proportion of correctly identified security events out of all events labelled as suspicious by the model.
- **Recall**, which indicates the proportion of correctly identified security events out of all actual security events.
- **F1-score**, representing the harmonic mean of the Precision and Recall metrics, which provides a balanced assessment of the model's performance on significantly imbalanced data.

During the evaluation, time intervals containing a security incident and time intervals representing exclusively normal system operation were analysed separately. This approach made it possible to independently assess the model's ability to identify security events whilst also monitoring the number of false positive detections during normal operation.

2.2.4 Interpretability of results

One of the main advantages of the proposed approach is the high interpretability of the results. The large language model not only flags suspicious events but also generates a textual description and explanation of them, which can assist a digital forensics analyst in their further investigation.

The result of the analysis is therefore not just a list of identified digital traces, but also a contextual interpretation of them, which enables a quicker understanding of the course of a security incident and supports the analyst's decision-making process.

Unlike traditional classification models, where the result is typically just a probability score or class, the LLM provides a natural language description of the analysed behaviour. This description includes the context of the event, its potential link to a security incident, and the rationale for selecting specific digital traces. This form of output reduces the burden of manually reviewing extensive timelines and enables the analyst to verify relevant events more quickly.

The aim of this study was to explore the potential for using modern machine learning and artificial intelligence methods to identify digital traces in the context of digital forensic analysis. The main objective was not to create a universal automated system that would replace the work of an analyst, but to verify the extent to which the selected methods can support the process of analysing large and complex datasets, in which the manual search for relevant events is time-consuming and practically limited.

As part of the study, two distinct approaches were examined – hierarchical clustering and analysis using large language models. These methods represent different perspectives on the identification of digital traces. Whilst clustering methods primarily work with similarities between objects and enable the discovery of naturally occurring structures in the data, large language models utilise the ability to interpret complex textual representations and provide contextual explanations of identified events.

Hierarchical clustering has proven to be a suitable tool for analysing large datasets in which it is not possible to define in advance the number or nature of individual clusters. In the field of digital forensic analysis, this feature is particularly significant when dealing with unknown activities, where the analyst often lacks sufficient information about the environment under analysis or expected patterns of behaviour. The clusters created enable the identification of similar groups of digital traces, the detection of recurring patterns, and support for the subsequent analysis of related events.

The significance of hierarchical clustering lies primarily in its ability to reduce the complexity of the analysed data. Instead of manually reviewing individual records, the analyst can work with a smaller number of representative groups that characterise similar behaviour. Such an approach can be useful, for example, when analysing a large number of commands executed by attackers, identifying similar methods of compromise, or detecting recurring activities across different incidents.

The second approach examined involved the use of large language models to interpret digital traces. Experiments showed that modern pre-trained language models are capable of handling complex forensic timelines and identifying events that may be related to a security incident. A key feature of this approach is the ability to combine individual events into a broader context, which distinguishes it from classical machine learning algorithms, which often work only with numerical representations or explicitly defined features.

In designing an LLM-based solution, it was also demonstrated that the successful application of large language models in the field of digital forensic analysis does not depend solely on the model itself; data preparation, appropriate representation of digital traces and verification of the resulting outputs also play a significant role. Processing the original forensic artefacts into a unified chronological timeline, selecting relevant attributes and

dividing the data into manageable blocks enabled a more effective utilisation of the language model's capabilities.

The results also highlighted the importance of interpretability when applying artificial intelligence methods in digital forensic analysis. Automated methods can significantly speed up the process of identifying relevant events; however, their outputs must remain comprehensible and verifiable by experts. For this reason, the ability to explain the reasons behind the model's decisions is one of the key factors in the practical application of these technologies. In the case of clustering methods, this primarily involves analysing the properties of individual clusters and their characteristic attributes, whilst for large language models it involves generating a contextual explanation of the identified events.

Experimental validation on three independent datasets has shown that the proposed methodology is capable of identifying groups of digital traces associated with a security incident, distinguishing them from normal system behaviour, and providing the analyst with interpretable information about the nature of the identified activities. The results suggest that hierarchical clustering of time windows is a suitable tool for facilitating initial orientation within extensive forensic timelines.

The proposed approaches have demonstrated that machine learning methods and large language models can provide significant support for digital forensic analysis, particularly when working with large datasets. Their main benefit lies in their ability to reduce the amount of manual work, highlight relevant digital traces and provide the analyst with the information needed to understand the incident under investigation more quickly.

At the same time, however, these methods cannot be regarded as a complete substitute for expert judgement. The results of automated analysis must always be interpreted within the broader context of the investigation and verified using available forensic evidence. The future of artificial intelligence in digital forensic analysis therefore lies not only in improving the accuracy of models, but also in developing transparent, explainable and practical tools that will work effectively alongside expert analysts.

Bibliography

- [1] C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," *Digital Investigation*, vol. 9, pp. S69–S79, 2012.
- [2] L. F. daCruz-Nassif and E. R. Hruschka, "Document clustering for forensic analysis: An approach for improving computer inspection," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 46–54, 2012.
- [3] X. Du and M. Scanlon, "Methodology for the automated metadata-based classification of incriminating digital forensic artefacts," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pp. 1–8, 2019.
- [4] M. AbuJarour, R. B. Tareaf, and V. Hafner, "Using machine learning to prioritise forensic artefacts in email evidence," in the *2025 International Conference on Information Networking (ICOIN)*, pp. 201–206, 2025.
- [5] H. Studiawan, C. Payne, and F. Sohel, "Graph clustering and anomaly detection of access control logs for forensic purposes," **Digital Investigation**, vol. 21, pp. 76–87, 2017.
- [6] E. Marková, P. Sokol, and K. Kováčová, "Detection of relevant digital evidence in forensic timelines," in the *14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI) 2022*, pp. 1–7, 2022.
- [7] H. Studiawan and F. Sohel, "Anomaly detection in a forensic timeline using deep autoencoders," *Journal of Information Security and Applications*, vol. 63, p. 103002, 2021.
- [8] F. Breiteringer, H. Studiawan, and C. Hargreaves, "SOK: Timeline-based event reconstruction for digital forensics: Terminology, methodology, and current challenges," *Forensic Science International: Digital Investigation*, vol. 53, p. 301932, 2025.
- [9] R. Adila, H. Studiawan, and F. Breiteringer, "Forensic event reconstruction with process mining," *IEEE Access*, 2026.
- [10] L. Schmidt and S. Schinzel, "Towards automated and robust forensic event reconstruction," in the *2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 200–210, 2025.
- [11] P. Sokol, Ľ. Antoni, O. Krídlo, E. Marková, K. Kováčová, and S. Krajčí, "A formal concept analysis approach to understanding relationships in digital evidence," *International Journal of Approximate Reasoning*, vol. 159, p. 108940, 2023.
- [12] H. Studiawan, F. Breiteringer, and M. Scanlon, "Towards a standardised methodology and dataset for evaluating LLM-based digital forensic timeline analysis," *Forensic Science International: Digital Investigation*, vol. 54, p. 301982, 2025.

- [13] Y. Wei, K.-P. Chow, and S.-M. Yiu, "Insider threat prediction based on an unsupervised anomaly detection scheme for proactive forensic investigation," *Forensic Science International: Digital Investigation*, vol. 38, p. 301126, 2021.
- [14] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.
- [15] M. R. Ackermann, J. Blömer, D. Kuntze, and C. Sohler, "Analysis of agglomerative clustering," *Algorithmica*, vol. 69, no. 1, pp. 184–215, 2014.
- [16] Magnet CTF 2019 Windows Desktop, 2019, [online 12 May 2026]. Available at: <https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2019%20CTF%20-%20Windows-Desktop.zip>.
- [17] Magnet CTF 2022 Windows, 2022, [online 12 May 2026]. Available at: <https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2022%20CTF%20-%20Windows.zip>.
- [18] Data LeakageCase, [online 12 May 2026]. Available at: https://cfreds-archive.nist.gov/data_leakage_case/data-leakage-case.html.
- [19] DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*, 2019 [online 25 September 2025]. Available at: [10.48550/arXiv.1810.04805](https://arxiv.org/abs/1810.04805).arXiv:1810.04805 [cs]
- [20] QWEN TEAM. Qwen3.6-35B-A3B. 2026. [online 12 May 2026]. Available at: <https://qwen.ai/blog?id=qwen3.6-35b-a3b>.
- [21] GOOGLE DEEPMIND. Gemma 4. 2026. [online 12 May 2026]. Available at: <https://deepmind.google/models/gemma/gemma-4/>.
- [22] Case 001 – the stolen Szechuan sauce, 2020, [online 12 May 2026]. Available at: <https://dfirmadness.com/the-stolen-szechuan-sauce/>.
- [23] Plaso Team. (n.d.). *Plaso documentation*. [online 12 May 2026]. Available at: <https://plaso.readthedocs.io/en/latest/>.