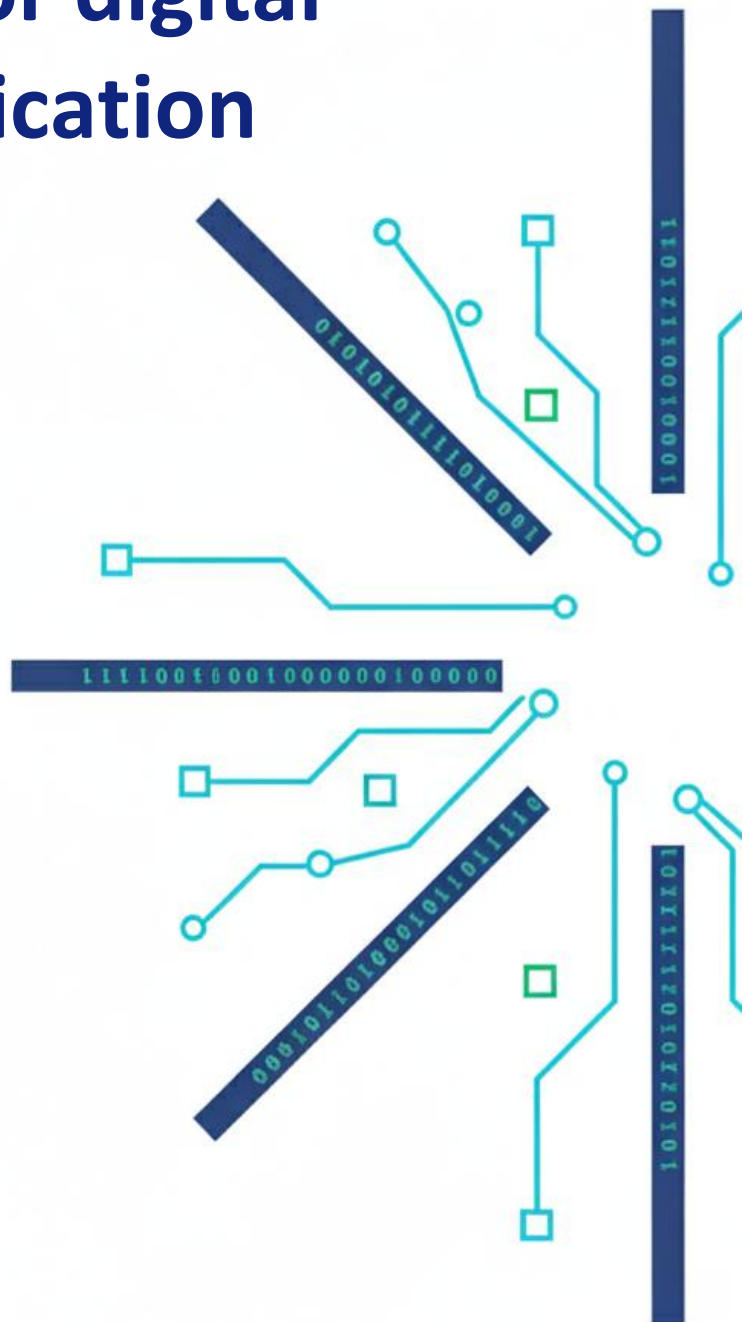


D19 – Method for digital evidence identification



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 Theoretical Introduction	5
1.1 Anomaly detection	6
1.2 Unsupervised anomaly detection	6
1.3 Types of anomalies.....	6
1.3.1 Point anomalies	6
1.3.2 Contextual anomalies	7
1.3.3 Collective anomalies.....	7
1.4 Categorisation of anomaly detection methods	7
2 Application of Selected Methods	9
2.1 Anomaly detection using unsupervised machine learning methods	9
2.2 Data sets and pre-processing.....	9
2.3 Anomaly detection methods used	10
2.3.1 Principal Component Analysis (PCA)	11
2.3.2 Local Outlier Factor (LOF).....	11
2.3.3 Subspace Outlier Detection (SOD)	11
2.3.4 Empirical-Cumulative-Distribution-Based Outlier Detection (ECOD)	12
2.3.5 Copula-Based Outlier Detection (COPD)	12
2.3.6 Isolation Forest.....	12
2.3.7 Isolation-based Nearest Neighbour Ensemble (INNE)	12
2.3.8 Lightweight On-line Detector of Anomalies (LODA)	12
2.4 Experiment setup and comparison of anomaly detection methods	13
2.4.1 Experimental setup	13
2.4.2 Evaluation of Detection Performance	14
2.4.3 Comparison of computational complexity	14
2.4.4 Interpretation of results	15
Summary	19
Bibliography.....	21
Appendices.....	23

Project description

The project **Automation of Digital Forensic Analysis and Incident Response** (hereinafter referred to as 'ADFIR') 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 effective 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 concepts of data anomaly detection, which enable isolated forensic traces to be transformed 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 various 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 Theoretical Introduction

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.

A significant part of the output involves the identification of anomalies. However, before detection can take place, it is essential to clearly define the concept of an anomaly in the context of the data being analysed. Behaviour that appears unusual from the perspective of a global dataset may represent legitimate activity by a specific user or system, whilst, conversely, normal behaviour may, in a particular context, indicate a security incident. For this reason, it is not possible to define an anomaly solely as a statistical deviation from normal behaviour; it is also necessary to take into account the nature of the analysed environment and the significance of individual digital traces.

As manually labelling all available datasets would be both time-consuming and impractical, unsupervised learning-based anomaly detection methods were employed, which enable the identification of deviations without the need for predefined classes. These methods are particularly suitable in the field of digital forensic analysis, where the available data often consists predominantly of legitimate events and only a small number of potentially significant or malicious activities.

As part of the analysis, several approaches to anomaly detection were examined, each representing different principles for identifying deviations in the data. The evaluation covered linear models based on transforming data into a more suitable space, methods based on the local similarity of objects, probabilistic models utilising the distributional properties of the data, and ensemble methods combining multiple detection approaches. The aim was to verify the ability of individual methods to identify non-standard events in complex and heterogeneous data, whilst also comparing their suitability for the needs of digital forensic analysis.

For this reason, thorough data pre-processing was crucial; this is described in detail in the ADFIR-D13-KPB2-02 output 'Method for dataset creation', which ensured that the heterogeneity, quality and complexity of the analysed data were preserved. Pre-processing involved adapting the input data into a format suitable for individual algorithms, removing irrelevant information, and creating a representation that enables the effective analysis of individual characteristics of digital traces.

A key aspect of using machine learning methods in the field of digital forensic analysis is not only the ability to identify 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 analysing the properties and characteristics that contribute most significantly to the flagging of events as potentially anomalous.

1.1 Anomaly detection

Anomaly detection is a critical component of digital forensic analysis. Its main objective is to identify patterns, events or records in the data that do not correspond to the expected (normal) behaviour of the system. These non-standard occurrences are referred to as anomalies or outliers and, in a forensic context, often represent traces of a cyber incident, unauthorised access or data manipulation[1] .

1.2 Unsupervised anomaly detection

In real-world forensic scenarios, we rarely have access to pre-labelled data that would precisely define what constitutes an attack and what constitutes normal operation. Therefore, **unsupervised anomaly detection** is primarily used[2] .

This approach works without any prior knowledge of anomalies. It is based on two fundamental assumptions:

1. The vast majority of data in the dataset represents normal, legitimate system behaviour.
2. Anomalies are statistically very rare and differ significantly from normal data in terms of their characteristics[1] ,[2] .

Unsupervised algorithms analyse the internal structure of the data and look for clusters with high density (the normal state). Anything that lies outside these dense clusters, or belongs to clusters that are too small and isolated, is flagged as an anomaly[2] .

1.3 Types of anomalies

In forensic practice and the analysis of digital traces, we distinguish between three basic categories of anomalies[1] :

1.3.1 Point anomalies

This is the simplest type of anomaly. It occurs when **a single specific event or data point** differs significantly from all other data in the entire dataset.

- *Example* : A file of extremely large size in a folder that usually contains only small text logs, or a one-off attempt to transfer a huge volume of data from a server.

1.3.2 Contextual anomalies

A data point is considered anomalous only **in relation to a specific context**, although in a different context it would be entirely normal. When identifying anomalies, it is crucial to evaluate not only the value itself (behavioural attribute) but also the context (e.g. time, location).

- *Example*: A user logging into the system at 2.00 pm on a working day is normal. The same user logging in from the same IP address, but at 3.00 am on Christmas Eve, exhibits strong contextual anomalies.

1.3.3 Collective anomalies

These occur when **a sequence or group of events** collectively exhibits anomalous behaviour, even though each individual event on its own would be considered entirely normal. The anomaly arises only from their interconnection and structure in time or space.

- *Example*: A single failed login attempt is a common user error (not an anomaly). However, 500 failed login attempts within 30 seconds from different accounts constitute a collective anomaly that clearly indicates a brute-force attack.

1.4 Categorisation of anomaly detection methods

Unsupervised anomaly detection methods can be divided into four main categories based on their mathematical and algorithmic principles[3] ,[4] :

- **Linear models for outlier detection (Linear Models)**: These approaches assume that normal data lie in a low-dimensional linear subspace. Anomalies are identified based on their distance from this subspace[5] .
 - *Examples*: Principal Component Analysis (PCA), One-Class Support Vector Machines using Stochastic Gradient Descent (SGDOCSVM).
- **Proximity-Based Models**: These methods analyse the spatial distribution of the data. They assume that anomalies are located in isolated, low-density regions, far from their nearest neighbours.
 - *Examples*: Local Outlier Factor (LOF)[6] , Subspace Outlier Detection (SOD)[7] , Rotation-based Outlier Detection (ROD)[8] .
- **Probabilistic models**: These approaches model outliers based on an estimate of the probability density distribution of the data. Points with an extremely low probability of occurrence within a given distribution are identified as outliers.

- *Examples:* Empirical-Cumulative-distribution-based Outlier Detection (ECOD)[9] , Copula-Based Outlier Detection (COPOD)[10] .
- **Ensembles and combined frameworks (Outlier Ensembles):** These utilise a combination of multiple models or techniques (known as ensembles) to achieve more robust results and reduce the rate of false positive detections. They often operate on the principle of isolating outliers from the rest of the data.
 - *Examples:* Isolation Forest, Isolation-based Anomaly Detection Using Nearest-Neighbour Ensembles (INNE)[11] , Lightweight on-line detector of anomalies (LODA)[12] .

2 Application of Selected Methods

This chapter presents the practical steps involved in applying 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.

2.1 Anomaly detection using unsupervised machine learning methods

An important aspect of identifying digital traces in the field of digital forensic analysis is the detection of unusual or suspicious behaviour within large datasets. However, such behaviour cannot always be unambiguously defined, as an event that appears anomalous from the perspective of the entire dataset may represent legitimate activity by a specific system or user. For this reason, unsupervised learning methods were employed, as they enable the identification of outliers without the need for manually labelled data.

Outlier detection is the process of identifying observations that differ significantly from other objects in the analysed space. Unlike classification methods, where individual classes must be defined in advance, outlier detection methods work solely with the characteristics of the data itself and seek out objects exhibiting non-standard behaviour.

This thesis examined a number of approaches to anomaly detection, which can be divided into four main categories:

- **linear anomaly detection methods,**
- **methods based on the similarity and distance between objects,**
- **probabilistic methods,**
- **ensemble methods based on a combination of multiple detectors.**

2.2 Data sets and pre-processing

The same datasets were used for the anomaly detection experiments as for the hierarchical clustering method. The analysed datasets consisted of digital traces obtained from the Windows operating system, which were transformed into a structured representation using the data pre-processing procedure described in the previous chapter, or in the output at (ADFIR-D13-KPB2-02-Method for dataset creation (EN)).

The basic unit of analysis was a time window representing the state of the system during a defined time interval. Individual operating system events were aggregated using selected aggregation functions, resulting in a matrix of attributes describing the characteristic properties of individual time windows.

Each time window was represented by a set of attributes obtained from forensic artefacts extracted using the Plaso tool. The attributes contained information on various types of system activity, such as file system events, system logs, user activities or operations performed.

As the available datasets did not contain a complete manual annotation of all anomalous events, it was not possible to use classical supervised learning methods. For this reason, only unsupervised learning methods were selected, which enable the identification of potentially suspicious objects directly on the basis of data characteristics.

2.3 Anomaly detection methods used

The libraries , **scikit-learn**[13] and **PyOD**[3] were used to implement the individual algorithms. The scikit-learn library provides implementations of several standard machine learning methods, including dimensionality reduction and outlier detection. PyOD (*Python Outlier Detection*) is a specialised library designed to detect anomalies in multidimensional data and contains a wide range of available detectors.

The following methods were tested as part of the experiments:

Method	Category	Main principle
PCA	Linear methods	Identification of deviations in principal component space
LOF	Proximity-based methods	Comparison of an object's local density with its surroundings
ECOD	Probabilistic methods	Analysis of empirical distribution functions
COPOD	Probabilistic methods	Detection based on copula distribution
Isolation Forest	Ensemble methods	Object isolation using random space partitioning
OTHERS	Ensemble methods	Detection based on isolation using nearest neighbours
LODA	Ensemble methods	Online detection using a combination of simple histogram models

Table 1: Anomaly detection methods

The methods were selected with the aim of covering the main categories of unsupervised anomaly detection. Linear, density-based, probabilistic and ensemble approaches were included in the experiments to enable a comparison of their behaviour on the same forensic datasets and to identify suitable characteristics of each category for digital forensic analysis.

2.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear transformation method that reduces the original attribute space to a smaller number of principal components. These components preserve the most significant variability of the original data.

In anomaly detection, PCA assumes that normal objects lie within the space defined by the principal components, whilst anomalous objects exhibit a greater deviation from this representation. The magnitude of this deviation can then be used as an anomaly score.

The advantage of PCA is its low computational complexity and suitability for high-dimensional data. A disadvantage is the assumption of linear relationships between attributes, which may not always hold true for complex forensic data.

2.3.2 Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) is a non-parametric method based on the analysis of the local density of objects. Each object is compared with its neighbourhood, defined as a set of nearest neighbours.

Anomalous objects typically have a significantly lower local density than their neighbours, which leads to a higher LOF score. The method is particularly suitable in cases where normal objects are found in areas of varying density.

The following parameters were monitored during the experiments:

- number of neighbours,
- the distance metric used,
- the assumed proportion of anomalies (*contamination*).

2.3.3 Subspace Outlier Detection (SOD)

The Subspace Outlier Detection method identifies outliers in subspaces of data. Unlike methods that operate on the full attribute space, it assumes that an outlier need not be visible in all dimensions, but only in a specific subset of attributes.

This approach is particularly suitable for high-dimensional datasets, where only a portion of the available features may be significant.

2.3.4 Empirical-Cumulative-Distribution-Based Outlier Detection (ECOD)

ECOD is a non-parametric method based on empirical cumulative distribution functions. The method does not require model training or complex parameter tuning, but assesses how likely the occurrence of a particular attribute value is given the entire data distribution.

An advantage of ECOD is its high interpretability, as the resulting score is directly related to the deviation of individual attributes.

2.3.5 Copula-Based Outlier Detection (COPOD)

COPOD is a deterministic method utilising the statistical properties of copulas and empirical distribution functions. Like ECOD, it requires neither a training phase nor hyperparameter optimisation.

The advantage of this method is the stability of the results and low computational complexity, which allows it to be used even with large datasets.

2.3.6 Isolation Forest

Isolation Forest is one of the most widely used methods for detecting anomalies in high-dimensional data. It is based on the principle that anomalous objects can be isolated more easily than normal objects.

The method generates a number of random decision trees, with objects that are isolated via shorter paths within the trees receiving higher anomaly scores.

2.3.7 Isolation-based Nearest Neighbour Ensemble (INNE)

INNE extends the principle of object isolation by utilising information about nearest neighbours. Compared to traditional distance-based methods, it is better suited to large and high-dimensional datasets, as it is less computationally intensive.

2.3.8 Lightweight On-line Detector of Anomalies (LODA)

LODA is an ensemble method based on a combination of several simple histogram models. It is designed primarily for the rapid analysis of large data streams and can also handle incomplete data.

The advantage of this method is its low memory and computational complexity, making it suitable for large-scale analytical tasks.

2.4 Experiment setup and comparison of anomaly detection methods

Based on the categories outlined above, the individual methods were further compared in terms of their parameters, computational complexity and the results achieved. A complete overview of the experimentally tested settings is provided in Appendix A, whilst the main characteristics of the individual methods are summarised in a separate table.

2.4.1 Experimental setup

To validate the proposed approach, experiments were conducted on prepared forensic datasets containing digital traces obtained from the Windows operating system. The individual algorithms were tested with various hyperparameter configurations, with particular attention paid to the ‘contamination’ parameter, which represents the estimated proportion of anomalous objects in the dataset. For methods based on local similarity, sensitivity to the number of neighbours and the distance metric used was also examined. For ensemble methods, the impact of the number of models created on the stability of the results was analysed.

The aim of the experiments was to identify settings that enable the effective differentiation of potentially suspicious events from normal system behaviour, whilst maintaining a reasonable computational cost of processing.

One of the parameters examined was **contamination**, which represents the estimated proportion of anomalous objects in the analysed dataset. This parameter is particularly significant in anomaly detection within the field of digital forensic analysis, where security-relevant events constitute only a small proportion of all available records. Incorrect setting of this parameter can lead either to the over-flagging of legitimate events as anomalous, or, conversely, to the overlooking of suspicious activities.

For each algorithm, in addition to the common ‘contamination’ parameter, specific hyperparameters influencing the behaviour of the particular method were also examined. For the **Local Outlier Factor** (LOF) method, the focus was primarily on the number of neighbours – **n_neighbours** – and the **distance metric** used, which determine how the local neighbourhood of objects is evaluated. For isolation-based methods, such as **Isolation Forest** and **INNE**, the number of models created (**n_estimators**) was examined, as this influences the stability of the resulting anomaly score. For methods working with subspaces, such as **Subspace Outlier Detection** (SOD), the parameters determining the selection of reference objects and the sensitivity of detection in individual dimensions were evaluated.

2.4.2 Evaluation of Detection Performance

The performance of individual methods was assessed by comparing the identified anomalous objects with reference events obtained through manual analysis of the datasets. Given the nature of digital forensic data, the aim was not to create a security incident classifier, but to identify time windows and events exhibiting non-standard behaviour.

The results showed that individual algorithms achieved varying levels of success depending on the nature of the data. Whilst some methods identified a larger number of potential anomalies, others provided more conservative results with a lower number of false positives.

The comparison also included the time complexity of the algorithms, which provides a theoretical estimate of their scalability with respect to the size of the input data. In the table, time complexity is expressed using *O-notation*, where the variable n denotes the number of objects analysed and d denotes the number of attributes in the data representation. This metric is particularly important for forensic datasets, which may contain a large number of events and require methods capable of processing extensive data within a reasonable time.

In addition to the theoretical computational complexity, the actual running times of the individual algorithms when processing the analysed dataset were also recorded. The measured values allow for a comparison of the practical usability of the individual methods, as an algorithm with low theoretical complexity may not always be the fastest in a specific implementation. The resulting execution time is influenced not only by the algorithm itself, but also by the library used, implementation optimisations, the size of the data, and the number of parameters tested.

2.4.3 Comparison of computational complexity

Another significant factor in selecting an anomaly detection method is the time required to process large datasets. For this reason, both the theoretical time complexity of the algorithms and the actual runtime of the experiments were monitored.

The probabilistic methods ECOD and COPOD exhibited low computational complexity and stable behaviour without the need for extensive parameter optimisation. In contrast, methods based on local object similarity or the creation of ensemble models required greater computational resources, but were able to detect even more complex forms of deviation in the data.

The performance of the individual methods was assessed based on the number of correctly identified anomalous objects compared with reference values obtained through manual labelling. Given the nature of the analysed data, it was not possible to unambiguously

identify all potential anomalies; therefore, the results were interpreted as the ability of individual methods to find a known set of suspicious objects. The table therefore shows the number of correctly identified outliers and the average number of correctly detected objects in repeated experiments with different parameter configurations.

2.4.4 Interpretation of results

The detected anomalies represented events or time intervals whose characteristics differed from the system's normal behaviour. However, the mere identification of an anomaly does not automatically imply the presence of a security incident. Some of the flagged objects corresponded to legitimate, albeit less frequent, user or operating system activities.

The results therefore serve primarily as a support mechanism for the digital forensic analyst, who can subsequently focus their attention on a smaller subset of potentially significant events. The combination of multiple anomaly detection methods proved to be a suitable approach, as the individual algorithms identified different types of anomalous behaviour and provided complementary results.

When interpreting the results, it is necessary to take into account the specific nature of digital forensic analysis. The detection of an anomaly does not constitute automatic confirmation of malicious activity, but rather the identification of an event whose characteristics differ from the rest of the behaviour in the dataset. The results of the individual algorithms therefore serve primarily to assist the forensic analyst in reducing the volume of data and identifying areas requiring more detailed manual examination.

The experimental results showed that the individual algorithms identify different types of anomalous behaviour. Methods based on local object density were more sensitive to local deviations, whilst probabilistic and ensemble approaches yielded more stable results when processing large datasets. From the perspective of digital forensic analysis, it therefore appears advantageous to combine multiple detection approaches that complement one another and enable the capture of a broader spectrum of potentially significant events.

The temporal relationship between the detected anomalies and the security incident is illustrated at **Figure 1**, **Figure 2**, **Figure 3**. The visualisation shows the anomalies identified by the individual methods used, with the red band indicating the time period of the security incident. It is clear from the figure that a significant proportion of the detected anomalies overlap with the period of the incident. This result suggests that the methods used were able to identify time intervals containing behaviour related to the incident, whilst also providing the forensic analyst with a narrower set of events suitable for further investigation.

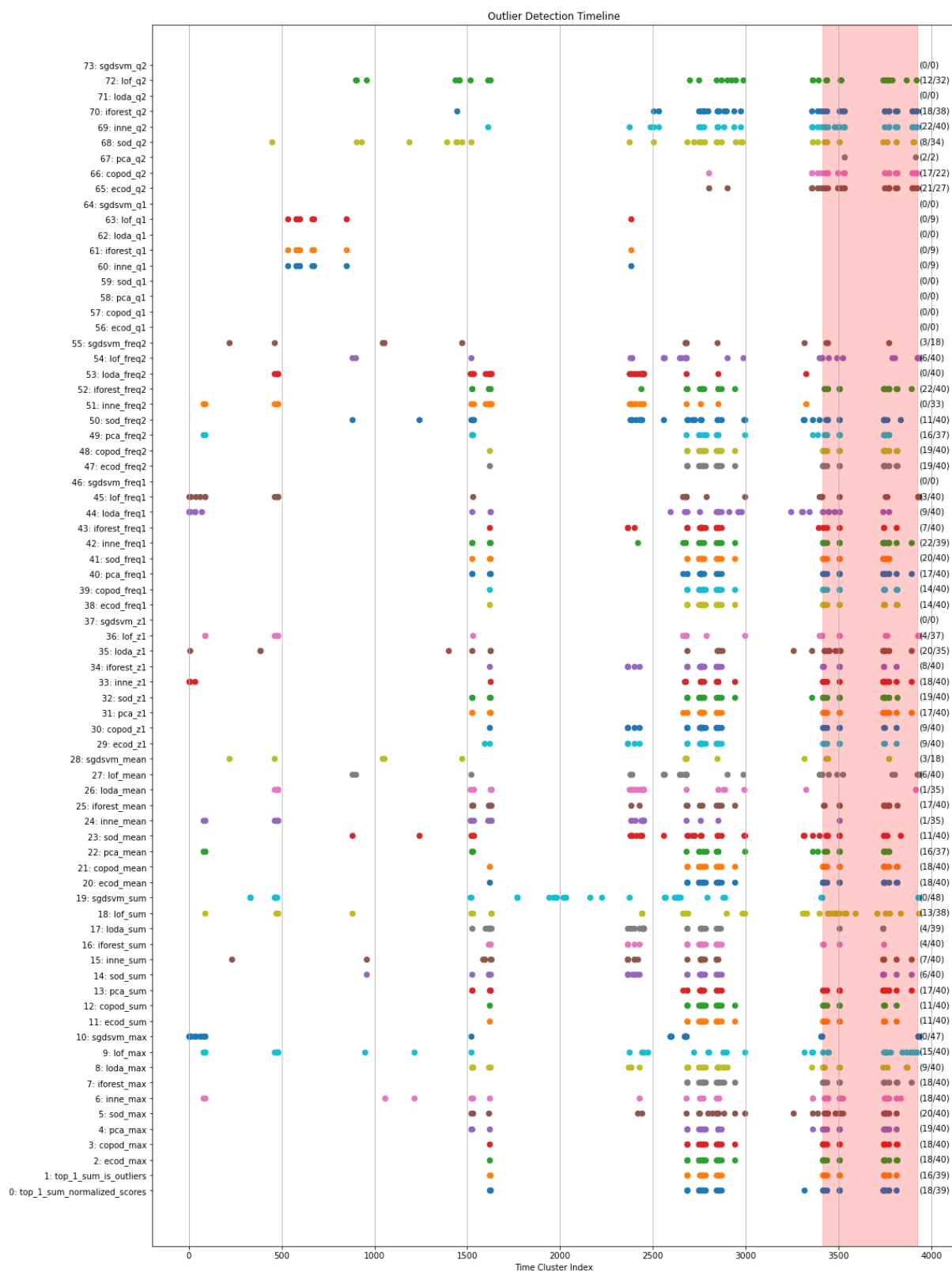


Figure 1 : Results of the methods for Stolen Szechuan Sauce DC

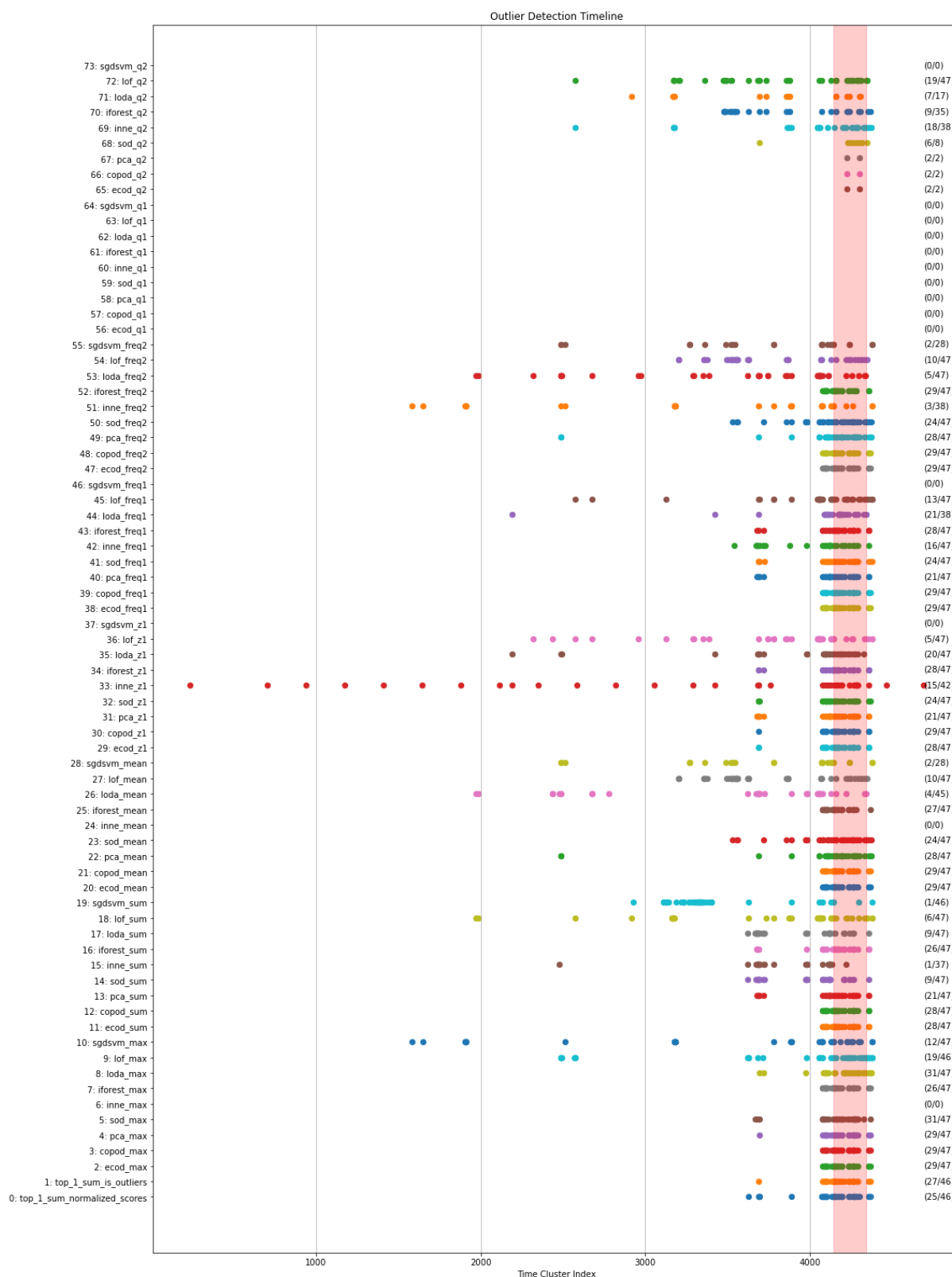


Figure 2 : Results of the methods for the 'Stolen Szechuan Sauce' desktop

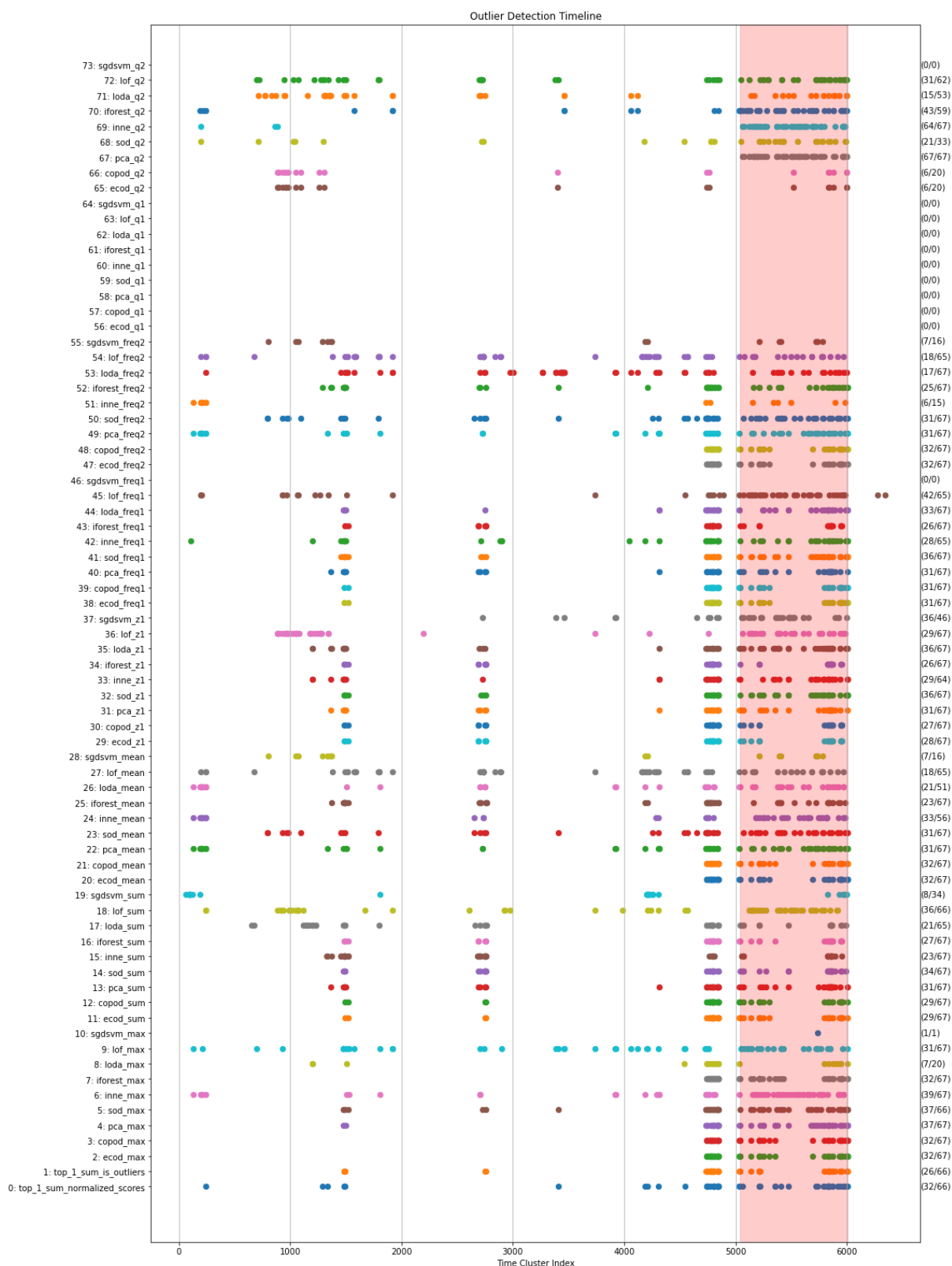


Figure 3 : Results of the methods for the Nist Data Leakage Case

Summary

This paper proposes and experimentally validates an approach to identifying potentially significant digital traces using anomaly detection methods based on unsupervised learning. The main objective was to investigate the possibilities of automated detection of anomalous behaviour in large-scale forensic datasets, where a complete manual annotation of all security-relevant events is not available.

In the first phase, the heterogeneous digital traces obtained from the Windows operating system were transformed into a representation suitable for algorithmic analysis. As individual forensic events often do not, on their own, provide sufficient context to determine their significance, the data were aggregated into time windows representing system behaviour over defined intervals. This approach made it possible to analyse not only individual events, but also broader patterns of behaviour and the interrelationships between individual attributes.

As part of the experiments, several categories of anomaly detection methods representing different analytical principles were compared. The methods tested included linear methods based on data transformation, methods utilising local object similarity, probabilistic approaches based on the distributional properties of the data, and ensemble methods utilising a combination of multiple detection models. This comparison made it possible to assess the differences between the individual approaches in terms of detection accuracy, computational complexity and practical applicability in the analysis of forensic data.

The results of the experiments showed that individual methods have different characteristics and their suitability depends on the nature of the data being analysed. Simpler and less computationally intensive methods allow for a rapid initial analysis of large datasets, whilst more complex methods are capable of identifying more intricate patterns of behaviour. For this reason, it is not possible to identify a single, universally best method for all forensic scenarios; rather, a more appropriate approach is to combine several analytical techniques according to the requirements of a specific investigation.

From the perspective of digital forensic analysis, anomaly detection is a significant tool, particularly when working with extensive timelines and large numbers of system artefacts. The output of these algorithms is not a direct confirmation of a security incident, but rather the identification of events or time periods that deviate from expected behaviour and warrant further analysis. This approach enables the forensic analyst to focus their attention on the most relevant parts of the data, thereby significantly reducing the volume of information processed manually.

The proposed approach therefore represents a supportive analytical tool that combines the capabilities of modern machine learning methods with the requirements of digital forensic practice. Its main benefit lies in its ability to automatically detect potentially significant anomalies within complex data, thereby facilitating more efficient event reconstruction, the

identification of suspicious behaviour, and the subsequent decision-making of the forensic analyst.

Bibliography

- [1] CHANDOLA, Varun, Arindam BANERJEE and Vipin KUMAR. Anomaly Detection: A Survey. *ACM Computing Surveys*. 2009, 41(3), 15:1–15:58. ISSN 0360-0300. DOI: 10.1145/1541880.1541882
- [2] AGGARWAL, Charu C. *Outlier Analysis*. 2nd ed. Cham: Springer, 2017. ISBN 978-3-319-47577-3. DOI: 10.1007/978-3-319-47577-6
- [3] ZHAO, Yue, Zainab NASRULLAH and Zheng LI. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*. 2019, 20(96), 1–7. ISSN 1533-7928.
- [4] S.J. Stolfo, F. Apap, E. Eskin, K. Heller, S. Hershkop, A. Honig, K. Svore, A comparative evaluation of two algorithms for Windows registry anomaly detection. *Journal of Computer Security* 13(4), 659–693 (2005). <https://doi.org/10.3233/JCS-2005-13403>
- [5] Shyu, Mei-Ling & Chen, Shu-Ching & Sarinnapakorn, Kanoksri & Chang, Liwu. (2003). A Novel Anomaly Detection Scheme Based on Principal Component Classifier. *Proceedings of the International Conference on Data Mining*.
- [6] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. *SIGMOD Rec.* 29, 2 (June 2000), 93–104. <https://doi.org/10.1145/335191.335388>
- [7] KRIEGL, Hans-Peter, Peer KRÖGER, Erich SCHUBERT and Arthur ZIMEK. Outlier Detection in Arbitrarily Oriented Subspaces. In: *IEEE 12th International Conference on Data Mining*. 2012, pp. 379–388. DOI: 10.1109/ICDM.2012.140.
- [8] Y. Almardeny, N. Boujnah and F. Cleary, "A Novel Outlier Detection Method for Multivariate Data," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 9, pp. 4052–4062, 1 September 2022, doi: 10.1109/TKDE.2020.3036524.
- [9] Li, Zheng; Zhao, Yue; Hu, Xiyang; Botta, Nicola; Ionescu, Cezar; and Chen, George. (2023). ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering*. 35. 12181–12193. 10.1109/TKDE.2022.3159580.
- [10] Z. Li, Y. Zhao, N. Botta, C. Ionescu and X. Hu, "COPOD: Copula-Based Outlier Detection," *2020 IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy, 2020, pp. 1118–1123, doi: 10.1109/ICDM50108.2020.00135.
- [11] Bandaragoda TR, Ting KM, Albrecht D, Liu FT, Zhu Y, Wells JR. *Isolation-based anomaly detection using nearest-neighbour ensembles. Computational Intelligence*. 2018; 34: 968–998. <https://doi.org/10.1111/coin.12156>
- [12] Pevný, T. Loda: Lightweight on-line detector of anomalies. *Mach Learn* **102**, 275–304 (2016). <https://doi.org/10.1007/s10994-015-5521-0>

- [13] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011

Appendices

ADFIR-D19-KPB4-01-02-Appendix A.xlsx