Facilitating Forensic Examinations of Multi-User Computer Environments through Session-to-Session Analysis of Internet History

David W. Gresty, Diane Gan, George Loukas and Constantinos Ierotheou

C-SAFE Centre, Dept. of Computing and Information Systems
University of Greenwich, UK
Email: D.Gresty@Greenwich.ac.uk

Abstract

This paper proposes a new approach to the forensic investigation of Internet history artefacts by aggregating the history from a recovered device into sessions and comparing those sessions to other sessions to determine whether they are one-time events or form a repetitive or habitual pattern. We describe two approaches for performing the session aggregation: fixed-length sessions and variable-length sessions. We also describe an approach for identifying repetitive pattern of life behaviour and show how such patterns can be extracted and represented as binary strings. Using the Jaccard similarity coefficient, a session-to-session comparison can be performed and the sessions can be analysed to determine to what extent a particular session is similar to any other session in the Internet history, and thus is highly likely to correspond to the same user. Experiments have been conducted using two sets of test data, where multiple users have access to the same computer. By identifying patterns of Internet usage that are unique to each user, our approach exhibits a high success rate in attributing particular sessions of the Internet history to the correct user. This can provide considerable help to a forensic investigator trying to establish which user was using the computer when a web-related crime was committed.

Keywords: Digital Forensics, World Wide Web, Session-to-Session Analysis, Context Analysis, Pattern of Life, Internet History Analysis

1. Introduction

During the course of a digital forensics examination, the investigator has a variety of locations and artefacts to search through to find the clues to show if a device was used, misused or contains the evidence required for the purpose of the investigation. In addition to the documents, pictures, media files etc. which could be the immediate target of the investigation, devices such as computers, laptops, tablets, smart phones etc., routinely have Internet connectivity, which can often also provide a treasure trove of investigative clues to how the device was used.

The Internet history is an ordered list of artefacts that contain a date, time and Universal Resource Locator (URL) address of websites or resources that were accessed. The history artefacts show the experienced investigator the types of websites that were accessed and the times of day that the users were active; they provide information about the email and social media contacts that a user has; they can show files, movies and pictures that have been downloaded. The Internet history is where the investigator gets to see the terms submitted to search engines, the poor spelling and the languages that the users speak.

Analysis of Internet history artefacts is however time consuming and to-date an ad hoc process. The artefacts may be few in number due to the small size of the storage medium, incomplete due to normal overwriting actions or 'private browsing' anti-forensics or even be quite extensive. Each of these challenges do not detract from the importance of the investigative clues contained within the Internet history artefacts. However, they do dictate whether the Internet artefacts are only usable by the investigator as clues to get a sense of how the device was used, or can be presented as useful evidence in their own right at a court or tribunal.

Above all, the Internet history artefacts show a user, an actual person, interacting with the device. Such interaction, shows the mental component of an action, the *mens rea*, of the person at the keyboard. Brenner et al. [2] highlight the R v Schofield case from the United Kingdom, where the prosecution was forced to dismiss charges for possession of unlawful pictures because Trojan Horse software was located on the defendant's computer and ultimately the analysis had not established responsibility for the creation of the unlawful pictures resting with the defendant, or indeed any actual person.

For the forensic investigator it is difficult to show the intent of the user of the system without placing the artefacts into contextual ordering. In the above case, the unlawful pictures were considered in isolation and consequently the intent of a user had not been distinguished from that of the activity of a Trojan Horse program. However, if for example in that case there were other artefacts showing search terms submitted to a web browser before the pictures were downloaded or link files after the download that showed the access of the pictures, then despite the fact that the Internet history and the link files were different artefacts to the pictures, we would see a timeline of artefacts that an analyst could contextually order. If downloaded pictures on Schofield's system were always preceded by search artefacts and followed by link files showing access this would also form a pattern, and if the pattern could be observed, occurring over a variety of times, then the existence of repetitive behaviour could be established.

1.1. One-Time and Repetitive Patterns

We propose in this paper that activity on a computing device, and specifically within the Internet history for such a device, can be classified as either one-time events or repetitive events:

- A One-time event could be a single event such as the moment malware was executed, or a short series of events that are never repeated such as someone searching the Internet for the phrase "how to make a bomb" then proceeding to view a number of websites that are relevant to the search term, but no subsequent search or similar web page access is located within the Internet history.
- Repetitive events show habitual patterns. Therefore, to be considered a repetitive event, an event must occur and re-occur at least at one other point. Repetitive events are temporally ordered sequences or clusters of activity within a temporal proximity to each other. Sequential patterns are such that A, B and C occur, in that order, within a timeframe. Temporal Clusters are where A, B and C may occur in any order, combination or repetition within a timeframe. For example, ACBAB.

Certain crimes or investigative goals lend themselves to the identification and analysis of repetitive or habitual behaviours, such as the accessing of indecent material or 'grooming' types of offences. Wherever there is a concern about who was the operator of a device at a particular point in time, even if that particular point in time is a one-time activity, such as the sending of an inappropriate email, if the one-time event is in close temporal proximity to a repetitive pattern then an investigator may be able to demonstrate the likeliness that the user at the time of interest is the same user at a number of other times, which can refute the "it wasn't me" defence, as there is certainly the appearance of a regular user of the device operating it at that time.

Within this paper we discuss related work to the analysis of Internet history artefacts and the profiling of digital devices. We describe our approach of aggregating the Internet artefacts into groups and then show how these can be broken into components which allow the aggregate groups to be compared to each other to see if there is overlap in the membership. The experimental section of this paper describes two different problems: finding the most effective time value for the aggregation into sessions and testing the effectiveness of the session-to-session comparison. As this paper reports on an ongoing research project, our evaluation and conclusions review the encouraging results from the experiments and highlight possible techniques to improve the performance of our approach.

2. Related work

One of the first attempts for a tool for forensic timeline analysis was Zeitline, introduced by Buchholz and Falk in 2005 [3]. Its purpose was to reconstruct artefacts and enable an investigator to create complex events, using searching and filtering to populate and analyse timelines. Different applications and different operating systems leave behind footprints of their activity. The approach by Khan and Wakeman [16] is to determine the footprint of applications on a system based upon the typical artefacts that are created in normal usage. These features are then used to train a neural network which could be used during a forensic examination to attempt to reconstruct a timeline of events concerning when applications were used.

In 2009, the Cyber Forensic TimeLab (CFTL) tool was developed by Olsson and Boldt in [21]. CFTL can parse a hard drive for known predefined artefacts to produce a histogram timeline. It does not automatically analyse the artefacts, but requires the analyst to make a visual correlation of different timelines overlaid to display clusters.

Another tool for forensic investigations, log2timeline, was reported in [8]. this tool creates a super-timeline by placing all the information into a monolithic list which can then be processed. This approach was endorsed by Carbone and Bean in the review of timeline creation utilities, but in their view too many irrelevant files are included [4]. Hargreaves and Patterson have developed a tool to reconstruct high-level events from low-level activity using temporal proximity pattern matching [11].

The cause and effect nature of event reconstruction has been studied, recently James and Gladyshev [15] have defined action instances, a state transition model where an action produces a trace. If traces can be identified, then actions can be implied because of the causal nature of certain state transitions on computer systems. In 2014, Zeitline was brought back to the forefront and enhanced with new features added by Inglot and Liu [12]. Chabot et al. [5] use knowledge management, semantic web and data mining to build a theoretical model, which they have called SADFC (Semantic Analysis of Digital Forensic Cases) . They propose that once implemented their tool will analyse the events and then build and analyse a graphical representation of the timeline. Khatik and Choudhary have developed a timeline visualization tool [17], which integrates log files from web servers, searches for an activity of interest and uses this to reconstruct the time line and finally generates a report which can be used in court.

The above timeline analysis tools have provided real benefits to forensic investigators. However, they are limited to the identification and presentation of known patterns of events. Interpretation of events and the identification of unknown events can be seen in Marrington's computer profiling [19], as well as the statistical clustering on file systems proposed by Sven et al. [24]. Their approach presupposes no prior knowledge of the system and aims to identify what applications and files are closely associated in time. Gresty et al. [7] used Principal Component Analysis (PCA) to simplify Internet history timelines from a large number of possible components, to a smaller number that was more accessible for a human investigator to review.

Outside of the area of traditional digital forensics investigations, there is interesting research into event reconstruction, management and display. Kiernan and Terzi [18] asserted that large event sequences must be reduced and simplified to view, whilst at the same time giving a global view of the activity to allow suspicious activity to be detected. Examples of this problem domain would be resource management, database optimisation. The authors propose techniques for the analysis of large-size audit logs which need to be digested and displayed to an investigator. Eagle and Pentland [6] assert that a person has structures, routines and patterns of behaviour, which when temporally, spatially and even socially contextualised can be easily identified. The authors term these underlying principal component-like behaviours as *eigenbehaviours*. Ye et al. [26] propose a notation called life pattern normal form (LP-normal form) and a life pattern framework to determine and mine location-based data about individuals and their mobile computing habits. The authors of this paper propose that the patterns refer to significant places in an individuals daily life, but these must be extracted from raw GPS data, using "stay point" detection and clustering. Schaefer et al. [23] describes event sequences and makes some notable distinctions between the time-synchronous events, and between aggregate events. The authors present different ways to visualise clusters of events, gaps and indeed show representations which are not timelines, but only event information. The approach by Al Awawdeh et al. [1] is a real-time agent for recording data as it happens rather than post-mortem style forensics. The authors discuss the problem of verbosity, which is the issue that unimportant details can be over-reported in logs and salient details are not given adequate prominence even though they are reported.

Hamid et al. [9] describe events as the interaction between "animate and inanimate objects" and highlight that the area of activity discovery is for the identification of repetitious patterns within sequences of data. The authors show that sequences of behaviour can show classes of activity, such as the sensors within a home showing someone moving from the kitchen to the stairway etc. Minnen et al. [20] describes *motifs* as sub-sequences within a longer sequence of data that have high similarity but notes that the problem with motif discovery is that the length, shape, size and scale of them are not known in advance.

Research on timeline analysis in digital forensics has traditionally focused on the identification of known sequences of events or the discovery of new ones. However, by looking at sequences of events in isolation one can miss clusters of behaviour that can provide useful information for a user. This is particularly true for a user's Internet history, which often exhibits habitual behaviour but not precise motif-type sequences. In the next sections, we present a method for aggregating whole sequences of timeline artefacts into sessions and performing session-session comparisons of clusters of simplified data. We demonstrate the usefulness of this approach for situations where multiple users share the same computer environment and we evaluate its performance for different configurations.

3. Approach

3.1. Temporal Aggregation for sessions

Schaefer et al. [23] outlined two useful methods of analysing temporal data, the *sequence* approach and the *aggregate* approach:

- Temporal sequence comparison. Patterns are identified within an ordered, typically long, sequence of data.
- Aggregate-against-aggregate comparison. A collection of grouped artefacts are compared against another collection of grouped artefacts.

For the analysis of Internet history timelines, or for any meta-data context analysis within digital forensic investigations, we propose an approach where 'session' temporal aggregates are compared against other 'sessions' to identify to what extent any of the sessions contain matching members or components. Once sessions have been compared and the like-for-like sessions have been grouped together, then the process of intra-session sequence analysis can be performed to identify whether specific patterns of components appear. This session-to-session grouping itself provides significant macro-level contextual analysis about the use of a device at any time, and temporal sequential analysis after this analysis, substantially reducing the quantities of sequential data to be processed.

The selection of the session temporal aggregates is therefore fundamental. We identify two approaches to selecting sessions:

- *Fixed length sessions.* Fixed periods of time are selected in advance, for example all artefacts in a window of 30 s, 60 s, or 60 min.

- *Variable length continuous activity sessions.* If two artefacts are closer together in time than a predefined temporal threshold, they are considered to be in the same session. Otherwise, the second artefact is considered the start of the next session.

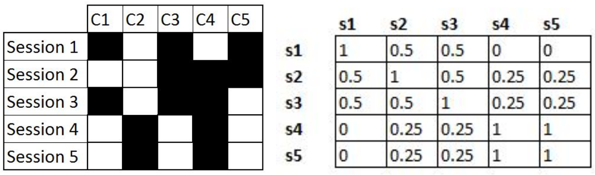
We would expect to see that the variable length approach produces a smaller number of sessions compared to the fixed-length approach (and this expectation was borne out during our experiments, figures 4 and 5), especially for sessions containing long periods of activity. The variable-length approach organically follows the activity from beginning to end of the session without artificially breaking up long sessions into smaller chunks. However, like-for-like comparison between sessions is open to some interpretation when using a variable length approach. Two sessions which could have the exact same component members and look at face value to be the same, could have very different characteristics. For example one session being two or three times longer than the other and having quite different behaviour at the beginning and end of the session.

The problem with using sessions is capturing the right amount of information that represents the 'behaviour' that is taking place at the time. The simplest example of this is where two users share the same user account on a computer, but each uses the computer for accessing very different website interests. Choosing a very large fixed-length size could easily capture the usage of the computer by both users, when there is very likely a desire to try and isolate the different access habits.

3.2. Components

Components are the events that are recorded within the sessions. Within an Internet history analysis the visit to a website domain 'Google' could be a component, or a specific sub-division may be desirable for the components such as 'google.co.uk' rather than 'google.com'. A file system analysis could make each directory a component, or each file creation, modification or access to a file type could be a component. In a wider pattern of life analysis of a home automation system the activation of lights, devices or other sensors could be recorded as components. Here, we focus on conventional Internet history components.

Components are established at analysis time and although the number of times a component is used may be recorded, for example multiple visits to the same website during the same session, this information is less significant in session-to-session analysis as it would be in a sequential analysis where the recurrence of the event 'A' in the sequence ABCAEFAG has significance. The binary condition of an event if it was or was not present during the session is sufficient for session-to-session. For example, if trying to attribute a particular session to a specific user who is known to be a motorcycle enthusiast, the number of times that a motorcycle-related website is accessed is substantially less significant than the fact that the motorcycle website was accessed at all.



	C1	C2	C3	C4	C5
Session 1	■	□	■	□	■
Session 2	□	□	■	■	■
Session 3	■	□	■	■	□
Session 4	□	■	□	■	□
Session 5	□	■	□	■	□

	s1	s2	s3	s4	s5
s1	1	0.5	0.5	0	0
s2	0.5	1	0.5	0.25	0.25
s3	0.5	0.5	1	0.25	0.25
s4	0	0.25	0.25	1	1
s5	0	0.25	0.25	1	1

Figure 1: A simple session component table with five sessions, five components and two different user behaviours (Left) and the corresponding session-to-session pairwise comparison (Right)

3.3. Plotting the Number of Sessions

We highlight two approaches to producing temporally aggregated sessions, a *fixed-length session* comprising of all the components within a fixed time window and a *variable-length session* containing all the components until a time gap between components is exceeded. The number of sessions produced by performing these aggregations can be plotted against the time in seconds for the fixed-length or variable-length threshold. Our assertion is that small values of time for the fixed-length sessions or variable-length thresholds will identify 'systemic' behaviour as the resulting sessions will be smaller slices of the timeline and frequently occurring events will be emphasised. Larger values of time however will allow a more comprehensive pattern of life as the user's repetitive behaviour will develop over the period of time and can be recognised if it recurs.

3.4. Jaccard Similarity Coefficient

By creating a binary condition for components a simple visual display can be made for the components per session as can be seen in Figure 1, which shows an example set of data containing five components (C1 to C5) and five sessions. Session 1 to 3 represent user 1, whereas sessions 4 and 5 represent user 2. Even with this example small set of data the repetitive pattern in sessions 4 and 5 and somewhat in sessions 1 and 3 stand out well visually.

The sessions, however, form a simple string which can have a pairwise distance comparison. For example session 1 [10101] and session 2 [00111] can be calculated to have a distance of 0.5 using the Jaccard similarity coefficient [13] (which is to say they share 2 of the 4 components). The following equation is for calculating the Jaccard Distance between two sets of data:

$$d_j(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{1}$$

The advantage of using Jaccard is that it only considers the components in sessions 1 and 2 that they share and does not consider C2, which is 0 in both cases. When dealing with some 4000 components all of them 0's such as seen in typical frequently used home computer Internet history, then using a pairwise comparison method such as the Hamming Distance [10], which produces a 0.999 similarity due to all the shared 0 components is undesirable and the Jaccard distance measure is substantially more useful.

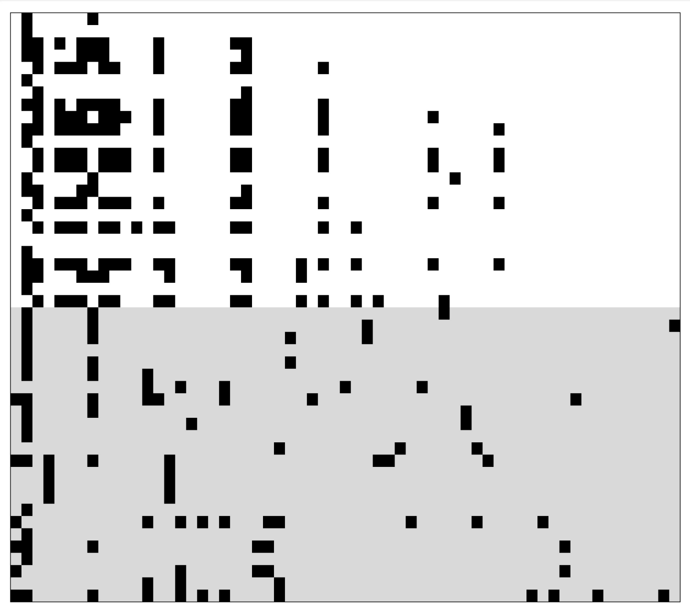


Figure 2: Sample of Components taken from the S-dataset showing 50 sessions and two users

In figure 2 we see a sample of real data displayed as a component table. The first half of the figure shows User 1 and the second half of the figure, in grey, is User 2. We can see that some components are clearly present in both users' history but the overall appearance is that these are two quite different sets of activity and that within each user there are clearly patterns of components that appear to be repeating.

3.5. Session-to-Session Patterns

Using the Jaccard similarity coefficient it can be seen in figure 1 that Session 1 and Session 4 share no components in common and consequently their similarity is 0.0. Indeed all sessions can be compared pairwise against all of the other sessions, as seen in Figure 1 (right).

Patterns are constructed by identifying groups of two or more sessions that are above a Jaccard distance measure. Although any value above 0.0 is potentially useful, the number of loosely associated sessions significantly increases as the acceptable Jaccard value is lowered. For example, at a level of 1.0, one session pattern is created: Pattern 1 = [s4 s5]. At a Jaccard level of 0.5, two session patterns are created: Pattern 1 = [s1 s2 s3], Pattern 2 = [s4 s5].

The objective of our session-to-session comparison is to identify a pair or more of sessions that belong to the same repetitive behaviour, and the inference is that this could be the same user. If the level of association is lowered sufficiently then any vaguely similar activity, potentially belonging to different users will be identified in our session-to-session pattern, and if too high a level only an exact match will be valid.

3.6. Conclusion of the Approach

We have shown in this section of the paper that there are two approaches to performing session selection and we have also discussed how to convert Internet history into components. We note that the number and frequency of components per session is not as significant within session-to-session analysis and this has allowed us to translate the presence or absence of a component within a session into a simple binary string.

The table of components shown in figure 2 derived using the principal component analysis of [7] is interesting in that it shows to an experienced analyst the presence of different types of repeating activity. However, such tables quickly become unwieldy. As such, a systematic, preferably quantitative comparison of these sessions is required. Also, it is important to consider an appropriate string comparison method. For this, we have selected the Jaccard coefficient, which deals solely with the comparison of the components that are present within the sessions, rather than a method which considers all possible components within the Internet history.

4. Experiments

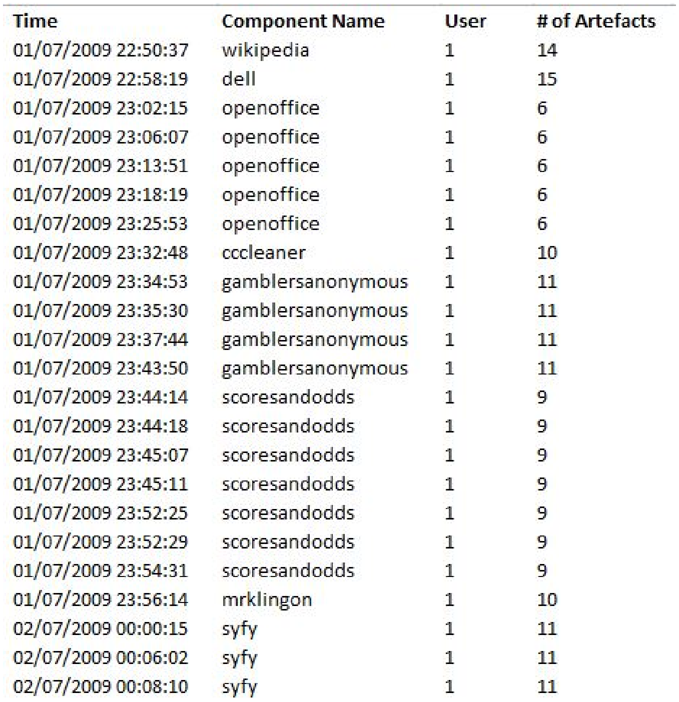
We present in this section two practical examples of our approach, and experimental tests to determine the effectiveness of two particular problems, namely the session selection using both fixed-length and variable-length methods and the session-to-session evaluation of the components at different tolerance levels for the Jaccard coefficient.

The aim of these experiments is to show that for test data where there is a known ground truth of which artefacts belong to which users, we can perform multiple variations of the experiments to identify the settings for the variables such that in the future an analyst with an unknown dataset can be confident, to a certain level of error, to have achieved the optimal session-level analysis.

4.1. Experimental Data

The data used in the experiments are Internet history timelines. The artefacts are a temporally ordered sequence of URLs. The component selection is therefore at the domain level part of the URL, rather than individual web page artefacts.

In both of the two datasets the scenario is that there is a single device and a single user account that multiple users have access to. In the one dataset, there are three possible users, and in the other dataset there are two possible users. This type of scenario, where the is a single machine with all the users having access to a single account is both realistic and not uncommon in law enforcement investigations in a domestic setting or commercial investigation setting where there is stand-alone machine with no, or poorly enforced access control.



Time	Component Name	User	# of Artefacts
01/07/2009 22:50:37	wikipedia	1	14
01/07/2009 22:58:19	dell	1	15
01/07/2009 23:02:15	openoffice	1	6
01/07/2009 23:06:07	openoffice	1	6
01/07/2009 23:13:51	openoffice	1	6
01/07/2009 23:18:19	openoffice	1	6
01/07/2009 23:25:53	openoffice	1	6
01/07/2009 23:32:48	cccleaner	1	10
01/07/2009 23:34:53	gamblersanonymous	1	11
01/07/2009 23:35:30	gamblersanonymous	1	11
01/07/2009 23:37:44	gamblersanonymous	1	11
01/07/2009 23:43:50	gamblersanonymous	1	11
01/07/2009 23:44:14	scoresandodds	1	9
01/07/2009 23:44:18	scoresandodds	1	9
01/07/2009 23:45:07	scoresandodds	1	9
01/07/2009 23:45:11	scoresandodds	1	9
01/07/2009 23:52:25	scoresandodds	1	9
01/07/2009 23:52:29	scoresandodds	1	9
01/07/2009 23:54:31	scoresandodds	1	9
01/07/2009 23:56:14	mrklingon	1	10
02/07/2009 00:00:15	syfy	1	11
02/07/2009 00:06:02	syfy	1	11
02/07/2009 00:08:10	syfy	1	11

Figure 3: Sample of processed Internet history from the S-dataset

S-Dataset

This is a single dataset constructed from three different workstations used in the M57-patents scenario of the Digital Corpora Project [25]. The resulting data is an office-based PC that has a sparsely populated Internet history for three different 'users' all with access to a single user account, but with a significant time gap between the users to ensure that no large variable-length threshold or fixed-length session could cause different user data to appear in the same session. This data simulates the use of a shared computer by three different staff members all working on different shift patterns, where there is no chance of overlap. A sample of the processed history data can be seen at figure 3. Each of the three users has approximately a similar size of Internet history in both time period and number of artefacts.

R-Dataset

The R-dataset is from a modern home PC with a single user account involved in a real law enforcement scenario, where the identity of the user at a particular time could be one of a possible two users sharing the same PC. The Internet history is extensive and lengthy periods of continuous usage are present. User 1 is the majority user of the PC whereas User 2 is a minority user with a considerably smaller number of accesses on the PC.

4.2. Plotting the Sessions

We have plotted in figures 4 and 5 the number of sessions that are available for analysis when the Internet history is aggregated using the two approaches and using the time groups of 30, 60, 120, 300, 600, 900, 1200, 1800 and 3600 s (half a minute, 1, 2, 5, 10, 15, 30 and 60 min respectively).

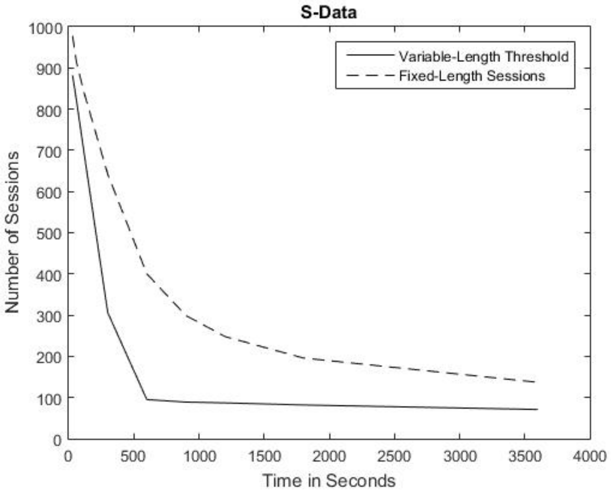


Figure 4: Graph showing for the S-dataset the number of sessions to analyse plotted against the size (in seconds) for the variable-length threshold or the fixed-length session size.

In figures 4 and 5 we see that a substantial reduction in the number of sessions occurs when a threshold value or fixed-size of 10 to 15 min is chosen. For Internet history analysis this seems reasonable given that a person can still be active on a computer reading pages or watching videos whilst there is an infrequent recording of artefacts onto the computer. If the timeline was of a different kind of artefact, for example file system artefacts, then a very different type of behaviour would be observed. Future work combining multiple different levels of artefacts could require their own session-to-session artefacts to accommodate that difference.

The number of sessions is always a greater number when using fixed-length session sizes rather than variable-length sessions. The variable-length data tends to fall off quickly, flatten and the number of sessions changes little regardless if a session length of 900 seconds or 3500 seconds is used. Similarly the fixed-length session ultimately flattens to a number of sessions that changes little after a session length of 1800 seconds is selected.

4.3. Data Reduction

The data was reduced by removing single occurrence components. Although the single occurrences may have investigative value as important one-time events, for the analysis of repetitive behaviour the single occurrences serve to only reduce the Jaccard distance between two sessions.

4.4. Correct Classification of User Activity

Our hypothesis for the experiments is that if two sessions have a high Jaccard Coefficient then they belong to the same user. To test this assertion we have used two

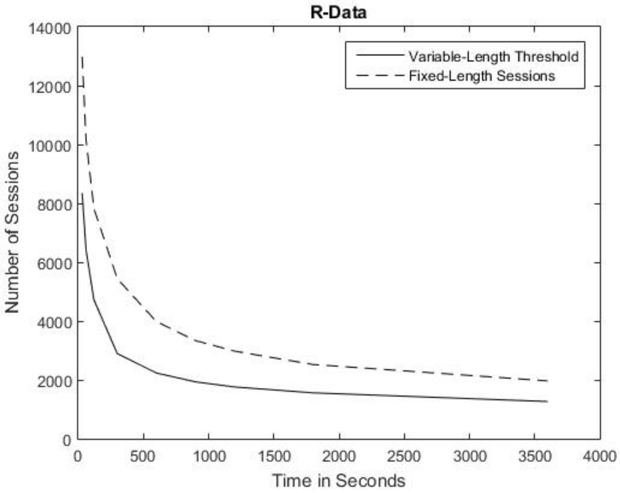


Figure 5: Graph showing for the R-dataset the number of sessions to analyse plotted against the size (in seconds) for the variable-length threshold or the fixed-length session size.

Internet history timelines as our sample test data. The histories have different characteristics and we do not propose that they represent 'model behaviour', indeed we note substantial further work is required to model what is 'normal user behaviour' in a number of settings such as an office environment, shared domestic environment etc.

Although we have referred to the results of the experiments as the correct user identification, the experiment is not identifying that session X belongs to User 1 and session Y belongs to User 2. What we are instead identifying is that if the two session patterns are above the Jaccard coefficient threshold in the experiment (0.25, 0.5, 0.75 and 1.0) and they belong to different users then that is a failure, whereas if the sessions belong to only one user then that is considered a correct identification of the user's self-similar behaviour. For example, the sample data in figure 1 (right), when comparing session 2 and session 4 we can see that there is a 0.25 similarity between the two sets of data. If these sessions represented different users then we would correctly identify them as separate users when using a threshold of 1.0, 0.75 and 0.5, but we would see them incorrectly identified as belonging to the same user when the threshold was lowered to 0.25.

Figures 6 and 8 show the variable-length threshold approach plotted against the S-dataset and R-dataset respectively and figures 7 and 9 show the fixed-length time values for the same datasets.

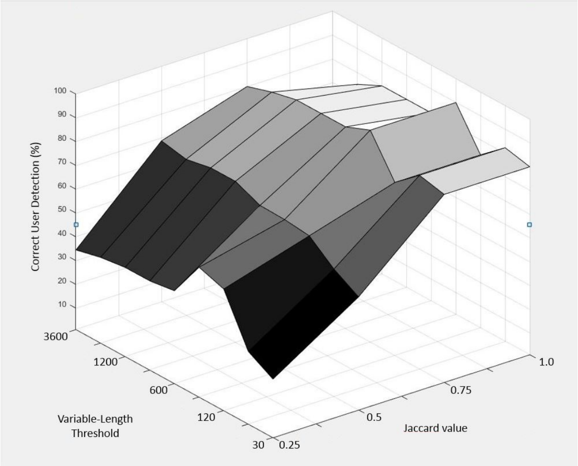


Figure 6: S-Dataset using the Variable Length approach

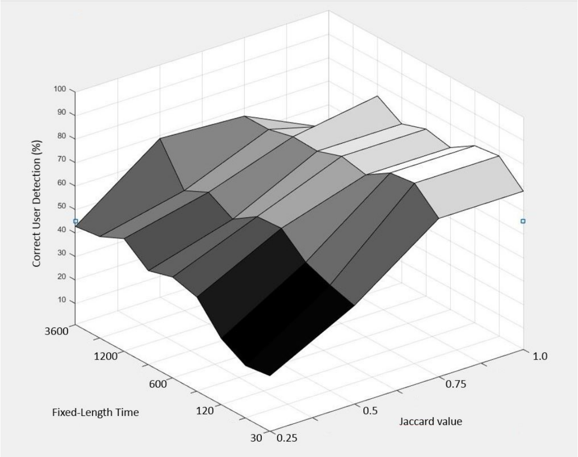


Figure 7: S-Dataset using the Fixed-Length approach

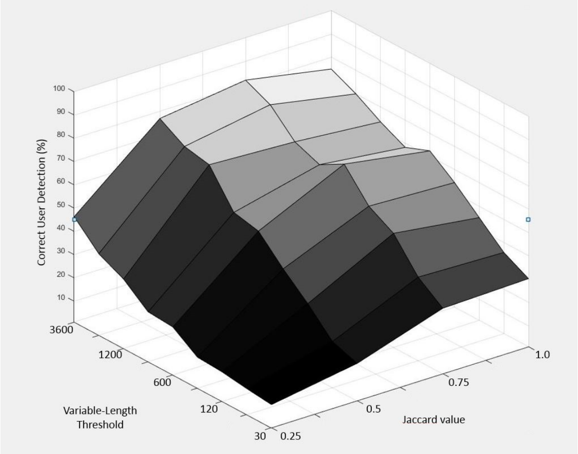


Figure 8: R-Dataset using the Variable-Length approach

5. Evaluation

5.1. Overall Performance of the Approach

The greatest overhead in the approach is computing the Jaccard distance coefficients for pairwise sessions. The number of sessions to compute is larger when using the fixed-length approach to session aggregation and when using short thresholds or fixed-lengths this can lead to significant computation time. Attempting to reduce the computational overhead by performing some kind of clustering

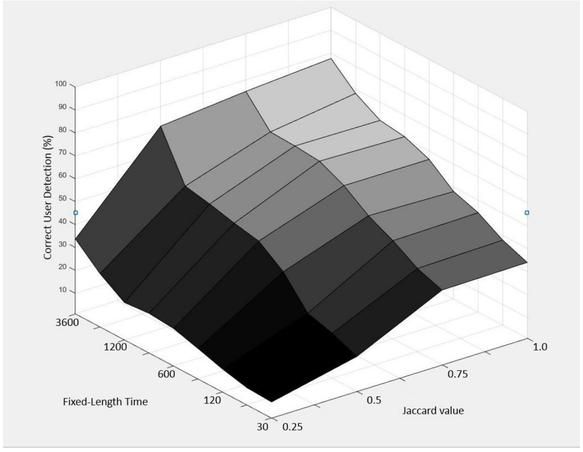


Figure 9: R-Dataset using the Fixed-Length approach

of components such as those suggested in Gresty et al. [7] is not particularly effective as the number of sessions is the main overhead. However, we note that such a reduction would potentially provide a more “juror friendly” visual representation of the Internet history than a full version, such as seen in the sample shown in figure 2.

The S-dataset was deliberately set up such that there was no chance of multiple different users being classified within the same session. The R-dataset did have a single example where the two users were accessing the original computer in close temporal proximity, and as such a erroneous classification did occur when the threshold was less than 600 seconds.

5.2. Fixed-length compared against Variable-length

The overall impression with respect to Internet history is that the variable-length approach performs substantially better than the fixed-length approach, which is somewhat expected as the variable-length approach better models the human-driven usage of the devices. We have not in this paper tested shorter, bursty artefacts that are indicative of systemic processes.

5.3. Jaccard Coefficient Levels

Within the graphs we see that with the variable-length approach and a Jaccard Measure of 0.75, we see optimal performance between 300 and 900 s. The association between sessions is not too restrictive (as at 1.0), yet the performance of correctly classifying users is much higher than at 0.5 and below.

5.4. Correct User Detection Rate

In the variable-length graphs for the S-dataset in figure 6, we see an 80-90% accurate classification rate in the 0.75 to 1.0 levels, and for the R-dataset in figure 8 a steadily increasing accurate classification rate from 56-80%. For an untrained, non-optimised approach that does not use any kind of profiling or classification of the sessions prior to analysis, we find these classification results very encouraging.

Within the long session-to-session patterns there were often a small number of “spoiler” sessions that contained frequently occurring components. The smaller size session-to-session patterns were the least likely to incorrectly classify the user, because as the Jaccard measure is raised closer to 1.0 the less sessions appear in the pattern and consequently fewer “spoiler” sessions appear in the pattern. The “spoilers” contain components that are not unique to an individual user within the dataset, but rather are commonly occurring activity one would expect to find within Internet history across a variety of users: within these patterns we find access to Google or MSN.

We have also experimented with removing the frequently occurring components. This dramatically increased the success rate, to 100% in some cases, but with the tradeoff that the number of sessions were reduced and in some cases, such as with the R-dataset where User 2, who was the minority user, was completely removed from the dataset by performing this reduction. In this case User 2 had characteristically an extremely repetitive behaviour and by removing the frequently occurring components the type of individual that we are hoping to detect was eliminated from detection.

It seems likely that a combination approach to dealing with the “spoiler” components where some form of prior knowledge or training outside of the analysis could be used to identify and eliminate generic components, coupled with eliminating the top X percentage of components. Further research will focus on the quantity of reduction and/or prior knowledge that is required to successful increase correct identification of users’ self-similar behaviour.

6. Conclusion and further work

We have presented an approach for temporally aggregating Internet history artefacts into sessions and we have argued that there is an investigative use for performing session-to-session comparisons, such that one-time events and repetitive behaviour within the Internet history can be identified. The approach and experiments presented within this paper shown an approach that is untrained, does not require any prior knowledge, and can be used by an investigator to identify and demonstrate habitual behaviour. Ultimately the goal of our research project is to produce and make available, a tool that can be used to test investigative hypotheses, such as “At time and date X, was there repetitive behaviour?”, “Are there any repetitive patterns that contain notable one-time events that are of interest to an investigator?” etc. Coupling such a tool with a systematic approach to testing testimony and factual evidence such as seen in [14], we believe would greatly assist an analyst or investigator and could be the difference between the investigator merely suggesting that something appears to be a common activity, as opposed to being able to demonstrate that something is indeed a pattern of life supported by evidence.

By reducing the Internet history to components we have shown that the history can be displayed, such as in figure 2, and even a human looking at the data can visualise the differences and similarities between sessions. However, with thousands of components and potentially tens of thousands of sessions it becomes impractical however to visually parse an entire Internet history. We therefore show that the components can be converted to a binary sting, which we are able to numerically test in terms of similarity between two sessions. Our experimental results have highlighted that different methods of session selection can produce considerably different types and sizes of sessions. For the analysis of Internet history we see that with the datasets used within this paper it is reasonable to use threshold interval to separate the sessions with values between 10 and 15 min and that variable-length sessions show the greatest accuracy at classifying individual user behaviour when there are multiple users of the same computer.

The experiments presented in this paper produced a 90% successful classification for one specific dataset and session aggregation method. However, we are of the opinion that classifying generic and commonly accessed websites would allow greater reduction of error patterns and greater identification of unique session patterns. Our initial experiments removing a percentage of the mostly frequently occurring components did yield very high correct classification rate at the cost of losing some important repetitive behaviour. Therefore, further work is required for the selection of the correct percentage of commonly occurring "spoiler" components to remove, possibly with the inclusion of prior or trained knowledge about what components are generic.

6.1. Future Work

The techniques proposed here are purely session aggregate, based upon a static value chosen at the beginning of the analysis. The result can produce extremely large data sets which are time consuming to process in pairwise order. We believe that the sessions can be reduced by session profiles, where sessions with similar characteristics such as length, duration, time of day and density of artefacts per time period, are processed together.

In the analysis of our experimental results, we noted that there is potentially a bias when using the Jaccard coefficient towards patterns being made up from sessions with few numbers of components, or with larger numbers of components that have a high degree of dependency. It is a bias in part by design, but as is highlighted above, selecting higher levels of Jaccard measures can be used to reduce the number of sessions in the session patterns. There could however be a level of dynamic selection such as if a session has few components then there may be a demand for it having a higher Jaccard value before it is added to the patterns. If the session has a large number or variety of components then a lower level of tolerance may be useful. The value on a pattern may be controlled by the complexity of the pattern. A suitable measure of complexity is a matter of further work.

A possible expansion of this work is the automatic classification of actual web-based behaviour, where the sessions are grouped and classified as noted within the paper but also a lookup is performed against the cached or recovered Internet history pages present on the device. This enhancement would require the ability to parse the pages and determine the content based upon the words that are present there. The cached web pages are quite often not available to an investigator. In such cases, a lookup may be performed against a resource such as the Internet Archive Wayback Machine [27] which can be searched for the closest time period that corresponds to the recovered Internet history artefacts on the device under investigation. This enhancement would allow an investigators to quickly identify the one-time events and repetitive events without attempting to interpret the URLs in the Internet history themselves.

The session-to-session approach provides a broad overview of a whole session, but does not adequately consider whether sub-session behaviours combine together to form the sessions. For this, we are developing component-to-component analysis to complement the session-to-session analysis.

The approach presented in this paper is a session-to-session comparison with no consideration to the order, sequence, volume and frequency of the components within the sessions. The next step of profiling two ostensibly similar sessions is to mine the sequential patterns within the session to identify useful repeating pattern sequences of the components. We are of the opinion that by identifying the sessions which are likely candidates to be similar using a session-to-session approach, the computation overhead examining intra-session will be substantially reduced.

References

[1] S. Al Awawdeh, I. Baggil, A. Marrington, F. Iqbal, "CAT Record (computer activity timeline record): A unified agent based approach for real time computer forensic evidence collection, Eighth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE), pp. 1-8, IEEE. 2013.

[2] S.W, Brenner, B. Carrier, and J. Henninger, "Trojan Horse Defense in Cybercrime Cases". Santa Clara High Technology Law Journal 21.1. 2004.

[3] F.P. Buchholz and C. Falk, "Design and Implementation of Zeitline: a Forensic Timeline Editor", DFRWS, 2005.

[4] R. Carbone, C. Bean, "Generating computer forensic supertimelines under Linux", SANS Reading Room, 1-136, 2011.

[5] Y. Chabot, A. Bertaux, C. Nicolle, and T. Kechadi. "Automatic Timeline Construction For Computer Forensics Purposes", IEEE Joint Intelligence and Security Informatics Conference, 2014.

[6] N. Eagle, A.S. Pentland, "Eigenbehaviors: Identifying structure in routine, Behavioral Ecology and Sociobiology, 63(7): 1057-1066. 2009.

[7] D.W. Gresty, D. Gan and G. Loukas, "Digital Forensic Analysis of Internet History Using Principal Component Analysis", 15th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, 237-242, Liverpool, UK, June 23-24 2014.

[8] K. Gudjonsson, “Mastering the Super Timeline With log2timeline”, SANS Reading Room, 2010.

[9] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, C.L. Isbell, “Unsupervised activity discovery and characterization from event-streams”, arXiv preprint arXiv:1207.1381. 2012.

[10] R.W. Hamming, “Error detecting and error correcting codes”, Bell System Technical Journal 29 (2): 147160. 1950.

[11] C. Hargreaves, J. Patterson, “An automated timeline reconstruction approach for digital forensic investigations”, Digital Investigation, 9: 69-79, 2012.

[12] B. Inglot, L. Liu, “Enhanced Timeline Analysis for Digital Forensic Investigations”, Information Security Journal: A Global Perspective, 23: 3244, 2014.

[13] P. Jaccard, “Etude comparative de la distribution florale dans une portion des Alpes et des Jura”, Bulletin de la Socit Vaudoise des Sciences Naturelles 37: 547579. 1901.

[14] J. James, P. Gladyshev, M.T. Abdullah, Y. Zhu, “Analysis of Evidence Using Formal Event Reconstruction”, Digital Forensics and Cyber Crime, pp. 85-98. 2010.

[15] J. James, P. Gladyshev, “Automated Inference of Past Action Instances in Digital Investigations”, International Journal of Information Security. Cryptography and Security. 2014.

[16] M.N.A. Khan, I. Wakeman “Machine Learning for Post-Event Timeline Reconstruction, First Conference on Advances in Computer Security and Forensics, Liverpool, UK. 2006.

[17] P. Khatik and P. Choudhary, “An Implementation of Time Line Events Visualization Tool Using Forensic Digger Algorithm”, JCSE International Journal of Computer Sciences and Engineering, 2(4), 2014.

[18] J. Kiernan, E. Terzi, “ Constructing comprehensive summaries of large event sequences, ACM Transactions on Knowledge Discovery from Data, 3(4), ACM. 2009.

[19] A. Marrington, “Computer Profiling for Forensic Purposes”, Queensland University of Technology Report, 2009.

[20] D. Minnen, T. Starner, I.A. Essa, C.L. Isbell, “Improving Activity Discovery with Automatic Neighborhood Estimation”, IJCAI. Vol. 7. 2007.

[21] J. Olsson, M. Boldt, “Computer forensic timeline visualization tool”. Digital Investigation, 6: 78-87, 2009.

[22] N.C. Rowe, S.L. Garfinkel, “Finding anomalous and suspicious files from directory metadata on a large corpus, Digital Forensics and Cyber Crime, Springer Berlin Heidelberg, 115-130. 2012.

[23] M. Schaefer, F. Wanner, F. Mansmann, C. Scheible, V. Stennett, A.T. Hasselrot, and D.A. Keim, “Visual pattern discovery in timed event data”. Visualization and data analysis, SPIE, San Francisco, 24 - 25 January 2011.

[24] S. Kälber, A. Dewald, and S. Idler “Forensic Zero-Knowledge Event Reconstruction on Filesystem Metadata”, Sicherheit, 331-343. 2014.

[25] K. Woods, C. Lee, S. Garfinkel, D. Dittrich, A. Russell, K. Kearton, “Creating Realistic Corpora for Forensic and Security Education”, 2011 ADFSL Conference on Digital Forensics, Security and Law

[26] Y. Ye, Y. Zheng, Y. Chen, J. Feng, X. Xie, “Mining individual life pattern based on location history, IEEE International Conference on Mobile Data Management. 2009.

[27] Internet Archive Wayback Machine, https://archive.org/web/.