2015

File system modelling for digital triage: An inductive profiling approach

Benjamin Rice
Australian Centre for Cyber Security, University of New South Wales, benjamin.rice1@defence.gov.au

Benjamin Turnbull
Australian Centre for Cyber Security, University of New South Wales, b.turnbull@adfa.edu.au

DOI: 10.4225/75/57b3ff9efb892
This paper was originally presented at The Proceedings of [the] 13th Australian Digital Forensics Conference, held from the 30 November – 2 December, 2015 (pp. 132-140), Edith Cowan University Joondalup Campus, Perth, Western Australia.

FILE SYSTEM MODELLING FOR DIGITAL TRIAGE: AN INDUCTIVE PROFILING APPROACH

LT Benjamin Rice, Dr Benjamin Turnbull
Australian Centre for Cyber Security, University of New South Wales, Canberra, Australia
benjamin.rice1@defence.gov.au, b.turnbull@adfa.edu.au

Abstract

Digital Triage is the initial, rapid screening of electronic devices as a precursor to full forensic analysis. Triage has numerous benefits including resource prioritisation, greater involvement of criminal investigators and the rapid provision of initial outcomes. In traditional scientific forensics and criminology, certain behavioural attributes and character traits can be identified and used to construct a case profile to focus an investigation and narrow down a list of suspects. This research introduces the Triage Modelling Tool (TMT), that uses a profiling approach to identify how offenders utilise and structure files through the creation of file system models. Results from the TMT have proven to be extremely promising when compared to Encase's similar in-built functionality, which provides a strong justification for future work within this area.

Keywords
Digital triage, digital forensics, device profiling, inductive profiling, file system, file system modelling, criminology, Enscript, dataset creation

Disclaimer
The views expressed are the authors' and not necessarily those of the Australian Army or the Department of Defence. The Commonwealth of Australia will not be legally responsible in contract, tort or otherwise for any statement made in this publication.

INTRODUCTION

Digital forensics is ultimately utilised to identify certain characteristics (Rogers, Goldman, Mislan, Wedge, & Debrota, 2006) or items of evidence (McKemmish, 1999; Rogers et al., 2006) from the devices of offenders. However, the abundance of devices and large amounts of data that belong to any one individual has increased to the point where effective and timely digital forensic analysis has become almost an insurmountable task (Roussev, Quates, & Martell, 2013). The solution to this problem is digital triage – a rapid, initial screen of potential target devices in order to assess their relative investigative worth (Roussev et al., 2013). In traditional scientific forensics and criminology, certain behavioural attributes and character traits can be identified and used to construct a case profile to focus an investigation and narrow down a list of suspects (Rogers, 2003). In the same way that case profiles can be developed for real-world criminal activities, the magnitude of the digital footprint offenders often leave on digital devices can also be used to create accurate and effective case profiles. The majority of profiling techniques currently utilised in digital forensic investigations adopt a deductive approach (Cantrell, Dampier, Dandass, Niu, & Bogen, 2012; Rogers et al., 2006). By developing an understanding of how users structure their file system on personal computers to facilitate their real-world criminal activities, inductive profiling techniques can be used to correlate the file systems of unknown computer systems to pre-established crime templates. This work seeks to address this through the development of a proof-of-concept tool chain that integrates with a commercial tool. This research is the first stage of a larger research project and will be used to generate data for future supervised learning systems.

BACKGROUND

Digital forensics is a relatively new field of research that was only formalised in the late 1990's (McKemmish, 1999; Pollitt, Noblett, & Presley, 2000). However due to the cultural trend of the rapidly increasing complexity and pervasiveness of technology, the increasing amount of data that can be extracted from digital devices is resulting in increasingly large forensic laboratory backlogs (Horsman, Laing, & Vickers, 2014). This is as a direct result from the number of criminal investigations that benefit from digital evidence, the number of devices an individual may own that could potentially hold evidentiary value and the increasing size and complexity of each device from an analysis perspective (Turnbull, Taylor, & Blundell, 2009). The increase in time required for digital forensic analysis causes delays in time critical cases where digital evidence may be crucial to providing a timely resolution for law enforcement officials and in the judicial process. Digital triage, a subset of digital forensics, was introduced to alleviate these issues by providing “a fast, initial screen of potential investigative targets in order to estimate their evidentiary value” (Roussev et al., 2013).

The key difference between digital forensic analysis and digital triage is the intended purpose of the investigative process. Although both digital forensic investigation and triage require the maintenance of evidentiary integrity, their aims differ. Digital forensic analysis is more concerned with determining a complete understanding of what has occurred on a computing device whereas the purpose of digital triage is to evaluate the evidentiary value related to an investigation in a timely manner (Cantrell et al., 2012; USDOJ, 2001). Digital triage can be used to gather information at a crime scene during initial search and seizure, providing time sensitive leads and a ‘psychological advantage’ for investigators when interviewing suspects (Rogers et al., 2006). In addition, it can also be used to prioritise devices submitted for forensic analysis to reduce case backlogs (Cantrell et al., 2012). Investigator-led triage (also known as administrative triage) is also a legitimate use of triage, allowing criminal investigators, with their deep knowledge of the case at hand to extract specific documents or prioritise full analysis as required (Shaw & Browne, 2013). On the other hand, technical triage refers to the software and hardware tools utilised to extract information from a device (Shaw & Browne, 2013).

There is no one solution for digital triage that is appropriate for all circumstances. As each case is inherently unique, different tools and techniques will yield different results. One technique commonly utilised in numerous triage models and in other areas of criminology is profiling. Profiling has proven to be an effective technique in criminal investigations and has roots tracing back to the ‘Jack the Ripper’ killings of the 19th Century (Rogers, 2003). Profiling has two forms: *deductive* and *inductive*. *Deductive profiling* is conducted after the offence and uses evidence to construct a behaviour profile for a particular case (Rogers, 2003), using specifics to build generalisations about a suspect’s expected behaviours or character traits. *Inductive profiling* draws on the characteristics of previous offenders to build evolutionary offender type profiles that can be applied to a larger population. This narrows down a list of subjects into a reduced subset of more probable offenders (Nykodym, Taylor, & Vilela, 2005).

RELATED WORK

Digital triage is a cost benefit analysis, maximising investigative outputs whilst minimising the time and resources required to analyse devices (Shaw & Browne, 2013). This approach is almost a subconscious action taken in criminal investigations, as resources are finite and time always works against the investigator or analyst.

Digital profiling and technical triage methods are relatively new areas of research within digital forensics. The benefits of digital profiling were only formally explored in 2003 (Rogers, 2003), with later work solidifying the benefits criminal profiling could provide (Nykodym et al., 2005). Arguably, the use of deductive profiling from the information gathered through digital forensics had already been occurring for a number of years; Pollitt’s previous work in computer forensics for the Federal Bureau of Investigation produced information that was incorporated into the traditional investigative process (Pollitt et al., 2000).

The first methodology to formally apply digital profiling within a triage framework was the Computer Field Forensics Triage Process Model (CFFTPM) (Rogers et al., 2006). The CFFTPM has an initial planning and triage phase that uses administrative triage to prioritise devices for analysis from case-specific information, followed by three stages of technical triage to develop a comprehensive device profile (Rogers et al., 2006). Holistically, these profiles are deductive in nature - they allow investigators to view the information rapidly gathered from a device and use their own insight and expert opinion to infer logical generalisations from the data. The CFFTPM is perhaps the most robust triage model that incorporates digital profiling and is one of the few digital triage models presented from research (Jiang, Yang, Lin, Zhang, & Liu, 2015).

As a forensic examination technique, profiling is difficult to subvert. The inherently habitual nature of human behaviour requires an offender to make a conscious effort to change their actions from those they are generally comfortable with. This requires a considerable amount of thought; an analyst can use digital profiling to detect commonalities during an investigation from a wide range of areas including, but not limited to: specific tools or toolkits utilised, language of the offender and their contacts from communications, location data, file timestamps or computer usage information (Foster & Liu, 2005).

Other triage approaches attempt to produce actionable intelligence by utilising automation with profiling. One such technique is Five Minute Forensics (5MF), used to rapidly classify the category of a user for each device analysed during a forensic investigation (Grillo, Lentini, Me, & Ottoni, 2009). Through minimal interaction with the device and the forensically sound extraction of a small number of files, the 5MF technique attempts to rapidly profile a user into one of five different user categories (Grillo et al., 2009).

More recent research combines aspects of the CFFTPM and 5MF into a Semi-Automated, Digital Triage Process Model (SADTPM) (Cantrell et al., 2012). Although case and evidence modelling have been identified as areas where future research is needed, (Nance, Hay, & Bishop, 2009) there has been little work conducted into the development of specific crime templates. The relative benefit that could be gained from this area is still yet to be fully understood.

The use of automation to aid in the rapid categorisation of devices has not only been explored by the SADTPM, with other work adopting a machine learning approach to classify digital media (Marturana & Tacconi, 2013). Well suited to the laboratory environment, their methodology utilises a large feature set and a number of crime-specific variables to extract data from a device. This data is then processed into a reduced data subset, normalised and the device classified using a binary categorisation on whether a plausible connection exists between the device and the target crime. Although this research utilises a large crime-specific feature set for analysis, these features are generally system-wide file statistics and the benefits from any type of profiling is limited to the crime-specific feature set.

The Case-Based Reasoning Forensic Triager (CBR-FT) is another tool developed for digital triage that presents a novel approach by locating evidence using generalised file system paths (Horsman et al., 2014). The CBR-FT targets specific locations in a file system hierarchy that are most likely to contain evidentiary data and improves its algorithm over time. This tool has merit in that it leverages off well tested and proven principles from psychology of profiling human behaviour to increase the success rate of rapidly finding data in a digital triage scenario (Horsman et al., 2014)

Each of these technical triage methods incorporate some form of categorisation, profiling or automated learning technique to improve the precision and recall of data returned during an investigation. However, none of these tools adequately model how an offender interacts with the underlying file system hierarchy and structure of a device. The CBR-FT is perhaps the closest tool that uses inductive profiling principles and increases its effectiveness over time, however, the tool is limited to generalised file system locations and can easily miss evidence in deep file path locations or uncommon storage areas. Wherever an offender chooses to store incriminating data, the structuring habits of that offender can be captured with the TFT through file system model generation.

There is little work towards digital triage methods that employ an inductive profiling approach. This work seeks to address this in-part through the introduction of the Triage Modelling Tool (TMT), which leverages file system structuring behaviours to identify devices with a greater likelihood of containing investigative value.

THE TRIAGE MODELLING TOOL

The purpose of the TMT is to provide a tool that can model a file system and provide information on the structure and sub-structures contained within it. This information can then be used in a profiling capacity to rapidly identify areas of interest for greater scrutiny during digital triage. The TMT developed by the authors provides a novel approach to file system model generation, improving on similar pre-existing solutions such as the in-build Encase ‘Export Folders’ function. The model adopts an inductive profiling approach as it is most suited to file system modelling. While deductive profiling has its merits, it is based on inferring logical conclusions from provided case data. For example, an investigator could easily deduce that an offender is a fan of a certain sporting team if their internet search history has numerous hits for that team. However, inferring logical conclusions from a file system representation is nigh impossible. For an investigator or analyst, viewing a directory and its associated subdirectories provides no immediately identifiable beneficial information.

The TMT has two components. The first component is the *Text Extractor* written in the proprietary Encase scripting language, *Enscript*, which provides a mechanism for extracting only the minimum amount of information required to recreate a model of a file system from the current investigation. This information is utilised by the second component of the TMT, the *File System Modeller* Python script which recreates a representation of the data originally selected from Encase. Both of these components are available for download at:

https://github.com/AustralianCentreforCyberSecurity/Triage-Modelling-Tool

The licensing specifics are also found on the GitHub site.

CONCEPTUALISATION

TMT is designed, in its initial stages, to assist and augment administrative triage in a laboratory environment. Often this triage is investigator-led and conducted as a precursor to a full forensic analysis. TMT relies on the use of existing forensic tools for integrity preservation. Its use will assist criminal investigators and forensic analysts to direct their efforts to areas more likely to benefit them.

The second component of TMT is as the initial data creation tool in a larger, ongoing research project. Unsupervised machine learning systems require tagged and verifiable datasets to effectively train (Pedregosa et al., 2011). One benefit of the TMT tool is that over time, the tool will create a larger number of crime-specific datasets that can be used as training data for future machine learning development. Once these datasets have been obtained, there will be future research to leverage these in a field-based digital forensic triage tool.

DEVELOPMENT OF THE TMT

The Text Extractor Enscript operates within Encase to maintain forensic integrity of data and allows a forensic analyst to select individual subdirectories, an individual user's entire home directory or an entire file system - the amount of customisation is up to the discretion of the analyst. Once the desired file system areas to be modelled have been selected, the Text Extractor Enscript can then be run to export the required information to the Python script.

The Enscript works by first defining an iterator to operate on the selected files and folders from the case view. The iterator is explicitly told not to calculate hashes of any data it extracts to increase efficiency. Once the iterator is established, the export path is defined and built on the local machine to store the output from the script. This location can be modified, but by default exports to the current case's export directory.

With all selected items from the case view, the Enscript iterates through every item and extracts the full path, name and category for each folder and file. The filename is then written to a string with the same name as the file or folder from the case, but appended with a Globally Unique Identifier (GUID) to ensure that the integrity of all entries remain intact and no entries with identical names are overwritten at the export location. The full path and category of each entry are also written to a string. A file is then generated in the local machine and uses the filename with GUID as the identifier. Finally, the text document is populated with the content from the second string - the full path and category - before being closed. This occurs until all items have been processed and a text file representing each file and folder has been created on the local machine.

The Python script has two major functions. The first function accepts input from the user for the file system directory location and the expected root file's name. This ensures the correct location for the text files exported from Encase is correct, but also strips any unnecessary files or folders that exist above the desired root folder from all exported files. This function then imports, reads, formats, splits and writes both lines from each text file into a list and provides an error check for how many files were processed.

The second function also accepts two items of input - the export location for the file system model to be generated at and the type of crime the model is most closely associated with. With this information, the function iterates through the list generated from the previous function and creates any entry with a category of 'Folder'. As the list is not ordered by the root directory listing, the function also checks to ensure all parent folders have also been created for each folder entry and creates them if needed. Once all folders have been created, the function then iterates through the list and populates each folder with any entry with a category of 'File'. Error catching and correction also occurs for NTFS and Windows 7 OS compatibility issues. After this function has finished populating the generated model with files, the script terminates.

TEST CASES

To test the required functionality of the TMT, four test cases were used. These test cases displayed the required functionality of the TMT by modelling the entire user space for one randomly selected user from the disk image, as well as modelling the immediate folders and files surrounding an item identified as evidence deep within the file system. The two disk images used to conduct this analysis are outlined below (digitalcorpora.org, 2015a).

- Test Case 1, 2 - nps-2008-jean.E01, Encase Image File Size: 1.464841GB (digitalcorpora.org, 2011)
- Test Case 3, 4 – terry-2009-12-11-001.E01, Encase Image File Size: 10.34532GB (digitalcorpora.org, 2015b)

The results of the test cases were measured against Encase's inbuilt 'Export Folders' function. This function was found to have the most similar functionality to the TMT and has the added benefit of being directly compatible with the software rather than being an external plugin or Enscript.

The TMT was designed, developed and tested on a Windows 7 Enterprise SP1 64-bit Operating System. The machine used had an Intel®Core™ i7 CPU 860@2.80GHz processor with 8GB RAM. All results were stored in the C:\ Drive, which for a 2GB-2TB volume has a minimum allocation unit size of 4KB.

Three performance metrics were used to measure the effectiveness of the TMT against the file system modelling Encase could produce:

- Size: This refers to the total size of all files (the entire model) exported by Encase. For the TMT, size refers to the collective size of all text files exported with the Enscript.
- Size on Disk: This refers to the amount of disk segments allocated by the OS for all files exported by Encase. For the TMT, this is the amount of disk segments allocated to store the exported text files. For the NTFS file system format, the minimum cluster allocation size was 4KB (Microsoft, 2015).
- Efficiency: This is the total time for Encase to export all folders and files. For the TMT, this is the collective time required to run, process and then generate a file system model from both the Enscript and Python script. For testing, the required user inputs for the Python script were predefined to ensure accuracy.

In the table and graphs below, TMT refers to the Triage Modelling Tool and ENC refers to the Encase export option.

Test Case Performance Metrics Comparison			
	Size	**Size on Disk**	**Efficiency**
Test Case 1 (User – C:/Users/Jean)	TMT output 125.24x less (1,749KB compared to 219,051KB)	TMT output 11.26x less (20,509KB compared to 230,928KB)	TMT 43s faster (11.35%)
Test Case 2 (Evidence File, Custom Root: Jean\Start Menu\Programs)	TMT output 2.69x less (8.65KB compared to 23.3KB)	ENC output 1.28x less (128KB compared to 100KB)	TMT 0.795s faster (39.75%)
Test Case 3 (User – C:/Users/terry)	TMT output 262.74x less (4,098KB compared to 1,076,700KB)	TMT output 22.51x less (49,161KB compared to 1,106,792KB)	TMT 17m40s faster (55.85%)
Test Case 4 (Evidence File, Custom Root: terry\AppData\Local\Google)	TMT output 373.72x less (890KB compared to 332,607KB)	TMT output 32.47x less (10,400KB compared to 337,670KB)	TMT 3m29s faster (55.73%)

Table 1 - Test Case Parameter Comparison of Encase vs TMT

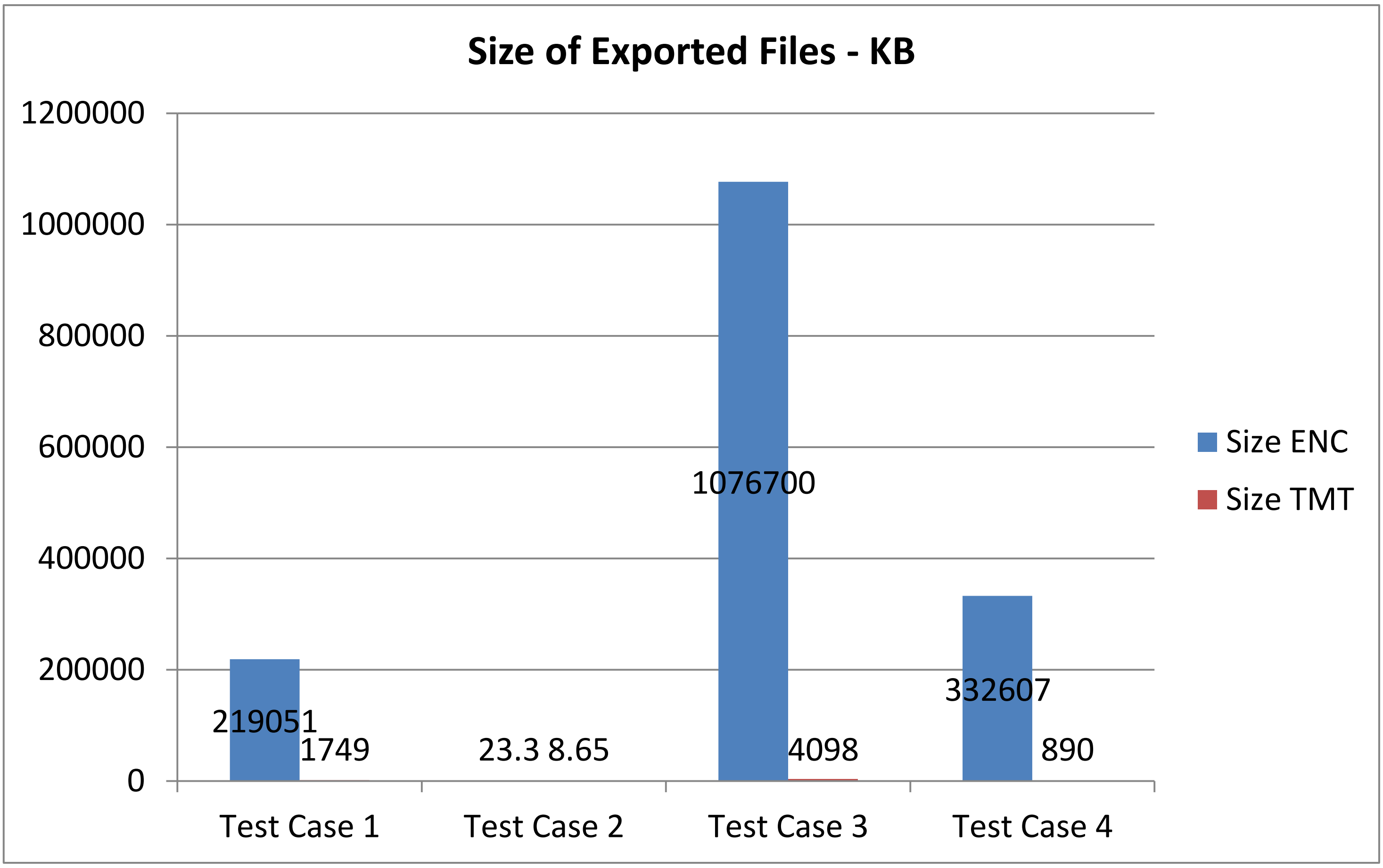

Figure 1 - Size of Exported Files of Encase vs TMT

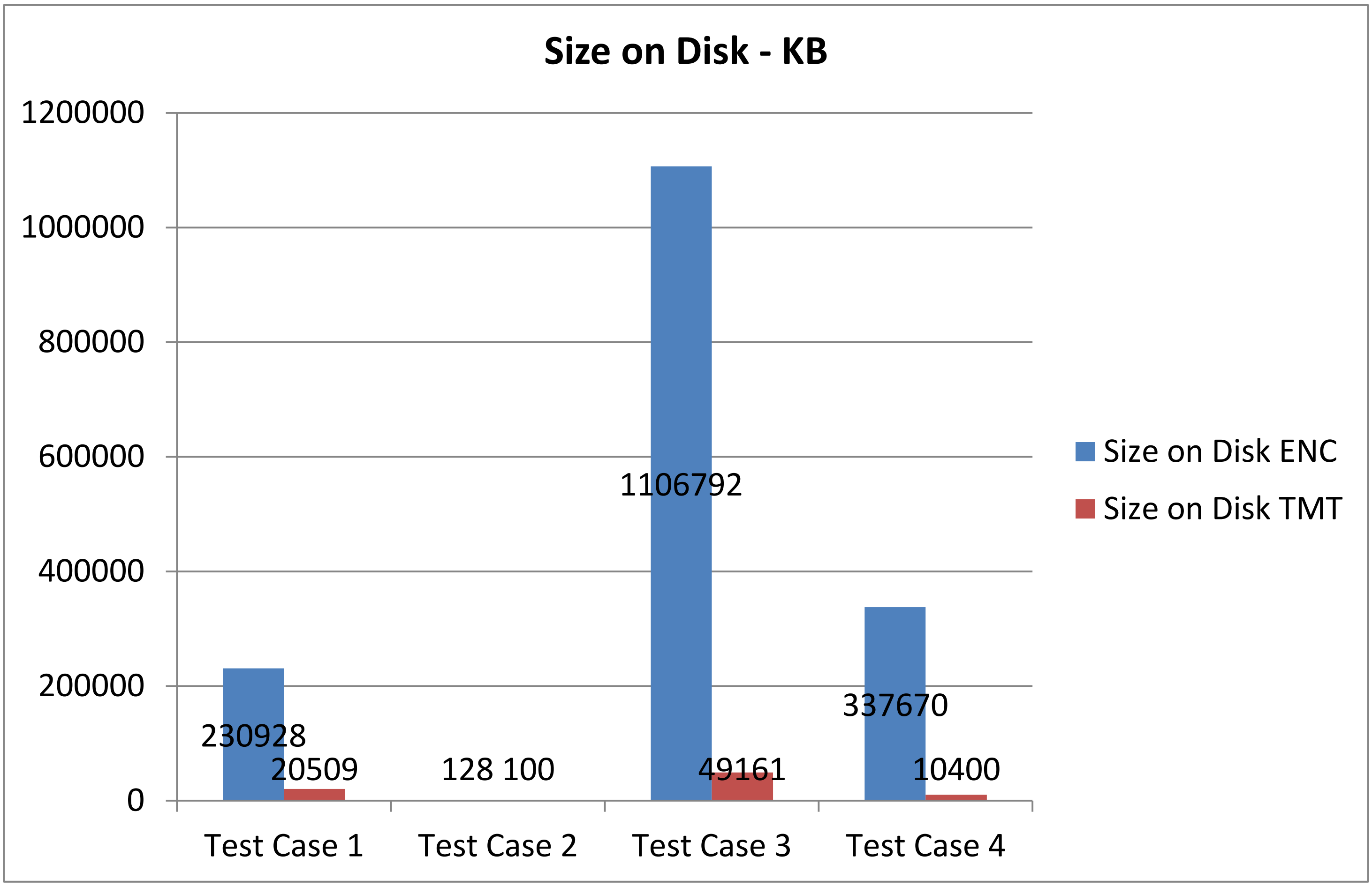

Figure 2 - Size on Disk of Encase vs TMT

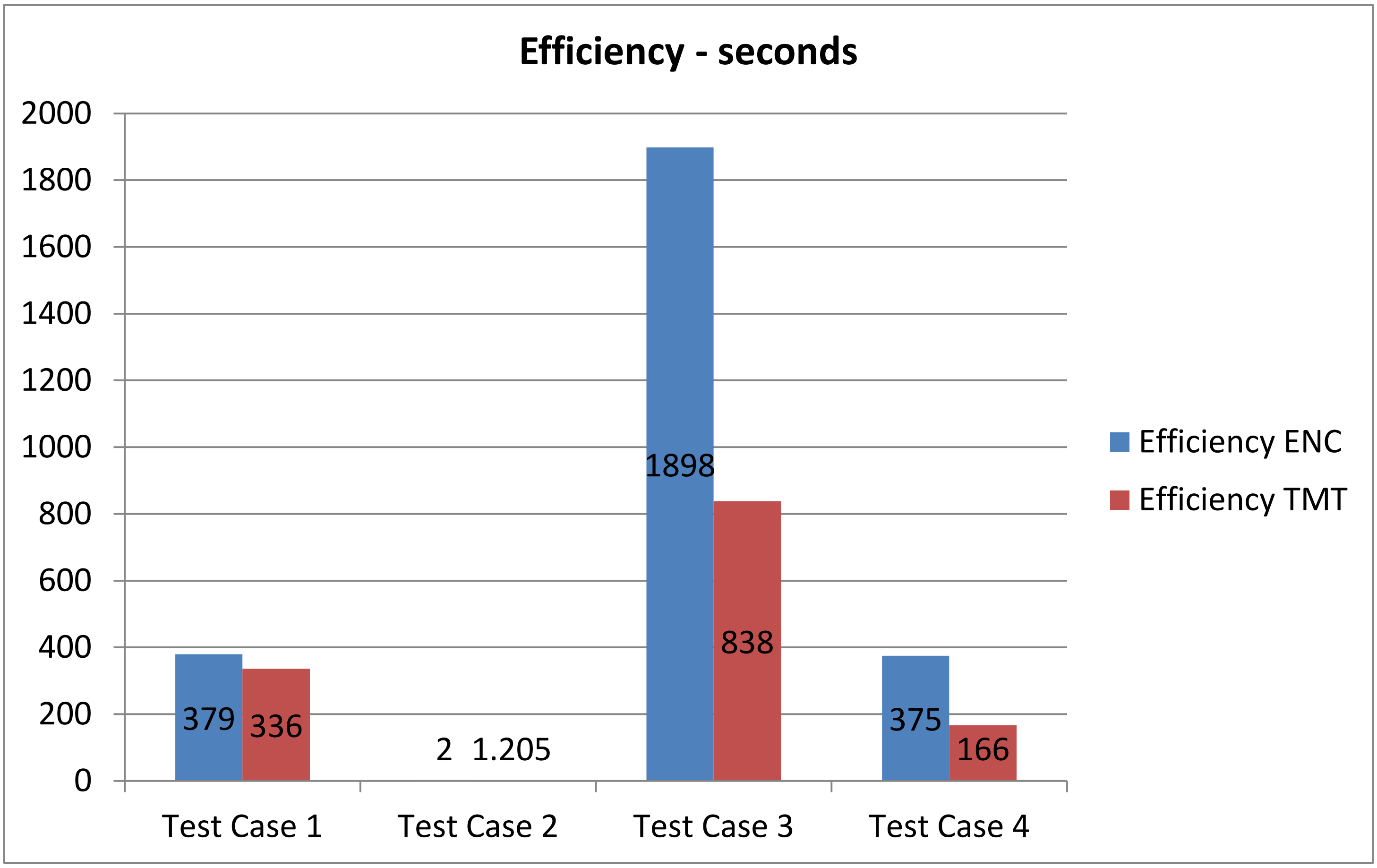

Figure 3 - Efficiency of Encase vs TMT

Displayed in Table 1 and Figures 1-3 above, the results from each test case were promising as in 11 out of 12 performance metrics compared, the TMT performed better than the Encase function. As the minimalistic design and implementation of the TMT only extracts what is required to rebuild the file system hierarchy, the size of exported files and size allocated on disk is extremely scalable for larger models. For extremely small models with files that have amounts of data that can be contained within the cluster allocation unit size for NTFS file system formats, the exported TMT text files representing folders require a 4KB segment allocation (Microsoft, 2015), increasing the TMT model size to greater than the Encase export. In reality, these types of models would generally have less profiling benefit. The size on disk is also only a concern between processing of the Enscript and Python script. Once the Python script has processed all text files and rebuilt a representation of a file system segment, these files can be deleted and the final size on disk of the model is 0 bytes. As the final model generates folders and files with only a name as metadata, NTFS can store the entire directory listing and all metadata within its master file table (MFT) and does not have to allocate a 4KB disk segment to any part of the generated model (NTFS.com, 2015). Figure 4 below displays a standard entry within the MFT - all metadata can be stored within the 'data or index' field of the entry.

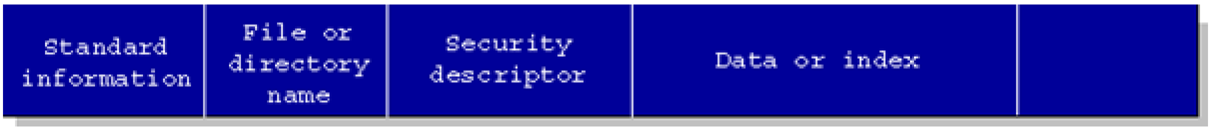

Figure 4 - NTFS MFT Entry

As discussed, the results of the tests are promising. As a proof-of-concept, this work provides a basis for future digital forensic profiling work.

CONCLUSION AND FUTURE WORK

This paper outlines the benefits and first stage of an inductive profiling approach for digital triage through the use of a file system model generation tool. As a proof-of-concept implementation, the TMT has displayed extremely positive results which encourage the continual development of the tool to encompass a greater variety of operating systems, file system formats and forensic programs.

The most important area of future work is the development of a file system model data store and correlation database to support the inductive profiling approach. As file system models are generated for different types of crime, a centralised data store with machine learning algorithms could be utilised to develop file system templates for different crime types. These crime templates could then be used to determine the amount of correlation between an unknown file system and the crime template database, providing a rapid means of classifying devices in a laboratory environment.

REFERENCES

Cantrell, G, Dampier, D, Dandass, Y. S, Niu, N, & Bogen, C. (2012). Research toward a Partially-Automated, and Crime Specific Digital Triage Process Model. Computer and Information Science, 5(2).
digitalcorpora.org. (2011). nps-2008-jean.E01. Retrieved from http://digitalcorpora.org/corpora/disk-images/nps-2008-m57-jean/

digitalcorpora.org. (2015a). Digital Corpora - Producing the Digital Body. Retrieved from http://digitalcorpora.org/

digitalcorpora.org. (2015b). terry-2009-12-11-001.E01. Retrieved from http://digitalcorpora.org/corpora/disk-images/m57-patents/

Foster, J, & Liu, V. (2005). Catch me, if you can, Blackhat Conference Briefing. Retrieved from http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-foster-liu-update.pdf

Grillo, A, Lentini, A, Me, G, & Ottoni, M. (2009). Fast User Classifying to Establish Forensic Analysis Priorities. Paper presented at the Fifth International Conference on IT Security Incident Management and IT Forensics, Stuttgart, Germany.

Horsman, G, Laing, C, & Vickers, P. (2014). A case-based reasoning method for locating evidence during digital forensic device triage. Decision Support Systems, 61, 69-78.

Jiang, J. G, Yang, B, Lin, S, Zhang, M. X, & Liu, K. Y. (2015). A Practical Approach for Digital Forensic Triage. Applied Mechanics and Materials, 742, 437-444.

Marturana, F, & Tacconi, S. (2013). A Machine Learning-based Triage methodology for automated categorization of digital media. Digital Investigation, 10(2), 193-204.

McKemmish, R. (1999). What is Forensic Computing. Australian Institute of Criminology trends and issues in crime and criminal justice, 118.

Microsoft. (2015). Default cluster size for NTFS, FAT, and exFAT. Retrieved from https://support.microsoft.com/en-us/kb/140365

Nance, K, Hay, B, & Bishop, M. (2009). Digital Forensics - Defining a research agenda. Paper presented at the Proceedings of the 42nd Hawaii International Conference on System Sciences, Hawaii.

NTFS.com. (2015). NTFS Master File Table (MFT). Retrieved from http://ntfs.com/ntfs-mft.htm

Nykodym, N, Taylor, R, & Vilela, J. (2005). Criminal profiling and insider cyber crime. Digital Investigation, 2(4), 261-267.

Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research, 12, 2825-2830.

Pollitt, M, Noblett, M, & Presley, L. (2000). Recovering and Examining Computer Forensic Evidence. Forensic Science Communications, 2(4).

Rogers, M. (2003). The role of criminal profiling in the computer forensics process. Computers & Security, 22(4), 292-298.

Rogers, M, Goldman, J, Mislan, R, Wedge, T, & Debrota, S. (2006). Computer Forensics Field Triage Process Model (CFFTPM). Journal of Digital Forensics, Security and Law, 1(2).

Roussev, V, Quates, C, & Martell, R. (2013). Real-time digital forensics and triage. Digital Investigation, 10(2), 158-167.

Shaw, A, & Browne, A. (2013). A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. Digital Investigation, 10(2), 116-128.

Turnbull, B, Taylor, R, & Blundell, B. (2009). The anatomy of electronic evidence–Quantitative analysis of police e-crime data. ARES'09 International Conference on Availability, Reliability and Security.

USDOJ. (2001). United States Department of Justice - Electronic Crime Scene Investigation: A Guide for First Responders. Technical Working Group for Electronic Crime Scene Investigation.