EM-DMKM Case Study
Computer and Network Forensics

Dominik Cygalski and Mohammad Ghufran

June 21, 2012

Contents

1 Introduction

Case Study in Computer and Network Forensics should start with a definition of this discipline. This is, though not a straightforward task. Different definitions can be found in the literature [15, 7] and even the name itself varies throughout different publications. To explain this situation brief overview of the evolution of this discipline is necessary [19].

With the advent of computers new type of evidences appeared on the crime scenes – digital evidence. At the beginning mostly restricted to analysis of stand alone computers, computer forensics soon needed to enter the domain of *Network Forensics* as computers became more networked. Nowadays, in the era of ubiquitous computing evidences are being found in a variety of devices such as mobile devices, video cameras, flash drives, routers, tv, etc. which raise a question if the discipline can still be called *Computer Forensics*. We decide to refer to the broadest possible understanding of various flavours of the discipline under generic name of *Digital Forensics*.

Main contributions to the development of the field at its early stages were made by the law enforcement organisations. Hence it was, for the most part, a practitioners driven discipline. Although this situation has been changing in recent years. As the forensics professionals require profound, multidisciplinary; training universities started to offer appropriate programs. Consequently, research community got involved which resulted in efforts to define research agenda for this field [10].

Distinction should be made between other fields that seem to be very close to Digital Forensics. A good example is Information security. They both work with digital material and law infringements. Whereas security focuses on how to prevent and protect systems against attacks, in the forensics stress is put on post-incident reaction [19]. This fact makes it similar to Incident Response field which is responsible for responding to security incidents within an organization and even common framework for these two fields has been proposed [5]. Although differences can be found – Incident Response focuses more on protecting an organisation and restoring compromised systems whereas in forensics the strong emphasis is put on using highly objective, scientific methods in order to provide evidences that can be further used in the court of law for litigation and legal action.

Common concepts being repeated in majority of definitions point out the steps that form a process. The number of steps in this process varies depending on the definition, though a core framework can be summarized as follows: identification, collection, preservation, analysis and presention of the computer-related evidence. Each phase poses various challenges, especially taking into account the fact that all activities should be performed ”in a manner that is legally acceptable by court” [15]. First steps from identification to preservation of digital data are beyond the scope of this Case Study. Thus, in this paper we will focus mainly on the step of analysis (and indirectly presentation) and we will try to apply state-of-the-art Knowledge Discovery and Knowledge Management practices to provide good quality outcomes of this step.

2 State of the art

Before any study of forensics, the first phase is the processing and preparation of data. The goal of the Analysis phase is to analyse the data prepared in the previous steps [5]. The data is prepared in the special manner through a sequence of steps that proceed the actual analysis. Two most important steps from our point of view are: harvesting and data reduction. Harvesting is known as gathering metadata about collected material such as timestamps, filetypes, authors and other attributes that can be deduced from different sources of data. The Goal of data reduction is to decrease the volume of collected material with various techniques (e.g. using hash tables of known good files) in order to ease actual task of analysis.st Once prepared in this way, temporal, multidimensional data set can be used by the investigator who can start to reconstruct the events that happened. The aim of this process is to try to identify the cause of the incident, victim(s) and culprit(s) as well as to find relevant proofs. If some hypothesis have already been established, proofs to support or refute them are searched for. One popular method is by exposing correlations between events, although extreme caution must be taken when drawing any conclusions and each of them must be supported by strictly scientific methods.

2.1 Different approaches

Since any sort of Digital Forensics Investigation involves considering a host of stochastic variables which are interconnected between themselves with numerous dependency relations, it is natural to use techniques such as Bayesian Networks to deal with the complexity of the situation and to be able to exploit reasoning and probabilistic inference mechanisms. There have been several examples in literature where the use and effectiveness of Bayesian Networks have been demonstrated. In [9] a Bayesian Network was set up for the case of illegal distribution of copyright protected multimedia material over a peer-to-peer network. A simple structure of the network was proposed by the authors comprising of three types of nodes: main hypothesis, sub-hypotheses and evidence nodes, which are the only observable nodes in the network, connected with each other using diverging connections. In order to minimize the subjectivity involved in assigning conditional probabilities to the network, the authors conducted a survey among domain experts to obtain corresponding probability distributions. Overall, work shows that Bayesian Networks are a useful tool and more importantly, they provide measurable interpretation of the digitaled evidence in support or refutation of certain hypotheses that is reliable and can be further used by a jury or a judge in the court. Moreover, in [14] a sensitivity analysis was performed on the model proposed in [9] that revealed reasonable stability of the model for missing evidences. Also, for the problem of subjectivity did not appear to be as crucial as expected for assigning conditional probabilities to the network. These studies show that Bayesian Networks perform fairly well even with missing data and with subjective conditional probabilities. This is vital in practical situations where not all variables are observable and the experts must be conservative in assigning conditional probabilities.

While the case of Bit Torrent piracy [9] demonstrates the use of dependencies of events through Bayesian Networks, it is evident that most of the situations which involve Digital Forensics also have temporal dependencies. [4] demonstrate the application of a Hidden Markov Model in the field of forensice. A Hidden Markov Model (HMM) is a statistical Markov model where the system to be modelled is considered to be a Markov process with unobserved states. The parameters of a HMM are of two types, the transition probabilities and transmission probabilities. The transition probabilities control the way the hidden state is chosen at time t given the hidden state at time t - 1. The emission probabilities govern the distribution of the observed variable at a particular time, given the state of hidden variable at that time. Hidden Markov Model (HMM) in Bayesian Networks are specifically utiliseable in situations where uncertainty in data is high and observations are incomplete; for example, speech, handwriting, and bioinformatics. [4] extend a Bayesian Network model with a Hidden Markov Model for predicting hypotheses and degree of criminal activity as it evolves over time. In the extended model, a Bayesian Network defines the structure of the observed variables while the Hidden Markov Model of degree one is used to model the evolution of the criminal activity over time. That is, the hidden nodes measure the "degree of criminality" for events of a given time interval which are influenced by both the present time interval *t* as well as the previous time *t-1*. Given this setup, observations for each time interval are applied to determine the most likely type of suspicious activity associated with the observations.

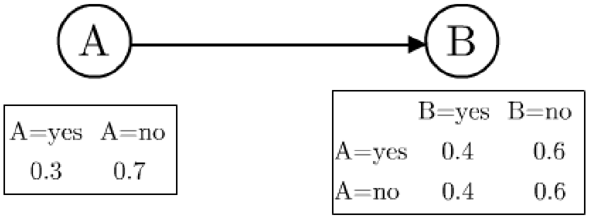
One of the challenges with temporal data is the association of different timestamped data with each other, specially when time scales are not defined and could vary. [6] illustrates the use of proximity matrices and clustering for timestamped events in order to find both similarities and outliers for a set of timestamped events. First, an algorithm for single scale analysis is presented, followed by an algorithm for multiple scale analysis. For the single scale method, two proximity matrices are computed; one each for proxomity between different time slices (time-to-time) and one for within time slice proximity (time-to-others). k-Means clustering is then performed on the matrix of eigenvalues obtained from the time-to-time matrix. For varying time scales, a method for aggregation of proximity matrices is proposed from the finest to more coarser levels of time slices, followed by similar clustering techniques described earlier. The outliers are identified if there are data points which deviate significantly from the clusters formed, normally forming an individual cluster. The authors demonstrate the utility and performance of using proximity matrices using practical data sets showing clusters where time slices are not known.

Another group of methods that are used in Digital Forensics is presented in the paper "The Trojan made me do it: a first step in statistical based computer forensics event reconstruction" [1]. In that work authors use stepwise discriminant analysis to decide whether files have been created intentionally or unintentionally on the suspect's computer. Unintentional creation might be explained by the presence of a malware software or a system being compromised. This is essential because the sole fact of existence of the illicit digital material is not sufficient in most cases of court proceedings. For instance, during the famous Enron trial in United States of America in the first decade of 20th century a digital version of a memo was found on the computer of a manager and constituted a major evidence of the guiltiness of the respondent. However, the plaintiff had to prove that the electronic document was in fact opened by the manager and even though they succeeded to show that using files' metadata they failed to prove that the manager actually saw the content of the second page of the document which contained the crucial information – the defendant claimed that monitors that were in use at that time displayed only one first page right after opening a document. The scenario adopted in [1] is about finding a set of child pornography pictures on the suspect's computer and trying to establish whether or not they were downloaded intentionally. For this purpose the supervised data is used which was obtained during conducted experiment and stepwise discriminant analysis is used to learn the model on variables that are mostly metadata collected from illicit files such as: average of the difference between files creation times, number of thumbnails that exist for contraband images, number of images created within five minutes of visit to contraband website or number of references to contraband items stored on local disk in the Recent Folder. The results show that classical regression methods can be exploited to determine whether a digital evidence was created intentionally by the suspect which can greatly assist the jury while deciding on the verdict.

2.2 Bayesian Networks

Bayesian Networks became more popular over the last few years. The concept originated in the Artificial Intelligence community and it constitutes a very important tool for those who have to deal with the uncertainty problem. In [2] professor Eugene Charniak even compares Bayesian Networks' importance to the importance of resolution theorem for logic community. Mostly known under the name of Bayesian Networks they are also known under other names such as belief networks, knowledge maps or probabilistic networks. They have been successfully applied in many fields mainly to support decision making (for instance medical diagnosis) but not only limited to that – speech recognition is a very good example of exploiting their capabilities while dealing with temporal dimension at the same time.

Bayesian Networks are a type of graphical model [17] whose main goal is to simplify the joint distribution of multiple random variables using additional information about dependencies between variables. Two main components constitute a Bayesian Network: a directed acyclic graph (DAG) and a set of conditional probability distributions.



A=yes	A=no
0.3	0.7

	B=yes	B=no
A=yes	0.4	0.6
A=no	0.4	0.6

Figure 1: Simple unfaithful network.

Directed acyclic graph (DAG) – graph is a structure that consists of the set of nodes and set of edges. An edge connects two nodes and can be directed. A directed acyclic graph is a graph whose edges are directed and there exist no directed cycle. A directed cycle can be described as a path that starts and ends in the same node following the direction of the edges. Use of directed edges results in distinction between parent and children nodes – in Figure 1 node A is a parent node and node B is its child node. Nodes represent random variables from the domain and traditionally are discreet although there exist methods for handling continuous variables too. Edges correspond to dependencies between variables.

Conditional probability distributions – describe quantitative distributions of the variables. For discreet variables distributions will be presented as tables like in Figure 1. The notion of faithfulness is important at this point. A Bayesian Network with given DAG and conditional probability distributions are said to be faithful if and only if every conditional independence relationship valid in distributions can be read in the DAG [12]. An example of an unfaithful network can also be seen in Figure 1 – graph structure suggests that both variables are dependent while given distribution indicates their independence.

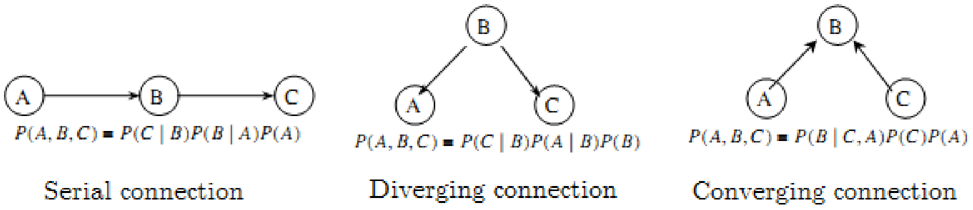

A
B
C
P(A, B, C) = P(C | B)P(B | A)P(A)
Serial connection
B
A
C
P(A, B, C) = P(C | B)P(A | B)P(B)
Diverging connection
B
A
C
P(A, B, C) = P(B | C, A)P(C)P(A)
Converging connection

Figure 2: Connection types. Diagram from [8].

Three basic structures can be identified while constructing a Bayesian Network: serial/linear connection, converging connection and diverging connection. All three are presented in the Figure 2. They provide different information about dependencies between variables. In serial connection and in diverging connection variables A and C are dependent but they become independent when B is given. On the contrary, in the converging connection variables A and C are independent but they become dependent when B is given. Converging connector is also called a V-structure.

The most interesting thing about Bayesian Network is that they allow to perform probabilistic inference on the models. Probabilistic inference means finding the distributions of variables given a set of observed variables. Observed variables are those whose state is known – they can also be referred to as the evidences. Because accurate inference often is difficult in big and strongly connected networks due to computation complexity there are methods that allow approximation of real values such as monte carlo, inference by ancestral simulation or Gibbs sampling.

Learning a Bayesian Network is not a simple task. Two main approaches are learning a structure using data and experts' knowledge. There exist algorithms that can both construct a structure of the network and estimate probability distributions. Experts are often involved when there is not sufficient data to estimate good parameters.

2.3 Extensions of Bayesian Networks

In previous subsection, definitions and general working of Bayesian Networks (BNs) have been given. While there are many applications of BNs, the standard approach and the model may not be useful in some situations. However, many variations and improvements of Bayesian Networks have been proposed which are more usable in specific situations. Since a BN is a probabilistic model, different variations using different variations and representations of the same probabilistic model have been proposed.

In this point, we discuss some of the most prominent and state of the art extensions of Bayesian Networks.

2.3.1 Causal Bayesian Networks

Probabilistic dependence, which is at the core of Bayesian Networks does not necessarily imply causality. However, quite often, the graphs constructed by experts are based on causality since judgements are more meaningful, reasonable, and hence reliable. Another advantage of building Bayesian Networks on causal relationships is the ability to represent and respond to external or spontaneous changes. For example, to remove one of the causes from a representation, simply removing all the edges connecting to that node can be removed. Similarly, if new causal relationships are added, only parts of the network probabilities need rework. This considerably reduces modelling time and effort [13].

It is for these reasons that Causal Networks are quite often used, specially in scenarios where the graphs are provided by experts because of the nature of reasoning of the experts. As already mentioned, in a usual Bayesian Networks, a relationship $A- > B$ is not causal. However, causal relationships can be identified using the Essential Graph (CPDAG). In a CPDAG, the edges that are directed are causal relationships [13].

2.3.2 Bayesian Multinets

Bayesian Networks have been widely used as classifiers and many classifiers based on Bayesian Networks exist - such as Naive Bayes and Tree Augmented Naive Bayes [11]. In any graphical model, there are four distinct components, in general: the semantics, the structure, the implementation, and the parameters. Based on training data, a particular semantic could potentially be selected or perhaps even a new one discovered. Keeping the structure constant, there are a variety of ways to implement the dependencies between random variables, such as conditional probability tables, neural networks, and decision trees. Overall, a good assignment of all the parameters must be found when creating a graphical model. The ultimate goal is to identify a system for probabilistic inference that is computationally efficient, accurate, and informative about the given problem domain.

As an extension, when the model is time dependent in nature, models like dynamic Bayesian Networks (and more specifically dynamic Bayesian Multinets) are frequently proposed. They are widely used in the fields of pattern recognition which is inherently a time dependent problem.

In a classification problem, the goal is to discriminate individual date points of a collection of domain variable into one of the various defined classes. When the Bayesian Network semantic is chosen as a graphical model, the structure must be learned from a set of training data (or through the help of domain experts). When using BN as classifiers (BNC), a new structure of (in)dependencies between domain variables is learnt using the available data. However, a BN requires that the relations among the domain variables be the same for all values of the class variable. This not only causes the model to be limiting but is also considerably more complex for structure learning and interpretation. In contrast, a Bayesian Multinet (BMC) allows different relations between variables. This means that the dependence and independence relations of variables for one value of the class variables is not required to be the same for other class values.

A Bayesian Multinet Classifier for a set of variables $\{A_1...A_n\}$, is composed of a set of Bayesian Networks, $\{B1, \ldots, B_{|C|}\}$, where each network corresponds to one value of the $|C|$ possible values of the class C. To implement this idea, the training data set is partitioned and new training subsets are obtained for each value of the class variable. The resulting probability distributions P_{B_i}, for each of B_i networks, approximates the joint distribution of the variables, given a specific class, that is, $P_D(A_1...A_n|C = c_i)$. The set of networks combined with a prior probability on C, $P(C)$, is called a Bayesian multinet. Formally, a multinet is a tuple $M = \langle C, B_1 \dots B_k \rangle$ where P_C is a distribution on C, and B_i is a Bayesian network over $\{A_1, \dots, A_n\}$ for $1 \leq i \leq k = |C|$. A multinet M defines a joint distribution:

$$P_M(C, A_1, \dots, A_n) = P_C(C) \times P_{B_i}(A_1, \dots, A_n) \text{ when } C = c_i.$$

When learning a multinet, we set $P_C(C)$ to be the frequency of the class variable in the training data, that is, $P_D(C)$, and learn the networks B_i in the manner just described. For the classification phase, a class is chosen that maximizes the posterior probability $P_M(C|A_1, \dots, A_n)$. A example of Bayesian Multinets is shown in the figure below.

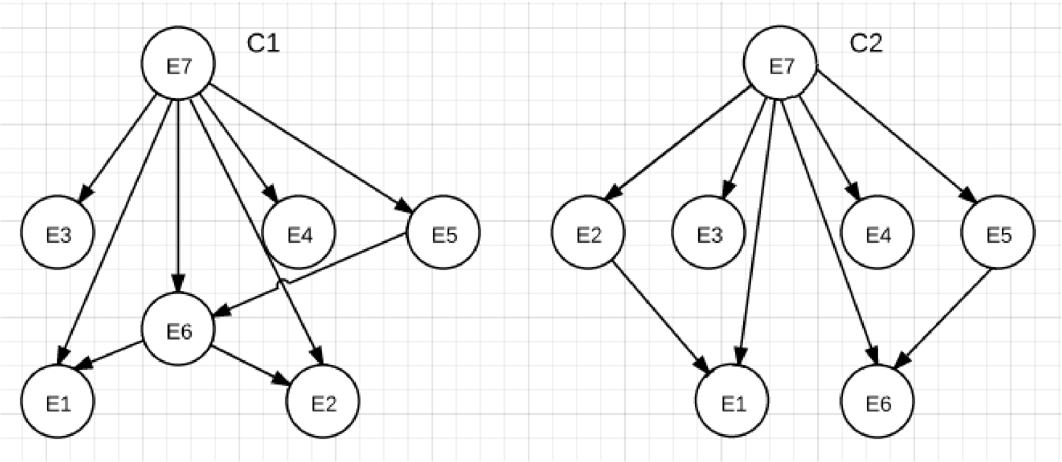


Figure 3: Example of an example Bayesian Multinet with two classes.

The BMC can be viewed as a generalization of any type of BNC when all the networks of the BMC have the same structure of the BNC [11]. While each of the individual networks for each class value will still need to be learned, the Bayesian Multinet model is generally less complex compared to a typical Bayesian Network Classifier. This is because each network will have a lower number of nodes than in a BNC since each sub-network models a simpler problem. This also means that the computational complexity of the BMC is also typically smaller. The accuracy is also higher when compared to a BNC since each network models a more specific set of data and ince both the complexity of structure learning and number of probabilities to estimate increases exponentially with the number of nodes in the structure [3].

The structure of each of the local networks is usually learned using a joint probability-based score that is less specific to classification, i.e., classifiers based on structures providing high scores are not necessarily accurate. This is specially true since these scores are less discriminative for the learning of multinet classifiers because the network structure is computed using only samples from one specific class and ignoring samples from the other classes. Improvements for the Bayesian Multinets have been proposed. Bayesian class-matched multinets (BCM^2) avoid the problem of losing discriminative power by learning local networks using a detection-rejection measure [18].

Instead of learning from only the training patterns for a specific class, each training pattern is labeled as *native* or *foreign*. The structure is then evaluated using a scoring mechanism based on the ability of the structure's ability to detect native and rejecting foreign patterns. In other words, both true positives (number of correct detections) and true negatives (number of correct rejections) on a training set are taken into consideration. This provides more discriminatory power to the model which is especially useful in classification problems and further improves performance of classification.

3 Experiment

3.1 Scenario

There is a plethora of possible scenarios that may involve digital investigation. All physical crimes such as murders, kidnappings and robberies may involve digital forensics if an electronic device is found on the crime scene, because it may contain vital information that help in solving the case. There is also whole variety of so called digital crimes such as violations of copyright law, violations of corporate computer usage policy, forgeries and also cyber terrorism. In general, digital investigation should be considered if a digital device is either a target or instrument of a crime or it is found on the crime scene. For the purpose of this Case Study a following simple scenario will be used.

The perpetrator finds a target on facebook. He can see that his target is a customer of a well known electronic book store. He identifies an e-mail address of the target on facebook that he uses as a login to the electronic book store. Then, he tries to break the password using brute force method with dictionary obtained from information available on target's profile page. After several iterations he gains access to the account – correct password is the name of the target's dog. He purchases one book and downloads it in pdf format to his computer. He is able to pay for the transaction because credit card number is stored in the system.

Three days later company that is a owner of the electronic book store receives a phone call from the customer who informs them that his bank account was debited for the purchase that he had not made. Employer of the company checks the account of the customer and sees that there was a transaction made three days before. Because customer denies that he made it a Security Officer of the company is informed and account's history is checked. In the log of login history important information is found – there is a successful login attempt preceded by several failure attempts, all made from different IP address than usual for this customer. In this situation Police is informed and with the help of local ISP the suspect is found. In his flat a laptop is found and collected as an evidence. This laptop, together with server from book store constitute a bounded operational system which needs to by analysed in the digital investigation.

3.2 Bounded system reconstruction

To recreate the bounded operational system and to simulate different scenarios and sequence of events, we setup two virtual machines. The first machine is designated the Server. This virtual machine has Ubuntu 12.04 installed with services that are to be expected on a regular webserver such as Secret

Shell Login (SSH), an opensource database server (mysql) etc. On this machine, we also setup an opensource Web Store with capability of handling online transactions. This Web Store is a popular webstore available online which is used widely by developers and small businesses. It is implemented Ruby on Rails web development stack and is setup to utilize mysql server installed on the system.

Ruby on Rails is one of the most popular platforms for development of web application today. It is preferred because of the ease of use, speed of development, flexibility, performance and due to its modular approach. It also provides extensive logs for each request that is processed by the servers. For all kinds of web applications, it is a common practice to store logs several days old or in some more sensitive cases (where information security is important and financial activity is involved) months in order to track back a sequence of events.

It is worth noting here that logging is part of almost all of the systems available and is a standard practice. The advantage of using Ruby on Rails is that the logs and the information is collected into a single log file and is easier to analyse and to recreate the sequence of events as they happened. All standard industry solutions also provide extensive logging but the information may be slightly more scattered. For instance, PHP/Apache/Mysql, the most popular web development utilised on the web provide extensive information but this information is scattered in three different logs for each of the application.

For the purposes of this experiment, we utilize the Webserver logs available as default to extract realtime information as well as past history of activities from the system. These logs give information about the interaction of users with the webserver. For each web request the server processes, information about the user (if a member of the site), IP address from where the request originated, the action that they performed (e.g. opened the login page), various variables or inputs that they provided in posted forms are stored. Also stored are information from within the system such as the queries that are run in the database, warnings and errors provided by the server.

The second virtual machine is the computer of the suspect. For simplicity, this machine also has Ubuntu Operating System installed on this system. We use this system to make requests to the server in order to simulate the act of infiltration. Both the Web Server and the computer of the suspect are on the same network. In this way, we are able to distinguish the activities of the suspect from the customer's activities. While this may seem to be simplistic, it does not affect the outcome if the particular machine is not on the same network. The only difference being that in a practical scenario (where the suspect is not on the same network), the IP address of the suspect's system would not be from the local domain addresses but instead would be a regular IPv4 address. However, tracking of the computer would still be possible. Using this system, we can perform various investigative tasks such as view browser history, analyse the hard disk and detect events which may support of negate whether the suspect is guilty.

3.3 Information Extraction

In any forensic study, whether it is specifically digital forensics or not, securing and extracting data is one of the most important and crucial steps. This step is often the determining factor in the success or failure of the prosecution. Not only is it important to secure, and collect data, it is also imperative to extract useful information from the raw data. Unlike a typical investigation, data from a digital

”crime scene” can be extremely voluminous and have intricate correlations. Also, useful information (evidence) could be hidden, be present in different forms, or may even be obfuscated and seem harmless. Due to the complex nature and vast possibilities, the data extraction step in digital forensics is often very challenging and requires work from case to case. For example, in a case dealing with internet fraud, data related to internet, web history, email logs are more interesting while in a case related to child pornography may require an in depth analysis of the data stored (or deleted) from the storage media.

As the field of Digital Forensics has developed, many tools have been developed in order to extract interesting information from systems seized from crime scenes. Many tools exist for performing analysis of various filesystems, collecting meta data of files and even deleted files. However, these tools are often for specific information and analysis. In many cases, specific information needs to be extracted and different tools (sometimes new) may need to be developed. While the main goal of this study is not to perform in-depth data recovery phase of the Digital Forensics, some data extraction has been employed.

In the scenario presented in this case study, one of the important pieces of evidences is the presence or absence of certain files believed to have been illegally obtained by the suspect. In order to achieve this, we utilize an opensource software named Digital Forensics Framework (DFF). This software has the capability of loading evidence files and performing various operations on them, including finding deleted files, and extracting meta data. Since DFF has the capability of loading and evaluating disk images made by Virtual Machines, we use it to evaluate disk images of our virtual machines to look for files we are looking for - which may or may not be deleted.

Another source of valuable information on systems are the logs that are maintained by the operating systems. One of the most important of these logs are the logs maintained by the authentication systems which contain information about successful logins, failed logins, remote sessions initiated or attempted etc. These logs are widely used in order to obtain information about intrusion detection and very often have sophisticated warning systems. Unauthorised access could explain several possible situations in this case study. For instance, if the computer of the suspect was remotely accessed, or if his password was ’guessed’ using a brute force attack; that would be considered strong evidence against the conviction of the suspect. Similarly, it is possible that the webserver of the Online Webstore was compromised and the same or a different suspect got access to sensitive information like client’s passwords. In this case also, the information about unauthorized access can prove vital evidences in Digital Forensics. However, the information stored in operating system logs is not readable and it is difficult to extract structured and timed events from them. In order to utilise information in the authentication logs, we have written a simplistic parser that extracts relevant information from the authentication logs (in Ubuntu versions used in our virtual machines /var/log/auth.log). The output of these logs is a list of events in a CSV file with the username, timestamp, method of access and the response (success/failure) from the authentication system. This gives the information in a structured format and its easy to apply filters and to extract patterns.

Standard webservers, whether they are commercial or opensource, also maintain detailed logs of each request. Ruby on Rails, which is the framework of choice for this case study, also maintains such logs. In order to extract information from these logs, we have written a parser to extract information for each request including the IP address, the page that is requested and the user information associated with the request. This could also be used to identify patterns easily. Specific information, such as purchases of users and their originating IP addresses can also be obtained from the database tables related to purchases.

However, since the goal of this study is not to do an extensive information retrieval from systems under Forensics Study, most of the information remains to be observed manually as we study the behaviour and output of our models.

3.4 Constructing a Bayesian Network

In this part we will focus on constructing a Bayesian Network for the proposed scenario. First thing to do is to enumerate propositions to be considered as a part of the network:

H seized computer was used as a tool for stealing victim's credentials to the electronic book store and illegally purchasing a book

$\mathbf{H}_1$ stealing victim's credentials using brute force attack was performed

$\mathbf{H}_2$ purchase of the electronic book was performed using stolen victim's account

$\mathbf{E}_1$ a victim's credentials are present on suspect's computer

$\mathbf{E}_2$ a dictionary used for brute force attack is found on suspect's computer

$\mathbf{E}_3$ a victim's facebook profile is present in suspect's web browser history/cache

$\mathbf{E}_4$ a website of electronic book store is present in suspect's web browser history/cache

$\mathbf{E}_5$ a file containing purchased book is/was present on suspect's computer

Looking on the H node one can see that it is not a goal of forensics analysis to decide on suspect's guiltiness or innocence. This decision belongs to the court which takes into account not only digital evidence but the whole big picture containing other evidences and testimonies. Thus, the forensics investigation focuses on proving whether certain events occurred or not - in this case we want to establish whether or not the suspect's personal computer was used as a tool for committing a crime.

3.4.1 Topology of the network

According to [16] one can distinguish between hypothesis and evidence nodes in the network. Hypothesis nodes present hypothetical scenarios of what happened and are never observable whereas evidence nodes correspond to the empirical observations made based on the collected evidence material. The arcs are directed from hypothesis to the evidence according to the top-down way of bayesian inference. Using forensics jargon they are said to go from explenans to explenandum. H node constitute main hypothesis and H_1 and H_2 are sub-hypotheses that tries to explain the way the H happened. Nodes E_i are evidence nodes. All of the nodes are discreet binary nodes with two states *yes* and *no*. Structure of the proposed network is presented in the Figure 4 and it is the same structure as used in [8].

3.4.2 Probability distributions

There are several ways of assigning probability distributions to the networks. While having sample data it is possible to learn the model from it using for example Maximum Likelihood (ML) method. Often, there is not enough data or missing values are present thus experts knowledge may be mixed

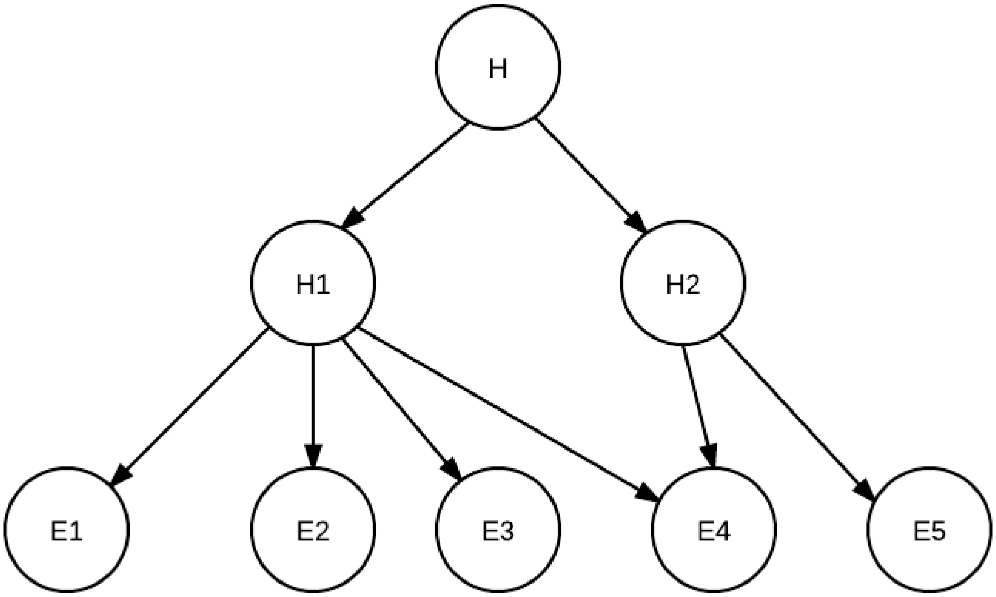


Figure 4: Network structure.

with data to learn the model using for instance Maximum Aposteriori (MAP) method. In some cases, while there is no data available only experts' knowledge may be used.

While assigning conditional probabilities it is important to pay attention and not fall into so called *fallacy of the transposed conditional*. "The fallacy occurs when, from the fact that if A has occurred, then B occurs with a high probability, it is erroneously concluded that if B has occurred, then A occurs with high probability" [16]. Following example illustrates this situation. A man was seen running away from the house in which a crime was committed. It is believed that probability that the man was running away given that he committed a crime – P(the man was running away | the man committed a crime) – is high because it is expected that he would want to leave the crime scene as quickly as possible. On the other hand, probability that the man committed a crime given that he was running away – P(the man committed a crime | the man was running away) – is not necessarily high because there are many possible explanations of the fact that he was running away and the sole fact of running away does not make the commitment of the crime more probable.

Because we have no data to learn the parameters of the network we assign them based on common sense knowledge trying to act as the experts. The probability distributions for the network are presented in the Table 3.4.2. A prori probability of H being *yes* is set to 0.7 because we assume that in 70% of police interventions in cooperation with ISP the suspect is found guilty. Remaining 30% may represent more complex situations like involvement of third person – possibly even from inside of the book store company for instance manipulating with the log files. Similarly, we can assume that in 60% of the infringements in the similar circumstances brute force attack is used to obtain credentials of the victim and again in 60% of cases perpetrators simply purchase items after using victims' stolen

H_1	YES	NO
H - YES	0.60	0.40
H - NO	0.05	0.95

H_2	YES	NO
H - YES	0.60	0.40
H - NO	0.02	0.98

E_1	YES	NO
H_1 - YES	0.60	0.40
H_1 - NO	0.01	0.99

E_2	YES	NO
H_1 - YES	0.80	0.20
H_1 - NO	0.05	0.95

E_3	YES	NO
H_1 - YES	0.55	0.45
H_1 - NO	0.15	0.85

E_5	YES	NO
H_2 - YES	0.90	0.10
H_2 - NO	0.10	0.90

E_4		YES	NO
H_2 - YES	H_1 - YES	0.80	0.20
	H_1 - NO	0.70	0.30
H_2 - NO	H_1 - YES	0.60	0.40
	H_1 - NO	0.05	0.95

Table 1: Probability distributions in the network from Figure 4

account. In case the seized computer was not used as a tool of this malicious activity probabilities of events H_1 and H_2 are close to 0. For the evidences, the probability that an evidence is present on the seized machine is high given that its parent hypothesis node value is *yes* and vary from 0.6 to 0.9 and rather low given that its parent hypothesis node value is *no* – value from 0.01 to 0.15. These assumption are ment to be rather careful although it is impossible to avoid subjectivity with this approach. However, as shown in [14] this structure has reasonable stability for the variation of parameters, thus we can expect that it will provide reliable results.

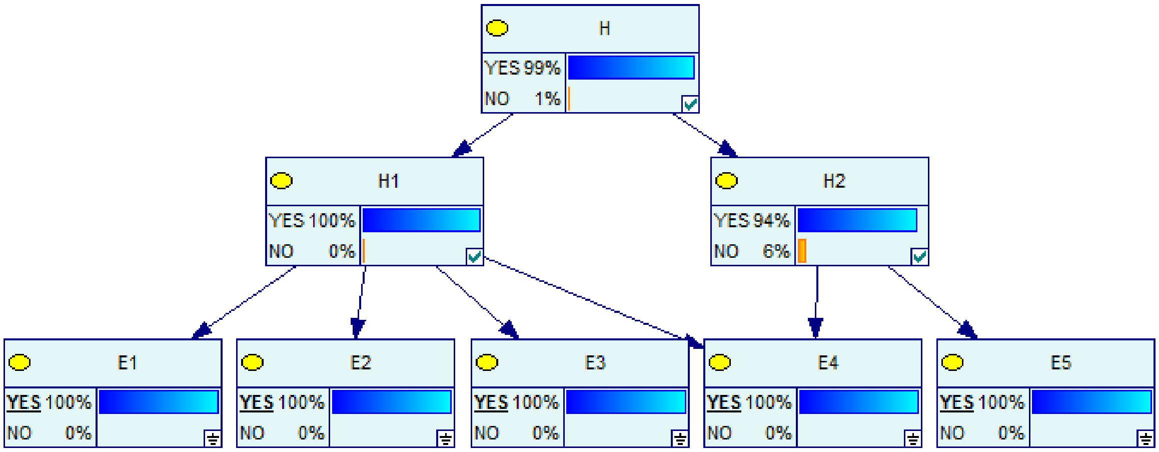


Figure 5: Network when all evidences are found.

3.4.3 Network implementation

To check the behaviour of the proposed network was implemented using GeNIe 2.0 software tool that provides graphical user interface and allows for interaction with the network. As expected, when all evidence are found the probability that hypothesis H is equal to *yes* is 99% – Figure 5. In this representation each node (random variable) is presented as a box containing a bar chart with corresponding probability values – in this case all evidence nodes are set to 100% which means that all evidences are found. In case no evidence is found the network assign probability value of 30% to H equal to *yes* and 70% to H equal to *no* which again has a natural interpretation that when no evidence is found it is very probable that the seized computer was not used for performing malicious activity – Figure 6. Verdict remains stable when one of the evidences is missing and vary from 97% to 99%. Network also allows to check which of the evidences is the most relevant for the case – to do this it is enough to check which of them influence the H node the most when found. In our case the most relevant evidence seems to be E_1 – finding victim's credential on seized computer. This single evidence causes raise of probability that node H takes value *yes* to 96% when all the other nodes are unobserved. The lest relevant evidence is E_3 which causes raise of probability that node H takes value *yes* to 84%.

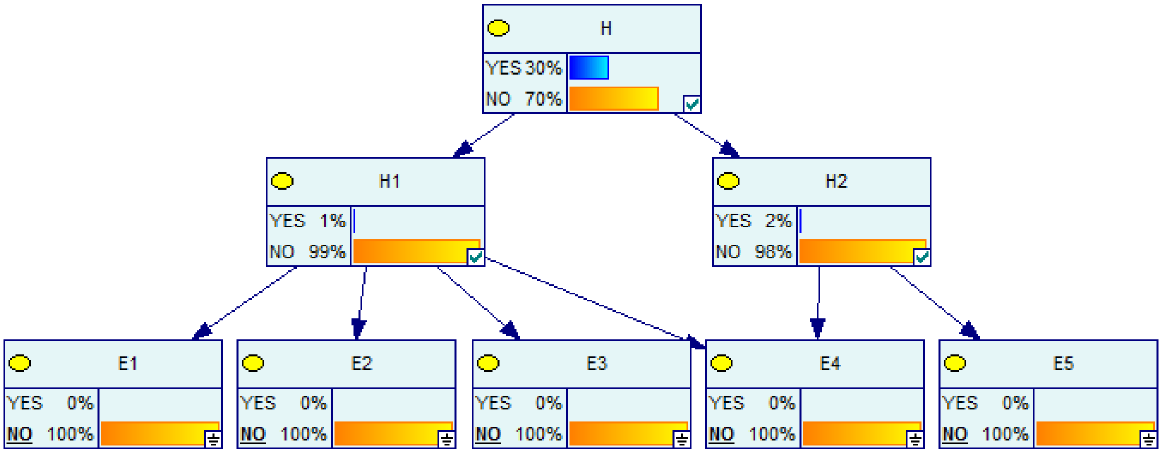


Figure 6: Network when no evidence is found.

3.5 Constructing a Bayesian Network — multi-nets approach

The network proposed in the previous subsection gives adequate results but seems too simplistic to fully grasp complexity of a real life situation. Some of the variables have temporal dimension and also dependencies between them may vary depending on the situation. It is also difficult to include various different hypothetical scenarios in the structure of the network due to the quickly growing complexity of the structure and probability distributions. Because the situation we are trying to model also looks similar to the classification task in which values of H node correspond to class values, we will try to apply multinets approach.

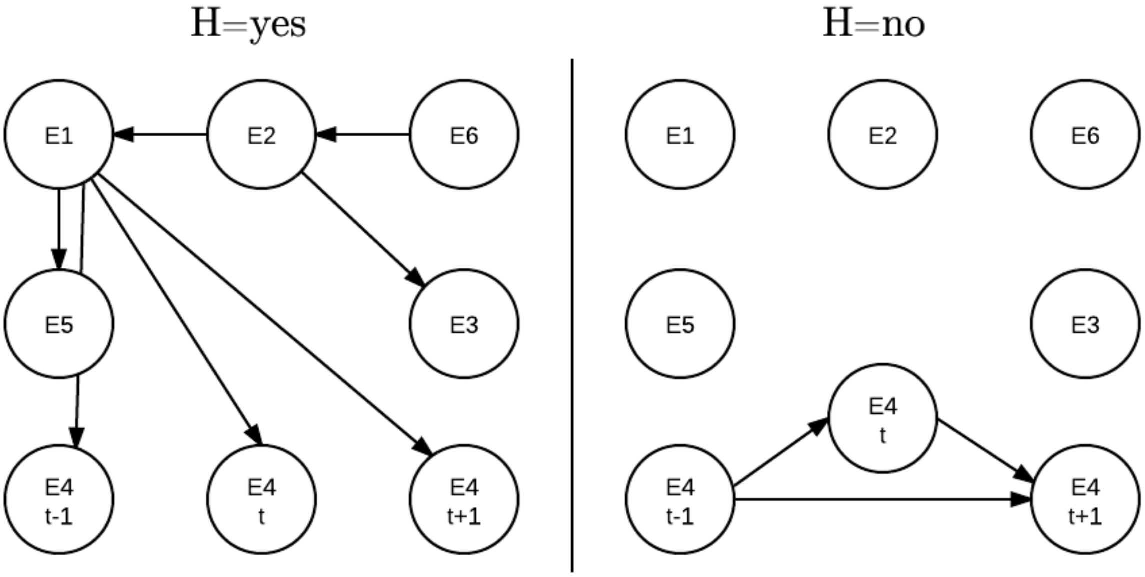


Figure 7: Multinets structure.

E_1	YES	NO
E_2 - YES	0.60	0.40
E_2 - NO	0.10	0.90

E_4t-1	YES	NO
E_1 - YES	0.60	0.40
E_1 - NO	0.30	0.70

E_2	YES	NO
E_1 - YES	0.60	0.40
E_1 - NO	0.40	0.60

E_4t	YES	NO
E_1 - YES	0.90	0.10
E_1 - NO	0.10	0.90

E_5	YES	NO
E_1 - YES	0.70	0.30
E_1 - NO	0.10	0.90

E_3	YES	NO
E_2 - YES	0.55	0.44
E_2 - NO	0.20	0.80

E_4t+1	YES	NO
E_2 - YES	0.10	0.90
E_2 - NO	0.50	0.50

E_6	YES	NO
	0.50	0.50

Table 2: Probability distributions – multinet H *yes*

3.5.1 Topology of the network

In this case two networks will be created – one for the case when the value of the H node is equal to *yes* and another to the case when the value of H node is equal to *no*. To simplify the nodes H_1 and H_2 will be no longer present. New nodes are added:

E4$_{t-1}$
E4$_t$
E4$_{t+1}$
E$_6$ brute force attack detected on server.

E4$_t$ indicates now presence of the electronic book store in the web history/cache of the suspect at the time when the crime was committed (the time can be checked in the server logs) and t-1 and t+1 indicate the presence of the same website in the web history/cache of the suspect in the time before and after committing a crime respectively. Each network is characterized by different dependency relationships between variables. If the seized computer was used as a tool for committing a crime then it is likely that the perpetrator would stop visiting electronic web store after the illegal act. On the other hand, if this computer was not used for the malicious purpose than the probability of visiting the website of the store depends whether or not this website was visited before, e.g. if the person is interested in this book store. Also in this case most of variables seem to be rather independent from each other. Structures for two proposed multinets are presented in Figure 7.

3.5.2 Probability distributions

Precise values of probability distributions are presented in Table 3.5.2 for multinet corresponding to H value of *yes* and in Table 3.5.2 for multinet corresponding to H value of *no*. Our intuition for assigning

E$_1$	YES	NO
	0.01	0.99

E$_3$	YES	NO
	0.20	0.80

E$_6$	YES	NO
	0.50	0.50

E$_2$	YES	NO
	0.01	0.99

E$_5$	YES	NO
	0.08	0.92

E$_4t-1$	YES	NO
	0.60	0.40

E$_4t$	YES	NO
E$_1$ - YES	0.90	0.10
E$_1$ - NO	0.10	0.90

E$_4t+1$		YES	NO
E$_4t-1$ - YES	E$_4t$ - YES	0.90	0.10
	E$_4t$ - NO	0.80	0.20
E$_4t-1$ - NO	E$_4t$ - YES	0.80	0.20
	E$_4t$ - NO	0.10	0.90

Table 3: Probability distributions – multinet H *no*

those values was following. Given that seized computer was used for stealing victim's account using brute force attack it is more probable that a dictionary used for the attack will be found on it. Finding dictionary increases probability of finding victim's credential and this fact increases probability of finding a stolen book and the suspect's computer. On the contrary, given that seized computer was not used for the malicious activity variables are considered independent and the probability of them to happen is low. Both structure of the networks and assigned probability distributions are highly subjective. We tried to follow common sense assumptions although these parameters would have to be either learned from data or consulted with specialists in the field.

3.5.3 Network implementation

For implementing multinets also GeNIe 2.0 software was used. Diagram for testing hypotheses in case when all evidences are found is presented in the Figure 8. There is no H node in these networks hence interpretation is not obvious and requires additional computations. Each of the networks represent $P(E_1...E_6|H)$ and we want to find $P(H|E_1...E_6)$ and those two probabilities are proportional to each other. $P(E_1...E_6|H)$ can be obtained using GeNIe 2.0 software – *Probability of Evidence* function. Thus to compute $P(H|E_1...E_6)$, $P(E_1...E_6|H)$ need to be diveded by normalization factor equal in our case to $P(E_1...E_6|H = yes)P(H = yes) + P(E_1...E_6|H = no)P(H = no)$. After performing computation probability that value of H is *yes* is equal to 100% and probability that value of H is *no* is equal to 0%. When no evidence is found (with E$_4t-1$, E$_4t$, E$_4t+1$ set to *yes*) respective values are 5.85% and 94.15% which can be interpreted that in this case seized machine was not used for malicious activity. Multinet works as expected and gives similar results to previous approach, although deeper comparison of the performance is difficult because of the subjectivity of the structure and parameters.

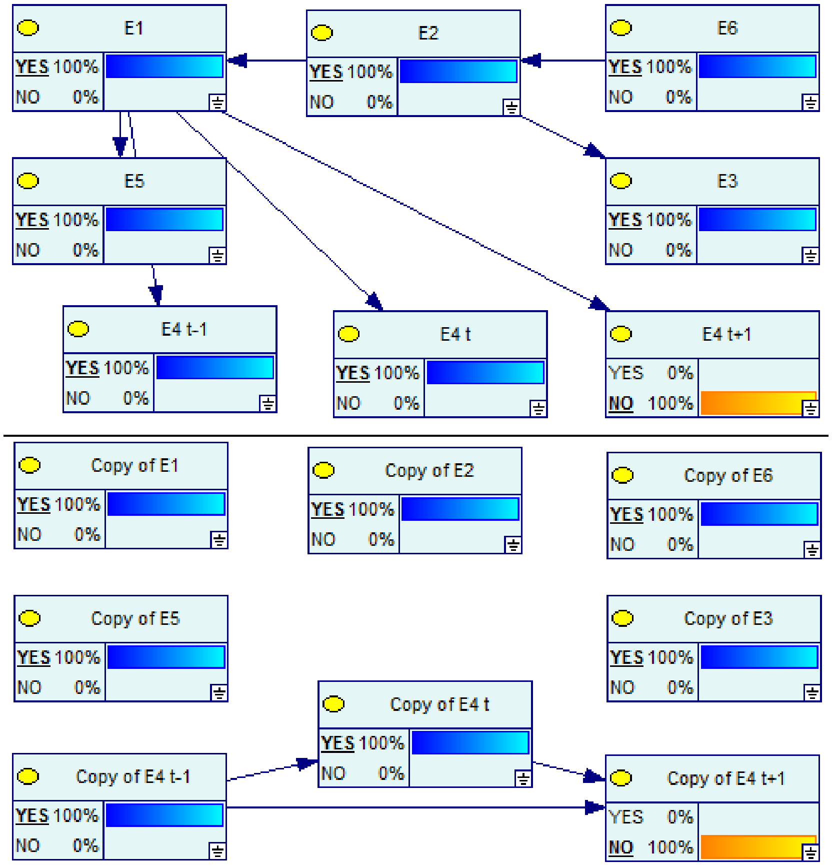


Figure 8: Multinets implemenation.

3.5.4 Extensions

Moreover, multinets approach for modelling the situation allows for convenient extensions of the model and testing alternative hypotheses. Let's imagine another probable scenario in which a suspect's computer is used as a proxy for commiting the crime. There is a need for looking for new evidences like malware software installed in his computer. Also dependencies between varibles in this case would be different and also new evidences could appear.

4 Conclusion

Digital Forensics is a complex field in which many areas can be distinguished where analytics and computer science techniques can be applied. In this Case Study we focused on the analysis and interpretation of the evidence and how Bayesian Networks can support this task. We have developed a crime scenario and recreated the situation described in it. After collecting evidences from the simulated bounded operational system we proposed several approaches from the Bayesian Network domain to support/refute certain hypotheses about the scenario. Firstly, we followed the approach described in [8] with a simple approach and then we proposed more robust model to simulate the situation based on Bayesian Multinets. The simple structure of the first approach may not seem robust enough to model complex situations. For instance, the structure implies limitations to extensions that one is more likely to encounter in a real world investigation with many more variables and complex interactions. However, its simplicity is a huge advantage because it can easily be presented and understood by judges and jury in the court and the results, structure and the dependencies depicted are easy to interpret. This cannot be said about the multinets approach which may not convince decision makers in the court because its concept may be not well understood even though it provides more robust, precise and flexible way of modelling crime circumstances. The process of obtaining the probabilities may be too complicated to interpret and understand. The choice between these two models is a trade-off between simplicity and robustness. As a court trial is often strictly dependent on psychological and sociological aspects of interactions between people participating in it, simple models can have greater impact on the proceedings and thus it may be preferred and more complex models introduced over time.

The focus of this study was to investigate and demonstrate various Bayesian Network techniques in the field of Digital Forensics which still remains to be a new field in Computer Science. Future works, extending on this case study could be an experiment which more complex scenarios with a higher number of variables and compare the simple and the more specific structure like Multinets. A study of the impact of Hidden Causes on the Multinet approach may also be worthwhile.

References

[1] Megan Carney and Marc Rogers. The trojan made me do it: a first step in statistical based computer forensics event reconstruction. *International Journal of Digital Evidence*, 2:1–11, 2004.

[2] Eugene Charniak. Bayesian networks without tears. *AI MAGAZINE*, 12(4):50–63, 1991.

[3] David Heckerman Dan Geiger. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82:45–74, 1996.

[4] Olivier De Vel, Nianjun Liu, Terry Caelli, and Tiberio S. Caetano. An embedded bayesian network hidden markov model for digital forensics. In *Proceedings of the 4th IEEE international conference on Intelligence and Security Informatics*, ISI'06, pages 459–465, Berlin, Heidelberg, 2006. Springer-Verlag.

[5] Felix C. Freiling and Bastian Schwittay. A common process model for incident response and computer forensics. In *IMF'07*, pages 19–40, 2007.

[6] Tina Eliassi-Rad Christos Faloutsos Hanghang Tong, Yasushi Sakurai. Fast mining of complex time-stamped events. `http://eliassi.org/papers/tong-cikm08.pdf/`.

[7] Veena H Bhat; Prasanth G Rao; Abhilash R V; P Deepa Shenoy; Venugopal K R; Patnaik L M. A data mining approach for data generation and analysis for digital forensic application. *IACSIT International Journal of Engineering and Technology*, Vol.2, No.3, June 2010.

[8] K.P.; Frank Y.W. Law; Pierre K.Y. Lai Michael, Y.K. Kwan; Chow. Computer forensics using bayesian network: A case study. The University of Hong Kong.

[9] Frank Y.W. Law Pierre K.Y. Lai Michael Y.K. Kwan, K.P. Chow. Computer forensics using bayesian network: A case study. `http://www.cs.hku.hk/cisc/forensics/papers/BayesianNetwork.pdf/`.

[10] Kara L. Nance, Brian Hay, and Matt Bishop. Digital forensics: Defining a research agenda. In *HICSS*, pages 1–6. IEEE Computer Society, 2009.

[11] Moises Goldszmidt Nir Friedman, Dan Geiger. Bayesian network classiers. *Machine Learning*, pages 131–163, 1997.

[12] Olivier Pourret Pierre-Henri Wuillemin Patrick NAÏM, Philippe Leray. *Les Réseaux bayésiens.* 2011-07.

[13] Judea Pearl. *Causality Models, Reasoning and Inference.*

[14] Michael Y.K. Kwan Kam-Pui Chow Frank Y.W. Law Pierre K.Y. Lai Richard E. Overill, Jantje A. M. Silomon. Sensitivity analysis of a bayesian network for reasoning about digital forensic evidence. `http://www.cs.hku.hk/cisc/forensics/papers/10_07.pdf/`.

[15] Mohd Taufik Abdullah; Ramlan Mahmod; Abdul Azim Ab. Ghani; Mohd Zain Abdullah; Abu Bakar Md Sultan. Advances in computer forensics. *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.2, February 2008.

[16] F. Taroni. *Bayesian Networks And Probabilistic Inference in Forensic Science.* Statistics in Practice. Wiley, 2006.

[17] M. Petkovic V. Mihajlovic. Dynamic bayesian networks: A state of the art. `http://doc.utwente.nl/36632/1/0000006a.pdf/`.

[18] Boaz Lerner Yaniv Gurwicz. Bayesian class-matched multinet classifier.

[19] R.F.; Marks D.G.; Pollitt M.M.; Sommer P.M. Yasinsac, A.; Erbacher. Computer forensics education. *Security & Privacy, IEEE*, 1 Issue:4:9, July-Aug. 2003.