FORENSICS BASICS ON MACHINE TO BRAIN INTERFACE

The area of forensic investigations and the search for evidences in crime scenes have been always the center of attention of all parties involved in crimes. Footprints and many other techniques have been exploited in crimes investigations. However, those classic evidences are not always available in all crime scenes. Nevertheless, the brain of the crime's preparator is one evidence that is always there during the crime. By searching the memories in the suspects' brains for any crime-relevant information, a subject is determined to be innocent or guilty. This is done by an evolving technology that is nowadays a hot research topic worldwide; Brain-Computer Interfaces (BCI) which is defined as ***"a communication system that does not depend on the brains normal output pathways of peripheral nerves and muscles***". The communication between the human brain and computers is still under research for it folds numerous applications in the medical field, military and computer advancement. Perhaps the most recent potential application of the BCI technology is its use in forensic applications. The concept of brain computer interface as a forensic tool is as follows: a suspect takes a guilty knowledge test (GKT) which contains crime related information.

A Brain-Computer Interface (BCI) acquires brain signals, analyzes and translates them into commands that are relayed to actuation devices for carrying out desired actions. With the widespread connectivity of everyday devices realized by the advent of the Internet of Things (IOT), BCI can empower individuals to directly control objects such as smart home appliances or assistive robots, directly via their thoughts. However, realization of this vision is faced with a number of challenges, most importantly being the issue of accurately interpreting the intent of the individual from the raw brain signals that are often of low fidelity and subject to noise. Moreover, pre-processing brain signals and the subsequent feature engineering are both time-consuming and highly reliant on human domain expertise.

BCI-based cognitive interactivity offers several advantages. One is the inherent privacy arising from the fact that brain activity is invisible and thus impossible to observe and replicate. The other is the convenience and real-time nature of the interaction, since the human only needs to think of the interaction rather than undertake the corresponding physical motions (e.g., speak, type, gesture). However, the BCI-based human-thing cognitive interactivity faces several challenges. While the brain signals can be measured using a number of technologies such as Electroencephalogram (EEG), Functional Near-Infrared Spectroscopy (FNIR), and Magnetoencephalography (MEG), all of these methods are susceptible to low fidelity and are easily influenced by environmental factors and sentiment status (e.g., noise, concentration). In other words, the brain signals generally have very low signal-to-noise ratios, and inherently lack sufficient spatial or temporal resolution and insight on activities of deep brain structures. As a result, while current cognitive recognition systems can achieve about 70-80% accuracy, this is not sufficient to design practical systems. Second, data pre-processing, parameter selection (e.g., filter type, filtering band, segment window, and overlapping), and feature engineering (e.g., feature selection and extraction both in the time domain and frequency domain) are all time-consuming and highly dependent on human expertise in the domain.

Components OF BCI

The purpose of a BCI is to detect and quantify features of brain signals that indicate the user's intentions and to translate these features in real time into device commands that accomplish the user's intent. To achieve this, a BCI system consists of four sequential components:

1. Signal acquisition
2. Feature extraction
3. Feature translation
4. Device output.

These four components are controlled by an operating protocol that defines the onset and timing of operation, the details of signal processing, the nature of the device commands, and the oversight of performance. An effective operating protocol allows a BCI system to be flexible and to serve the specific needs of each user.

- **Signal Acquisition**

Signal acquisition is the measurement of brain signals using a particular sensor modality. The signals are amplified to levels suitable for electronic processing.The signals are then digitized and transmitted to a computer.

- **Feature Extraction**

Feature extraction is the process of analyzing the digital signals to distinguish pertinent signal characteristics (i.e., signal features related to the person's intent) from extraneous content and representing them in a compact form suitable for translation into output commands. These features should have strong correlations with the user's intent.

- **Feature Translation**

The resulting signal features are then passed to the feature translation algorithm, which converts the features into the appropriate commands for the output device (ie, commands that accomplish the user's intent).

- **Device Output**

The commands from the feature translation algorithm operate the external device, providing functions such as letter selection, cursor control, robotic arm operation, and so forth. The device operation provides feedback to the user, thus closing the control loop.

Digital Forensics

All kinds of storage and digital media (i.e. hard disks, USB memory sticks, zip drives, digital cameras, mobiles, etc.) are subject to investigation while doing a forensics analysis. As a result, the volume of data obtained when conducting an electronic crime scene investigation will inevitably increase. Yet, the true value of data is substantiated based on its ability to produce useful information that might help in decision-making.

In most cases, DF analysis is a conventional manual process where analysts or commonly called e-crime investigators would familiarize themselves with the domain, and with the help of some tools, they would generate reports and summaries to describe data insight. Nevertheless, it might be difficult for the e-crime investigator to comprehend the useful information on the collected data in its original form. Therefore, the collected data can be transformed into an understandable form by using some tools such as SANS SFIT, Sleuth Kit, and LastActivityView. Once the data transformation process is completed then it can be analyzed using some statistical tools. However, when the size and dimension of data increase, costs of investigation rise, or even when forensic analysis cases become more complicated; the manual process may quickly deteriorate. Therefore, when the data exploration goes beyond abilities of typical manual analysis e-crime investigators might look for a more reliable knowledge discovery method to process data. Machine Learning (ML) comes into sight as an effective method to facilitate producing possibly useful knowledge for decision makers. ML is the method of analyzing datasets from different viewpoints and reforming them into meaningful forms.

Normally, DF analysis aims to identify the perpetrators of digital incident by determining, classifying, and most importantly reconstructing the events related to a digital crime. Event reconstruction is a crucial process when conducting a DF investigation in the sense that it will facilitate the preparation of a timeline of a digital incident. In addition, it can provide some digital evidence about a cyber-crime.

Digital evidence can be defined as: reliable information obtained from reliable sources confirming or denying the occurrence of a cyber-crime. In general, event reconstructing is the process of reordering of incidents occurred on a storage or digital media device. There are several sources where reliable information can be gathered for reconstructing cyber-crime events such as web browser history-files, cookies, temporary files, log files, and file system. System files remain an important source of information for such a purpose. The routine operation of computer system might result in modifying system files. Identifying the system files affected by a digital-crime is an essential phase towards facilitating event reconstruction process. DF investigators can make use of the metadata associated with a file. Yet, system files might be manipulated by several applications. Therefore, identifying the exact sequence of actions affecting a file system is also important in order to ascertain that specific activities correspond to reliable application program(s) or to some sort of malicious program(s).

NEURAL NETWORK BASED DIGITAL FORENSICS

The set of experiments aims to assess the capability of Neural Network techniques in classifying which files have been manipulated by a particular computer program after analyzing the features (footprints) that a program leaves when accessing, updating, modifying, or deleting a specific file system. Determining the set of files affected by an incident would facilitate reconstructing previous events.

- **DATASETS DESCRIPTION**

A training dataset contains several footprints (features) related to file system activities (file system metadata) and system event audit log entries have been collected. File system, audit logs entries, and registry information are deemed to be the primary source for pinpointing digital evidence [19]. One advantage of combining the features from these three resources is that if one feature is messing or damaged in one source it may still be available in one another.

- **DATASET PREPARATION**

The quality of dataset items has a great impact on the performance of the produced ML models. Therefore, dataset preparation is an essential step towards building efficient and accurate models. In addition, data preparation would increase generalization ability of the generated models.

- **MODELLING NEURAL NETWORK**

An Artificial Neural Network (ANN) is an information processing technique that is inspired by the way the biological nerve system processes information. A trained ANN model is said to be an expert in the field it is applied to because it can accurately forecast the class value of new unseen examples. ANN proved its superiority in different domains and that can be attributed for different reasons such as nonlinearity, adaptiveness, generalization ability, and fault tolerance. There are several methods for training ANN models. The backpropagation algorithm is the most frequently used method. Backpropagation is usually implemented by using the feedforward architecture. The main idea in feedforward is to propagate the error through the hidden layers to update the weights of the model. The ANN architecture adopted in this research is the feedforward multi-layer perceptron, which consists of input, hidden, and output layers.

Below figure depicts the main phases of creating such architecture.

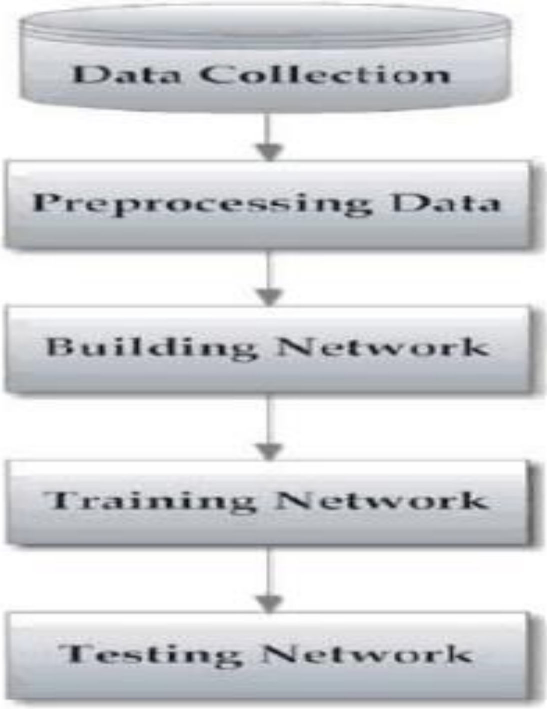

Fig- Phases of creating the DF model