USER ATTRIBUTION IN DIGITAL FORENSICS THROUGH MODELING KEYSTROKE AND MOUSE USAGE DATA USING XGBOOST

by

Shruti Gupta

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Polytechnic Institute

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Marcus K Rogers, Chair

Purdue Polytechnic Institute

Dr. John A Springer

Purdue Polytechnic Institute

Dr. Umit Karabiyik

Purdue Polytechnic Institute

Dr. Richard Adeyemi Ikuesan

Zayed University

Approved by:

Dr. Kathryne A. Newton

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ACC	Accuracy
ANN	Artificial Neural Networks
ANOVA	Analysis of Variance
BEA	Behavioral Evidence Analysis
CPU	Central Processing Unit
EER	Equal Error Rate
FAR	False Acceptance Rate
FMR	False Match Rate
FNMR	False Non-Match Rate
FN	False Negative
FP	False Positive
FRR	False Rejection Rate
GBM	Gradient Boosting Machine
GPS	Global Positioning System
GUI	Graphical User Interface
HCI	Human Computer Interaction
ML	Machine Learning
NN	Nearest Neighbor
OOB	Out Of Bag (data)
OS	Operating System
PCA	Principle Component Analysis
PII	Personally Identifiable Information
RBFN	Radial Basis Function Networks
SU-AIS BB-MAS	Syracuse University and Assured Information Study - Behavioral Biometrics Multi-device and multi-Activity data from Same users
SVM	Support Vector Machine
TP	True Positive
TN	True Negative

ABSTRACT

The increase in the use of digital devices, has vastly increased the amount of data used and consequently, has increased the availability and relevance of digital evidence. Typically, digital evidence helps to establish the identity of an offender by identifying the *username* or the user account logged into the device at the time of offense. Investigating officers need to establish the link between that user and an actual person. This is difficult in the case of computers that are shared or compromised. Also, the increasing amount of data in digital investigations necessitates the use of advanced data analysis approaches like machine learning, while keeping pace with the constantly evolving techniques. It also requires reporting on known error rates for these advanced techniques. There have been several research studies exploring the use of behavioral biometrics to support this user attribution in digital forensics. However, the use of the state-of-the-art XGBoost algorithm, hasn't been explored yet. This study builds on previously conducted research by modeling user interaction using the XGBoost algorithm, based on features related to keystroke and mouse usage, and verifying the performance for user attribution. With an F1 score and Area Under the Receiver Operating Curve (AUROC) of .95, the algorithm successfully attributes the user event to the right user. The XGBoost model also outperforms other classifiers based on algorithms such as Support Vector Machines (SVM), Boosted SVM and Random Forest.

1. INTRODUCTION

This chapter provides an introduction to the motivation behind the research study, the significance of the study, the research question, the assumptions, limitations, and delimitations of the study.

1.1 Background

The Industrial Era of computing began in the 1940s [1] but the processes to investigate computers were not defined until much later [2]. This is because until the 1980s, computers were an industrial appliance, and *personal computers* were not owned within a typical household. By the mid-90s investigators recognized the need to define new tools and processes to analyze digital information, with the International Organization on Computer Evidence (IOCE) being established in 1995 [3]. Over the next decade, the use of technology became ubiquitous, and so did the relevance of technology-related evidence, or *digital evidence* in legal scenarios. Locard's Principle of Exchange states that any individual interacting with a crime scene will leave behind some traces of their presence [4]. In a digital environment, this implies that any user interaction with a computer leaves behind digital evidence. Digital evidence is evidence related to all digital devices including cell-phones, gaming consoles, GPS systems, digital cameras, and online websites such as Facebook, YouTube etc. [5]. Census data published in 2018 reported that 89% of American households used a computer [6], indicating the widespread significance of digital evidence.

As digital evidence gained prominence in the courtroom, the formalization of digital forensics made great strides. Forensics refers to the use of science and technology to investigate and establish facts for the courts of law [7]. This included the development of processes and tools for digital evidence collection, analysis, and presentation [2]. Over the past few decades, commercial vendors have produced several hardware and software tools that gather and analyze digital artifacts [8]. Standard digital forensic techniques consist of examining these artifacts to gather information related to the person using the device. However, this pertains to the digital user or the *username*. Determining the actual person using the device at the time of the offense may be the most significant challenge in the digital forensic science

environment [9]. If the computer is used by a single user, it is easy to link the username to the user. However, in cases of a shared computer or compromised devices, it may be difficult to establish the identity of the person actually using the device. This limitation of traditional authentication requires the integration of the forensic attribution process with other means of non-repudiation [10]. It requires a means of user identification, or user attribution, to specifically identify 'who did what' on the system under investigation [11].

Given the nature of digital evidence and the complexity of digital environments, evidence analysis and expert testimony often require innovative approaches [12]. Two such approaches for user attribution, to link the *user* and the *username*, is the analysis of the physiological and behavioral characteristics of the user. Physiological characteristics include commonly used biometric modalities such as fingerprint scans, iris scans etc. Behavioral characteristics rely on modeling user behavior through analyzing the interaction of the user with the system.

Like other biometric features, all humans show patterns of behavior that are unique [13]. They exhibit repeatable and identifiable routines, and these are more obvious when the behavior is temporally, spatially and socially contextualized [14]. As early as 1895, it was noticed that telegraph transmitters could be identified by their manner of keying messages. Operators often knew the transmitter on the other end simply by their typing patterns [15]. This predictability of human behavioral characteristics has since frequently been studied and used in different contexts. One of the applications is to detect users being impersonated on a computer, with the occurrence being known as a *masquerade attack*. Another application is in the field of digital forensics. Gupta, Rogers, Elliot, *et al.* [16] explored the notion of human behavior on a computer or *computer behavior* to assist with digital investigations. The researchers used a desktop recording tool to observer the users' computer behavior. Specifically, the sequence of steps used by any user to perform a specific set of tasks on a computer. An experiment with 60 users was conducted, where users were given a set of tasks to complete in a Windows environment. The tasks were chosen to replicate common, everyday activities on a personal computer. The study showed that the recorded activity of users exhibited uniqueness and consistency in how tasks were performed. In other words, users showed distinct habits in how they interacted with the computer. The conducted research exploited these computer habits to model user behavior based on

computer interaction, so that unknown user events are attributed to the right user. Assuming that each user is considered as a *class* that the user event needs to be attributed to, this can be seen as a classification task. This was done using the machine learning algorithm, XGBoost, which has shown exceptional performance in such classification tasks [17].

1.2 Scope

This research assessed whether user profiles can be created based on how users interact with the graphical user interface on a computer. This was done by first training the system with some events that are attributed to a user (known events) and then identifying the specific user for an unknown event. Data was collected from users by asking them to perform a set of tasks that represent general tasks done on a computer (e.g. answering questions by typing responses). All the users were asked to do the same tasks, so the focus is on *how* users perform the tasks and not *what* they are doing. Only data about how the tasks are done was analyzed, without relying on personally identifiable information (PII) from the participants.

1.3 Significance

1.3.1 The Need for User Attribution

As mentioned earlier, digital evidence is pervasive today. Any computer can be seen as a future site for digital evidence collection [18]. This applies not only to crimes that target computers but also computer-assisted and computer-incidental crimes such as murder or kidnapping [19]. This omnipresence of digital forensics highlights the need for proactive preparation to gather the most effective digital evidence [20] [21]. As seen in Figure 1.1, digital forensic readiness is an important aspect, and a precursor to a successful digital forensics investigation. Re-purposing security mechanisms towards providing forensic value is a gradual yet inevitable shift in today's digital society [10].

Digital evidence has its limitations. If a person is logged into another person's account, there needs to be a way to differentiate between them. Relying on login information alone may not be successful as that could be shared or compromised. As Katz [22] says, it is more important than ever that forensic science is equipped to identify the actual user on digital

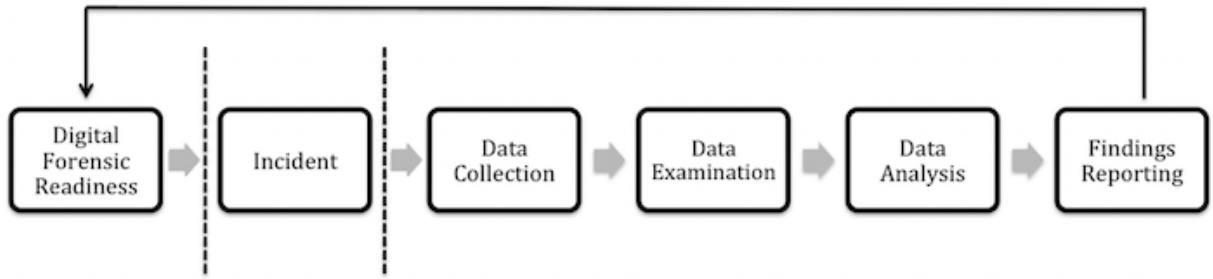

Figure 1.1. Digital Forensic Investigation Lifecycle [20]

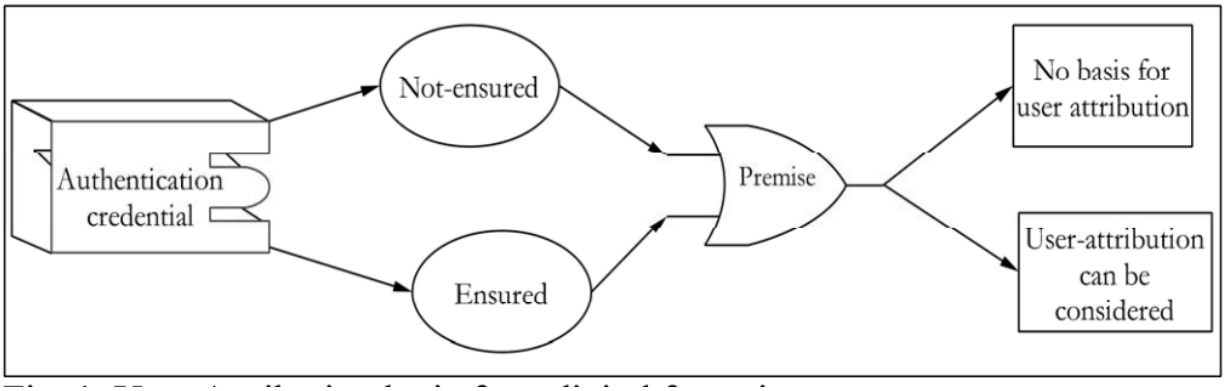

Figure 1.2. The role of user attribution in digital forensics [10]

devices through studying user behavioral traits. Figure 1.2 illustrates how user attribution can be used to establish greater confidence in the user's identity during the investigation lifecycle. Goldring [23] has talked about the importance of extracting psychological traits from technical information. Several researchers have recommended the inclusion of behavioral analysis as part of digital forensics investigations [24], [25], [26].

This study used the SU-AIS BB-MAS (Syracuse University and Assured Information Security - Behavioral Biometrics Multi-device and multi-Activity data from Same users) dataset [27] to extract mouse and keystroke usage data for the users. It examined these keystroke and mouse characteristics of the user to establish a user profile. The behavior of the unknown user events was compared to the *known* behavior profile for user attribution. Intrusion detection systems, especially masquerade attack detection techniques, work on the same principle. A baseline behavior of the system is established and user or system activity

is compared to the base-line to check for anomalous behavior. It relies on the concept that each person has a unique cognitive process and their personality, behavior, and environment connect in a way to leave a distinct impression on the environment [28]. This impression can be considered as a *print* that the user leaves on a machine, which can then be used to identify the user, like how fingerprints have been traditionally used [29].

While user profiles have been extensively discussed in the context of masquerade detection, masquerade detection and other computer security techniques have some very elemental differences in goals when compared to computer forensics procedures. With computer security, the system is running in real-time and the tool needs to run continuously in order to detect patterns in real-time. In a computer forensics scenario, only collection of data is conducted continuously and analysis of data to find patterns is done post hoc if needed. The lack of real-time constraints allows for long computation times, as long as it assists the examiner. Also, it allows for complex and heavy algorithms with less emphasis on performance than in computer security. In computer security, systems are designed to be as autonomous as possible. However in computer forensics, human intervention to examine the specialized evidence is not only necessary, but encouraged. Machine learning techniques can be customized by the examiner according to the current investigation [30]. Also, most masquerade detection techniques are designed for enterprise networks with machines containing critical data. This justifies the high performance and storage overhead for logging user activity. These techniques also rely on collecting large amount of intrusive data to create user profiles, introducing potential conflicts with user privacy [22].

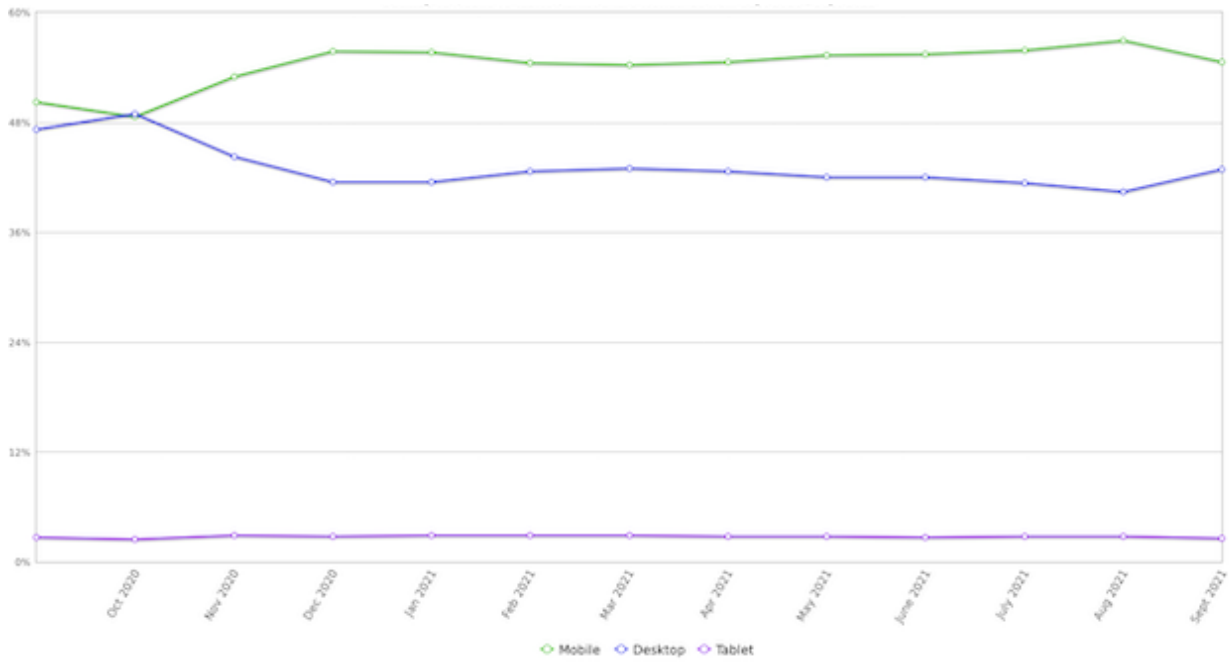

Figure 1.3. Global market share for digital devices in September, 2021 [31]

As mentioned in the previous section, this study focused on desktop computers only. With 42.87% of the market share as of September, 2021, desktop computers are still very relevant and prevalent personal digital devices. This compares to 54.61% and 2.52% for phones and tablets respectively [31]. As seen in Figure 1.3, even the past year has shown a steady trend in the use of desktops [31]. With an estimated 80% of phones having biometrics enabled in 2020 [32], the problem of user attribution for digital forensic purposes may be less relevant for cell phones than for personal computers.

This study targets computers in general use, which may or may not contain digital evidence in the future. It provides a novel proof of concept to identify a user based on how they perform common tasks (e.g. word processing). This will allow the creation of user profiles to distinguish between users without relying on large amounts of personal data. User profiles also allow investigators to link different user events and establish patterns of habitual behavior. This can help to establish *Mens Rea* or intent by proving repetitive behavior. Habitual behavior, combined with personal information can provide significant and useful circumstantial evidence to investigators. Digital forensics evidence should be

planned in advance and not just rely on available evidence at the time [20]. This is an important consideration for moving towards an enhanced state of digital forensic readiness.

The need for user attribution has already been seen in a digital investigation. This was encountered in [33] where the appellate court reversed the conviction (for knowingly possessing child pornography) because the computer was accessed by multiple persons. The defendant's computers were found to contain 112 images containing child pornography. However, the defendant, his wife, and his terminally ill father (who had since passed away), all had access to the computer including shared access to usernames, passwords and a Yahoo account. Thus, there was no way of connecting the images to the defendant Moreland, and not to his father or someone else. With computers often shared between household members, it is feasible that such situations may be seen again in the future. There are some intuitive solutions to the problem of user-attribution:

- Physical Fingerprints: Fingerprints can be seen as a way to connect digital and physical users. However, it can't be assumed that only one user has handled the device, and the traditional method of dusting for fingerprints may not provide very useful information.

- User artifacts: The usual method of establishing the user on a system involves examining artifacts that can provide information about the email accounts, social networking accounts and other personal information that links to a user. However, if there are claims of a device being compromised then it could also be claimed that user accounts were hacked. This would make it an infeasible approach.

There are additional user attribution techniques. There is a growing body of literature addressing the shortcomings of traditional security measures relying on one-time authentication [34]. Industry and academia are looking towards a 'Zero-Trust' model, based on continuous authentication. These security goals are a larger focus of most user attribution research, with a smaller focus on digital forensics. Considerable research has discussed the need for digital forensics readiness and as mentioned, also on behavioral biometrics with a security focus. The practical significance of the interconnection between both, focusing on the application of behavioral biometrics for forensics readiness is an important, yet relatively

unexplored area [10]. Chapter 2 discusses the body of literature in this area, both from a user attribution for information security and from a digital forensics perspective.

1.3.2 The Need for Advanced Data Analysis Techniques such as Machine Learning

While the focus is on the need for user attribution, there is another lens that drives the significance of this research - the need for advanced and automated analysis techniques through the use of approaches like machine learning [35]. Through machine learning, patterns can be observed in large amounts of data to model behavior or observe criminal activity. It can be considered to be a foundation on which behavioral forensics can be established [36]. As Guarino [37] puts it, digital forensics is now a *big data* challenge and law enforcement needs to start rethinking several established principles and processes. It will require innovative methods and research showing validation of these new methods. Especially the reliance on validation will be greater than repeatability in the strictest sense i.e. obtaining the same results when repeated in the same test environment [37]. This reliance on validation emphasizes the need for research studies that report clearly on the accuracy of proposed techniques through established metrics such as the F1 score or the Area under the Receiver Operator Characteristic (AUROC) curves.

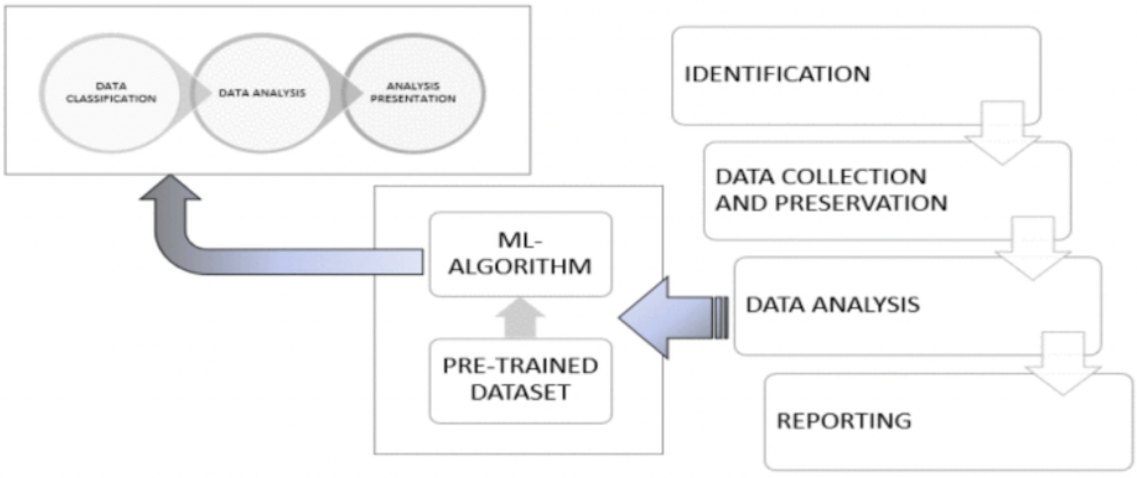

Figure 1.4. Automated Digital Forensics Framework proposed by Qadir and Noor [38]

Several researchers have discussed the need to include machine learning techniques into digital forensics [38], [37] [36] [35]. Especially with the ever-increasing amount of data, there is an increased need for criminal investigations to adopt automated analysis techniques [38]. Bhatt and Rughani [35] have even proposed that a new field of digital forensics, *machine learning forensics* is required, which can focus on developing methodologies and frameworks exploring and investigating machine learning techniques in digital forensic investigations. One such framework, proposed by Qadir and Noor [38], includes machine learning into *analysis* phase of the digital investigation cycle as seen in Figure 1.4. It includes training the algorithm with any available data and then using the algorithm to make predictions, without entirely relying on human inference.

The use of machine learning techniques in digital forensics is not unexplored. Machine learning has been extensively used in areas of malware analysis, image/video forensics, network forensics, and file-related forensics (including memory forensics and mobile forensics) [38]. However, there has been limited focus on using machine learning techniques for user attribution in digital forensics, with most research in this area driven by Ikuesan [10] [11]. Even then, machine learning is a continuously evolving field with new and improved algorithms being proposed every few years. There is still a lack of a clear understanding of machine learning, in its applicability to digital forensics [30]. There is a need for digital forensics techniques to evolve hand-in-hand with the evolving data and data science landscape [38]. This evolution seems to be more matured in the field of *computer security* but remains relatively immature in the digital forensics space [30]. The current best technique for most classification and regression, XGBoost, was only proposed in 2015 [17] and has not been used for user attribution in digital forensics yet. Given that the algorithm has not been used, there is also a gap in the literature for metrics reporting on the accuracy and sensitivity of using this algorithm for user attribution. Establishing these error rates are an important step towards acceptance of these techniques by the scientific community and eventually in the court of law. The *Procedures and Data Collection* section of this dissertation dives deeper into the evolution and working of the machine learning approach used.

1.4 Research Question

This research explored whether user interaction could be modeled to distinguish between users on shared desktop computers using machine learning approaches, specifically the XGBoost algorithm. The study developed and tested an XGBoost model that *learned* user behavior and attributed events to users through a multi-modal approach using keystroke and mouse data.

The research question was: Can the XGBoost machine learning algorithm be used to develop a classifier that can attribute a user activity on a desktop computer to a specific user, based on a multi-modal approach relying on keystroke and mouse usage data?

The research question was answered through testing the following hypothesis:

- A machine learning model can be developed using XGBoost, relying on keyboard and mouse usage information, that correctly attributes a user activity to the right user.

1.5 Assumptions

The assumptions for this study include:

- The tasks assigned to the participants simulate common tasks performed by users on computers.
- Participants performed the tasks themselves.
- Participants followed the instructions provided.
- Participant behavior for each event was not influenced by other events. This is supported by Imsand [39].
- Participants were honest about their computer expertise and the daily duration of their computer usage.
- The results achieved in the controlled lab environment can be applied to a real world setting.

1.6 Limitations

The limitations for this study include:

- The tasks performed by the participants will only cover a subset of usual activities performed on a computer.
- While previous research supports the assumption that it shouldn't, the behavior of a participant might be influenced by prior activity.
- Users will be given the same set of tasks to minimize the impact of other variables. This may be different from the real world scenario.
- This research relies on secondary analysis of published data, which was only collected in a Windows environment.

1.7 Delimitations

The delimitations for this study include:

- The study focused on user behavior in a Windows graphical user interface environment and might not work for users who extensively work on a command line environment.
- The study was conducted on desktop computers and may not be applicable to other digital devices.
- The study conducted secondary analysis using keystroke and mouse data as collected through the desktop activity from the SU-AIS BB-MAS (Syracuse University and Assured Information Security - Behavioral Biometrics Multi-device and multi-Activity data from Same users) dataset. See Figure 5.3 for details.
- The SU-AIS BB-MAS dataset collected data over a single session. This may introduce limitations when attributing events to users that have been collected over multiple sessions.

1.8 Definitions

For the purposes of this research, the following terms are defined (in order of their usage):

***False Acceptance Rate or False Match Rate*:** The percentage of identification instances in which unauthorized persons are incorrectly accepted [40]

***False Rejection Rate or False Non-Match Rate*:** The percentage of identification instances in which unauthorized persons are incorrectly rejected [40]

***Equal Error Rate*:** The percentage of identification instances when the *false acceptance rate* and the *false rejection rate* are the same [40]

***Masquerade Attack*:** A cyber attack in which an unauthorized user makes use of an authorized user's access to an account [41]

***User Attribution*:** The process of assigning a specific user to an object, where the object can be a computer artifact or activity [11]

***Machine Learning*:** Discipline involving the use of machines to perform tasks that were previously performed by humans [42]

***Behavioral Biometrics*:** Identification of the user through the detection of the behavioral features of the user, such as signature, voice, and keystroke dynamics [43]

***Human Computer Interaction*:** Field dealing with how users interact with digital devices [44]

***Keystroke Dynamics*:** Analysis of keystroke usage of users, usually with the goal of identifying them through this usage [45]

***Digraphs*:** Combinations of two letters, used in the area of keystroke dynamics [46]

***Key Press Time*:** Duration of the time interval when the key is pressed down or held down by a user [47]

***Artifical Neural Networks*:** Machine learning technique evolved from the notion of simulating a human brain in learning [48]

***Support Vector Machines*:** Supervised machine learning technique that uses hyperplanes for for prediction tasks [49]

***Supervised Learning*:** Learning technique in which the model is training with some known data [50]

***Unsupervised Learning*:** Learning technique in which the model does not undergo any training with known data [50]

***Training Set*:** Dataset used to train classifiers in supervised learning techniques [51]

***Validation Set*:** Dataset used to obtain an unbiased estimate of the model's abilities, before using it for prediction [51]

***Testing Set*:** Dataset used to measure the performance of a developed machine learning [51]

***Overfitting*:** Occurrence of a large gap between the training error and the testing error i.e. a model that is too specific to the training data [52]

***Underfitting*:** Occurrence of a model's inability to obtain a low value on the training error i.e. the model is not closely aligned with the training data [52]

***Hyperparameters*:** Parameters that control the learning process and helps to obtain the value of the model parameters that a learning algorithm ends up learning [53]

***Regularization*:** Machine learning technique used to prevent overfitting [54]

***Generalizability*:** Ability of a machine learning model to make predictions on unseen data [55]

***Accuracy*:** The measure of correctly classified objects or instances [56]

***Precision*:** The ratio of the correctly classified instances to the total number of classified instances [56]

***Recall*:** The ratio of the correctly classified instances to the total number of instances in the class [56]

***F1 Score*:** A weighted ratio of precision and recall that addresses both false positive and false negatives [56]

***Ensemble Learning*:** A single classifier that is a combination of several individual classifiers in order towards obtaining better performance [57]

1.9 Summary

This chapter provided the scope, significance, research question, assumptions, limitations, delimitations, and other background information for the research project. The next chapter provides a review of the literature relevant to the thesis.

2. LITERATURE REVIEW

The concept of attribution has been a concern among security and forensic researchers [58]. Attribution refers to the method of assigning causation (event or action) to a known effect (actor, source, recipient) [11]. Attribution in this case doesn't only refer to the system causing the effect (system attribution), but should also focus on the actual individual behind a given system (human attribution) [59]. This chapter provides a review of the literature related to identifying users *at the keyboard* on a computer.

As mentioned earlier, there is a large body of research discussing user behavior modeling to either (1) detect an illegitimate user on a system or network, (i.e. detect masquerade attacks) or (2) determine whether a legitimate user is performing unauthorized activities, i.e. detect insider threats. As discussed below, there has also been research towards the application of traditional criminal profiling techniques, to develop behavioral or psychological criminal profiles of offenders in computer-targeted or computer-assisted crimes [60][61]. However, there has been limited research on modeling user behavior on a computer or *computer behavior* specifically for the purposes of discriminating between different users on the computer, towards aiding a forensic investigation [62].

The literature review first presents a general discussion on the research conducted towards modeling user behavior. The review then dives into the previously conducted research through the lens of:

- Nature of the feature data used to build user profiles
- Main motivations or suggested uses of conducted research (i.e. whether researchers were focused on security goals such as masquerade attacks or on modeling user behavior for forensic purposes)

2.1 User behavior modeling

A user profile can be described as a description of a user's interests, characteristics, behaviors, and preferences [63]. Digital behavioral analysis, or creating behavioral profiles for digital forensic purposes, is a relatively new field [64]. Casey [60] and Rogers [61] made the

case early that even though digital forensic science was still emerging at the time, the main goal of the forensic process, namely identifying the suspects, also applies to the digital realm. Similar to how a profile of a suspect would be created for traditional crimes, researchers have proposed taxonomies to model the social, behavioral, and psychological traits of cyber offenders [65], [66],[25].

Profiling of this nature is focused on getting additional information about the motive, developing personality profiles of cyber criminals, and linking crimes to criminals [67]. A key feature in the user profiling discussed above is that it is usually focused on digital artifacts that can help establish the identity of the offender through either deductive, inductive, or more commonly - hybrid reasoning [61]. Deductive approaches like the Behavioral Evidence Analysis (BEA) proposed by Turvey [24] analyze evidence from a specific case to derive user profiles describing personality and behavioral traits of the probable offender. These approaches use specific information to predict more general characteristics that might be applicable to the suspect. Al Mutawa, Bryce, Franqueira, *et al.* [68] proposed the use of BEA to build digital profiles that focus on the behavior and motivations of cyberstalkers. Krone [65] proposed a typology and Rogers and Seigfried-Spellar [69] suggested using internet artifacts to create behavioral profiles of child pornographers. Inductive approaches start with generalized theories about offenders and attempts to apply those theories to the current case [67].

There is another goal of establishing digital profiles that slightly deviates from the goals of criminal profiling discussed thus far. As opposed to inferring information about the personality, motivation, or activities of the offender, user profiles can be created to allow discrimination between different users in a digital environment. In such user profiling, the emphasis often shifts from *what* the user was doing on the digital device to *how* they interact with the machine and what their normal usage patterns are. Early cognitive researchers have suggested that users show uniqueness in how they perform simple tasks [70]. This relates to the social cognitive theory which asserts that a user's personality or individuality, behavior, and the environment, all interrelate and influence each other [71]. Research has shown that this influence of the cognitive and thinking style of users on their behaviors or activities also extends into the digital environment [72].

Computers are now ubiquitous in most developed societies. Similar to activities performed in a non-digital space, human beings interact with computers in many different ways and show unique features in the form of strategies, styles of usage, knowledge, and skills applied. Models can be formalized to quantify these traits, which can provide a means to provide more information about the identities of the person using the computer [44]. While it may not provide enough concrete evidence for identification by itself, it may allow discrimination from other users by observing how the user interacts with the systems and the applications. For seasoned users, this behavior might be subconscious without being aware, providing a promising feature set to model behavior [73]. Users that are not native to the computer or to the digital environment in which they are operating are expected to show different behavior from the usual users [74].

The main focus of this literature review is to highlight key research in this area, focusing on user discrimination or attribution through user behavior profiling by modeling user habits in a digital environment. This involves extracting certain features to model what *normal* behavior would be for a user and then comparing with other users to verify if users can be discriminated in a manner that is consistent and reliable enough to support an investigation. There are several challenges that researchers have had to overcome towards this goal. Data capturing digital behavior can be very noisy, and researchers have to account for outliers without overfitting the data [75]. While considering approaches to model user behavior, it is also important to account for the evolving nature of the user behavior [73].

The following sections explore the various research studies that have attempted to address these issues and present approaches to discriminate between users in a digital environment. The review initially focuses on research that targets the state of the system such as running processes, applications, etc. to model a user's habits. It then explores studies that have used the commands typed by the user as an indicative feature of their behavior. Given the prevalence of graphical user interfaces, it then pivots to the promising results seen by modeling the users' interaction with graphical interfaces. While the majority of the studies focus on security related goals, the literature review finally targets the studies that are explicit about the forensic motivations of such user behavior profiling towards user discrimination.

2.2 System State and Configuration

User profiling based on the state of the system as a user works on it, has been proposed by several researchers. A user at work is assumed to have an approximately uniform set of tasks to perform each day, which would imply that the resources on the system are also being used in a uniform manner. A deviation from the observed system state can be assumed to be a different user. This section explores research studies that model user behavior based on analysis of the system state. This can include different metrics such as the titles of the windows open, process table activity, and system calls sent to the operation etc.

All multitasking operating systems use process tables to allocate resources and run different functions. Therefore, using the process table to determine the user activity, and build behavioral profiles, initially showed promise. However, using the processes introduces a need to filter out system related processes and extract only those processes related to user activity to form accurate user profiles [23]. Window titles can be used to represent user activity on the system. Goldring [23] suggested that the process identifiers associated with open windows can be connected to the process table to filter out the noise and log process activity related to user processes. The researcher uses a feature set consisting of titles of Windows being used, process table information and system timings and uses a support vector machine for analysis.

Rybak and Mosdorf [76] capture the differences in users based on their activity on the computer. They model this activity based on system behavior represented by the number of interrupts per second. They captured data using the *vmstat* program on a Linux computer. Their captured metrics included the number of pages free, swapped and cached pages, the number of buffers, the number of blocks read and written, and different uses of CPU (Central Processing Unit) times. However, they used the number of interrupts per second as the main variable for detailed analysis as it represented the system activity the best.

Similarly, Li and Manikopoulos [77] used data from the process table to represent user activity on a system. They trained and tested the system using data from 35 sessions of 4 users. They used another 4 users to exclusively test the system. They used a support vector machine (SVM) for training and testing, achieving a detection rate of 63% and a false

positive rate of 3.7%. Song, Salem, Hershkop, *et al.* [78] used system level characteristics such as information from the process table, registry activity and file actions to build user profiles. They used Fisher feature selection to optimize classification. Once features were selected, they used a Gaussian mixture model to train the system. They claim that using Fisher feature selection surpassed other comparable SVM-based methods and showed an improvement of 17.6 % over their earlier approaches.

Frequency attributes of system calls have also been used to model user behavior. Liao and Vemuri [79] used text categorization based on frequency attributes of system calls. Hu, Liao, and Vemuri [80] modified their research to use SVMs based on frequency of system calls. Chen, Hsu, and Shen [81] compared analysis using SVMs and Articial Neural Networks (ANN) and found that classification schemes based on SVMs outperformed those using neural networks. Wang, Zhang, and Gombault [82] used weighted frequency attributes combined with distance measures such as Nearest Neighbor (NN), k-NN, and Principal Component Analysis (PCA) to achieve promising results.

Given the prolific mobile phone usage today, researchers have also been focusing on modeling user behavior by capturing system data from cell phones. With most smartphones including embedded sensors such as GPS, Bluetooth, accelerometers etc., this can provide a wealth of additional information that was previously unavailable on traditional personal computers. Ye, Zheng, Chen, *et al.* [83] suggested the use of GPS (Global Positioning System) data to model *normal* user behavior or *life patterns* as they call it. While they don't specifically discuss the applicability to digital forensics or even in the field of information security, they proposed that GPS data could be mined to extract user behavior patterns and this can be used to build a digital profile of that user's behavior.

Grillo, Lentini, Me, *et al.* [84] have conducted research to build user profiles by using machine learning on feature vectors, which among other things uses elements related to user habits. The feature vectors related to the user habits include the programs installed, the chronological use of the installed programs, the order of visiting websites etc. Also, use skill level is modeled as well as user interests. However, instead of trying to build a profile for each user on the system, their goal is to develop a triage model to obtain a "class" of the user so that future direction of the investigation can be prioritized. They use different machine

learning approaches. They classify 25 vectors to test their classification scheme. They get greatest success with the BayesNet Algorithm (100%), followed by Naives Bayes (92%), and obtain a success rate of 84% using the J48 Decision Tree classifier.

Li, Clarke, Papadaki, *et al.* [85] used one month of mobile phone log data from applications, voice calls and text messages from 106 participants in order to build user profiles towards detecting masquerade attacks. For each of the log types, they report equal error rates of 13.5%, 5.4%, and 2.2% respectively. They used the following features for each of the logs:

- For general applications, the application name, date of initiating application, and location of application usage
- For voice calls, the telephone number, the date, and the location of the calls was selected as features
- For text messages, the receiver telephone number, the date, and the location of the texting occurrence were the chosen features

As evident by the selected features, Li, Clarke, Papadaki, *et al.* [85]'s approach had severe limitations in terms of privacy, especially with the use of personally identifiable information such as telephone numbers. Zhang, Yan, Yang, *et al.* [86] suggested that their approach helped to alleviate some concerns around privacy by utilizing system data from mobile phones sensors for their approach to model user behavior. Their framework used:

- Frequency based features such as the amount of activity on WiFi, cell towers, Bluetooth and overall application usage
- Entropy based features which focus on not just the frequency of the activity but on the distribution as an additional feature
- Conditioning the features on specific times and specific locations

They collected data from 22 users over 2 months and found an overall performance of 81.3% when all the features were considered. Figure 2.1 illustrates the framework proposed

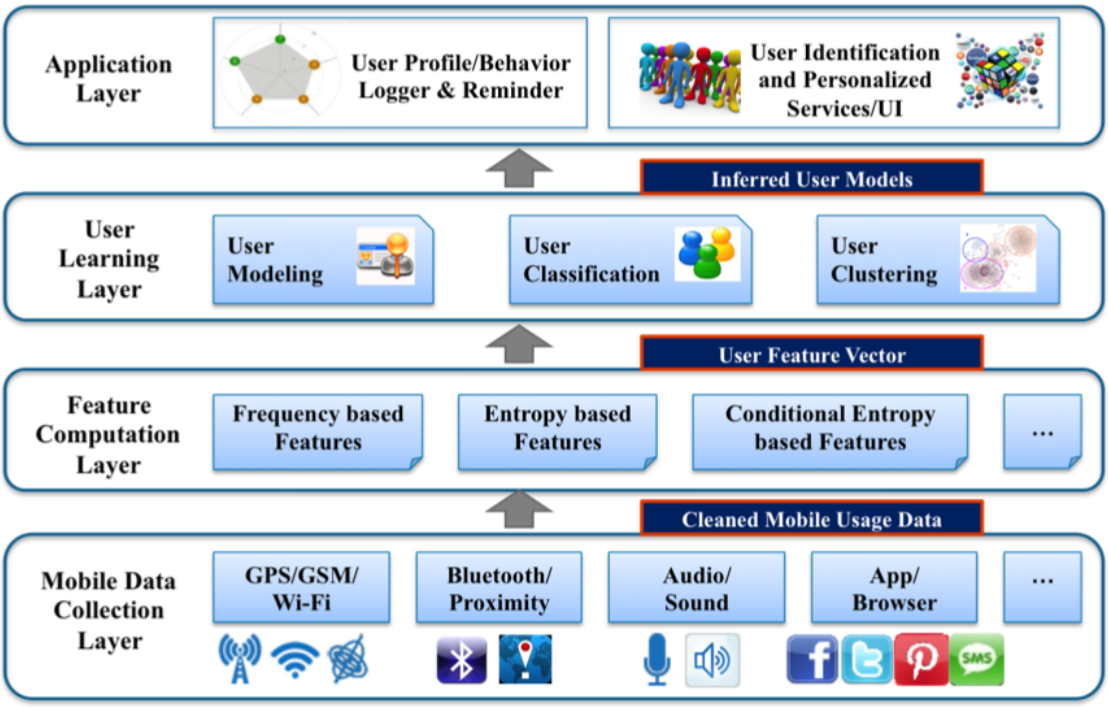

Figure 2.1. User behavior modeling framework suggested by Zhang, Yan, Yang, *et al.* [86]

by the authors towards creating a digital *fingerprint* based on system information collected from mobile phones.

Singh, Mehtre, and Sangeetha [87] modeled user behavior using an ensemble hybrid machine learning approach using Multi State Long Short Term Memory (MSLSTM) and Convolution Neural Networks (CNN) based time series anomaly detection. The researchers used spatial-temporal behavior features (e.g. number of logons per day, first access time, number of running processes, total bytes downloaded each day etc.) to build a profile of *normal* usage for a user, and then used these profile to detect any deviation from usual behavioral to indicate an insider threat.

The biggest issue with system-level analysis is the amount of data logged. All these methods are proposed with the goal of masquerade detection on critical systems. The goal of the present study is to propose the proof of concept of an extensible mechanism that might

be included in future operating systems towards enhancing digital forensic investigations. Logging system data such as system calls provides a huge overhead which may not be feasible. As an example, 112 *sendmail* messages generated over 1.5 million system calls [88].

2.3 Command Line Activity

Earlier studies profiling user behavior focused on modeling behavior based on command line activity of the user. This was usually done on UNIX-like systems where most users mainly worked on the command line.

Lane and Brodley [75] conducted early research which used command line history from 7 participants to build profiles. They tested different similarity measures and found that measures that gave greater weight to adjacency were the most useful. Schonlau, DuMouchel, Ju, *et al.* [41] also profiled users based on command lines by collecting 15,000 commands from 50 users. Maxion and Townsend [89] expanded their research and used the same dataset but instead of injecting random data from outside to simulate attackers, they used every user's session as an attack session for other users. They claimed a 55 % improvement over Schonlau, DuMouchel, Ju, *et al.* [41]'s results.

Yung [90] suggested an adapted version of Schonlau, DuMouchel, Ju, *et al.* [41]'s approach using a Naive Bayes classifier by introducing a feedback mechanism which requests feedback from the user when there is suspicious activity. The author claims that introducing feedback reduces false alarms by 30 %. Wu and Huang [91] used Schonlau et al.'s dataset but performed classification using principle component analysis to reduce the dimensionality. Their main contribution was the illustration of how the false detection rate is impacted by similarity of commands used. Shim, Kim, and Gantenbein [92] used command lines where truncated command lines were parsed as tokens of four and weight frequencies of patterns occurring together were used to build the profile. They trained the system with 2 users and tested it with 8 users. Their results were promising because they were resistant to noise and needed less data to be logged.

Using profiles based on command line history is the easiest to log and with truncated command lines, complexity is the least. However, data logged in this manner is not useful

anymore [23]. Most users today use a graphical user interface to interact with the system which makes it difficult to rely on obtaining adequate command line data to build and compare profiles.

2.4 Human Computer Interaction

Human computer interaction (HCI) is the field that deals with how humans interact with computers. The nature of this interaction is relatively unique to each individual and this individuality can be exploited to develop a means to identify the individual on the computer [44]. Unlike approaches that focus on system data, these techniques shift the focus from *what* is being done by the user to *how* the user is doing something. At least in the current state, this alleviates some privacy concerns as the data collected does not include personally identifiable information (PII).

2.4.1 Keystroke Activity

Analysis of keystroke activity or *keystroke dynamics* was one of the earliest modalities explored towards establishing a digital behavioral biometric, based on human interaction. Researchers have proposed many different approaches based on the features used, the feature extraction techniques, and classification methods [93].

Gaines, Lisowski, Press, *et al.* [46] conducted initial research in this area. They had seven typists type a paragraph of prose and recorded the times between successive keystrokes. They had the same typists repeat the experiment after four months. They examined the probability distributions of certain commonly occurring two-letter combinations, known as *digraphs*. They found that a set of five digraphs could be used to differentiate between users. Umphress and Williams [94] conducted an experiment with seventeen users. They used the average time interval between keystrokes (which they call the mean keystroke latency) and also average of digraph times similar to those used by Gaines, Lisowski, Press, *et al.* [46]. They obtained a false rejection rate of 12 % and false acceptance rate of 6%. They concluded that while keystrokes themselves aren't robust enough as an authentication mechanism, they can be combined with other metrics to provide strong authentication. Hammon and Young

[45] who introduced the term *keystroke dynamics*, experimented with metrics like the time between keystrokes, the time taken to type a certain number of characters, or the pressure applied to keys.

Joyce and Gupta [95] used the same metrics as Umphress and Williams [94], which is the time between keystrokes or the latency time. However, they used login credentials for authentication. The authentication signature consisted of the user-name, password, the first name, and the last name. The mean latency was calculated to build profiles. They tested this by training the system with 33 users, with 27 users then acting as intruders on six targets. Their study reported a false positive rate of 7 % and a false negative rate of less than 1 %. Obaidat and Sadoun [47] introduced an additional parameter of the *key press time* or *key hold time*, which is the duration of time for which a key is pressed by the user. They used 15 students to test their approach which used neural networks based classification techniques such as the fuzzy ARTMAP, radial basis function networks (RBFN), and learning vector quantization. They reported an identification accuracy of 100% within their test dataset. They found that a classification scheme that combined key latency (or interkey time as they called it) and key hold times was more efficient than schemes using these metrics individually. Monrose and Rubin [96] expanded on the research done by Joyce and Gupta [95] to use variable feature sets obtained by factor analysis, expanding the number of participants to 63 and exploring the impact of different classification schemes. They found that the weighted probabilistic classifier performed the best with a success rate of 87.18 %.

This section outlines a few of the prominent research projects in this area, but many other approaches have been suggested by researchers. One of the drawbacks of using keystrokes as the authentication metric is that with the advent of graphical interfaces, many operation system use mouse movements to interact with the screen. This might make it difficult to collect enough data on a computer used for general use.

2.4.2 Mouse Activity

Using data about mouse usage towards user authentication gained greater traction with the advent of computers with graphical user interfaces. The first prominent research study

in this area was by Pusara and Brodley [97]. They conducted a study with 18 students collecting data metrics such as screen co-ordinates for each mouse movement, mouse wheel rotation, mouse clicks and the movement of the mouse outside the client application. They conducted discrimination of a user from others as well as pair-wise discrimination of users. They reported very promising error rates with a false positive rate of .43 % and a false negative rate of 1.75 %. Garg, Rahalkar, Upadhyaya, *et al.* [98] used similar metrics as Pusara and Brodley [97]. However, Garg et. al allowed their users to use the computer flexibly with their own choice of tasks. Pusara and Brodley [97] required subjects to browse specific pages on a website. Garg, Rahalkar, Upadhyaya, *et al.* [98] reported a higher false negative of 3.85%.

Ahmed and Traore [43] combined the approaches of keystroke and mouse dynamics by using metrics like the key interval and the key press time as well as mouse metrics such as average mouse speed for a particular distance and for certain directions. They use a supervised learning approach where each successive user session is used to improve the reference profile of the user. This eliminates the need for explicit training of the user. They obtained a false negative rate of .65 % and a false positive rate of 1.31 %. Bhukya, Kommuru, and Negi [99] also combined keystroke and mouse features such as mouse clicks, mouse entrance and exit, wheel rotations, keys pressed, and keyboard shortcuts used. They suggest that their work is the first attempt to profile users based on their GUI data on a K Desktop Environment (KDE). They use a one-class SVM approach for classification among three users. They obtained a 86 % detection rate with a 2.93 % false positive rate and 11.77% false negative rate. Garg, Upadhyayal, and Kwiat [100] use features such as mouse speed, distance, angles, and number of clicks to build user profiles. They use support vector machine (SVM) for user classification and report a detection rate up to 96 % while testing their classification algorithm among three users.

Given the ease of data collection and high probability of uniqueness, mouse dynamics have been gaining traction as a promising biometric modality for forensics [101]. While the currently reliability falls below the .001 FAR and 1.00 FRR established by the European Standard for commercial biometric technology [102], combined with other modalities, mouse dynamics shows promise towards enhancing user profiles for forensics purposes [62].

2.4.3 GUI Usage analysis

While there is extensive research on biometrics based on metrics related to muscle control, such as keystroke dynamics and mouse dynamics, biometric techniques related to how users interact with the graphical user interface are not as well explored [44]. Imsand and Hamilton [103] suggested that users could be differentiated on a system on the basis of how they do a task, instead of what they were doing on the system. They conducted an experiment with 31 users using similarity matching using the Jaccard Index for classification. When they used a customized attack threshold for each user, they achieved a false negative rate of 6.27 % and a false positive rate of zero.

Camiña, Monroy, Trejo, *et al.* [104] conducted a preliminary study to illustrate that user behavior can be modeled based on how users navigate the folder structures. They used six subjects with three acting as legitimate users and data from other three used to simulate data from a masquerade attack. While the size of the experiment was too small to draw any reliable conclusions, they found that the user profiles were more consistent and reliable for experienced users with good directory organization.

Gupta, Rogers, Elliot, *et al.* [16]'s behavioral study used 60 participants, with 30 participants used to determine the most promising features and the next 30 participants used to verify whether users were consistent in how they navigated the graphical user interface to perform a fixed set of tasks. Their findings supported Camiña, Monroy, Trejo, *et al.* [104]'s - users that spent more time on the computer, showed more uniqueness and consistency in their computer behavior. While Gupta, Rogers, Elliot, *et al.* [16]'s was an observational study and did not extract and model user behavior, it reaffirmed the idea that computer habits can possibly be used to model user behavior, which may assist in forensic investigations.

Saljooghinejad and Rathore [105] proposed a method which combined keystroke information, mouse data, and GUI interaction data of the user. Instead of using overall keystroke and mouse data, they just use data pertaining to how the user interacts with each application. This includes features like the average number of mouse exits and entrances in an application, the number of shortcuts used per application etc. They use window-related data such as the average number of times the windows are resized per application, the average

number of times that the user switches between windows etc. They tested their system with data from three users and found success rates of 91.54 % with an average false positive rate of 9.3 % using the k-NN classifier.

Touchscreen biometrics is an active area of research, where user interaction with a graphical user interface is modeled, but relying on gestures (such as 'swipes') as opposed to interaction through the use of a keyboard or mouse [106], [107]. Similar to other user attribution techniques, touchscreen biometrics relies on the notion that each user behaves uniquely in their interaction with a touchscreen. Research in this area has shown that this behavior is highly discriminative, with a high inter-class variance. However, it shows lower intra-class variance, especially for behavior captured over different days [108]. Table 2.1 illustrates examples of research in this area. In this context, SD performance refers to instances where users were authenticated within the same day. DD performance refers to authentication on a different day. Antal, Bokor, and Szabó [109] used touchscreen biometrics to estimate age and gender of the user.

2.5 Stylometrics

There has been much research in the user attribution through the use of stylometrics. Many authors have discussed the application of stylometric analysis towards forensics purposes.

De Vel, Anderson, Corney, *et al.* [119] explored the possibility of mining emails to create behavioral profiles of authors based on structural patterns and linguistic characteristics of emails. They used a Support Vector Machine classifier to train the system with the writing behavior of three authors. They ascertain that while their sample size consists only of three authors, most digital investigations usually involve two or three suspects so their results are promising.

Chaski [120] used stylometrics to differentiate between users through the means of syntactic analysis. They analyzed samples of approximately 2000 words from ten authors. They measured similarity on the basis of three metrics: punctuation, lexicon, and syntax and used Linear Discriminant Analysis for classification.

Kucukyilmaz, Cambazoglu, Aykanat, *et al.* [121] performed mining of chat logs to identify unique writing behavior among users. However, they don't explicitly mention any forensics uses of their work. They use different classification algorithms for term-based and style-based classification. They use term-based classification for user and message attributes and style-based for content within the chat logs. They also explore the impact that the author has on such classification techniques.

Orebaugh and Allnutt [13] performed stylometric analysis on chat logs. Their data consisted of logs from four authors and they used attributes such as abbreviations, emoticons and special characters to attribute works to authors. They performed experiments with different attributes and different classification algorithms and found that differentiating between authors based on the abbreviations used, using a Naive Bayes Classifier provided the most promising results.

While experiments in this area have shown promising results and can be useful in specific scenarios, it heavily relies on the availability of textual data. If a user consistently types large volumes of text in each login session, this can be a feasible approach towards user attribution. However, since that is not the case, stylometric analysis might not prove to be too helpful towards differentiating between multiple users on a shared computer.

2.6 Other

Cheng and Chen [122] proposed an approach based on a user's 'interest level' in a particular file. They use the cloud model to account for the fuzziness of user preference. They build the user profile based on how much interaction a user has with a certain files. They see that the graphical representation of the different users' clouds showed enough variability to determine when the users are the same users or not.

Pannell and Ashman [123] suggested an approach that combined elements from system state, GUI usage habits and typing patterns. They used metrics consisting of number of applications running, number of windows open, CPU and memory usage data for each application, websites viewed by the user and keystroke analysis. They found that they obtained best results when they used a combined metric, which is consistent with other research in

the area. With goals of intrusion detection, their main focus was the performance of the system in terms of time to detect intrusion.

2.7 Forensic Goals

As mentioned earlier, while the past twenty years has seen an explosion in research focused on user behavior modeling, the majority of the studies mainly focus on information security goals such as detection of masquerade attacks and insider threat. However, there have some been some research studies that specifically discuss profiling of digital behavior towards assisting law enforcement by establishing a stronger link between the *user* and the *username*.

Colombini and Colella [124] recognized the challenge and significance of building an association between the criminal and the computer on which the crime has been committed. The researchers suggest an approach using features such as files accessed, hardware and software installed, websites visited, etc. to create a digital footprint that can help to link different user profiles to the same criminal.

While initially targeting security-focused authentication [102], Shen, Cai, Maxion, *et al.* [125] expanded their research towards using computer interaction behavior for forensic analysis. They propose that the interaction between users and computers can be used to determine demographic traits, with recognition rates ranging from 82.11% to 87.32%. They extracted keystroke and mouse movement features and used a weighted random forest classifier to infer five demographic traits (gender, age, ethnicity, handedness, and language) from user interaction behavior.

Govindaraj, Verma, and Gupta [126] propose a digital forensics readiness framework based on extracting and analyzing ads on mobile devices to retrieve user-specific information and using this information to build user profiles. They suggest that the following user information may be extracted from advertisements on mobile phones (both iOS and Android):

- App name and version of the ad that was clicked
- List of device capabilities

- Name of network operator
- User-provided age
- User-provided gender
- Ad publisher account ID
- Type of network used (e.g., 3G, 4G or Wi-Fi)
- User-set system language
- App-supplied keywords
- User location
- Time zone
- User demographic information
- User emotional state (e.g., anger, fear, sadness, depression or hopelessness)

They found that the nature of information extracted from ads might vary based on how frequently the ads were clicked. This extracted data can be used to develop user profiles could potentially be utilized to predict a user's identity both proactively and reactively in digital investigations.

As mentioned in the previous section, there has been significant discussion on utilizing mouse movements towards profiling users but most research is security-focused as opposed to forensic focused. Ernsberger, Ikuesan, Venter, *et al.* [62] propose the application of mouse dynamics for forensic purposes. Their experiment consisted of capturing mouse usage related features for 11 participants, freely browsing the internet without specific tasks. They used different classifiers and determined that the path used by the mouse can be used as a measurable and reliable feature towards user attribution in digital forensic frameworks.

Ikuesan and Venter [11] furthered the research towards using mouse dynamics-based behavioral biometrics. They used three existing datasets with mouse features and developed a set theory-based adaptive two-stage hash function and multi-stage rule-based semantic

algorithm to determine the feasibility of extracting a unique behavioral signature for forensic usage. They also determined that it has use as a complementary modality but additional work is required before it can be used independently in forensic litigation.

Clarke, Li, and Furnell [127] address the need for identifying specific users in digital investigations and suggest that network traffic can be utilized to model user interactions and generate a discriminatory feature set enabling more reliable user identification for law enforcement. Their study collected data from 46 users over two months and achieve average recognition rates of 90%. Given that their approach only relies on network traffic, the authors believe that it preserves privacy relative to other user profiling methods.

Adeyemi, Abd Razak, Salleh, *et al.* [72] suggest the use of a digital fingerprint based on human thinking style. Their study collected server-side network data and self-reported thinking styles from 43 respondents. They extracted cluster dichotomies from five thinking styles and then used supervised learning techniques to distinguish individuals on each dichotomy. They illustrated that network data analysis can provide additional *psycho-social* information about the users, which can strength digital profiles used by law enforcement.

Xue, Li, Zhang, *et al.* [128] uses association rules to model user behavior for digital forensics. They propose that association mining of computer usage data can find connections between different attributes in a dataset and can uncover dependencies between valuable multiple attribute domains. Their dataset consisted of two months of computer usage data from three users. They use a Frequent Pattern Growth algorithm to define associate mining rules for the users' behavior. This allows them to speculate on the activity of each user and gain greater confidence in the user's identity.

They suggest that their associate mining algorithm can be used in computer forensic analysis through the following steps:

- Gather original evidence from target computer
- Preprocess original database into standard form that can be used for mining data
- Select the characteristic attributes that need to be mined, based on the historical crime information, and set attribute weight values respectively

- Use the mining algorithm to mine the weighted feature database and general association rules, which can then be used to obtain additional behavioral information about the criminal suspects

An example would be that the mining rules could indicate that the user frequently works late at night or has been continuously working on a single Word document for several months, which can lead to possible speculation that it may be a student working on a thesis.

2.8 Analysis Techniques

As seen through the review of related studies, most research in the area focuses on detecting an anomalous user on the computer for intrusion detection. The research focuses on using different metrics such as the user's typing habits, the commands typed, or the state of the system as they use the computer. There are many different names for this task that are fundamentally similar but the authors call them by different names. The approaches have been described as anomaly detection, outlier detection, novelty detection, exception mining etc. [129]. The underlying principle of profiling user behavior in a digital environment applies both in security and forensics disciplines.

Data mining is the technique of converting large amount of data into useful information. There are many different technologies or domains that contribute to data mining. Figure 2.2 illustrates some of the different domains. Among these domains, statistical and machine learning approaches have been explored towards the task of user attribution. Each of these approaches will be discussed in the next sections.

2.8.1 Statistical Approaches

Statistics is concerned with the collection, analysis, interpretation and presentation of data, which makes it closely related to data mining. Statistical approaches attempt to explain the behavior of data through random variables and associated probability distributions [130].

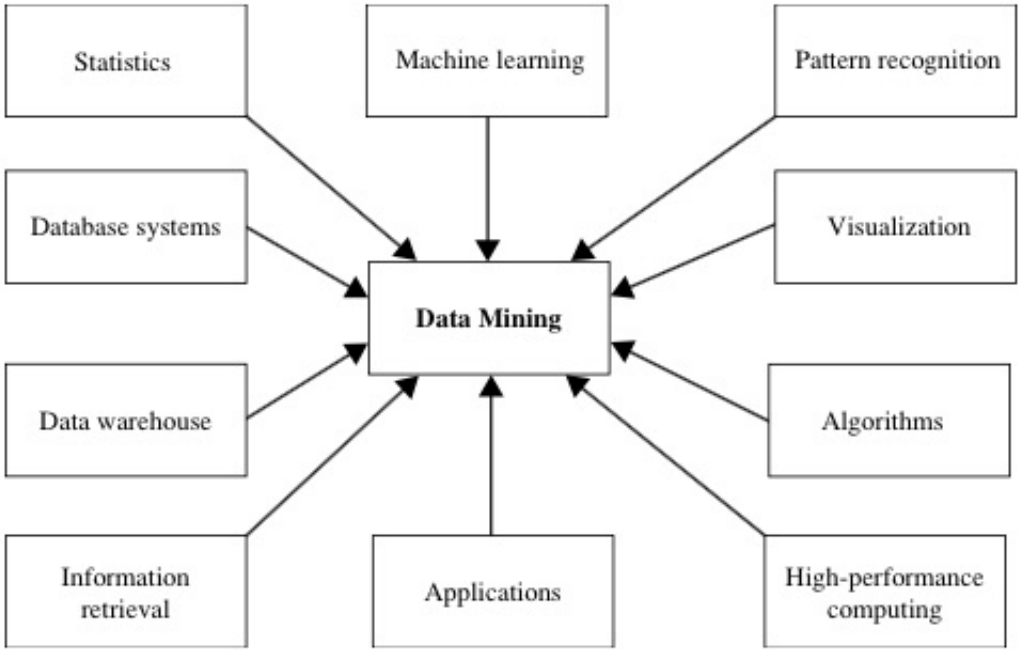

Figure 2.2. Different domains contributing to data mining techniques [130].

Linear Discriminant Analysis

Linear discriminant analysis can be considered as a supervised learning method based on a statistical approach. Supervised learning indicates prior knowledge of the class labels on some training data. Linear discriminant analysis (LDA) performs classification by maximizing the ratio of between-class variance to the within-class variance [131]. It requires that the classes are linearly separable and for data to be normally distributed. Linear discriminant analysis has been used frequently in the past for classification of textual data with promising results. Chaski [120] used linear discriminant analysis for their user classification based on stylometric analysis of written text.

Principle Component Analysis

Principle component analysis (PCA) is a dimensionality reduction technique. Its goal is to explain the variance in data. Assuming that the dataset is seen as a set of points in the high-dimensional data space, PCA provides a projection of the data to principle components

that account for the largest variability in the dataset. Using only the first few components obtained can reduce the dimensionality of the data. PCA is not a classification algorithm per se, but forms the basis of many classification approaches [131]. Wu and Huang [91] and Shim, Kim, and Gantenbein [92] used PCA for their user classification.

Bayesian Algorithms

Bayesian classifiers are statistical classifiers. They provide a probabilistic measure of class membership for a given tuple. Bayesian classifiers are based on the Bayes' theorem. If X is a data tuple, H is the hypothesis that a tuple belongs to a class C. Then the probability measure calculated by the Bayesian classifier is $P(H/X)$, which describes the probability that a tuple X belongs to class C given the attributes that known of X [130]. Bayesian classifiers such as the Naive Bayes classifier have shown comparable performance to neural network classifiers and show high accuracy for large databases. Yung [90], Monrose and Rubin [96], Orebaugh and Allnutt [13] used a Naive Bayes classifier for the classification of users.

2.8.2 Machine Learning Approaches

As mentioned earlier, machine learning investigates the best techniques in order to make a computer *learn* so that it can make intelligent decisions. The following machine learning techniques have been explored with user attribution goals.

k-Nearest Neighbor Classification

The k-Nearest Neighbor (k-NN) method has been widely used for pattern recognition. Assume that each tuple with n attributes in the dataset is a point in an n-dimensional space. The k-NN classifier works by searching for k number of tuples closest to the unknown tuple. The closeness can be defined in terms of different distance metrics such as Euclidean distance [130]. k-NN is an example of instance learning or lazy learners where all processing occurs when the algorithm is presented with a test tuple. Wang, Zhang, and Gombault [82] and Saljooghinejad and Rathore [105] used the k-NN classifier and reported promising results.

Support Vector Machines

Support vector machines (SVMs) finds the maximum margin classifier from the set of classifiers that separates a set of instances. This hyperplane is called the support vectors. The classifier uses the support vectors to classify test samples. If the samples are not linearly separable, then it transfers these instances to a higher dimension called the kernel space [132]. Several researchers have successfully explored the classification of users based on behavioral biometrics using SVM [119], [77], [80], [99], [100], [23]. Chen, Hsu, and Shen [81] compared analysis using SVMs and artificial neural networks (ANNs) and found that classification schemes based on SVMs outperformed those using neural networks.

Neural Networks

Backpropagation is a neural network learning algorithm. In simple terms, a neural network is a set of inputs and outputs with each connection having a weight associated with it. The back-propagation algorithm iteratively learns a set of weights for prediction of the class label of tuples. It performs learning on a multi-layer feed-forward neural network. As seen in Figure 2.3 the multi-layer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer. [130]

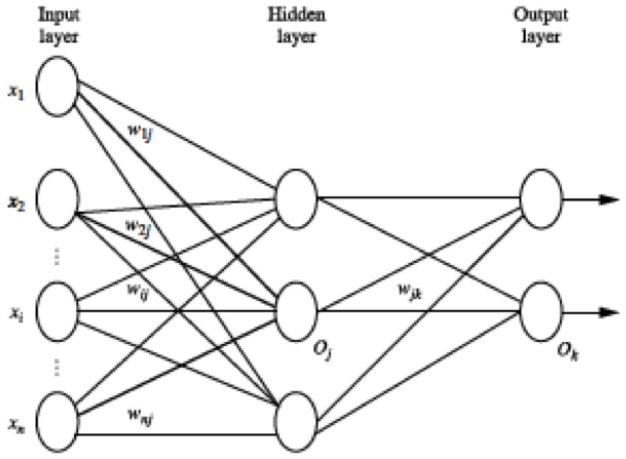

Figure 2.3. Example of Multilayer Feed Forward Network [130].

Obaidat and Sadoun [47] classified users with a reported accuracy rate of 100%, based on keystroke data using neural networks based classification techniques such as the fuzzy ARTMAP, radial basis function networks (RBFN), and learning vector quantization. They reported an identification accuracy of 100% within their test dataset. Imsand, Garrett, and Hamilton [133] also used artificial neural networks to model user interaction.

2.8.3 Similarity Matching

Similarity measures (also known as proximity measures) are used within different classification and clustering algorithms to compute the extent to which data tuples are similar [130]. Lane and Brodley [75] and Imsand and Hamilton [103] used similarity matching for their user classification techniques.

2.8.4 Discussion on Classification Techniques

The earlier sections provided an outline of the data mining techniques used towards similar user profiling tasks. There is no correct answer to determine the data mining technique for any scenario without trying different techniques. Data mining tasks usually involve trying several different options and using cross-validation techniques to choose one technique or a combination of techniques. When a large number of positive examples are available, and a good estimate is known of what the positive examples in the future will look like, there is a large overlap between anomaly detection algorithms and supervised learning techniques.

While Naive Bayes has shown promising results in the literature, it relies heavily on conditional independence. It also cannot learn the interactions between features. A Support Vector Machine with a Gaussian radial basis function (RBF) should provide a robust and efficient option to model data for a multi-modal feature set combining keystroke and mouse usage information. It can be assumed that the computer behavior being modeled is non-linear because of the inter-dependency between programs in an operating system like Windows [76]. The choice of the kernel does not have a big impact on the accuracy of the SVM classifier. An SVM with a Gaussian RBF kernel gives the same decision hyperplane as a type of neural

network known as a radial basis function network. However, there are many advantages of choosing SVMs over ANNs in this scenario.

One big advantage of SVMs over ANNs is that neural networks can converge to local minima whereas, in theory SVMs provide a global and unique solution. Support vector machines have a simple geometric interpretation and give a sparse solution. Also, their computational complexity does not depend on the dimensionality of the input space. ANNs use empirical risk minimization, whilst SVMs use structural risk minimization. The biggest advantage of SVMs over ANNs, which also accounts for its popularity, is that it is less prone to over-fitting [130]. To classify multi-class data, such as the data in discussion, single class SVMs can be combined with the help of error-correcting codes.

SVMs are also suitable for high-dimensional data without requiring the application of dimensionality reduction techniques such as PCA, even in scenarios where the number of features exceeds the number of classes. In terms of performance, SVM has low performance overheads because the decision functions use a subset of the training samples. SVMs are also less prone to the effects of outliers. Lastly, SVMs can be robust even when the training samples are biased. Given these considerations, SVMs have been a popular choice for user attribution tasks in digital forensics [11].

With that being said, the field of machine learning is continuously evolving with new, promising approaches being developed. One such approach is *gradient boosting*, with XGBoost or *eXtreme Gradient Boosting* as a specific implementation of gradient boosting. While there doesn't appear to be much research comparing the use of XGBoost and SVMs towards user attribution, XGBoost has shown tremendous performance in complex classification tasks and out-performed SVMs in similar predictions [134]. The proposed research will use XGBoost as the classifier to model user behavior. The next section provides more details on the evolution and working of the XGBoost algorithm.

2.9 Digital Forensics Readiness Framework

Most digital forensic investigations focus on what happens during and after an investigation [135]. Assuming the required evidence exists, digital forensic investigations can use

the evidence to build and prosecute the case. If the evidence doesn't, the suspect can't be charged and prosecuted. The quality and availability of evidence was seen as a passive aspect, one that couldn't be controlled more than ensuring that appropriate digital forensics procedures were used as part of the investigation.

Rowlingson *et al.* [135] was one of the early researchers to discuss the importance of not only what happens during a digital forensic investigation, but also the setup and events prior to undertaking an investigation. He introduced the notion of collecting evidence in advance of a crime, so the availability of evidence can be enhanced proactively, as opposed to only responding in a reactive manner. This proactive approach, known as digital forensics readiness, is a method that collects, preserves, pre-processes and stores potential information that may have evidentiary value or provide corroborative insight during investigations [11]. This approach can be useful towards a $1 : N$ identity matching as opposed to the $1 : 1$ matching provided by approaches such as psychological profiling [11]. Digital forensic readiness is also an important aspect of an organization's security strategy [21]. Proactive digitally forensics evidence management is required to ensure that the require evidence can be collected from the environment, while minimizing any disruption to business. This requires planning and testing the processes ahead of time, while often influencing security processes across the organization (e.g., log collection and auditing processes). In fact, digital forensic readiness is usually a requirement through regulatory norms such as the Sarbanes-Oxley Act [136]. In spite of this, organizations are frequently unprepared and often don't have policies and processes to ensure availability of data that can provide evidentiary value in court [137].

Ikuesan and Venter [10] proposed the Behavioral Biometric Digital Forensic Readiness Framework (BBDFRF) that includes augmentation of user attribution through the use of behavioral biometrics. Figure 2.4 shows the forensic readiness phase that includes behavioral biometrics.

Similar to the phases in a digital forensic investigation, and as outlined by Ikuesan and Venter [10], the BBDFRF consists of 4 phases:

1. Acquisition phase - As seen in the figure, this phase includes two sub-processes.

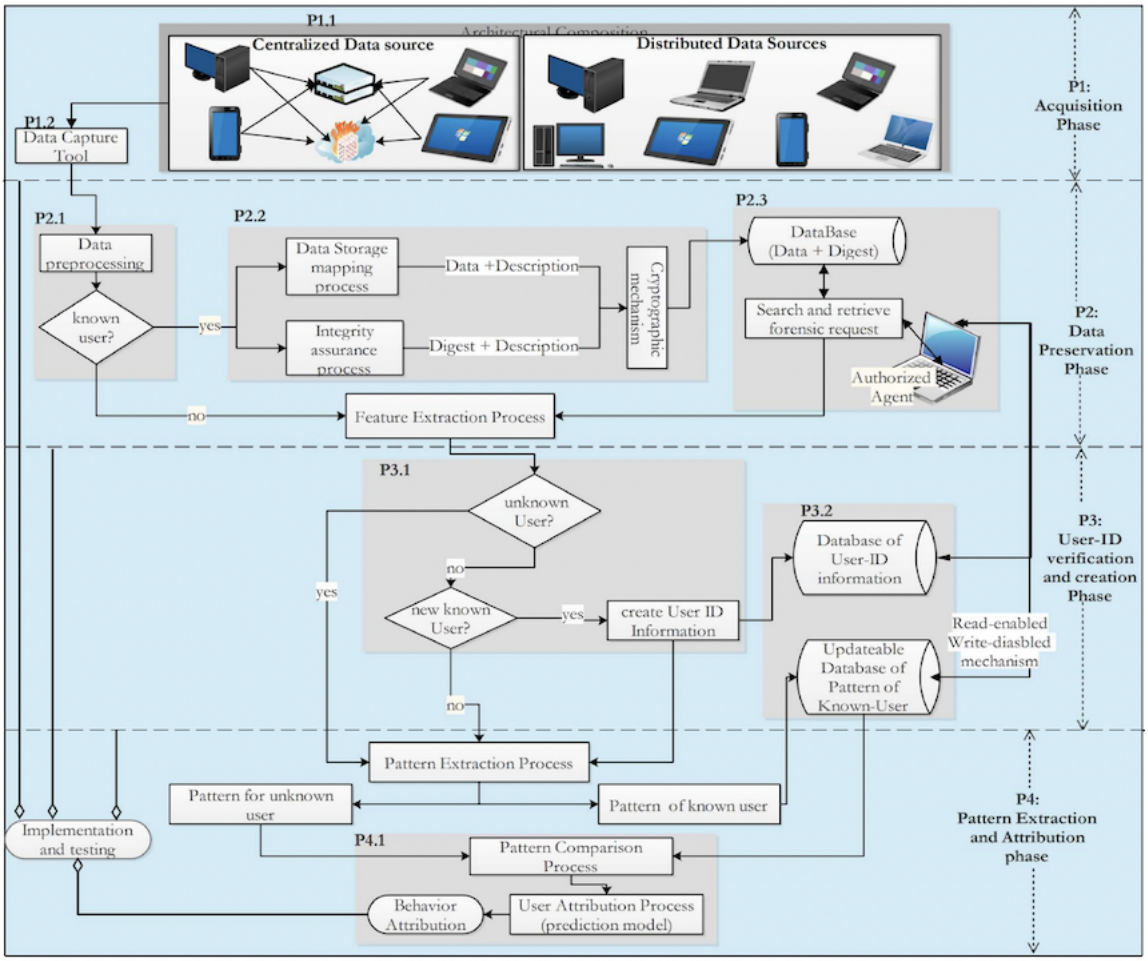

Figure 2.4. Behavioral Biometric Digital Forensic Readiness Framework. [10]

- P1.1: data collection architecture - this includes the identification of potential behavioral evidence. This can include uni-modal biometrics or multi-modal biometrics that is a combination of different features such as keystroke and mouse dynamics. This would also include details about the storage architecture, e.g., a centralized approach or a distributed approach.
- P1.2: development and deployment of capture tool - this include the acquisition tool for each type of behavioral biometrics under consideration. Other considerations would include the device compatibility, data encryption techniques, and privacy attributes.

2. Data Preservation phase - This phase includes three sub-processes.

- P2.1: data preprocessing and data representation - the nature of the behavioral biometrics used would determine the preprocessing. As an example, a multi-modal biometric might have more complex preprocessing.
- P2.2: forensic assurance - verifying the integrity of the stored data to ensure that it is forensically useful, keeping trade-offs between cost and performance in mind.
- P2.3: data storage mechanism - describing the specifications of the data storage, including details of access controls and decryption algorithms in use.

3. User-ID Verification and Creation phase - This phase includes the following sub-processes.
   - P3.1: User-identity verification and/or creation - verification is done if there is previously known information about the user. If there is no previous data, user creation details are stored in this phase of the process.
   - P3.2: storage mechanism of user-identity information - storing data about identity of such as the user patterns of known users
4. Pattern Extraction and Attribution phase - This phase is the main phase that conducts user attribution such that a given action/event can be attributed to a specific user without a very low possibility of repudiation. As proposed in the following section, this usually involves the development of a classifying or attributing technique that uses a statistical or machine learning algorithm and stored patterns of known users, in order to attribute a specific action to an unknown user. Metrics that are commonly used to test the accuracy of machine learning algorithms, such as equal error rate, F-1 scores, receiver operating characteristics curves, false accept rates, false reject rates etc. can be used to report on the confidence in the assigned attribution Ikuesan and Venter [10].

This framework shows how feasible approaches for behavioral biometrics can be used to improve the nature of data available prior and during digital investigations [10]. This data can be used towards reconstruction of the event, response planning, training and retraining processes [72].

2.10 Summary

This chapter provided a review of the literature in the field of user attribution in general and then specifically with an emphasis on digital forensics. As discussed, while there are many studies in the general area, there are few gaps that the current literature doesn't address:

- Most studies employ smaller sizes. Shen, Cai, Maxion, *et al.* [125]'s sample size of 58 users for mouse dynamics is one of the bigger sample sizes, but they only focused on whether demographic data can be predicted on the basis of keystroke and mouse patterns.
- The studies that are forensic-focused do not employ a multi-modal approach.

The proposed research study will greatly enhance the sample size with 117 participants. It will also explore a multi-modal approach using both keystroke and mouse dynamics. The next chapter provides the framework and methodology to be used in the research project.

Table 2.1. Related works in touchscreen biometrics [34]

Study	Users	Features	Classifier	SD%	DD%
Frank, Biedert, Ma, *et al.* [107]	41	27	SVM,kNN	EER: 2.0-3.0	EER: 0.0-4.0
Serwadda, Phoha, and Wang [110]	190	28	Ten different classifiers (best logistic regression, SVM, and random forest)	-	EER: 13.8-36.0
Xu, Zhou, and Lyu [108]	32	37	SVM	EER: 10.0	Acc: 70.0-100.0
[109]	71	15	SVM, random forest, kNN	-	-
Zhang, Patel, Fathy, *et al.* [111]	50	27	SVM, sparsity-based	EER: 4.1-5.9	EER: 4.9-14.4
Mondal and Bours [112]	-	15	ANN	FNMR: 0.0 FMR:0.08	-
Murmuria, Stavrou, Barbará, *et al.* [113]	73	5	StrOUD	-	EER:32.1-46.3
[114]	28	5	kNN, random forest	Acc: 88.0-92.0	-
Mahbub, Sarkar, Patel, *et al.* [115]	48	24	kNN, SVM, GBM random forest	EER: 22.1-38.0	-
Shen, Zhang, Guan, *et al.* [116]	71	22-27	SVM, random forest, kNN, ANN	FAR: 1.9-7.4 FRR: 2.7-8.6	FAR: 4.7-10.9 FRR: 5.7-13.5
Sitová, Šeděnka, Yang, *et al.* [117]	90	22-27	Scaled Manhattan, Scaled Euclidean	-	EER: 15.0-16.0
Kumar, Kundu, and Phoha [118]	-	-	Bayesian and MiniMax QCD	Acc: 80.1-89.6	-

3. DISCUSSION ON LEGAL AND PRIVACY CONSIDERATIONS

This chapter discusses the legal and privacy considerations of the suggested user attribution technique.

3.1 Legal Considerations

Forensics is defined as the use of scientific knowledge or methods to solve crime [138]. With the introduction of any new proposed techniques, there is need to analyze the admissibility of such evidence in a court of law [139]. Researchers have commented that while there is a large emphasis on the development of new methodologies, there has not been much formal evaluation into the degree of uncertainty of using these new techniques [140] [141]. There is a gap between what constitutes as proof between the technical community and the legal community [142].

Imagine a scenario where a protected network gets hacked. The network has an intrusion detection system in place that allows the administrators to trace the hackers, who are then arrested. During the preliminary hearing, the hackers' lawyers convince the judge that the techniques used to trace the hackers are not sufficiently tested and analyzed. Not only does the judge agree that the evidence is not enough, the charges are dropped and in return, the hackers press charges for defamation of character. This is an actual incident that happened at George Washington University [143]. In situations like this, it is very important for the techniques used in court to be tested, validated and peer-reviewed.

Until 1993, the Frye test was used to determine the admissibility of expert witness testimony related to scientific evidence. The basic premise of the Frye test was that for a scientific technique to be admissible, the scientific principles on which it is based should be generally accepted by the scientific community that is involved with it [144]. In *Daubert v. Merrell Dow Pharmaceuticals, Inc.* [145], the court decided that the Rule 702 of the Federal Rules of Evidence did not have to incorporate Frye's general acceptance test to establish

admissibility of scientific expert testimony, but instead used a flexible standard to establish scientific validity of procedures used. In Daubert, the guidelines proposed were:

- Judge is gatekeeper: The judge is responsible for ensuring that the scientific testimony is based on scientifically sound knowledge and principles.
- Relevance and reliability: The trial judge is responsible to ensure that the expert's testimony is relevant and is based on reliable principles.
- Scientific Knowledge : Scientific assertions made by expert witness will be accepted as scientific knowledge if it is based on the scientific method.
- Factors: Daubert made suggestions for several factors that can be taken into consideration to establish the validity of scientific procedures used:
  - whether the methods have been adequately tested,
  - whether they are peer-reviewed and published,
  - whether the techniques employed have a known error rate,
  - whether they are subjected to controls and standards, and
  - whether they are generally accepted within the relevant scientific community.

It is important to note that these guidelines were not meant as rules. It is not required that each of the guidelines is followed, and it is possible that the expert witness's testimony may be considered scientifically valid without one or more of the guidelines being fulfilled.

In *General Electric Co. v. Joiner* [146], the Supreme Court decided that an appellate judge should use an abuse-of-discretion standard of review to review a trial court's decision to admit expert witness testimony. The district court judge should keep the focus on the methodology and not on the conclusions. This was the second case that formed the basis of the Daubert standard. The third case was *Kumho Tire Co. v. Carmichael* [147] which established that the Daubert considerations did not only extend to all scientific testimony but to all other testimony which included scientific techniques, engineering techniques or other specialized knowledge.

The Rule 702 of the Federal Rules of Evidence was first amended in 2000 to include the Daubert Criteria. This was amended again in 2011. The current version of Rule 702 is [148, pg. 123]:

"RULE 702. TESTIMONY BY EXPERT WITNESSES

A witness who is qualified as an expert by knowledge, skill, experience, training, or education may testify in the form of an opinion or otherwise if:

(a) The expert's scientific, technical, or other specialized knowledge will help the trier of fact to understand the evidence or to determine a fact in issue;

(b) The testimony is based on sufficient facts or data;

(c) The testimony is the product of reliable principles and methods; and

(d) The expert has reliably applied the principles and methods to the facts of the case.

(As amended Apr. 17, 2000, eff. Dec. 1, 2000; Apr. 26, 2011, eff. Dec. 1, 2011)"

3.1.1 Legal considerations of similar techniques

The main emphasis of the Daubert standard is that the procedures used by the expert witness should be based on the scientific method. If there is any doubt about the scientific validity of the technique, it can be challenged and invalidated in court. Within the digital realm, this is an even bigger challenge because it is possible for the same data to provide different conclusions based on the analysis techniques employed by the analyst [141]. The method proposed in this dissertation bears similarity to other techniques such as psychological behavior profile analysis and also profiles created for intrusion detection.

1. Linkage analysis: Behaviors shown on a computer has some similarities to linkage analysis on psychological profiles. Linkage analysis refers to the method of linking two suspect profiles by analyzing the crime scene, the modus operandi, and signature behaviors [149]. There have been a few interesting cases in psychological linkage analysis relevant to the technique being discussed. In *Pennell v. State* [150], the expert witness testified that based on the similarity in the pattern, it could be determined that the crime was committed by a serial killer. The court determined that while the expert can testify to the similar patterns leading to a 'signature behavior', it would not

allow 'profile' behavior testimony. Their interpretation of profile analysis was that it attempts to link traits of a specific individual to general characteristics of a population. With respect to the technique under discussion, it would be the equivalent of creating user profiles and generalizing that the profile indicates a certain profession or certain gender. However, this research merely attempts to differentiate between different users on the computer without attempting to extrapolate these profiles to behavioral traits in a general population.

In the case of *State v. Fortin* [151] the expert witness testified that different crimes were committed by the same individual because there was an unusual pattern seen in both crimes. The witness was considered an expert on behavioral linkage analysis, but the court determined that the scientific reliability of his techniques was not sound. Only a few peers of the expert witness were familiar with his technique, so there wasn't adequate peer review. Also, the witness testimony was could be providing direct conclusions about the guilt. In that scenario, it seems like technique used by the expert witness was not based on statistical inference but rather based on experience. The number of similarities in the cases were overshadowed by the number of differences. There were no scientific determinations of the number and extent of similarities required to define an unusual pattern. Unlike the psychological analysis in this case, the computer behavior analysis technique uses scientifically established machine learning algorithms with empirical data about the extent to which two profiles can be considered similar.

2. Intrusion Detection systems: The computer behavior profile analysis technique in consideration uses similar techniques as post-event audit trails which detect unusual patterns using statistical anomaly detection. Similar to the profile analysis methodology under question, the key requirement in an intrusion detection case is to establish the identity of the perpetrator [142]. In cases involving intrusion detection, it is often challenging to obtain sufficient evidence because of the same challenges as mentioned for the profile analysis technique. Therefore, the technique depends on statistical analysis of complete logs with the requirement that the expert can provide a means to vouch

for the authenticity, completeness and relevance of the logs. As long as these challenges are met, intrusion detection techniques are known to withstand the challenge of scrutiny in court [142]. There has been research on integrating legal requirements with the evidence collected to obtain a pre-estimate of admissibility of the evidence in court [152]. While there are some difference in the techniques, research in intrusion detection systems supports the legal admissibility of computer behavior profiling techniques.

The Daubert criteria provides guidelines to allow testimony provided by the expert witness, which is usually based on scientific assertions. An assertion in science is either proven or unproven. When unproven, it is stated as a possible conjecture while providing the degree to which its occurrence is likely. However, in judicial situations, it is often the expectation that science provide answers that are absolutely conclusive, and free of uncertainty. In practice, such assertions are can rarely be made [153]. Faigman, Kaye, Saks, *et al.* [154] have commented that the debate shouldn't be focused on science or not-science, rather the focus needs to be on the specific method employed to draw an inference. And for such methods, the existence of data supporting the method in turn supports the expertise of the presenter.

3.2 Privacy Considerations

User data is prolific today. Organizations have already been modeling and using user data for attack detection and prevention systems. These approaches often combine both technical as well as psychological data characteristics [155]. As expected, there is a trade-off between using such user modeling techniques against the privacy of individuals that are being profiled [155]. This trade-off is also applicable and has been previously discussed in other user profiling scenarios such as intrusion detection systems [156], [157].

The privacy concerns depend on many specific factors such as the nature of the environment where data is collected. In a professional or academic environment, users are usually warned about, and accept the lack of privacy on computers that belong to the organization. Gathering data on such computers might mitigate some of the privacy-related considerations.

For data that is collected on computers where users can have a reasonable expectation of privacy, user-profiling techniques have bigger hurdles to overcome. Data anonymization

might not be feasible since there is a need to link to the user's identity as part of developing the *known* profiles. One of the approaches to solve this problem would be to adapt solutions that allow investigating agencies to revoke the anonymization under strict conditions [158]. This places a large burden of ethical behavior on law enforcement to ensure that the data is used within the strict ethical boundaries of the purpose for which it was intended. The advantage of the proposed technique is that it suggests the storage of data within the users' local devices (e.g. through the use of forensic-friendly extensible operating systems), allowing for more user control and oversight on how the data is accessed. The storage of the data would also have to ensure adequate protection through techniques such as encryption, preventing theft and misuse of user data. Access to this data can fall under the search and seizure boundaries of the user laptop itself, e.g., through securing the necessary warrants. At the same time, the nature of user data collected (e.g. keystroke timings and mouse coordinates) presents the advantage of having limited value outside of the specific task of correlating user activity on that computer and should help to alleviate concerns related to privacy violations.

4. MACHINE LEARNING OVERVIEW

As mentioned in previous sections, the high-level principle behind the methodology is to model the keystroke and mouse usage behavior of each user on a computer, in order to discern between different user sessions, even if they are logged into the same account.

Events that are repetitive in nature can show habitual patterns for users. These can either be temporally ordered or clusters with temporal proximity [159]. Such patterns can be used to discern the operator of the device at the specific instance of time, even if that instance was a one-time event. Based on temporal sequence or proximity with other events, the investigator might be able to build user profiles that provide the ability to demonstrate increased probability of a specific user [159]. If the pattern shows that it was the regular user, this can assist in refuting the 'it wasn't me' defence or inversely if there is anomalous behavior, it can provide indication that the operator at the time of the action was not the regular user as might be expected.

While relatively rarer in its applicability to digital forensics, the use of such *training* is not novel in the field of cybersecurity. Figure 4.1 shows the structure of a generic adaptive cybersecurity defense system, that relies on such learning.

4.1 Machine Learning

Samuel [161] introduced the term 'machine learning', describing it as the field that allowed computers to *learn* without having to be programmed explicitly. Learning refers to the process of finding statistical regularities or other patterns in data [162]. Mitchell [42] formalized the notion through the definition that, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

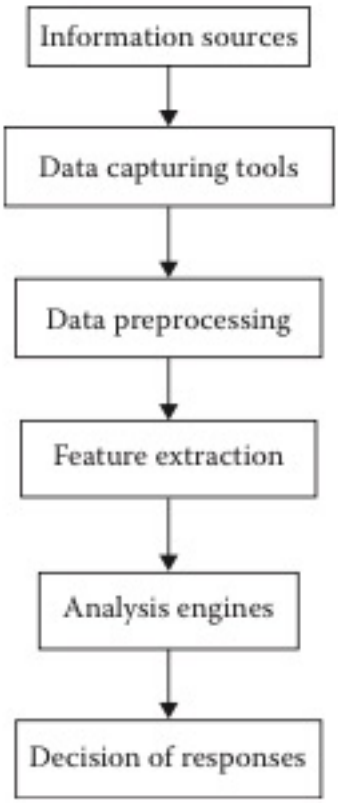

Figure 4.1. Adaptive defense system for cybersecurity [160]

Table 4.1. Development of Machine Learning [54]

1950	Alan Turing's Turing Test was an initial foray into machine learning. The test required a machine to convince a human that they were speaking to a human [163]
1952	Arthur Samuel created a self-learning algorithm that could learn as it played the game of Checkers with itself [161]
1956	Artificial Intelligence is conceptualized at a conference in Darthmouth organized by Martin Minsky, John McCarty, Claude Shannon, and Nathan Rochester
1958	Frank Rosenblatt proposed the concept of the perceptron, laying the foundation for Artificial Neural Networks (ANN) [164]
1967	Pattern recognition using the Nearest Neighbor algorithm is proposed [165]
1979	An 'intelligent' robot that could navigate obstacles, the Stanford Cart, was developed by Stanford University [166]
1981	Gerald Dejong proposed 'Explanation-based learning' training a computer to create rules for discarding useless data [167]
1985	Terry Sejnowski invented NetTalk, training it to pronounce English words [168]
1990s	Emphasis shifted from knowledge-driven to data-driven, with the goals now focusing on analysis and interpretation of large amounts of data [169]
1997	IBM invented the Deep Blue computer, which defeated chess champion Gary Kasparov
2006	Geoffery Hinton coined 'Deep Learning', referring to neural networks that learned through multiple layers of neurons [170]
2011	IBM's Watson computer used natural language to defeat a human competitor at Jeopardy [171]
2012	Google's Jeff Dean developed 'Google Brain', a deep neural network to detect patterns in images and videos [172]
2014	Facebook developed 'Deepface', an algorithm based on deep neural networks, to detect human faces in pictures [173]
2015	Amazon and Microsoft develop their machine learning platforms to support distributed processing of algorithms [174] [175]
2017	Widespread adoption across different industries with Google (Google Lens, Google Clicks, Google Home Mini etc.) and Apple (Apple Homepad) launching machine learning enabled devices

In this case, the task T refers to the process that an algorithm needs to perform [52]. $x \in R^n$ where each entry x_i is a feature. Considering the example of a classification task, this can be represented as:

Learn $f : R^n \rightarrow \{1, .., k\}$

$y = f(x)$: assigns the input to the category with numerical code y

The performance P can refer to [52]:

- *Accuracy*: The ratio of examples for which the model accurately predicts the category
- *Error Rate*: The ratio of examples for which the model inaccurately predicts the category to which they belong

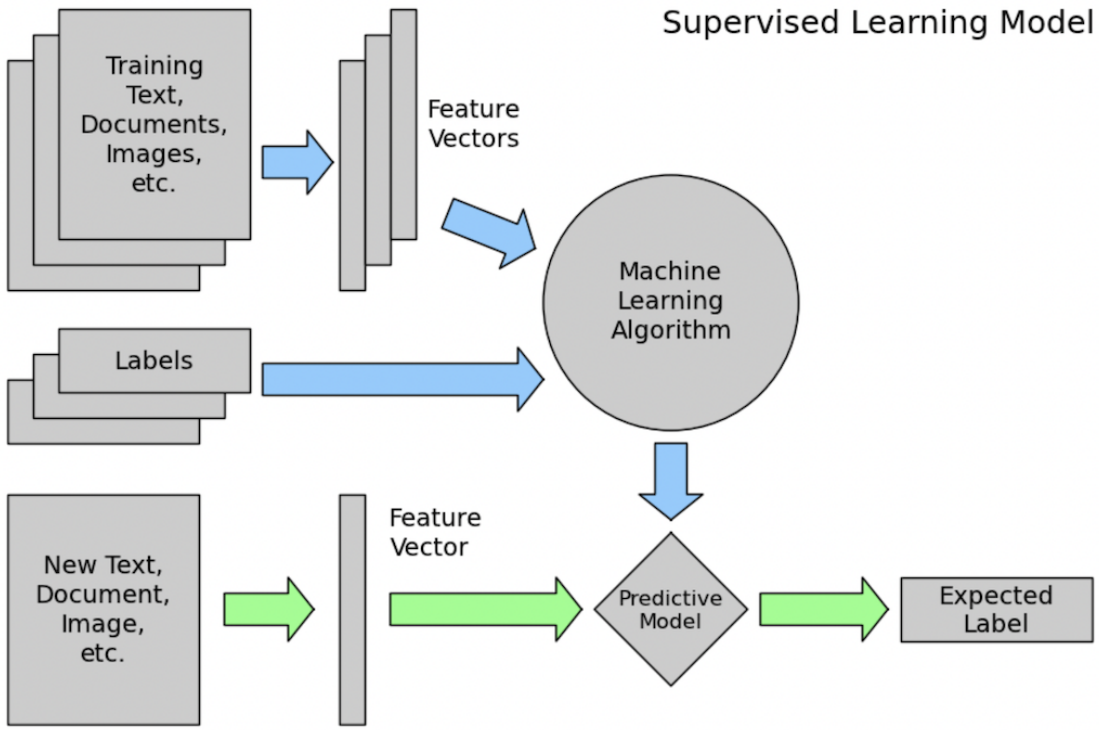

Figure 4.2. Supervised Learning Process [176]

The experience E can represent different approaches such as supervised or unsupervised algorithms [130][52]. Supervised learning techniques usually consist of some training data with labeled classes that can be used to train the system. The output of the learning

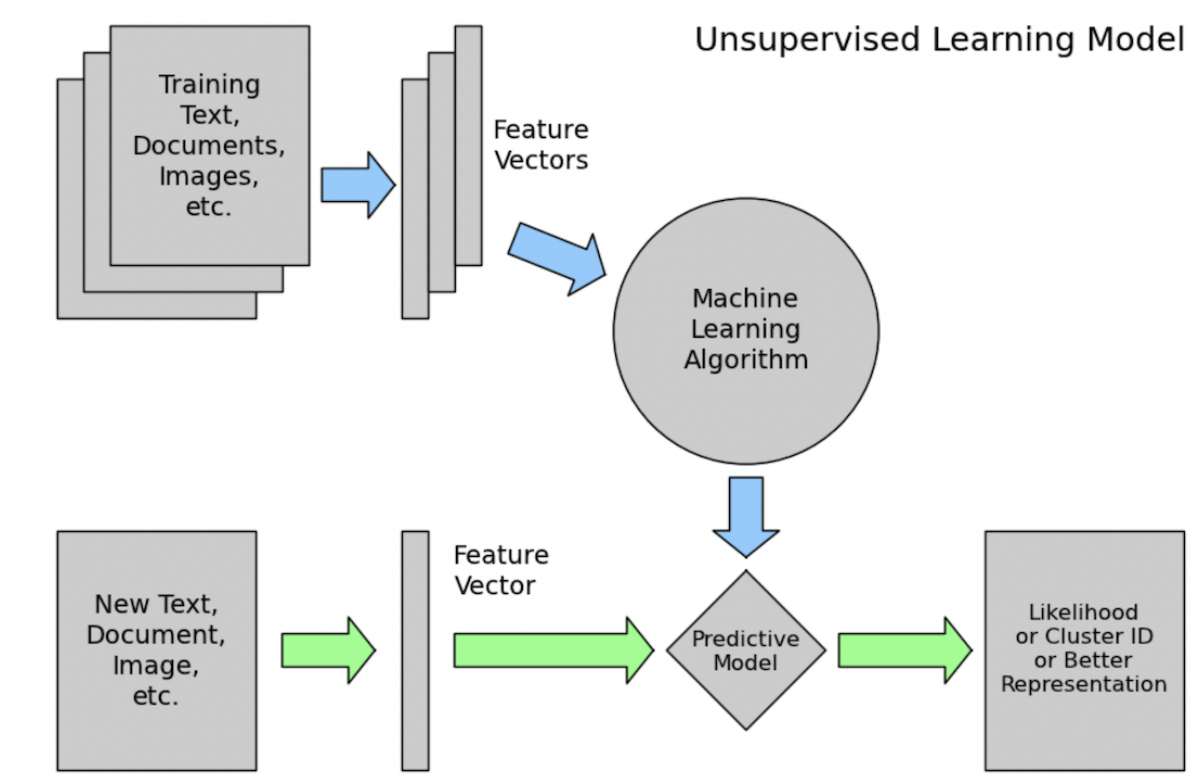

Figure 4.3. Unsupervised Learning Process [176]

technique can then predict the class label of the input variables. Classification and prediction are examples of supervised learning techniques, that rely on previously labeled data to make classifications or predictions. Figure 4.2 represents the process of learning using supervised techniques. Unsupervised techniques, such as clustering and associative rule mining, attempt to discover hidden patterns in the data without the use of labeled classes or training data. Figure 4.3 represents the process of learning using unsupervised techniques.Table 4.1 shows key developments in the field of machine learning, as it expanded its use and applicability across most disciplines and sciences today.

Data science is the moniker for a field that focuses on analyzing data through the unification of statistical, data analysis, and machine learning techniques [177]. Hernán, Hsu, and Healy [178] classified data science tasks as either descriptive, predictive or related to causal inference. More details are provided below. However, the *right* approach depends on the

specific problem. In some scenarios, traditional statistical approaches may be equally good or even better suited to the nature of the problem [179].

- *Description*: Descriptive tasks refer to tasks that provide additional information about available data. While on the surface, it appears that traditional statistical approaches are well-equipped to provide such information, machine learning may assist in discovering unknown relationships between multiple sources of data, leading to advanced analyses not previously obtained through traditional means [179]. Unsupervised learning is especially useful for such tasks as they can discover hidden patterns in data, without a specific outcome [179]. With the explosion of data available and used on a daily basis, this is very relevant to digital forensics today [37], and can be used in many areas - e.g., discovering patterns of criminal activity [180], discovering anomalous network activities, separation of audio noise etc.

- *Prediction*: Predictive tasks focus on events that may happen in the future, based on available data. Depending on the nature of the tasks, traditional statistical approaches like logistical regression may be used to test developed hypothesis and provide confidence levels. However, if the available dataset is large and complex, machine learning approaches (that may overlap with statistical methods) might be more suitable since they have fewer required underlying assumptions (e.g. linear relationships, absence of multi-collinearlity) [179]. In the field of digital forensics, there are many such scenarios of predictive tasks, such as image recognition, malware analysis, fraud detection etc. where machine learning is providing advanced analyses opportunities that were previously unavailable [37].

- *Causal Inference*: Causal inference tasks are related to assigning a *cause* to an event. It allows us to say that y happened because x happened. Statistics and Probability Theory has been the foundation of causal inference in forensic science, including digital forensics. However, machine learning approaches are required for the creation of dependable models of causality and inference, especially in cases where the available information is incomplete or uncertain [181]. Machine Learning is enabling complex investigation approaches that were previously not feasible [182].

Another lens to approach this from is the data science problem that machine learning task is assisting to address [183]. Some key problems that machine learning techniques can help to solve:

- *Classification Problem*: problems related to assigning an input to an output, where the output consists of fixed number of classes. The research problem of the proposed study, of attributing a user session to the correct user, is an example of a classification problem.
- *Anomaly Detection Problem*: problems related to analyzing a pattern of data to detect outliers or changes from expected behavior.
- *Regression Problem*: problems related to predicting future numerical or continuous data.
- *Clustering Problem*: problems related to identifying structures within data and clustering based on observed patterns.
- *Reinforcement Problem*: Problems associated with taking actions based on past experience, to maximize the defined reward function.

4.1.1 Building Machine Learning Models

Machine learning models have significantly evolved in both speed and capabilities, allowing complex mathematical calculations on large amounts of data [162]. The previous section briefly touched on the stages of building an adaptive, machine learning-driven cybersecurity system. This section explores the steps involved in more detail, considering the more commonly used supervised learning algorithms. As seen in Figure 4.4, the process aligns closely with the setup in traditional statistical experiments.

- *Data collection, processing, and feature engineering*: The first step would be to collect data, ensuring diversity and coverage. In user attribution, this could refer to ensuring diverse demographics are included as participants. Once data is available, data may

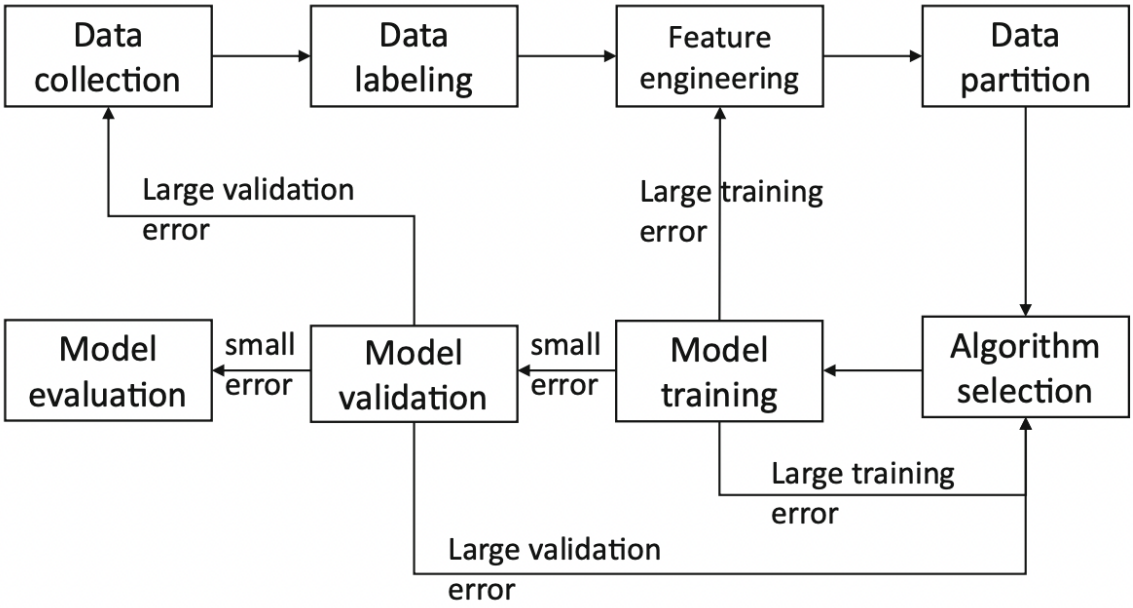

Figure 4.4. Typical workflow to build a machine learning model [51]

require preprocessing and labeling, especially in instances of supervised learning. Depending on the nature of the data and the problem, often this labeling requires human labelers. Depending on the datasets used, techniques may have to be applied to handle missing or outlier data. Batista and Monard [184] and Hodge and Austin [129] have proposed techniques to address each of those areas respectively. Feature engineering can refer to feature extraction from the data, reducing dimensionality, and normalizing features [51]. Feature selection can be guided by the expertise of the researcher, but may sometimes require a *brute-force* approach where all the features are used and the features with lowest contributions are removed. This approach requires larger preprocessing overheads [185]. New features, may have to be created from the basic feature set, in a process called feature transformation/construction [186].

- *Algorithm selection, model training, and hyperparameter tuning*: Algorithm selection is a critical step in developing machine learning models. It is common practice among researchers to test several preliminary classifiers, before choosing one that shows promise [187]. For supervised approaches, a common is to use two-thirds of the dataset for train-

ing and a third for testing [187]. Another approach is to use three disjointed subsets of the data - the training set used to optimize the model and algorithm, the validation set is used to choose the ideal hyperparameters and validate the performance, and finally the testing test is used to measure performance of the developed model [51]. Assuming that model M_1 is trained using hyperparameter λ_1, then model M_2 is trained using hyperparameter $\lambda_2 > \lambda_1$ and tested on the validation test. If M_2 is worse than M_1, this is repeated with a $\lambda_3 < \lambda_1$ and so on. This tuning is performed until an acceptable level of performance is seen on the validation set and the test set. Most machine learning algorithms require initial manual setting of the hyperparameters [183]. In another technique called cross-validation, the training set is divided into three equal and mutually-exclusive subsets and for each subset, the classifier is trained on the union of the other subsets [187].

In scenarios where the model performance is not acceptable in spite of attempts at hypertuning, it is may require additional investigation on the suitability of the chosen algorithm and feature set. If the performance is poor on both testing and validation datasets, this may indicate *underfitting*, or that the algorithm or selected features are not expressive enough for the training data. If the model performed well on the training set but poorly on the validation set, this is indicative of *overfitting* and this issue is known as generalizability. To take generalizability into consideration, the algorithm needs to balance the training error objectives with the complexity of the learner [55]. Approaches like regularization that limit the complexity of the model, enhancing the size of the dataset etc. can help to address overfitting issues.

4.1.2 Assessment of Machine Learning models

As seen in Figure 4.4, the last step in the workflow to build a machine learning model is to evaluate the model's performance to determine success. This can be assessed using the following metrics:

- *Accuracy*: this refers to the percentage of correctly classified instances. In the definition below, *TP* refers to True Positive, *TN* refers to True Negative, *FP* refers to False

Positive, and *FN* refers False Negative. Assuming a 2-class classification scenario, Figure 4.5 illustrates these concepts.

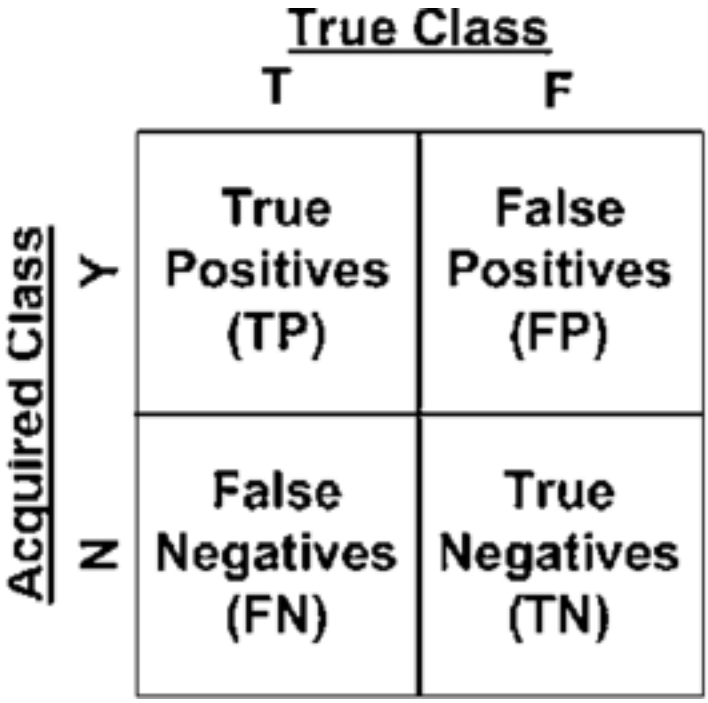

Figure 4.5. Classification of two classes [188]

$$Accuracy = (TP + TN)/(TP + TN + FP + FN))$$

While accuracy is an important indicator, but in the scenario that the class size ends up being very unequal, a bad classifier may still give a high accuracy. As an example, if class A contains 90% of the objects, and class B consists of 10%, a classifier that only recognizes objects from class A would still have a high accuracy [189]. To address this shortcoming, other indicators such as *precision* and *recall* are used, through a summarizing indicator known as the *F1 score*.

- *Recall*: This is concerned with how many of the relevant instances did the model capture through labeling it as positive?

$$Recall = TP/(TP + FN)$$

- *Precision*: This is concerned with how precise the model is i.e. how many instances were actually positive out of those that were predicted as positive?

$$Precision = TP/(TP + FP)$$

Precision and *recall* measure the *confusion* of a classifier. *Precision* reflects how *optimistic* a classifier is in its estimates or how often the classifier to add objects to a class. *Recall* on the other hand refers to how *pessimistic* a classifier is, or how often it neglects to add objects to the class that they belong to [189]. This naturally leads to the preference that each of these indicators should tend towards 1. To achieve a more rounded metric, F1 scores are used, a modified average that would tend to 1, if both *precision* and *recall* are close to it. It should be noted that a simple average of *precision* and *recall* would not give the same results.

- *F1 Score*: the weighted average of *precision* and *recall*. This score takes both false positives and false negatives into account and provides a realistic measure of

$$F1 = (2 * (Recall + Precision))/(Recall + Precision)$$

4.1.3 Supervised Learning

As discussed in the previous section, in supervised learning techniques, classes are predetermined by the human, with some subset of the available data labeled with the class that they belong to [162]. It is sometimes referred to as *Learning with a Teacher*, *Learning from Labelled Data*, or *Inductive Machine Learning* [55]. The main goal of supervised learning algorithms is to find patterns and construct mathematical models that can create a mapping between inputs and outputs. These models are then tested for predictability, while accounting for the variance from the initial data that was used for training. There is significant overlap between supervised learning and traditional statistical approaches. Jiang, Gradus, and Rosellini [179] outlines the overlap in Table 4.2.

Table 4.2. Overlap of terminology between traditional statistics and supervised learning [179]

Traditional Statistics	**Machine Learning**
Prediction	Supervised learning
Predictors/covariates/independent variables	Features
Outcome/dependent variables	Output/Target
Categorical outcome predictions	Classification
Continuous outcome predictions	Regression
Number/overlap of predictors	Dimensionality
R-squared	Coefficient of determination
Sensitivity	Recall
Positive predictive value	Precision
Contingency table	Confusion Matrix

Assuming that the data is a set of n pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in X$ and $y_i \in Y$, and the learning algorithm $f()$ maps the the input space X into output space Y. The learning process is called training and the pair $\{x_i, y_i\}$ consists of a sample where x_i represents the feature vector and y_i represents the output. Once trained, the learnt model $f()$ is applied to the next input set $x_j \in X$ and the outputs are predicted [51].

If the output consists of a finite set of discrete values that indicate the class labels of the input, this learning technique is used for classification. If the output consists of continuous values instead, this is a regression task [55]. The focus for this proposal is classification, specifically attribution of user sessions to classes representing users. In classification tasks, the output set can be considered to consist of k categories: $Y = \{1, 2, 3, \ldots, k\}$. The learning algorithm needs to develop a function $f()$ that can identify the one or multiple of the k categories that x_j belongs to [51]:

$$f : X -> \{1, 2, 3, ..., k\}$$

The input X can be in one of many different formats. In this research study, the input vector X is d-dimensional vector and $X = R^d$.

There are many supervised learning algorithms. As mentioned earlier, there is no *best* algorithm that can fit all the scenarios. With that being said, XGBoost, introduced in 2016, has proven to perform exceptionally well, and is used by leading applications in industry applications [190]. Figure 4.6 shows the evolution of XGBoost from Decision Trees. This study will develop the model using XGBoost, an algorithm that is yet to be explored towards user attribution in digital forensics. Given the vast number of supervised learning approaches, the next few sections restrict its scope to and provide an overview of the techniques that evolved from Decision Trees to XGBoost as seen in Figure 4.6.

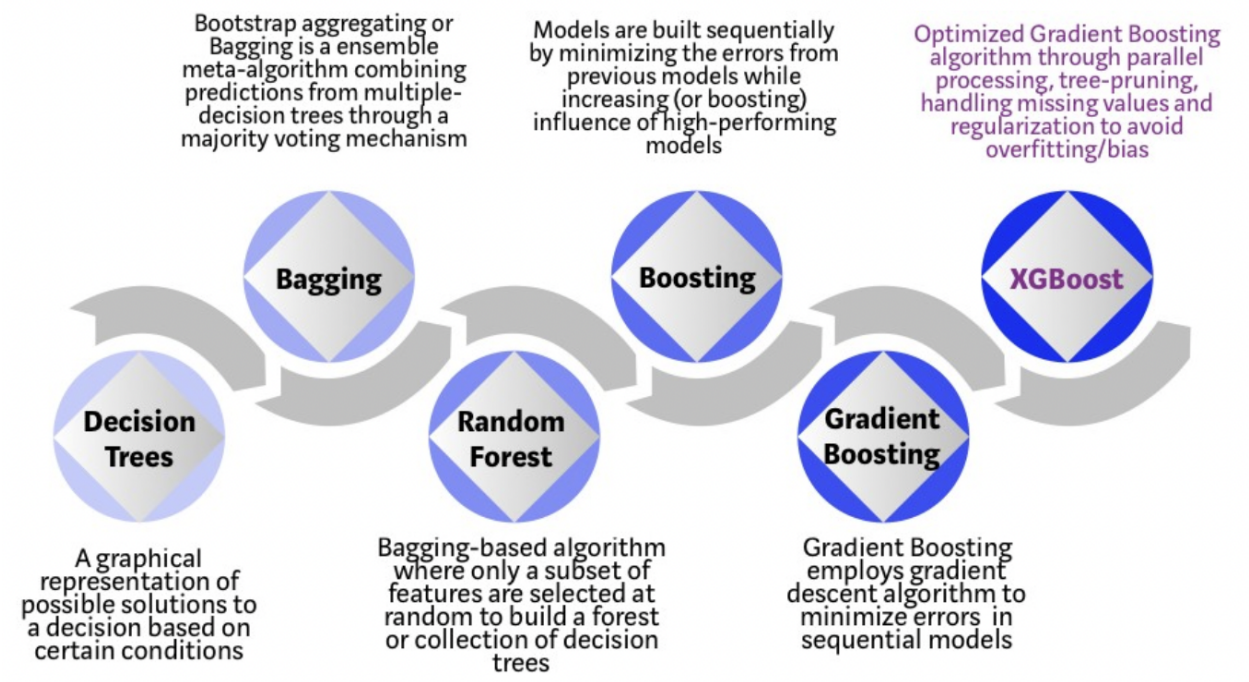

Figure 4.6. Evolution of the XGBoost algorithm from Decision Tree Learning [190]

Decision Trees

Approaches related to decision trees have received significant attention recently for their ability to represent the most complex problems given sufficient data [54]. While Artificial Neural Networks perform best in prediction tasks with unstructured data (e.g. image recog-

nition), algorithms that are based on decision trees are best performers for small-to-medium structured datasets [191][192][193].

Decision trees use sorted feature values for classification, where each node represents a feature and each branch represents a value that the feature can assume [187]. Figure 4.8 shows an example of a decision tree. The training set is shown in Figure 4.7. Based on the example decision tree, an instance x_i\{at1, at2, at3, at4\} = \{a1, b2, a3, b4\} would be assigned to the *Yes* class. The feature that is determined to be the biggest contributor is used as the root node [194].

at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Figure 4.7. Training Set for Decision Tree [187]

Decision trees can be used for both classification and regression and are commonly called Classification and Regression Trees (CART). For categorical outputs, as seen in the example above, the prediction focuses on the most commonly occurring class in a node. For contiguous outputs, the observation that belongs to a node is assumed to have a mean of the response values within that node [179]. The divisions can be made on many different criteria such as the best accuracy (for classification) or minimizing the residual sum of squares (for regression) [179]. Decision trees provide the advantage of high visual interpretability and are useful for visualizing relationships between different variables.

Generally when models such as decision trees are combined, or built upon with other processing, and used as a single classifier, such an approach is known as ensemble learning

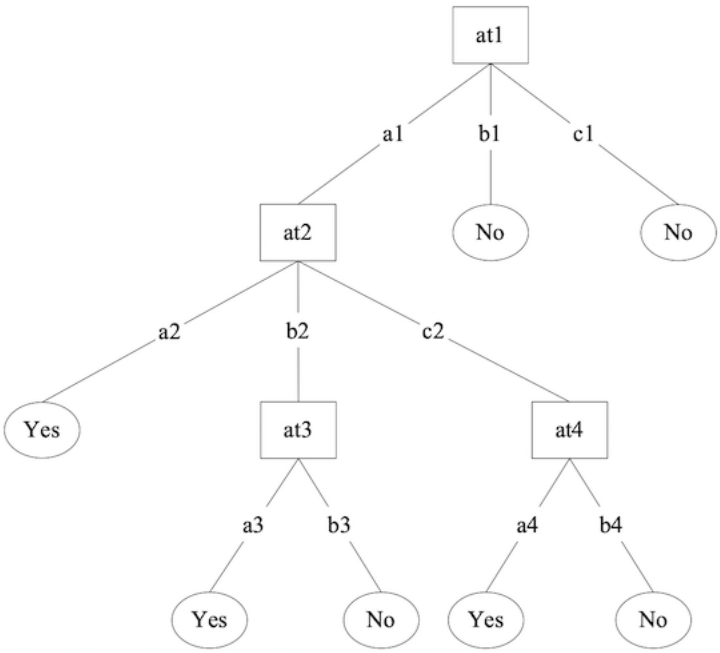

Figure 4.8. Decision Tree [187]

[195]. It i based on the principle that a combined committee of classifiers will perform better than a single classifier [196]. This definition appears to be fluid, since ensemble approaches have been suggested that include multiple classifiers running in parallel, without combining into a single classifier [197].

Ensemble approaches are currently the global standard for predictions, especially in case unstable learners such as neural networks and decision trees [199] [200]. Assuming that the variance measures the diversity of the classifier, it has been shown that performance is directly proportional to the diversity of the classifiers [201]. Restated, this can be explained that ensemble classifiers are more likely to be right, such that when they are wrong, they are wrong in different ways. Figure 4.9 represents a comparison between learning algorithms that use a single learner against ensemble approaches with different algorithms. While the success of ensemble approaches does not align with traditional statistical notions of *Garbage In Garbage Out* [197], they offer several advantages that make them the state-of-the-art approach in learning tasks:

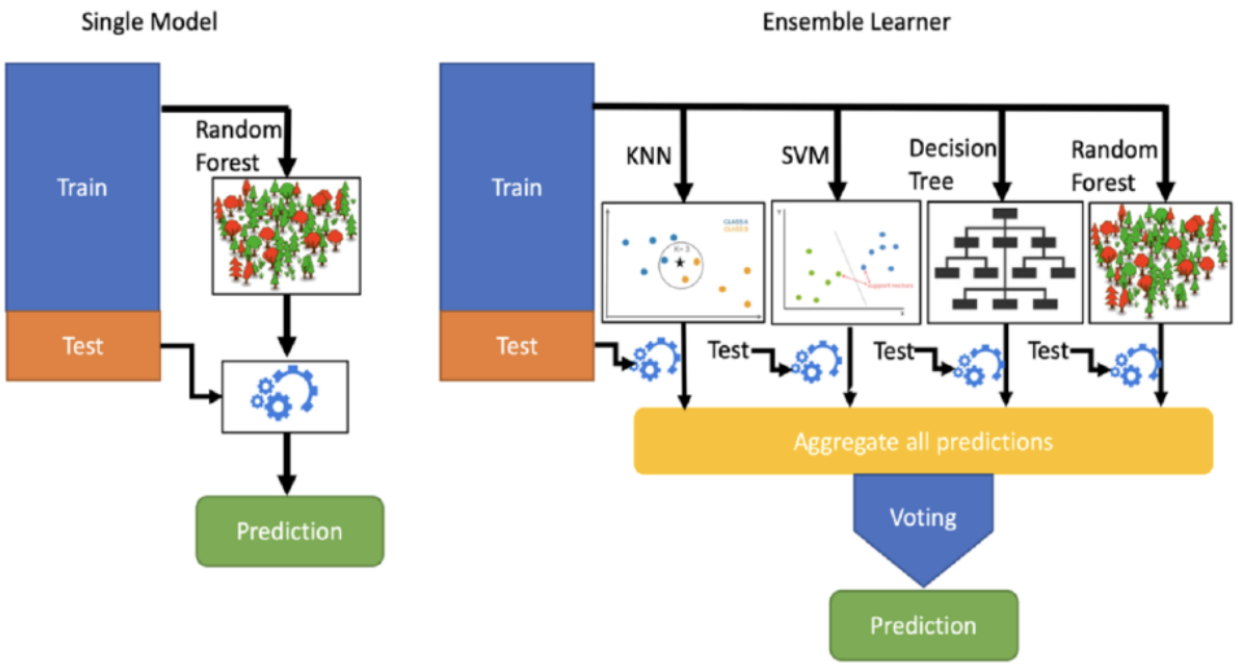

Figure 4.9. Single models against ensemble learners [198]

- Ensemble learners allow diverse algorithms to be applied to a diverse data, before the right model and the right data is determined. It allows optimization during the analysis run, and is inherently non-parametric [202].
- Without the correct data and algorithm known ahead of time, ensemble learners minimize errors and biases that are introduced when working with known data and algorithms.
- Statistical theory cannot account for data with high levels of complexity. Ensemble learners, without full visibility into the underlying theory, provides exceptional performance, essentially out-pacing the known theory in this field [203].

The next sections provide an overview of ensemble techniques of bagging, boosting and gradient boosting that led to the evolution of the XGBoost algorithm, which the proposed model will be using.

Bagging

Bagging, introduced by Breiman [199] is an acronym for *Bootstrap Aggregation*. As the name suggests, it combines or aggregates predictions from multiple decision trees or Artificial Neural Networks (ANN). Figure 4.10 shows an example of a bagging ensemble, with the training set represented by D and a query sample q. The steps would be as follows [196]:

1. Training sets $T_{\mathrm{i}}(\mathrm{i} = 1, S)$ are generated for an ensemble of S classifiers, from D using bootstrap sampling.

2. While the training data is D_{i}, data that is not included in the training set and will be used for validation is notated as $D_{v\mathrm{i}}$. $D_{v\mathrm{i}}$, or data that is not used for training, may be called the Out of Bag (OOB) data.

3. S classifiers, $f_{\mathrm{i}}(, D_{\mathrm{i}})$, are trained using the D_{i} training sets, with overfitting being controlled with the validation sets, D_v.

4. The S classifiers, $f_{\mathrm{i}}(, D_{\mathrm{i}})$ generate q predictions as seen in Figure 4.10.

5. An aggregating function is used to to aggregate these S predictions, into a single prediction $f_{\mathrm{i}}(q, D_{\mathrm{i}})$. This can be represented as:

$$f_E(q, D) = F(f_1(q, D_1), f_2(q, D_2), f_3(q, D_3)..., f_S(q, D_S))$$

   Regression ensembles can be aggregated using a simple average and classification ensembles can be aggregated using a weighted average:

$$f_E(q, D) = \sum_{\mathrm{i}=1}^{S} w_{\mathrm{i}} \times f_{\mathrm{i}}(q, D\mathrm{i})$$

   where $\sum_{\mathrm{i}=1}^{S} w_{\mathrm{i}} = 1$

Unlike regression methods, bagging is optimized for the best prediction. In linear regressions, the optimization is based on least squares and on minimum the variance r^2. However, in

bagging, an overall optimization of the prediction, allows for a high level of generalization and performs well [199].

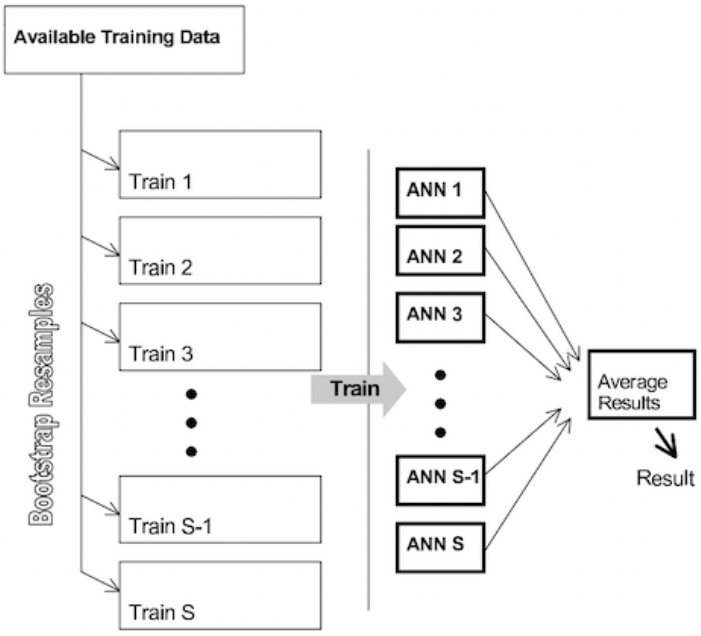

Figure 4.10. Overview of a bagging ensemble [196]

Random Forest

Random forest is an extension of the bagging technique that combines multiple trees into a single random forest, through creation of bootstrapped copies of the data and estimating a single tree for each bootstrap [204]. These are then averaged together to provide more accurate and stable predictions [205]. It differs from bagging in that in randomly selects subsets of features in each sample, decorrelating the trees. Similar to other techniques it can be used for categorical predictions, in classification tasks or continuous response, where it is a regression task [206].

Cutler, Cutler, and Stevens [206] provides outlines the algorithm for random forests as follows. Assuming that the training data $D = \{(x_1, y_1), ..., (x_N, y_N)\}$ with $x_\mathrm{i} = (x_{\mathrm{i},1}, ..., x_{\mathrm{i},p})^T$. For j = i to J :

1. A bootstrap sample D_{j} of size N is taken from D.

2. A tree is fitted tree using binary recursive partitioning, using the bootstrap sample D_{j} is used as the training data:

   (a) The observations are started in a single node.

   (b) the following steps are repeated recursively for each un-split node until the stopping criterion is met:

      i. m predictors are selected from the available p predictors.

      ii. The best binary split among all the splits on the m predictors is obtained.

      iii. That selected split is used to split the node into two descendant nodes.

      iv. To make a classification prediction at a new point x, the following is used:

$$\hat{f}(x) = \arg\max_{y} \sum_{\mathrm{j}=1}^{J} I(\hat{h}_{\mathrm{j}}(x) = y)$$

      where $\hat{h}_{\mathrm{j}}(x)$ is the prediction at x using the $\mathrm{j}th$ tree

Using only a subset of features not only helps to make to make the trees more independent and reduce variance errors, but also slightly increases the speed when compared to bagging. However, more sophisticated techniques are now available, which may show better performance depending on the specific nature of the problem. Gradient-boosted trees generally show higher prediction accuracy than random forests. A trained forest may also need significant storage memory since it retains information from several hundred individual trees.

Boosting

Boosting updates the weight of the observation based on the last classification, using a sequence of weak models that perform better than they would independently [42] [207]. Unlike bagging, it does not combine classifiers but uses them sequentially, improving on the results of the previous classifier. It *boosts* the higher performing models while minimizing the

error of the weaker models. AdaBoost stands for adaptive boosting and is only of the most commonly used boosting approaches [183]. The principle behind boosting is represented in Figure 4.11. Boosting focuses on the errors near the decision boundary, highlighted in the figure, in building the next classifier in the sequence [196]. It focuses on training samples that were not classified correctly and adjusts the sampling distribution of the training data. Assuming that the available training data is represented by $(x_1, y_1), (x_2, y_2), ..., (x_{|D|}, y_{|D|})$, with $x_\mathrm{i} \in X, y_\mathrm{i} \in Y = \{-1, +1\}$, Schapire [207] outlines the boosting algorithm as follows:

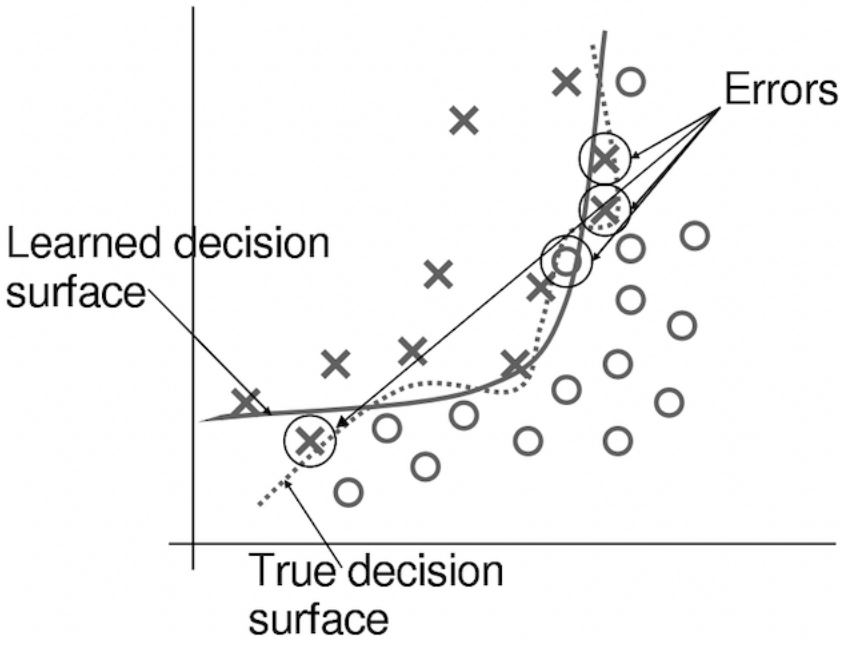

Figure 4.11. Difficulty in classifying errors near the decision boundary [196]

1. Initialize the sampling distribution $P_1(\mathrm{i}) = 1/|D|$
2. For each $t = 1, ..., T$:
   (a) Train the classifier using distribution P_t.
   (b) This classifier represents the hypothesis $h_t : X \to \{-1, +1\}$.
   (c) The estimated error of this classifier is $\epsilon_t = \sum_{\mathrm{i}:h_t(x_\mathrm{i}) \neq y_\mathrm{i}} P_t(\mathrm{i})$.
   (d) Let $\alpha_t = 1/2 \ln \frac{1-\epsilon_t}{\epsilon_t}$

(e) Update $P_{t+1}(\mathrm{i}) = \frac{P_t(\mathrm{i})}{Z_t} \times \begin{cases} \mathrm{e}^{-\alpha t}, & \text{if } h_t(x_\mathrm{i}) = y_\mathrm{i} \\ \mathrm{e}^{\alpha t}, & \text{if } h_t(x_\mathrm{i}) \neq y_\mathrm{i} \end{cases}$ where Z_t is a normalization factor that's adjusted to ensure that P_{t+1} is a distribution.

(f) These classifiers are trained while $\epsilon_t < 0.5$.

3. The classification produced this ensemble would be:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

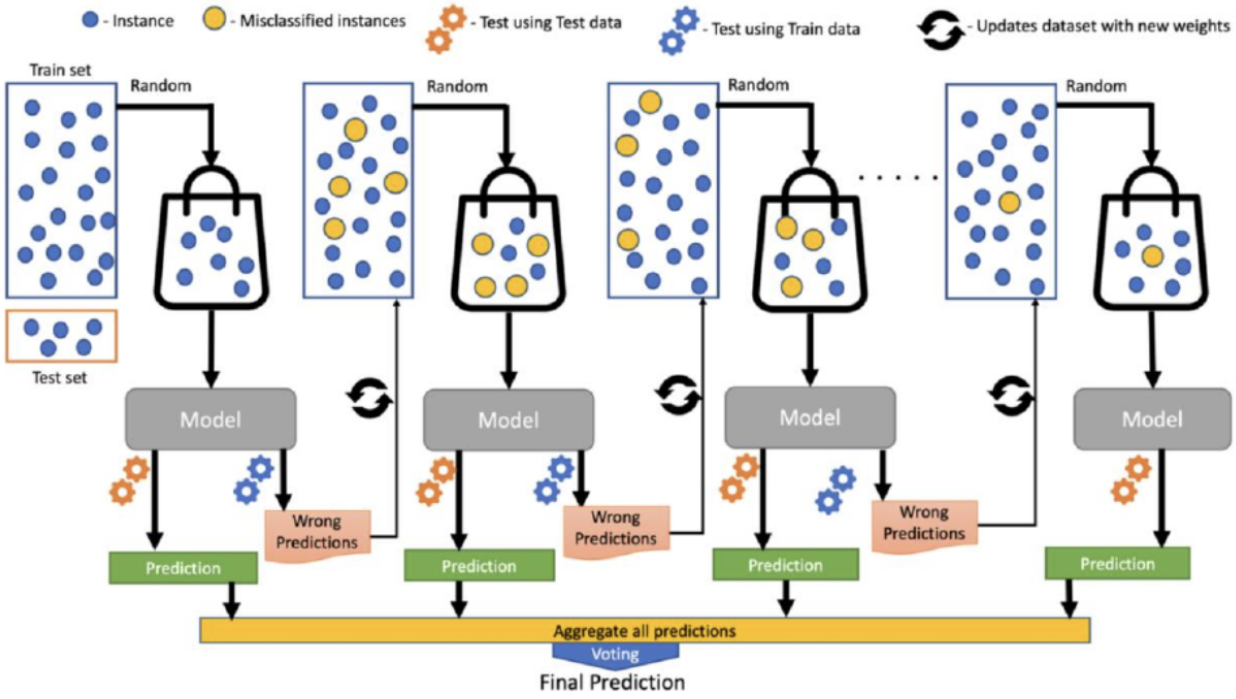

Figure 4.12. Illustration of details of a boosting algorithm [198]

Figure 4.12 shows the steps involved in the boosting algorithm described above. Given the role of classification in this ensemble, the algorithm proposed above only works for classification tasks [196]. However, other modified algorithms have been proposed to use for regression. Boosting with tree algorithms has proved to be very successful, due to their recursive nature. Their binary splitting rule can be used repeatedly throughout the tree and the dataset, allowing good use of the dimensions of the dataset for explaining predictions

and such types of inference [204]. There are many approaches for fine-tuning the setting on boosting algorithms and this can have a big impact on the model performance [197]. The next section discuss two such approaches - Gradient Boosting and XGBoost.

Gradient Boosting

As discussed earlier, boosting techniques were mostly driven by the algorithm, without a clear picture of the properties driving the performance [207]. There has been some speculation of the *boosting paradox* - the algorithms either outperformed other methods where applicable or they could be inapplicable entirely due to overfitting [208]. In order to link the algorithm performance to a statistical framework, boosting methods that relied on gradient-descent were proposed [209] [210]. The models using this approach are known as Gradient Boosting Machines, or GBMs.

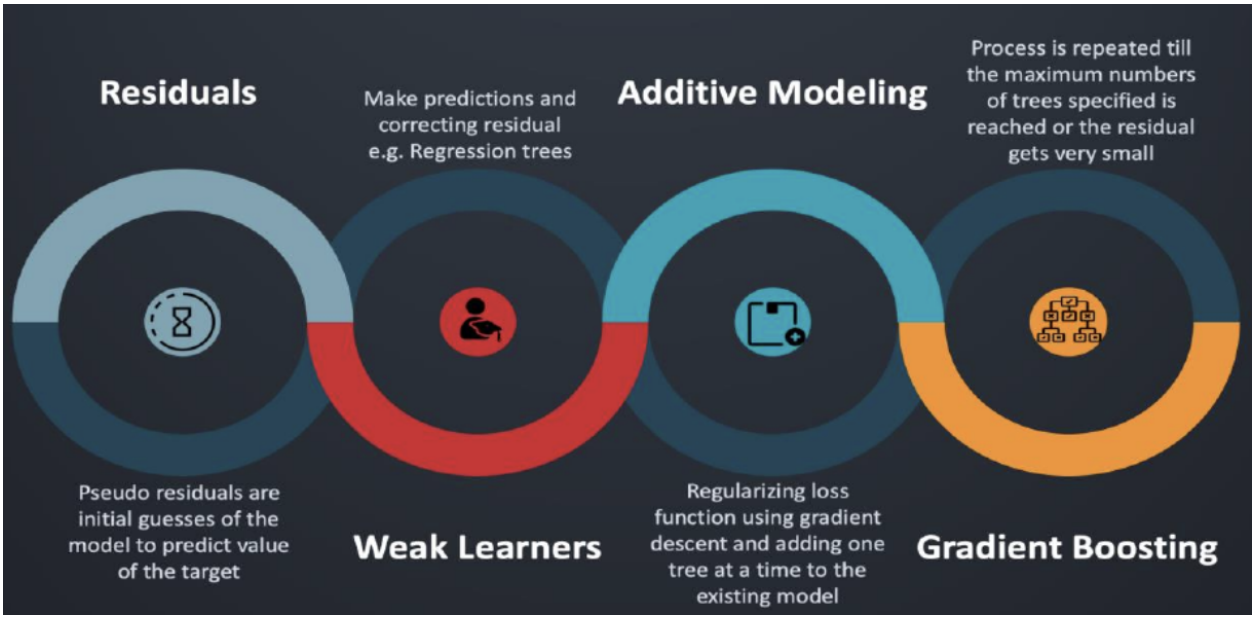

Figure 4.13. Gradient Boosting process flow Malik, Harode, and Kunwar [198]

Similar to general boosting, GBMs learning algorithm keeps updating iteratively to provide high accuracy. In each iteration, the new learners are developed such that they have maximum correlation with the negative gradient of the loss function for the ensemble [211]. The loss-function can be chosen by the researcher and this makes GBMs very customizable

depending on the nature of the task. Similar to other machine learning algorithms, choosing the appropriate loss function might require some trial and error. This process is illustrated in Figure 4.13.

Algorithm 1 Friedman's Gradient Boost algorithm

Inputs:

- input data $(x, y)_{i=1}^{N}$
- number of iterations M
- choice of the loss-function $\Psi(y, f)$
- choice of the base-learner model $h(x, \theta)$

Algorithm:

1: initialize $\widehat{f}_0$ with a constant
2: **for** $t = 1$ to M **do**
3: compute the negative gradient $g_t(x)$
4: fit a new base-learner function $h(x, \theta_t)$
5: find the best gradient descent step-size ρ_t:

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \Psi\left[y_i, \widehat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)\right]$$

6: update the function estimate:

$$\widehat{f}_t \leftarrow \widehat{f}_{t-1} + \rho_t h(x, \theta_t)$$

7: **end for**

Figure 4.14. Gradient Boosting algorithm proposed by Friedman [210]

Figure 4.14 illustrates the gradient boosting algorithm proposed by Friedman [210]. The algorithm can be customized into a more specific implementation through different options for Ψ(y, f) and $h(x, \theta_t)$.

XGBoost

XGBoost, proposed by Chen and Guestrin [17], is the fastest implementation of gradient boosting, both for classification and regression problems. It stands for *Extreme Gradient Boosting* and enhances hardware and software capabilities to increase the performance gradient boosting techniques in terms of speed and accuracy [198] through using the strengths of second order derivatives of the loss function, regularization and parallel computing.

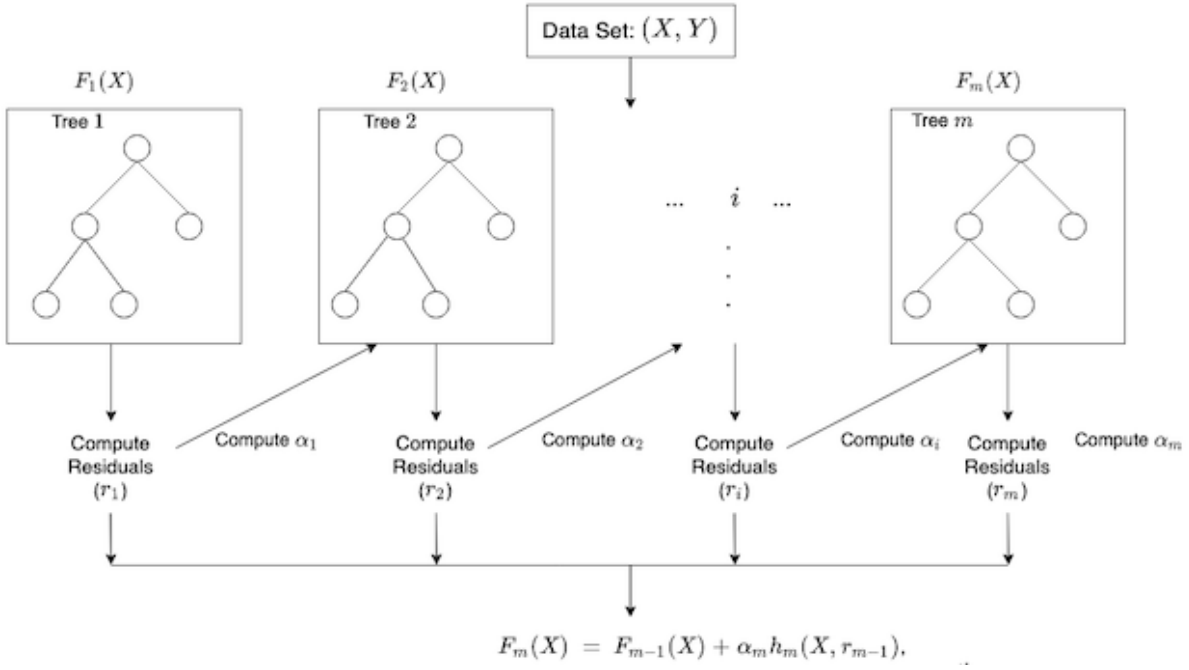

Figure 4.15. Illustration of the XGBoost Algorithm [212]

Figure 4.15 illustrates how XGBoost works, where α_{i} and r_{i} are regularization parameters and residuals computed with the i^{th} tree respectively and h_{i} is the training function used to predict the residuals r_{i} using X for the i^{th} tree. To compute the regularization parameter α_{i}, the residuals, r_{i} are used to obtain the following:

$$\arg\min_{\alpha} = \sum_{\mathrm{i}=1}^{m} L(Y_{\mathrm{i}}, F_{\mathrm{i}-1}(X_{\mathrm{i}}) + \alpha h_{\mathrm{i}}(X_{\mathrm{i}}, r_{\mathrm{i}-1}))$$

where $L(Y, F(X))$ represents the differential loss function.

While the algorithm discussed illustrates a regression task, a similar approach is used for classification tasks, where similar to other boosting approaches, the weights on the incorrectly predicted objects are increased (similar to the residuals r_i in regression algorithm discussion above) and it is passed to the next classifier. XGBoost has been a winner in several machine learning competitions through its several advantages over gradient boosted trees [17]:

- It is a more *regularized* gradient boosting, using advanced L1 and L2 regularization techniques. This improves the generalizability for the model.
- As the name suggests, XGBoost enhances performance of gradient boosting with high training speeds and the ability to parallelize over distributed clusters.
- It computes second order gradients i.e. second partial derivatives of the loss function, providing more information about the gradient direction and getting to the minimum of the loss function.
- It also handles missing values, and may potentially reduce the effort for data preprocessing.

Considering an example of choosing a candidate in an interview [190]. A decision tree classification process would involve hiring on the basis of the hiring manager's criteria such as years of experience, education, technical skills etc. A bagged tree can be envisioned as an interview panel where votes are collected and the candidate is chosen through a voting process. Instead of looking at all the *features* or qualifications of the candidate, if each member of the panel only focuses on a random subsets of qualifications and votes on that basis, that is an example of a random forest classification. Under the scenario, where instead of all interviewers being in a panel together, they interview the candidate sequentially, altering their questions based on feedback from the previous interviewers, this represents a boosted approach. Using a specific approach to minimize errors, e.g. using case interviews to weed out candidates that are less qualified, this aligns with a gradient boosting approach, where the case interviews are analogous to the gradient descent algorithm. Considering a gradient boosted approach where interviewers are using a specific approach of case interviews and giving feedback to the next interviewer, while being able to do this much faster and with better

prediction results, can be seen as a representation of the XGBoost Algorithm. Given its unparalleled performance and its limited use in digital forensics, the state-of-the-art XGBoost is being proposed to develop a learning model for user attribution.

5. PROCEDURES AND DATA COLLECTION

This chapter provides a description of the methodology used in this research. The next section explains how the methodology of this research differs from non-machine learning techniques, and the rationale for the performance measures used.

5.1 Hypothesis Testing versus Classification in Machine Learning

Two approaches can be taken in order to answer the research question. The traditional approach in the field of academic research has been through *hypothesis testing* where a null hypothesis is assumed and evidence is gathered that can allow researchers to reject this notion with *statistical significance*. An important feature of hypothesis testing is that the researchers don't have the *correct* answers for any subset of the population. Based on the available data, the main goal is to make *inferences* about the data sample and the general population [213].

On the other hand, classification is rooted in making *predictions* in the future, often based on available present data. In this approach, instead of making a hypothesis assumption and gathering evidence towards a decision, the decision making focuses on whether or not an instance can be assigned or labeled to a specific class, i.e. a true or false decision towards class membership [213]. The emphasis is not only on the evidence but also on the development of a model that can make future predictions. Depending on the nature of the problem, the explanation or cause of the behavior may not be as important as anticipating the behavior (e.g., in classifying spam emails). Figure 5.1 shows how the concepts for hypothesis testing and binary classification are related to each other. As seen, the concepts are closely related but the methods employed and goals (inference vs. prediction) are not always the same.

Given the different approaches in developing the decision rules, it is important to discuss how the performance evaluation of the two approaches differ. The decision rules or statistical tests to reject the null hypothesis are compared in terms of *power*. A significance level α is assumed, and the larger the power, the better the test [213]. The power of the statistical test may not always be observable through the data, and may require advanced statistical analysis. In such scenarios, it is not uncommon to choose the test that fits the underlying

Concept		Hypothesis Testing	Binary Classification
Binary question		Is the null hypothesis false? (unanswerable)	Does the instance have a label 1?
Binary answer	0 (no)	The null hypothesis is true (unobservable)	The instance has a label 0
	1 (yes)	The null hypothesis is false (unobservable)	The instance has a label 1
Decision rule		A statistical test that inputs data and outputs a p value, which is compared against a user-specified significance level α	A trained classifier that inputs an unlabeled instance's feature values and outputs a predicted label
Binary decision	0	Do not reject the null hypothesis	Label the instance as 0
	1	Reject the null hypothesis	Label the instance as 1

Figure 5.1. Conceptual difference between hypothesis testing and binary classification [213]

assumptions the best or to choose the test with the smallest p-value, i.e. the greater statistical significance in rejecting the null hypothesis.

For classification, the performance evaluation is more straightforward. As discussed in earlier sections, the approach involves using a subset of the data to test the model and evaluate performance in terms of metrics such as *accuracy*, *precision*, and *recall* that account for both false positives and false negatives. Another approach is through the Area Under a Receiver Operating Characteristics (AUROC) curve. In general, there is less subjectivity in these performance metrics against the traditional hypothesis testing approaches [213].

The preceding discussion does not imply that hypothesis testing and binary classification approaches are mutually exclusive. While they are different approaches, they can often

be used together depending on the nature of the problem. The present study uses binary classification performance metrics to report on the classifier's performance.

This research uses previously published data. The next section provides more details of this analysis approach, including the rationale and the considerations in the research methodology when using previously published data for analysis.

5.2 Secondary Analysis

This study used the Syracuse University and Assured Information Study - Behavioral Biometrics Multi-device and multi-Activity data from Same users (SU-AIS BB-MAS) dataset, published on IEEE DataPort.

Secondary analysis of data refers to new research that is conducted on data that has not been collected by the researcher themselves or was previously collected for different purposes [214][215]. It is an empirical method employing the same research method and principles as research studies that use primary data [216]. Glaser [217] suggested early that the independent research could rely on secondary analysis for, among other things, *lend new strength to the body of fundamental social knowledge* [218]. Since then it has been used with different motivations such as obtaining entirely new insights from previously published data [219] or to expand the scope of the initial analysis [219], [220]. It is not only commonly used by researchers, but also specifically graduate students conducting dissertation research [221]. While this data was not specifically available in the field of cyber forensics, an illustrative example can be seen in the field of social sciences where hundreds and potentially thousands of dissertations have used data from the Inter-university Consortium of Political and Social Research (ICPSR) [221]. Universities often encourage the use of secondary data and publish sources of secondary data on their dissertation support pages [222][215][223].

The use of secondary data for research has become more relevant with the newly available technologies and yet remains under-utilized [216]. Current capabilities support easier, faster, cheaper, and more secure data sharing, allowing complex analysis on large data repositories[214]. There are several advantages of choosing secondary research over primary research [215]. The quality of secondary data is likely to be much better, than that of data collected

by a graduate student[221]. This is because large-scale benchmark datasets provided online can often be infeasible in terms of cost and time for a graduate student, even with funding.

The use of secondary data also allows for better comparison between different research studies, and improves the repeatability of the study. If the underlying data used is different, it is more challenging to find errors in the analyses and interpretation of the results. This is especially true if the initial data is not publicly published [215]. It also provides the opportunity for different insight into previously conducted research, and enhances opportunities for proposing alternate approaches that weren't previously discussed [222]. It allows researchers to potentially discover relationships that were not predictable at the time on the initial data collection [224]. Guarino [37] believes that with the adoption of more machine learning approaches in digital forensics, there will be an increased reliance on previous validation performed by researchers, greatly increasing the importance of published data that can be independently validated as opposed to more black-box approaches.

Secondary data analyses should also be supported because the limited resources available to graduate students can limit the nature of the data they can collect. This often leads to cross-sectional data with small, nonrandom samples [221]. This also has effects on the ability to make causal inferences that can be generalized to a known population. The use of public datasets also provides a larger and more diverse sample than the researchers may be able to access through their own data collection [222]. It also increases the accessibility and provides more equitable opportunities for researchers in under-served communities, such as developing nations [222]. Data published on reputed public data repositories are usually vetted for the required approvals and alleviate some risk and approvals required, expediting the research process [222]. It also provides researchers with access to data where concerns like privacy may make the data challenging to collect or from populations that are not easily accessible to the research community [218]. It allows additional research on such potentially vulnerable populations without the repeated intrusion of data collection, given that the data is reused with the same ethical boundaries that were initially established [214].

Given the advantages discussed, there is often still hesitancy among researchers towards secondary analyses [221]. The data available may not adequately address the specific measures required to answer the research question. The data available through public datasets

may also be outdated or unwieldy to download, with different databases using different formats. There may be inadequate documentation without all the specific details of the sampling design and data collection. The publisher also may not be available to answer any queries on the data, which makes the use of such data often more intimidating [221]. Data published on reputed databases, such as IEEE DataStore, is professionally vetted and is usually expected to adhere to quality standards that may alleviate some of the concerns around formats and lack of documentation [223].

Dissertation committees may also disapprove of the use of secondary analyses [221]. One of the reasons can be that given their prior experiences dealing with the challenges discussed above, related to the availability, and complexity of using published data, they may discourage graduate students from selecting this approach. Another potential reason is that the dissertation process, in addition to focusing on a new contribution in the relevant field, also focuses on research training. Committees may feel that using secondary analyses may skip the training for the data collection phase. However, there is a general expectation that students conducting secondary analyses have had prior experience with and have learned the necessary skills related to the collection of data [221].

There is an expectation of due diligence on the researcher to ensure that the chosen dataset meets the requirements of their research question. Studies relying on secondary research should outline data collection processes used by the original researchers and provide full transparency on the processing of the data (e.g. approaches of dealing with missing data or outliers) [220]. Assuming that the data available fulfills the research requirements, using secondary data for dissertation research is considered a reasonable and acceptable approach [221]. The use of secondary data in empirical research to investigate research questions still generates new knowledge, as required for a dissertation [216]. Supporting the use of research using secondary analyses will also increase the motivation for data sharing in the research community. The availability of research data for public scrutiny and reanalysis as part of scientific enquiry is a cornerstone for good science [225].

As discussed in this section, secondary analyses contributes to the creation of scientific knowledge, similar to the primary analyses research method, while only differing in its reliance on existing data [216]. The SU-AIS BB-MAS is a benchmark dataset published in

2020, and is currently the most extensive and updated data available for user computer behavior. The researchers collect data from the same users over multiple tasks, providing a highly suitable secondary source of data towards answering the proposed research question. The researchers have included detailed information on the data collection process, provided raw and processed data and also include demographic data for the participants, making it feasible for use without a loss in quality of available information.

5.3 Details of Data Collection

This section provides details about the experiment setup and the process used for collection of data by the primary researchers at Syracuse University. In addition to the approvals obtained by the primary researchers, approval from the Purdue Institutional Review Board (IRB) was also obtained for the use of the published dataset.

5.3.1 Experiment Setup

As mentioned, the published SU-AIS BB-MAS dataset was used for this research. As part of the data collection, participants were recruited through email sent to all of the student, faculty, and staff body at Syracuse University. Data was collected between April 2017 and June 2017 after the researchers obtained the required Institutional Review Board approvals.

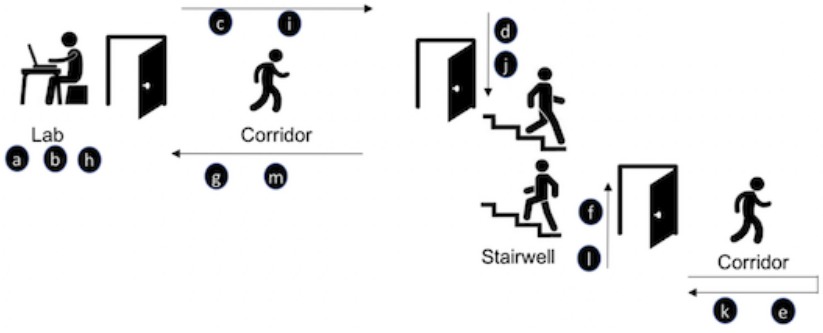

Figure 5.2. Data collection procedure used for the SU-AIS BB-MAS dataset. [226]

Upon arrival, each participant was asked to provide demographic and technology usage information. Each participant was assigned a unique identifier and performed actions such

as typing, swiping etc. 5.2 shows the different actions from a to m that was performed by each participant. Activities for each participant were recorded on four devices - a Desktop, a Tablet, and two phones (with one phone in the pocket and one in the hand). The details of each device used for data collection are as follows [226, p. 1]:

- Desktops: The researchers set up two identical desktop stations. Each with a standard QWERTY keyboard (Dell kb212-b), optical mouse (Dell ms111-p) and a Dell 21-inch monitor.
- Tablets: HTC-Nexus-9 tablets were used, a screen size of 8.9 inches, screen resolution of 1536 x 2048 pixels, device dimensions of 9 x 6 x 0.3 inches (Length X Width X Height) and weighed about 435 grams.
- Phones: Samsung-S6 and HTC-One phones were used in the data collection. The Samsung Galaxy S6 had a screen size of 5.1 inches and screen resolution of 1440 x 2560 pixels with body dimensions of 143.4 x 70.5 x 6.8 mm and weighing 138 grams, whereas the HTC-One had a screen size of 5.0 inches and screen resolution of 1080 x 1920 pixels with body dimensions of 146.4 x 70.6 x 9.4 mm and weighing 160 grams.

The data collection included 117 participants, in a session lasting between 2 to 2.5 hours each. The details of the tasks *a* to *m* as performed in a sequence and as outlined in Belman, Wang, Iyengar, *et al.* [27] has been explained below:

- As illustrated by task *a*, each participant was instructed to sit and use the desktop to type out two excerpts of fixed text, ten times each. Each excerpt had two sentences and an average of 112 characters as seen in A.1. As part of the same task, next the participant was given a shopping list of six items (see A.3) and was asked to browse and search for the best prices online and note their opinions. Next, each participant was provided a list of 12 questions of varying cognitive loads (see A.2) and were asked to types out their responses in order of their own preference, suggesting an interval of about fifteen minutes to complete the task. A keystroke and mouse movement recorder installed on the desktop was logging all actions of the participant during this task.

- For task b, the participant was provided with a tablet and asked to type the two sentences again (see A.1) and then respond to ten questions (see A.6). They were required to respond with a minimum of 50 characters. The task required participants to swipe vertically and horizontally between questions. During this task, the keystroke, touch, accelerometer, and gyroscope loggers were used on the tablet to log all typing, swiping, touch, and movement events.

- Next, the researchers provided the participant with a phone (Phone1) and instructed them to put the phone (Phone1) in their pocket, hold the tablet in their hand, and walk a predefined path. 5.2 shows the path walked by each participant. The tablet displayed buttons that the participants were required to press before and after passing through a doorway and also after they took the staircase. The tasks c, e, and g consisted of walking, and tasks d and f consisted of climbing downstairs and upstairs respectively. As each task was performed, c to g, the accelerometer and gyroscope values were logged for tablet and the phone (Phone1). The button presses (doorway and staircase) were also logged.

- Once task g was completed, the participant returned the tablet and received another phone (Phone2). Phone2 executed the same application as the tablet in task b for task h, where the participant had to type the two sentences (see A.1) and then answer ten other questions (see A.5). Similar to earlier, the questions were of varying cognitive load, responses had to be a minimum of 50 characters, and the participant employed horizontal and vertical swipes to navigate between questions. Phone2 logged all keystroke, touch, accelerometer and gyroscope values for typing, swiping, touch, and movement events.

- The participant then repeated tasks c to g, with the exception that Phone2 was held in the hand and Phone1 was in the pocket. These are labeled as tasks i to m in 5.2. All accelerometer and gyroscope values were logged on both Phone1 and Phone2. The pressing of buttons (doorway and staircase) by the participant was also logged on Phone2.

5.3 provides a summarized view of activity performed by each participant and the data that was collected, in each session. For the scope of this study, only features collected in task *a* will be used for classifying the users. Further research in this area may focus on other data collected through alternate digital devices.

Task	Device	Activity	Data	Duration (Approx.)
a	Desktop	Typing, Browsing	Keystroke and Mouse	50 min
b	Tablet	Typing	Keystroke, Swipe Accelerometer, Gyroscope	25 min
c	Tablet (in hand) Phone1 (in pocket)	Walking	Accelerometer, Gyroscope	5 min
d		Climbing down stairs		
e		Walking		
f		Climbing upstairs		
g		Walking		
h	Phone2 (in hand)	Typing	Keystroke, Swipe Accelerometer, Gyroscope	25 min
i	Phone2 (in hand) Phone1 (in pocket)	Walking	Accelerometer, Gyroscope	5 min
j		Climbing down stairs		
k		Walking		
l		Climbing upstairs		
m		Walking		

Figure 5.3. Summarized view of the data collection tasks performed by the participants [226]

5.3.2 Overview of Raw Dataset

The benchmark dataset includes 3.5 million keystroke events; 57.1 million data-points for accelerometer and gyroscope each; and 1.7 million data-points for swipes [226].5.4 shows the details of the dataset.

As seen by the data collected and the nature of the activities, the dataset aims to be reflective of the usual activities that a user might perform on such devices. While it is conducted in a controlled environment to minimize the impact of other variables, the nature of the activity, such as typing free text, allows for a more realistic model of behavior, as opposed to tasks that are completely fixed in nature.

Item		Description
Number of users		117
Activities		Typing, Gait and Swipes from ***the SAME USERS on all devices***
Devices (per participant)		Desktop, 2 Phones (hand and pocket), Tablet
Avg. and StD. (per user) of number of	keystrokes on Desktop	11760, 2132
	keystrokes on Phone	9415, 1463
	keystrokes on Tablet	8966, 1584
	keystrokes from a user on all devices	30153, 3880
Avg. and StD. (per user) of number of datapoints from Accelerometer and Gyroscope	on Pocket-Phone	30280, 21775
	on Hand-Phone	222949, 38500
	on Tablet	240473,39984
	from a user on all devices	492241, 69308
Avg. and StD. (per user) of number of datapoints from swipes	on Phone	8890, 4558
	on Tablet	6105,2759
	from a user on all devices	14995, 5634

Figure 5.4. Key details of the SU-AIS BB-MAS dataset [226]

5.3.3 Training and Testing

As discussed earlier, the main motivation of the proposed method is for utilization in forensic investigations. The science behind any such methods is absolutely critical, with the expectation that any methods employed should be reliable and repeatable [11].

Reliability in forensic science is measured through the use of error rates for specifically chosen thresholds. The error rates for any new suggested biometric must be known and accepted before consideration in forensics [227].

Studies that focus on user attribution usually report error rates such as the Equal Error Rate (EER), Receiver Operating Characteristic (ROC) curve, False Acceptance Rate (FAR), or the False Rejection Rate (FRR) towards reporting on the reliability ([11]. In addition to this, performance will also be reported through F1 scores, as explained earlier.

As discussed in the previous section, the process for obtaining these error rates usually involves choosing specific or a combination of biometric features. The training process uses preprocessed user data and tries to establish a pattern based on the relationship between the

features. This training model is then tested to generate error rates that can be indicative of the reliability of the proposed biometric [11]. Based on initial evaluation, an 80/20 split for training and testing respectively will be used.

5.4 Summary

This chapter provided the framework and methodology that was used for the research study. Future sections describe the findings and conclusions obtained through the research study.

6. PRESENTATION OF THE DATA

This chapter presents details about the data collected, the data processing steps, and the analyses performed on the processed dataset.

6.1 Demographic Data Description

Figure 6.1 describes the demographic data of the participants in the study. Given that data was collected in a university setting, most participants are between the ages of 19 and 30. This does not impact the study, and in fact bolsters support for the result, given that individuals with similar ages are more likely to show similar patterns or preferences in their computer usage, making it a bigger challenge to discriminate between them on these usage patterns.

Category		Size	Category		Size
Age in years	19 - 22	22	Daily usage of desktop in hours	0 - 1	17
	23 - 26	61		2 - 4	58
	27 - 30	28		5 - 7	28
	>30	06		8 - 12	12
Sex	Female	45		>12	2
	Male	72	Daily usage of phone in hours	0-1	3
Height in inches	≤60	6		2-4	51
	60-65	40		5-7	43
	65-70	43		8-12	16
	>70	28		>12	4
Spoken Languages	1	13	Daily usage of tablet	0 - 1	93
	2	64		2-4	20
	3	32		5-7	4
	>3	8	Typing style	Touch	31
				Visual	86

Figure 6.1. Description of Demographic data [27]

Another interesting observation from the demographic data is that it includes decent variability in the daily usage of the desktop (in terms of number of hours) that users self-reported. This is relevant because prior research [228][16] has shown that user behavior tends to be more consistent and discriminatory when users show a high daily usage (of 8+ hours). Given that majority of the participants in this study report less than four hours of daily usage, the results obtained are even more impressive.

6.2 Processed Data

This section discusses the additional processing on the collected data, prior to analysis. It first outlines the preprocessing performed on the raw dataset. Once a preprocessed dataset of the raw features is obtained, the first approach uses feature selection techniques to select a subset of raw features and performs analysis using this smaller set of raw features. In addition that that, the raw features were transformed into a more complex *engineered* feature set, and analysis was also performed on these newly constructed features. Figure 6.2 illustrates the different steps involved in the processing of data before analysis can be done using the chosen machine learning algorithm.

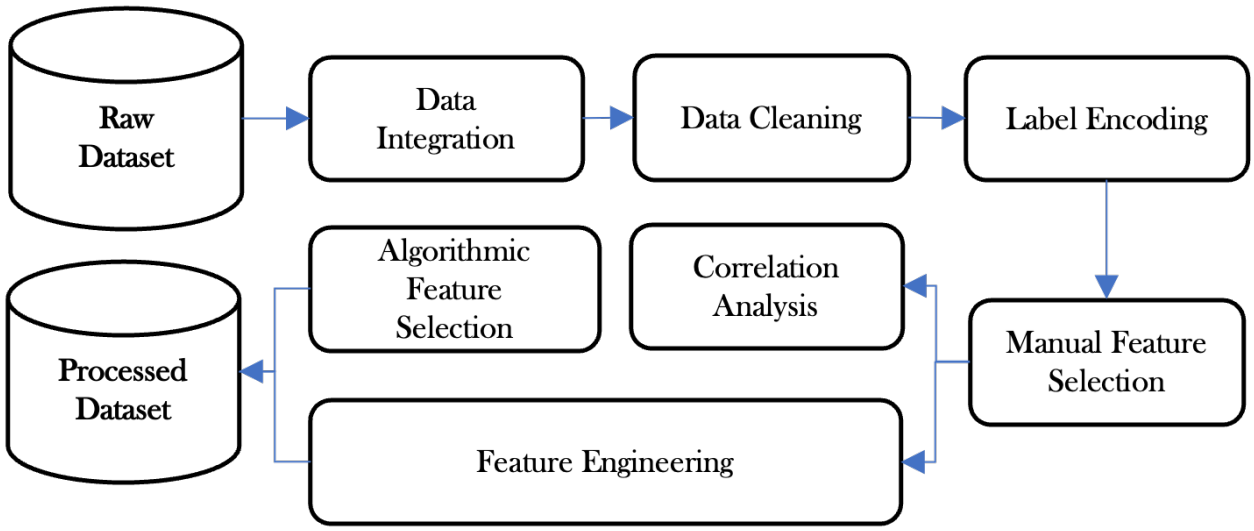

Figure 6.2. Steps involved in data processing

6.2.1 Preprocessed Data

As discussed in Chapter 5, data preprocessing is usually required for any analysis that uses machine learning. This is especially relevant in a secondary analysis that uses previously published data. The following preprocessing actions were performed on the dataset:

- *Data Integration*: The published dataset for the keyboard and mouse data was split into many files. A full outer join was used to bring combine them into a dataframe in order to allow processing. Once integrated, the initial raw data consisted of the following 22 variables:

    1. *EID*: event ID repeated four times for events related to the keyboard, mouse movement, mouse wheel, and mouse button (Integer)
    2. *key*: the key triggering the key-event (String)
    3. *direction*: the type of key-event (Integer, with 0 for press and 1 for release)
    4. *time*: the timestamp of the key-event and mouse-event (String in date-time format with millisecond resolution)
    5. rX_x and rY_x: the x and y coordinates relative to the active window for the events related to the mouse button (Integer)
    6. rX_y and rY_y: the x and y coordinates relative to the active window for the events related to the mouse movement (Integer)
    7. rX and rY: the x and y coordinates relative to the active window for the events related to the mouse wheel (Integer)
    8. pX_x and pY_x: the x and y coordinate on screen for the events related to the mouse button (Integer)
    9. pX_y and pY_y: the x and y coordinate on screen for the events related to the mouse movement (Integer)
    10. pX and pY: the x and y coordinate on screen for the events related to the mouse wheel (Integer)

11. *LR*: mouse button (Integer, 0 for left or 1 for right)
12. *state*: the type of button event (Integer, 0 for press and 1 for release)
13. *delta*: the direction of scroll (Integer, negative for scroll-down and positive for scroll-up)

- *Data Cleaning*: In the published dataset, null values were represented by the string value *NaN*. These values were replaced with *0* in order to convert the null values to numerical data. Given the small number of null values, interpolation was not required nor used to fill the missing data. As part of the data cleaning, the fields representing the *EIDs* for each of the event types (keyboard, mouse movement, mouse wheel, and mouse button) were dropped since they are labeling fields that are not meant for analysis. Similarly, the *time* field is discarded because the data does not represent a time-series and the field does not add any value for the current analysis. The updated data consisted of seven rows with duplicate values that were removed.

- *Label Encoding*: Labeling consisted of assigning a label to each instance, corresponding to each of the 117 users, since a supervised algorithm is being used. In addition to that, the string format of the *key* field is encoded to an integer format per the classifier format requirement. This is done by simply assigning a unique number to each letter in this field.

- *Preliminary Manual Feature Selection*: Manual feature selection, as the name suggests, involves manually choosing features that contribute the most towards attributing user activity to a specific user. There are multiple motivations behind focusing only on the highly contributing features and eliminating the rest. A big motivation is to prevent overfitting of the data as explained in Chapter 4. Overfitting the data to the algorithm can degrade its performance for new data. By only using the features that contribute to class membership, the chances of overfitting are reduced and it improves the performance of the model [229]. Another motivation is to simplify the model and reduce the training times. This would reduce the performance costs, as well as the data storage costs [230]. In addition to the discussed advantages, reducing the

number of features increases the interpretability of the data, and can lead to greater understanding of the observed behavior [231].

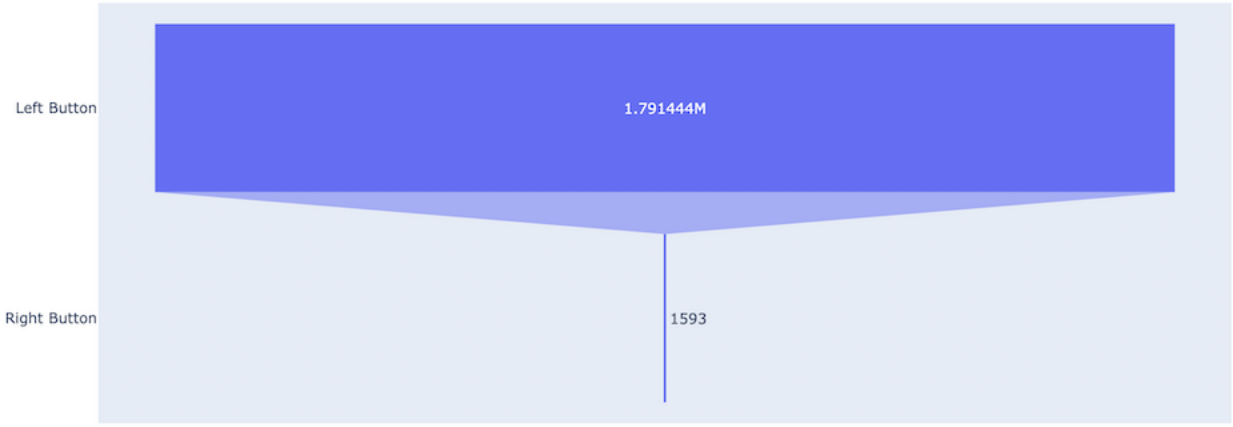

Figure 6.3. Visualization of counts of mouse button clicks

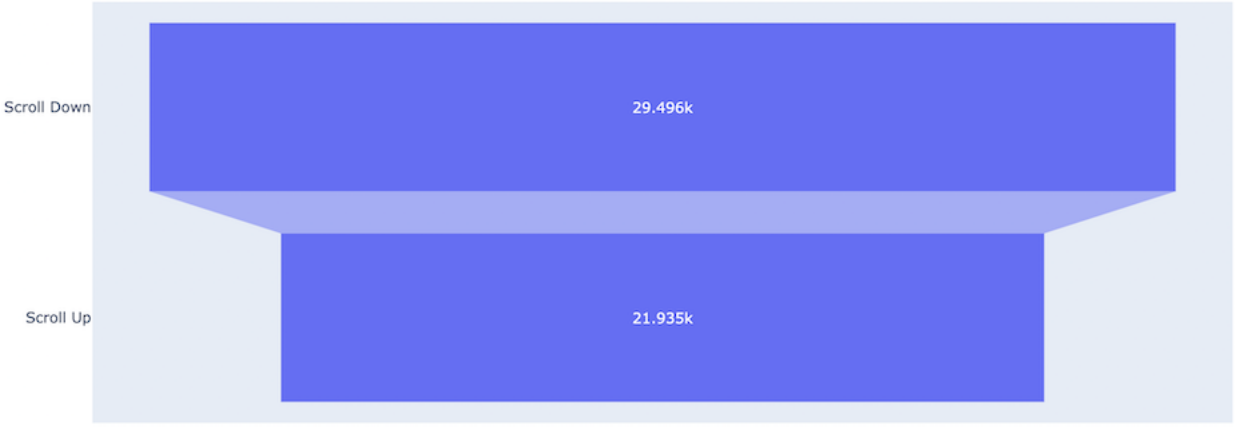

Figure 6.4. Visualization of counts of scroll direction

As a preliminary step based on initial review of the data, the *state* field was removed since the field only consisted of two values indicating a button press or button release and did not offer much information that could contribute to class membership The lack of variation in the data can be seen in Figure 6.3. While the scroll direction also contains only two possible values, the values are more equally distributed, as seen in Figure 6.4, so the field is not discarded as part of the preliminary analysis.

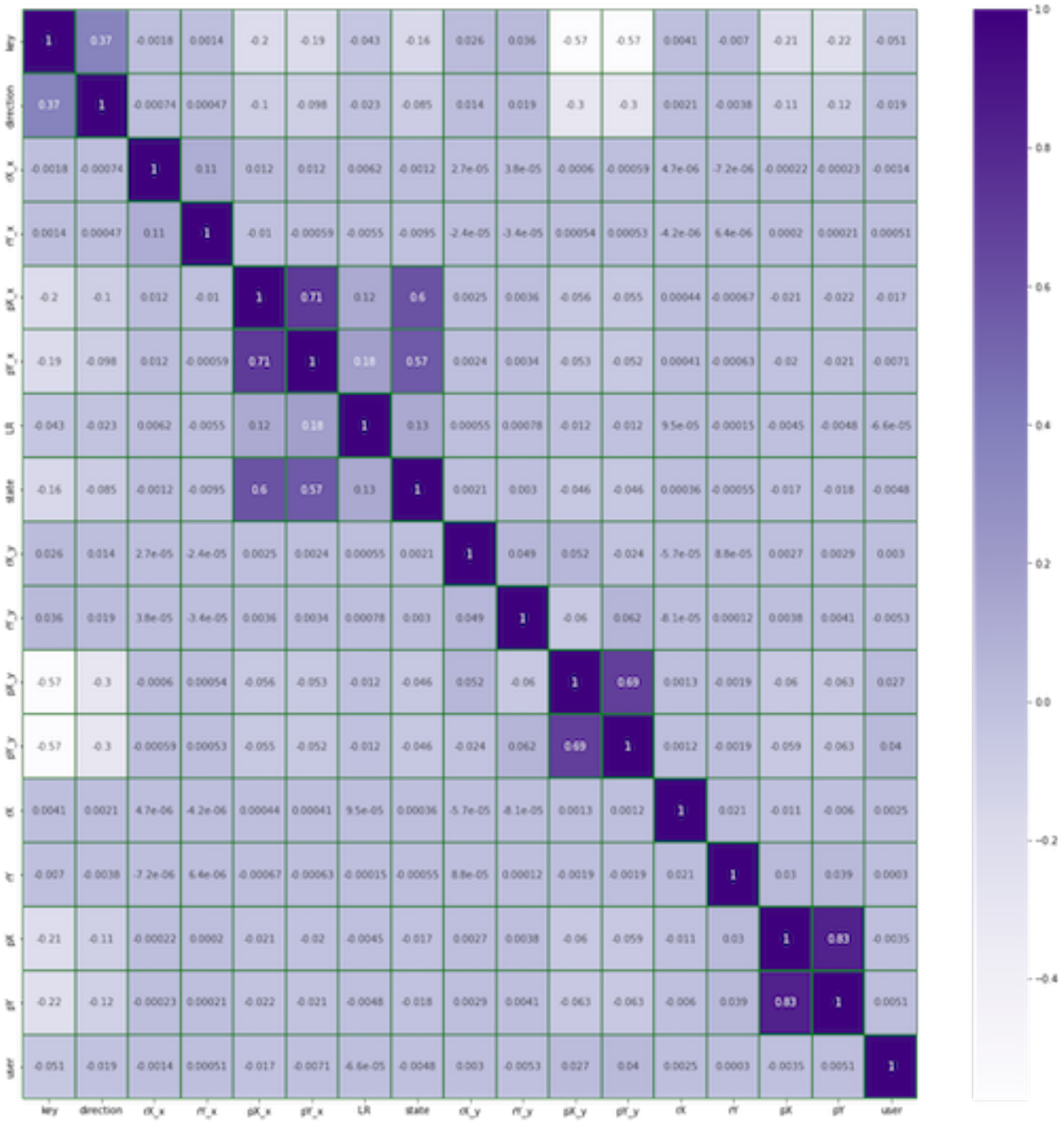

Figure 6.5. Correlation matrix for the 16 features after preliminary feature selection

After the preliminary data cleaning and selection, the updated dataset consisted of 16 variables or features. As seen in Figure 6.5, there are some variables that show a strong positive or negative correlation with each other. The high correlation between features indicates a linear dependence, resulting in a similar effect on the dependent

variable. If two features are strongly correlated, one of them may be dropped without impacting the performance of the classification model. Therefore, this indicates that the dataset might be a good candidate for further feature reduction techniques. Section 6.2.2 provides a more in-depth discussion about the results of the feature selection and the evaluation of the results obtained.

6.2.2 Feature Selection

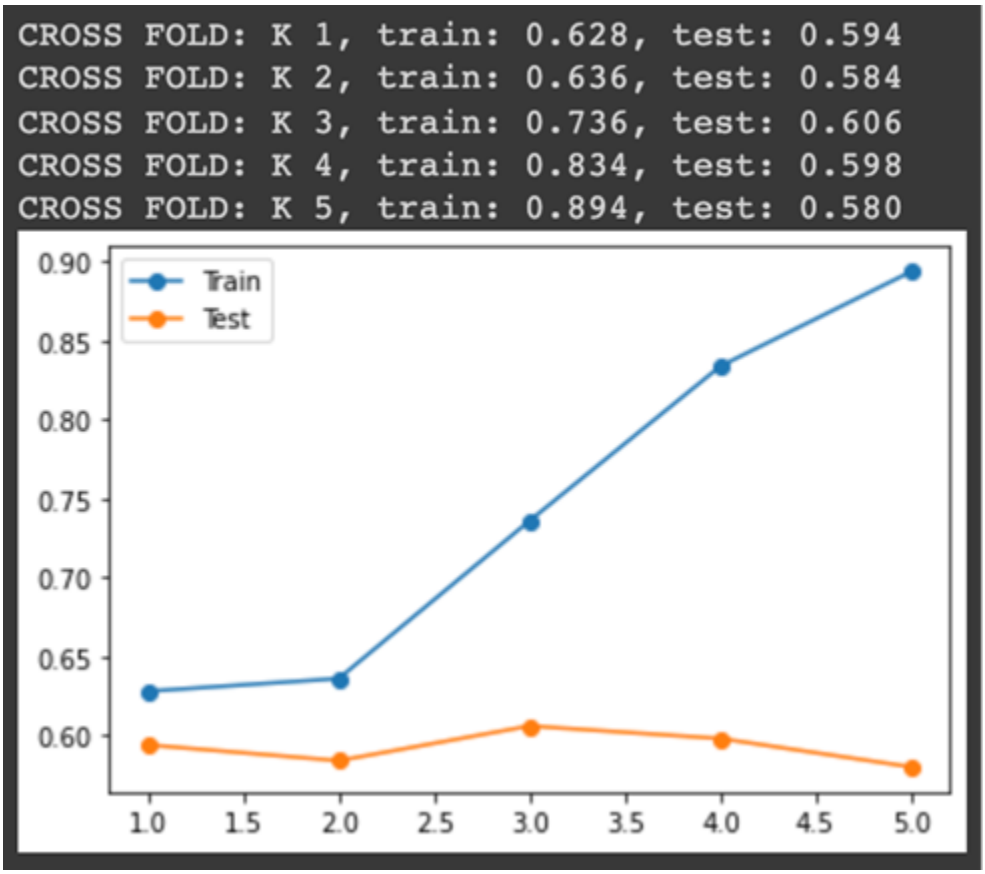

Figure 6.6. Training and testing score for 5-fold cross validation without feature selection

When a dataset is used in order to develop a model, it heavily depends on the features that are available as input. Including all the available variables as features introduces the *curse of dimensionality*, where the accuracy of the model increases till a certain threshold of increasing number of available features and then it starts decreasing [232].

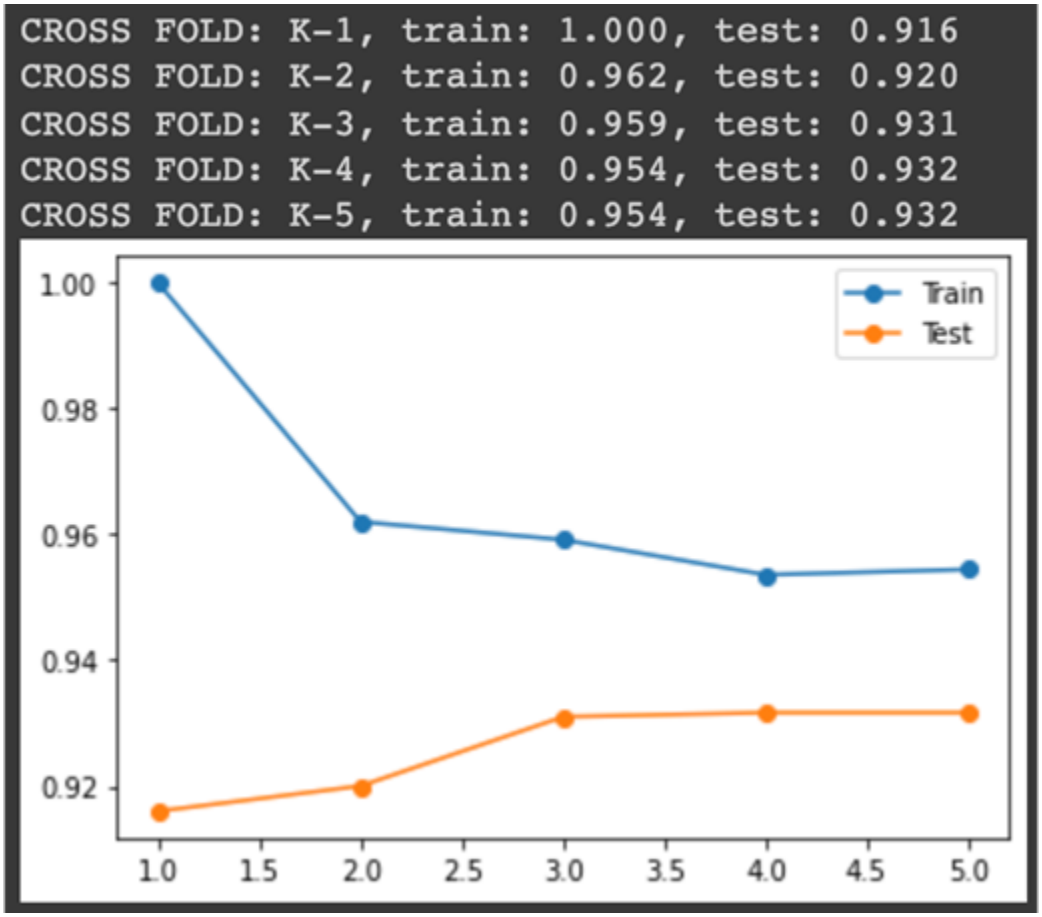

Figure 6.7. Training and testing score for 5-fold cross validation with feature selection

Statistical tests can be used towards feature selection. In general, a statistical test helps to determine whether there is significant difference between two populations. A comparison of the values of one variable with another, allows the determination of the contribution of the feature or variable towards class membership. The next subsections describes the additional tests used towards feature selection of this algorithm.

Feature Selection using Analysis of Variance

The statistical test, Analysis of variance (ANOVA), was used towards feature selection among the 16 remaining features. ANOVA can be used when one of the variables is categorical (the user class label) and the other variable is numerical (features that are selected). It is applicable when there are more than three independent groups of variables. It can assist

with determining whether the user class has an impact on the numerical features that are being assessed. It checks whether the variance between the different users is the same for each feature. An equal variance implies that the feature does not contribute to the class membership. The features that show the maximum variance were selected.

The ANOVA test statistic is calculated through:

$$F = \frac{\sum n_{\mathrm{j}}(\bar{x}_{\mathrm{j}} - \bar{x})^2/(k-1)}{\sum\sum(\bar{x} - \bar{x}_{\mathrm{j}})^2/(N-k)}$$

where n_{j} = sample size in the jth group, $\bar{x}_{\mathrm{j}}$ is the sample mean in the jth group, $\bar{x}$ is the overall mean, k is the number of independent groups, and N is the total number of observations in the analysis [233].

Table 6.1. F-scores for the full set of raw features

S.No.	Features	F-score
1	key	925.749
2	direction	112.214
3	rX_x	2.763
4	rY_x	1.885
5	pX_x	49.694
6	pY_x	39.946
7	LR	19.487
8	state	17.691
9	rX_y	20.177
10	rY_y	25.585
11	pX_y	559.421
12	pY_y	522.491
13	rX	7.41
14	rY	9.743
15	pX	265.603
16	pY	245.073

Table 6.1 shows the F-scores for the 16 features. It can be seen that the features *key, pX_y, pY_y, pX,* and *pY* show the most discrimination between the different users. This is not surprising since it can be expected that users show preferences in the keys that they use for performing the same tasks (e.g. using *Shift* instead of *Caps Lock*) and their mouse

coordinates are dependant on the placement of the application windows, even as they perform the same tasks. Based on this analysis, these five features are retained towards the analysis and the other features are ignored. This allows the model to shorten the training time and avoid overfitting, while performing at a high level of accuracy.

Feature Selection using Crow Search Algorithm

Machine learning approaches encourage the exploration of different techniques, either to discover more information or to validate known findings. Given that the ANOVA test shows that majority of the variable are not very discriminating in nature, feature selection was also repeated using the Crow Search Algorithm to validate the findings.

The Crow Search Algorithm, proposed in 2016 by Askarzadeh [234], has drawn the attention on many researchers since. It is inspired by the behavior of crows in nature, where crows hide their food and follow other crows in order to discover their stored food. Askarzadeh [234] uses this concept to propose an optimization algorithm as seen in Figure 6.8. The simplicity, easy implementation, and efficiency has made it a popular choice for feature selection in machine learning circles.

Analyzing the initial feature set of 16 features using the Crow Search algorithm provides the same output as the ANOVA test, with the 5 features indicated as those contributing the most to class membership i.e. identifying the user.

Figure 6.9 shows a summarized view of the features before and after the data preprocessing. A common approach to validate the feature selection is to analyze the performance with and without fine-tuning the features [229]. Figure 6.6 shows the training and testing scores for 5-fold cross validation done on all the full raw feature set. As the training scores increase, this has an adverse impact on the testing. This indicates that the model has been overfitted to the training data and may not perform well on other datasets. In contrast to this, Figure 6.7 illustrates the training and testing scores from 5-fold validation on the new feature set with the five features. This illustrates that the training scores are not reducing with each fold, i.e. the model is not overfitted and is expected to perform well on new datasets.

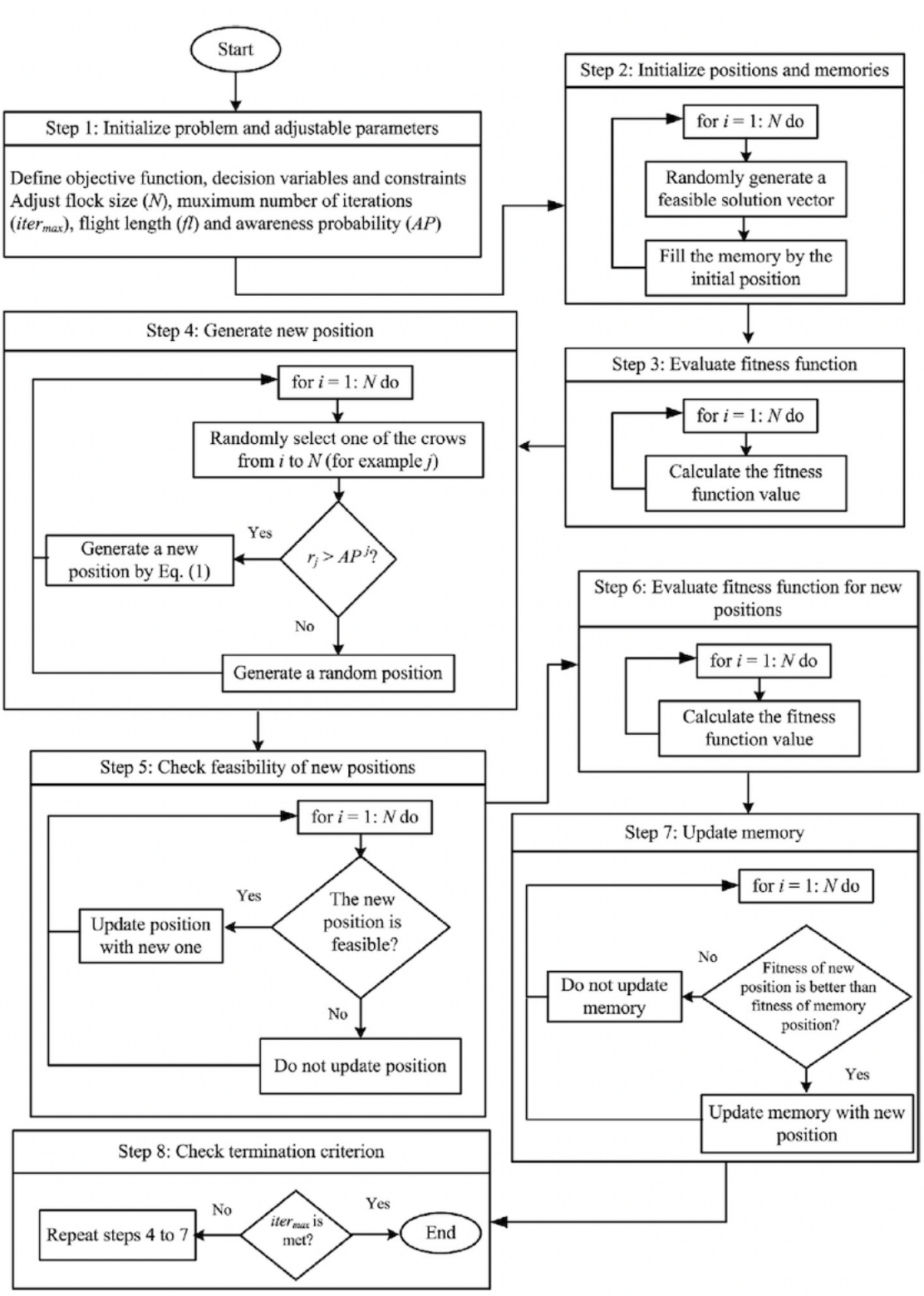

Figure 6.8. Flowchart for optimization using the Crow Search Algorithm. [234]

	Feature Size	Sample Size	Features			
			Desktop Keyboard	Mouse Button	Mouse Move	Mouse Wheel
Initial Dataset	22	1793191	EID, key, direction, time	rX_x, rY_x, pX_x, pY_x, LR, state, time	rX_y, rY_y, pX_y, pY_y, time	rX, rY, pX, pY, delta, time
Processed Dataset	5	1793184	key	***N/A***	pX_y, pY_y	pX, pY

Figure 6.9. Dataset before and after preprocessing

6.2.3 Feature Engineering

As discussed, often raw features may not contribute towards class membership but may be used to construct more meaningful features that improve the performance of the model. These features will be referred to as *engineered* features, to make the distinction from using the raw data as the feature set.

Engineered Keystroke Features

The published dataset includes engineered features for keystroke data (but not mouse data) [226]. As seen in Table 6.2, twelve common *unigraphs* or single keys are selected and the the key-hold time is calculated for each and used as a feature. Similarly, 18 common digraphs are selected and the temporal features showed are selected.

Table 6.2. Engineered Keystroke Feature set

Unigraphs	*"BACKSPACE", "SPACE", "a", "e", "h", "i", "l", "n", "r", "S" and "t"*	$Keyhold_{Ki}$ *:* K_i*Release - *K_i*Press*
Digraphs	*('BACKSPACE', 'BACKSPACE'), ('SPACE', 'a'), ('SPACE', 'i'), ('SPACE', 's'), ('SPACE', 't'), ('e', 'SPACE'), ('e', 'n'), ('e', 'r'), ('e', 's'), ('n', 'SPACE'), ('o', 'SPACE'), ('o', 'n'), ('r', 'e'), ('s', 'SPACE'), ('s', 'e'), ('t', 'SPACE'), ('t', 'e') and ('t', 'h')*	$Flight1_{Ki\ Ki+1}$ *:* K_{i+1}*Press - *K_i*Release* $Flight2_{Ki\ Ki+1}$ *:* K_{i+1}*Release - *K_i*Release* $Flight3_{Ki\ Ki+1}$ *:* K_{i+1}*Press - *K_i*Press* $Flight4_{Ki\ Ki+1}$ *:* K_{i+1}*Release - *K_i*Press*

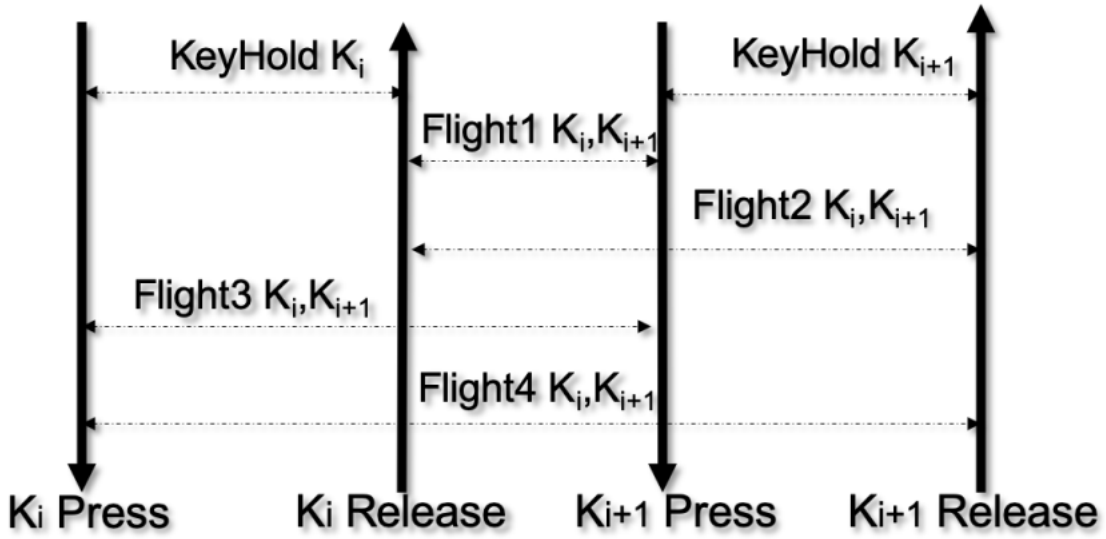

Figure 6.10. Explanation of different keystroke features [27]

Figure 6.10 illustrates the relationship between the different keystroke features that were used for analysis.

Engineered Mouse Features

Similar to the keystroke features, the raw mouse data was used to extract more sophisticated features. The following three features were extracted:

- *Distance* - This refers to the distance between consecutive mouse events
- *Time between Clicks* - As the name suggests, this feature was constructed by calculating the time between two mouse consecutive clicks.
- *Velocity* - This was calculated as the velocity of the mouse movement between consecutive mouse events.

Section 6.6 provides the details of the performance with the use of only the keystroke features, only the mouse features, and a combination of the keystroke and mouse features, both using the raw feature set and the feature set consisting of the engineered feature data.

6.2.4 Oversampling Data using the SMOTE technique

The number of instances are not very balanced between all the different users in the raw dataset. This is called *imbalanced class data*. This essentially leads to the scenario that model is not getting equally trained for the user with the few instances. This leads to degraded performance since it is important to see how the algorithm will perform for such a minority class. One approach can be to duplicate the instances in the minority class, but this is not helpful since it does not add additional training information for the model.

Chawla, Bowyer, Hall, *et al.* [235] proposed a technique called the Synthetic Minority Over-sampling Technique (or SMOTE) which helps to balance such imbalanced class data. Their approach creates synthetic samples in the minority class by:

- Choosing a random instance
- Selecting one of k nearest neighbors to that instance
- Drawing a line between the instance and this chosen neighbor
- Choosing features on this line segment

This technique solves the problem of the class imbalance while generating meaningful synthetic samples that are close to the actual feature space. The performance evaluation considers the performance both with and without the application of the SMOTE technique to the raw dataset. Given that the engineered dataset is fairly balanced, the SMOTE technique is not applied to it. If the technique is applied to balanced datasets, it increases the possibility of overfitting the data to the training set.

6.3 Final Dataset

This section presents the datasets that were used for analysis after preprocessing and processing was complete. As described in the previous section, the data processing was either done as feature selection or feature engineering.

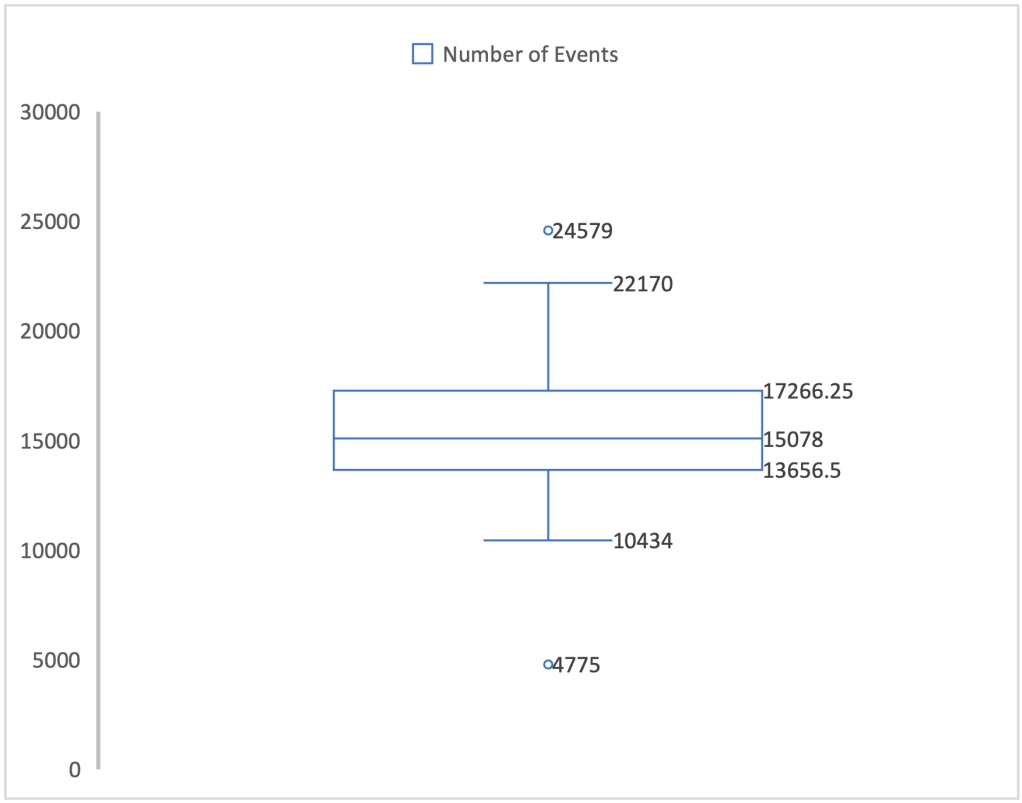

Figure 6.11. Number of events for each of the 117 users in the dataset

6.3.1 Dataset using Raw Features

As seen in Figure 6.9, the final raw dataset consisted of 1,793,184 events for the 117 users. A summarized view of the the number of events for each user can be seen in the box-plot in Figure 6.11. Only two out of the 117 users were outliers, and Section 6.2.4 discusses approaches to account for this imbalance.

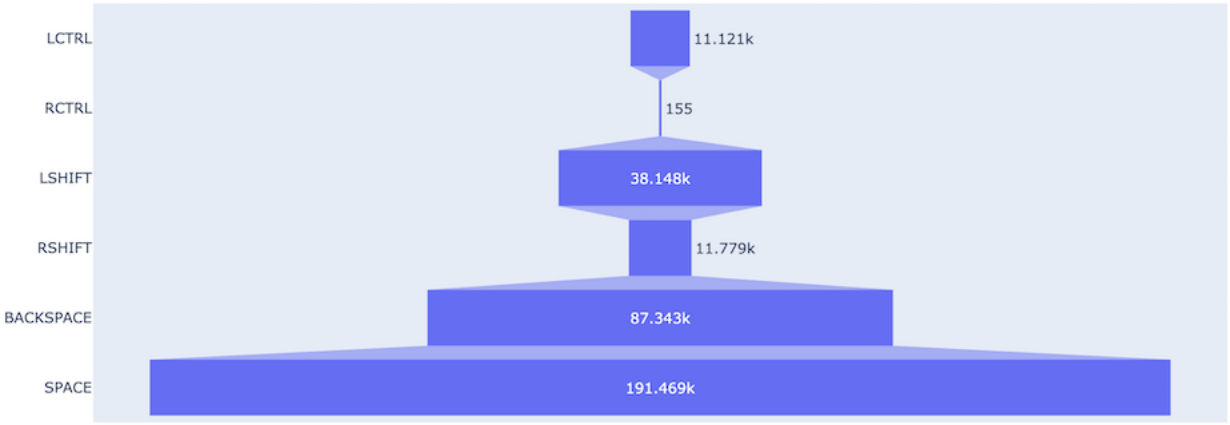

Figure 6.12. Visualization of key press counts for subset of keys

As mentioned previously and seen in Figure 6.9, the processed dataset consisted of five features: key, pX_y, pY_y, pX, pY where key represents the keyboard key that is pressed by the user, and the other features represent coordinates on the screen during events related to mouse movement and events related to using the mouse wheel respectively. Figure shows a visual representation of the counts of key presses for a subset of the keys, 6.12, giving an initial glimpse into user typing patterns.

6.3.2 Dataset Using Engineered Features

The engineered dataset consisted of the 14 keystroke features and three mouse features previously described. Table A.1 in the Appendix illustrate the 17 engineered features and a sample of events in this feature set.

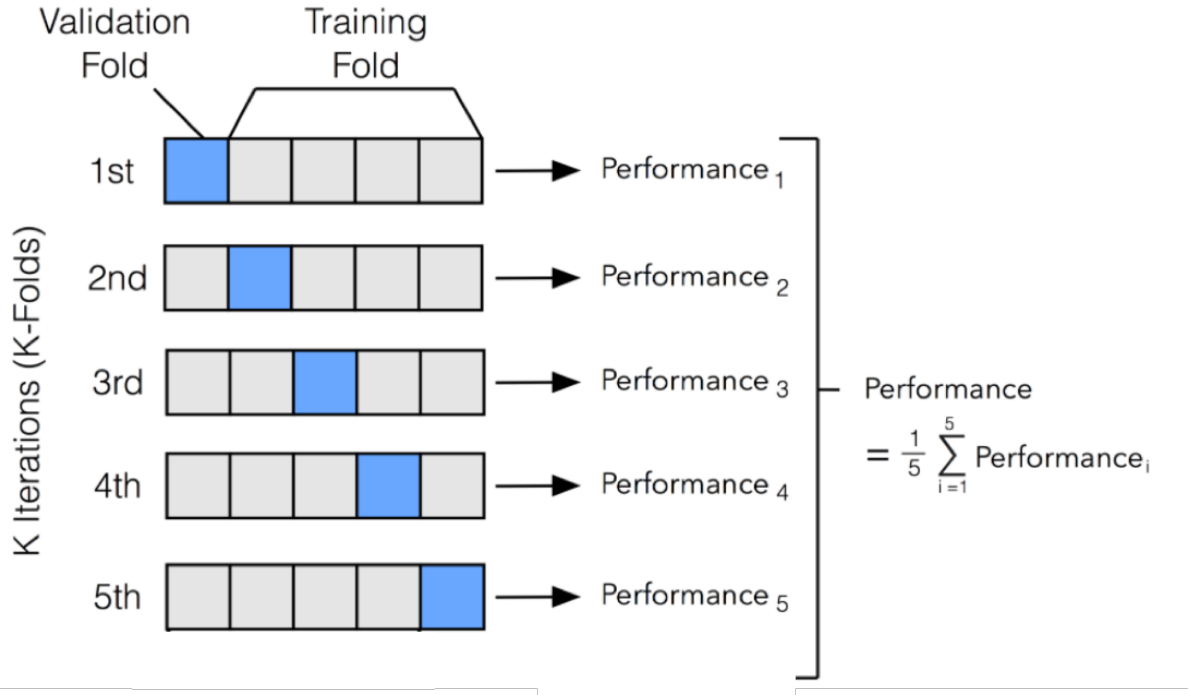

Figure 6.13. K-fold cross validation [236]

6.4 K-fold Cross Validation

K-fold cross validation is one of the most common methods of model evaluation. It allows each observation in the dataset to be tested, as opposed to only using a fixed subset for testing. The dataset is iterated over k times. In each iteration, the dataset is split into k parts with one part used for validation and *k-1* parts merged into a subset for training. Figure 6.13 illustrates k-fold cross validation.

An 80-20 training-testing split is used, using a random state of "1984". The random state provides a seed to for pseudo-randomizing. Chosen hyperparameter settings on the learning algorithm is used to fit the models to the training data in each iteration. When working with data of reasonable size (5000 to 10000 instances/rows), the conventional value of k is 10. The value of k as 5 is chosen for this study due to the large nature of the dataset i.e. the recommended number of folds increases inversely with the size of the dataset [236].

5-fold cross-validation provides 5 models fitted on unique training sets (that may be overlapping), which are then evaluated‘ on non-overlapping validation sets [236]. Eventually, the cross-validation performance is determined as the average of the k performance estimates from the validation sets. Unlike a simple train/test split, this method uses more training data and limits the pessimistic bias instead of earmarking a large amount of the dataset for testing purposes only [236].

6.5 Hyperparameter Tuning

Hyperparameters allow the tuning of machine learning algorithms in order to adapt the behavior of the algorithm to the specific dataset and research problem. While similar, they are distinct from parameters. They are the internal coefficients or weights for a model determined by the learning algorithm [237]. While model parameters are determined intrinsically, researchers configure the model through selection of the hyperparameters. It can be challenging to determine the value of the hyperparameters in order to tailor the model, and strategies such as random search and Grid Search have been defined towards identifying the values [238]. Tuning the hyperparameters has a time cost associated with it, making it preferable to choose a minimal subset of hyperparameters to tune.

6.5.1 Hyperparameter Tuning Algorithms

Grid Search is a common Hyperparameter Optimization (HPO) technique. As seen in Figure 6.14, it makes a complete search over the hyperparameter space of the training algorithm. The number of iterations is therefore the product of the possible values for the hyperparameters. Given that some hyperparameters may have unlimited values, some boundary needs to be usually specified when using Grid Search [239]. Since it searches the entire space, Grid Search can have large processing overheads. However, it can be parallelized since the hyperparameter values are usually independent of each other [238].

In contrast to Grid Search, Random Search selects random combination to train the model from a grid of hyperparameter values (Figure 6.15. This provides more control over

the number of iteration, and consequently, the computation time [240]. However, since the search is "random", the best results may not be immediately obvious.

Given that the time was not a big constraint and both approaches show comparable performance, the more predictable Grid Search algorithm was used to determine the hyper-parameters for the models.

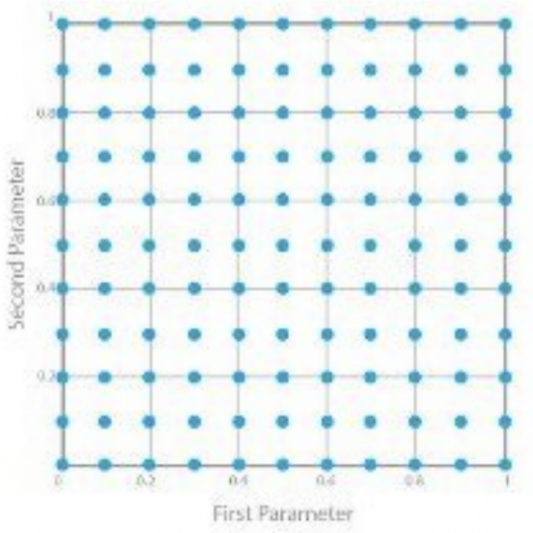

Figure 6.14. Illustration of parameter space using Grid Search optimization [238]

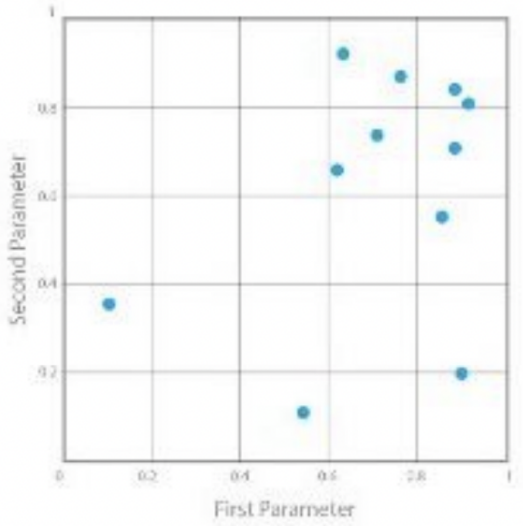

Figure 6.15. Illustration of parameter space using random search optimization [238]

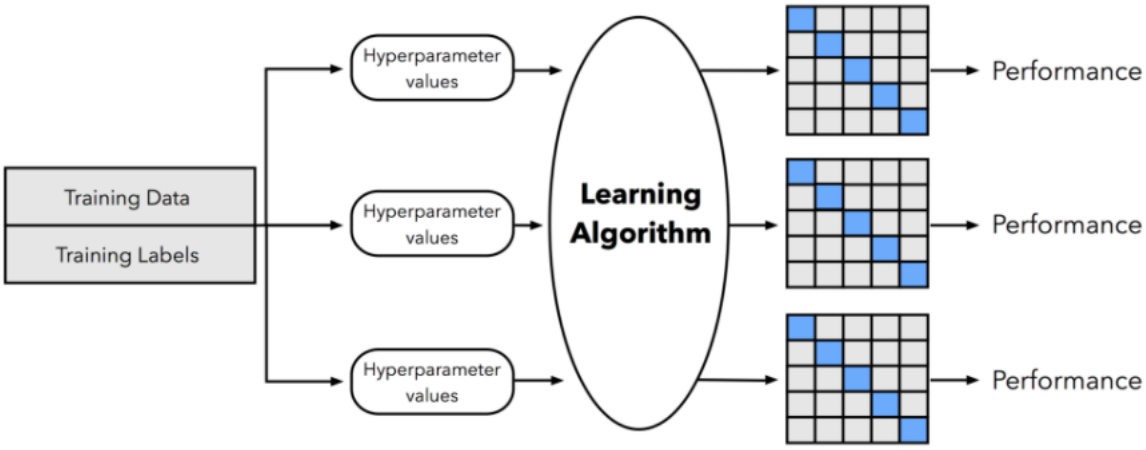

Figure 6.16. Hyperparameter Tuning [236]

6.5.2 Hyperparameter Tuning for XGBoost

XGBoost is very powerful in its ability to automatically tune thousands of learning parameters while providing some hyperparameters that can have a big impact on model performance. The hyperparameters chosen for tuning in this study are the *learning rate* and *gamma*:

- *learning rate*: This is similar to the learning rate in the gradient-boosting method. It refers to the shrinkage in the step size as updates are made to prevent overfitting. It shrinks the feature weights to make the boosting process more conservative and makes the model more robust. It has a range between 0-1, and has a default value of 0.3. [241]

- *gamma*: As discussed earlier, XGBoost is decision tree-based where each node splits into child nodes to reduce the loss function. The *gamma* hyperparameter specifies the minimum reduction in the loss to allow for the split. Similar to the learning rate, it makes the algorithm conservative with a larger gamma value leading to a more conservative approach [241].

An initial set of parameters is set as *learning rate*:$[0.001, 0.0001, 0.01, 0.0002, 0.1, 0.0003]$ and *gamma*: $[1, 0.1, 0.01, 0.001, 0.0001]$. As seen in Figure 6.16, for all 150 possible configurations, the 5-fold cross validation is applied on the training set, resulting in multiple models and performance estimates.

6.5.3 Hyperparameter Tuning for Support Vector Machines (SVM)

Just like for XGBoost, hyper-parameters can be tuned for SVM to improve the classification accuracy. Three major hyper-parameters that are generally used for SVM were tuned to determine the SVM model that can shows the greatest classification accuracy:

- *Regularization parameter, C*: The goal of the Support Vector Machine model is to establish the decision boundary that is generic enough to be useful for future classifications but also gives a high accuracy for the current dataset. This trade-off can be managed through the use of the regularization parameter *C*. *C* adds a penalty for each data point that is incorrectly classified by the algorithm. If the penalty is large, SVM tries to minimize the number of incorrectly classified instances [237]. This parameter can take on a range of values and has a dramatic effect on the shape of the resulting regions for each class. The *C* values of $[.1, 1, 10, 100]$ are considered as part of the optimization.

- *kernel*: Kernels can be considered the transformation functions that help to map the data to a higher dimensional plane. Some kernel options include linear, polynomial, sigmoid, or radial basis function (RBF). The Grid Search algorithm chooses between RBF and polynominal kernels during optimization.

- *gamma*: With respect to SVM, while the regularization parameter *C* aims for a high-variance and low bias, the *gamma* parameter aims for low-variance and a high-bias. A high *gamma* value can lead to overfitting of the model. If the *gamma* value is not provided by the user, then the library used chooses between *auto* or *scale* value where $auto = 1/n_{features}$ and $scale = 1/(n_{features} * X.var())$.

The Boosted Support Vector Machine algorithm (AdaBoost SVM) uses the same hyperparameter options as described above for Support Vector Machines.

6.5.4 Hyperparameter Tuning for Random Forest

Similar to the previous two algorithms, hyperparameters for the Random Forest algorithm help to optimize the model towards maximizing the performance:

- *max depth*: Given that Random Forest is a decision-tree based ensemble approach, this refers to the maximum depth that each tree can attain. Increasing the tree depth provides more nodes, allowing the tree to store more training data. But this can also lead to overfitting related issues. *Max Depth* values of $[10, 20, 30, 40, 50]$ are provided as options to the optimizer.
- *random state*: Similar to how pseudo-random generators work, this provides a 'seed' so that the results obtained from the algorithm can be reproduced. Setting an integer value as the random state tells the model to start with that seed each time.

Table 6.3. Hyperparameter values for the different algorithms using Grid Search (with the same values for both raw and engineered feature sets)

	SVM	**Random Forest**	**AdaBoostSVM**	**XGBoost**
gamma	**scale**	N/A	scale	**0.01**
kernel	**rbf**	N/A	rbf	N/A
max_depth	N/A	**40**	N/A	N/A
random_state	N/A	**1984**	N/A	N/A
learning_rate	N/A	N/A	N/A	**0.001**
C	100	N/A	100	N/A

The Grid Search optimizer is used to determine the ideal hyperparameters for the chosen algorithms. Table 6.3 shows the values of the hyperparameters obtained for the different machine learning algorithms. Using these optimal hyperparameter values obtained, the training set is then used to fit the model and obtain the machine learning model that can be used to attribute user activity to specific users. This has been illustrated in Figure 6.17.

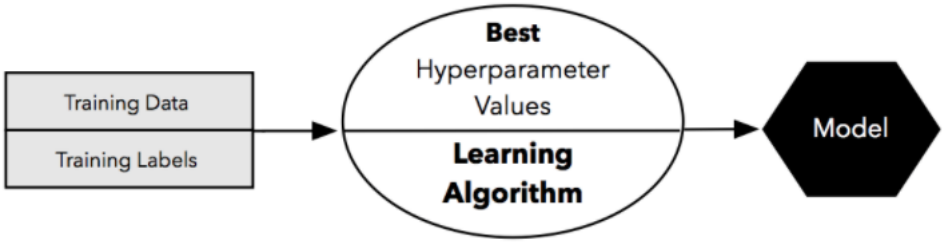

Figure 6.17. Model Fitting [236]

6.6 Performance Evaluation

This section provides an overview of the performance metrics observed based on the analysis conducted, both using the raw data features, as well as the engineered feature sets.

6.6.1 For analysis using raw features

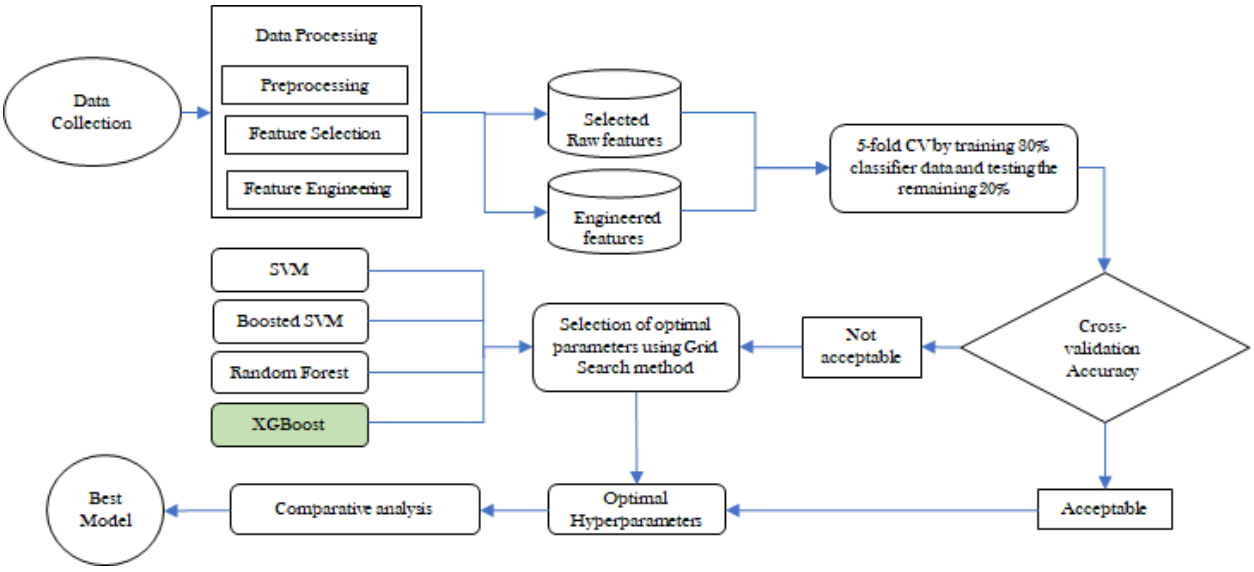

Figure 6.18. Overall process of model development and performance evaluation

Figure 6.18 provides an overview of the methodology and performance analysis. As seen in the figure, the analysis is performed with three other algorithms - Support Vector

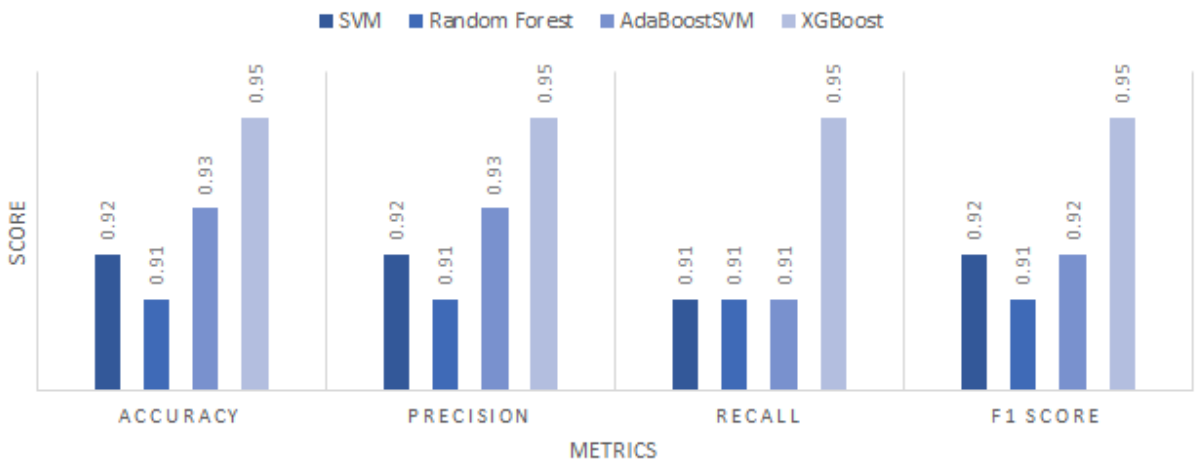

Figure 6.19. Performance using 5 raw features, after feature selection

Machines (SVM), Random Forest (RF) and Boosted Support Vector Machines. As seen in the literature review, Support Vector machines and Random Forest classifiers have been used by previous research. Boosted Support Vector Machines were also included so that the XGBoost model is not only compared to models used in previous research but also to other boosted approaches for a more robust comparison. The model fitted with the optimal hyperparameters, showing the best performance, is selected as the ideal classifier for user attribution.

Figure 6.19 shows the performance metrics for the four algorithms, using the raw selected features. With a mean F1 score of .95, it can be determined that XGBoost performs really well at attributing the user activity to the correct user. The figure also shows that XGBoost outperforms the other algorithms in terms of all the performance metrics of accuracy, precision, recall that were discussed in the previous chapter. Figure 6.20 shows the performance metrics after the application of the SMOTE technique. This shows some improvement for the other algorithms with the performance metrics for the XGBoost algorithm. This leads to multiple observations:

- XGBoost is fairly resistant to class imbalance issues and shows equally good performance without requiring additional oversampling methods

- Even when class imbalance is corrected, XGBoost outperforms the other algorithms
- XGBoost is not just better performing that Support Vector Machines (SVMs) but also outperforms ensemble techniques like Random Forest and boosted version of Support Vector Machines.

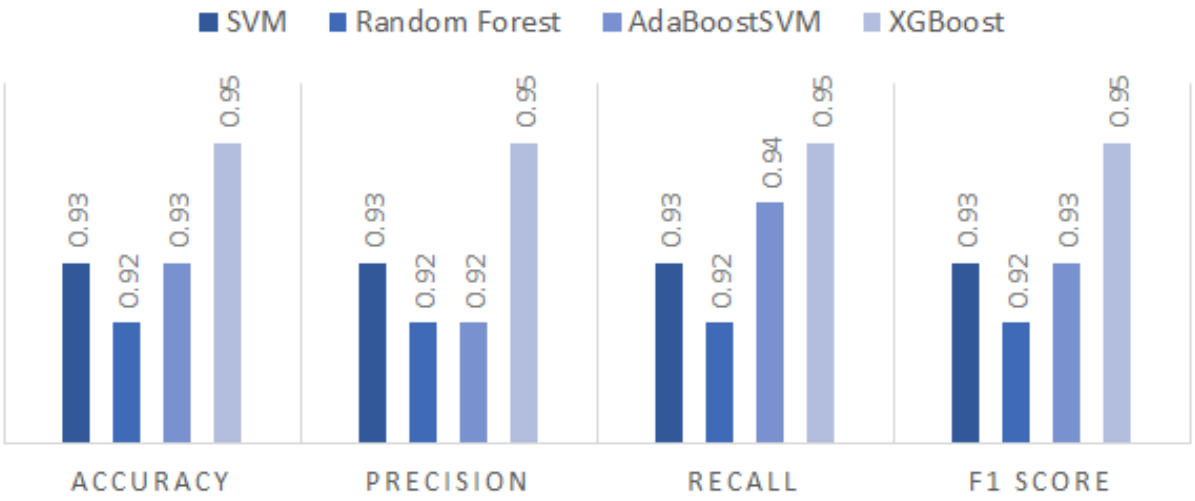

Figure 6.20. Performance using 5 raw features after feature selection and oversampling using SMOTE technique

The analysis was also performed on all the 22 features, without feature selection from the raw set of features, to validate assumptions about the overfitting of data. As seen in Figure 6.21, all the algorithms perform worse when all the features was considered. This is expected behavior due to overfitting as previously discussed. In addition to overfitting, another consideration for feature selection was performance cost in terms of the training time. Figure 6.22 and Figure 6.23 illustrates the impact on feature selection on the training time. It is observed that the training time is reduced across all the algorithms, with the XGBoost classifier having the lowest. If the training times are compared once the data is oversampled with the SMOTE technique (and increasing the number of training samples), all algorithms except XGBoost have greater training times that when the full feature set is used. To put it more simply:

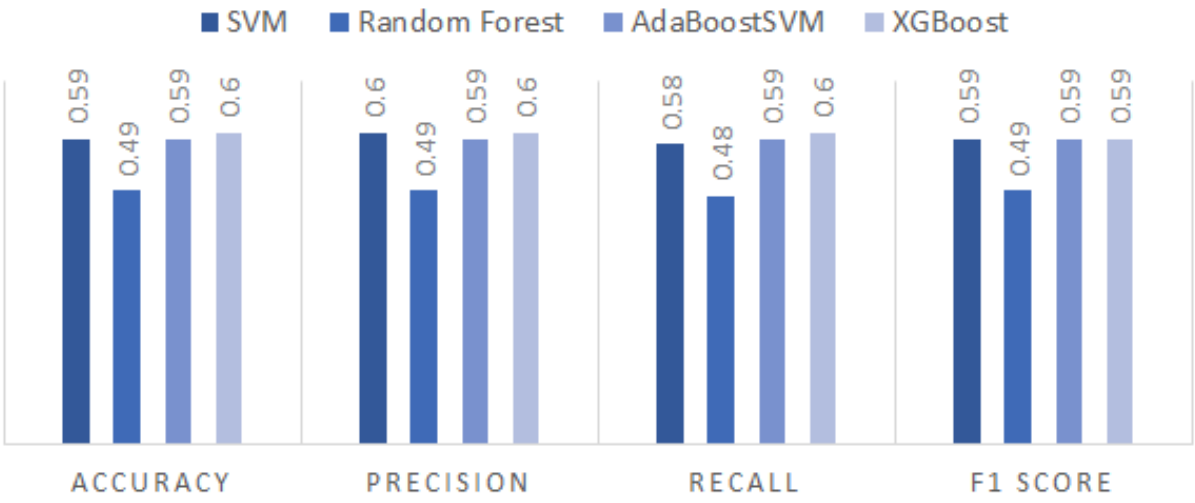

Figure 6.21. Performance using 22 raw features, without feature selection

- Without oversampling, feature selection techniques almost double the time performance of the algorithms
- For the XGBoost algorithm, feature selection reduces the training time, even when the features are oversampled using the SMOTE technique
- XGBoost outperforms the other algorithms once feature selection techniques are applied

Therefore, XGBoost not only outperforms the algorithms in terms of classification but also does so with the least time cost, once only the selected features are used. There isn't a defined threshold for how much *better* an F1 score should be, to be considered statistically significant improvement. However, previously published research studies in different areas have called out improved performances of classifiers in similar ranges of 1% improvement [242] [243]. For classification, not only is the model accurate but it shows high performance in term of how it deals with both false positives and false negatives.

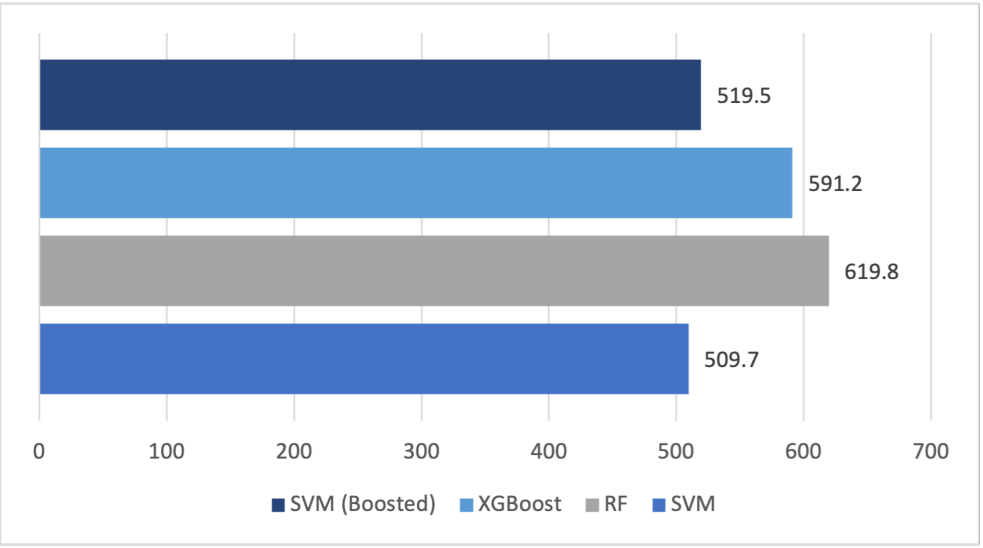

Figure 6.22. Training time using 22 features, before feature selection

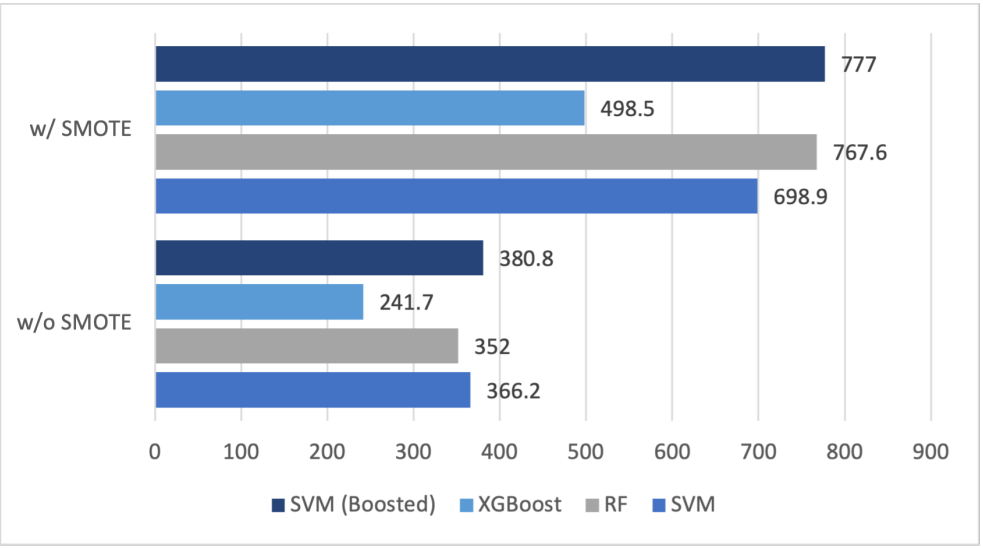

Figure 6.23. Training time using 5 features, after feature selection

6.6.2 For analysis using engineered features

Figure 6.24 shows the performance metrics when only the three engineering mouse features are used. With scores between 71% - 74%, the three mouse features perform relatively well considering they are discriminating between 117 users. However, it obviously does not perform anywhere as well as the combined raw feature set. Figure 6.25 shows the perfor-

mance when only the 14 keystroke features shown in Table 6.2 are used. With an F1 score of .89, the XGBoost model performs better than when only mouse features are used. Figure 6.26 shows the performance when both engineered keystroke and mouse features are used. With an F1-score of .9, while under-performing when compared to the raw data set, the model satisfies the conditions to disprove the null hypothesis and performs well towards attributing the events to the right user with 90% accuracy, precision and recall.

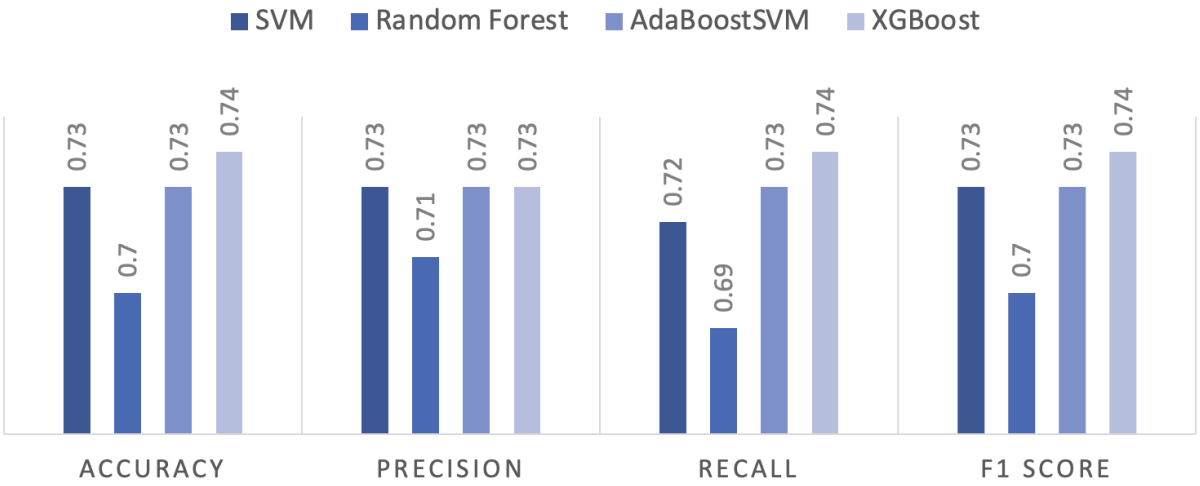

Figure 6.24. Performance using only engineered mouse usage features

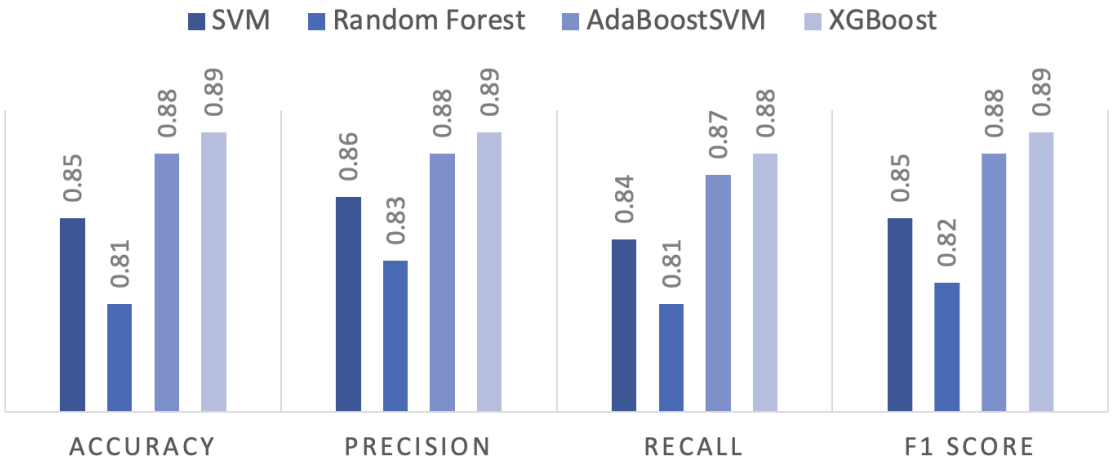

Figure 6.25. Performance using only engineered keystroke usage features

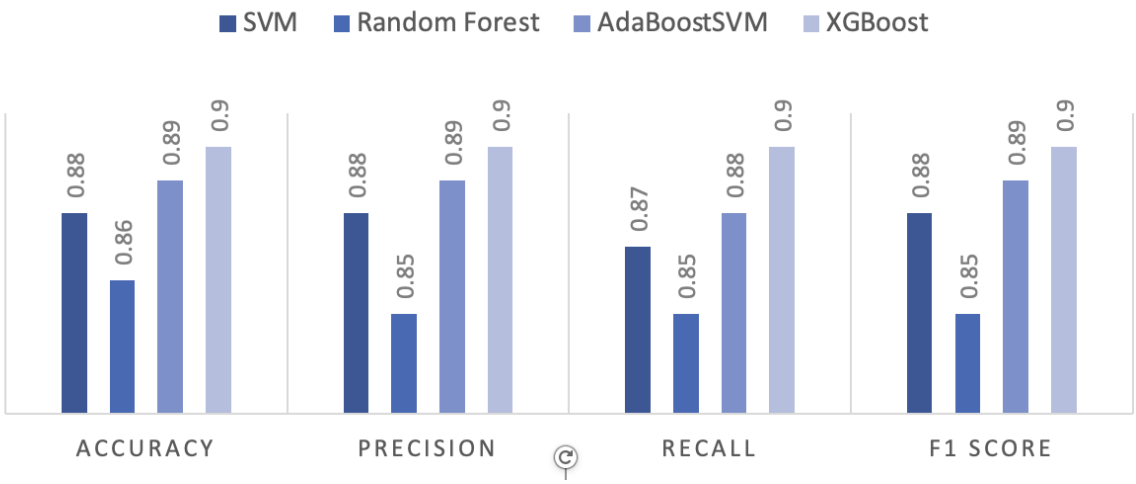

Figure 6.26. Performance using engineered keystroke and mouse usage features

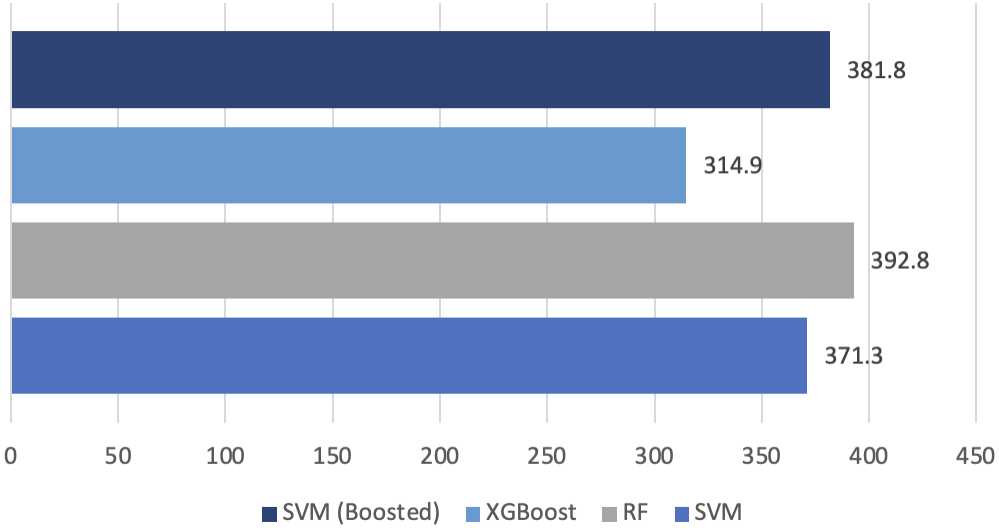

Figure 6.27. Training time when only using the 3 mouse features

Similar to the analysis with the raw feature set, the training times are compared for the different algorithmic models. As seen in Figures 6.27 and 6.28, and XGBoost still has shorter training times than the other algorithms under review. It is interesting to see that even though the number of features are much lesser (with 14 keystroke features against three mouse features), using only keystroke data has a better performance in terms of time. Figure 6.29 shows that when considering analysis using both engineered keystroke and mouse

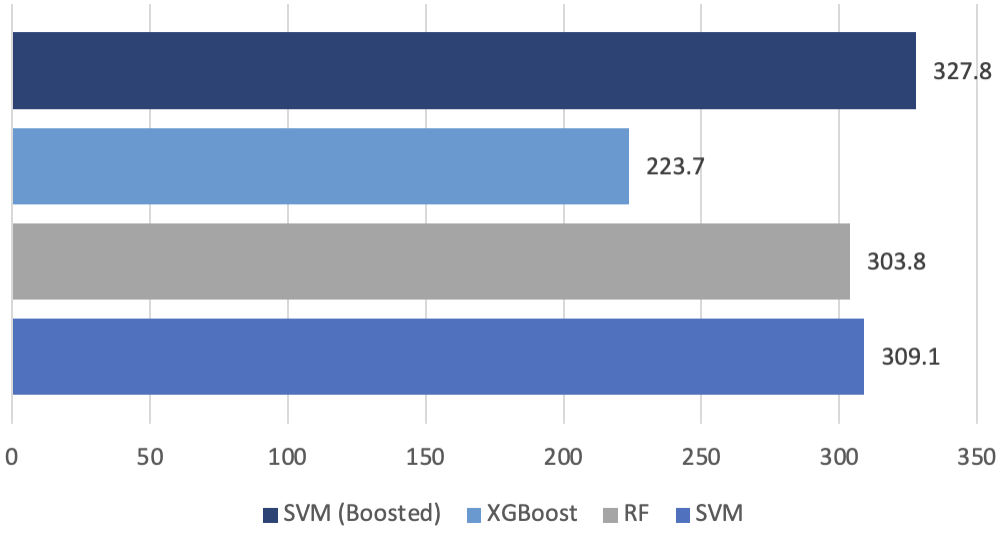

Figure 6.28. Training time when only using the 14 keystroke features

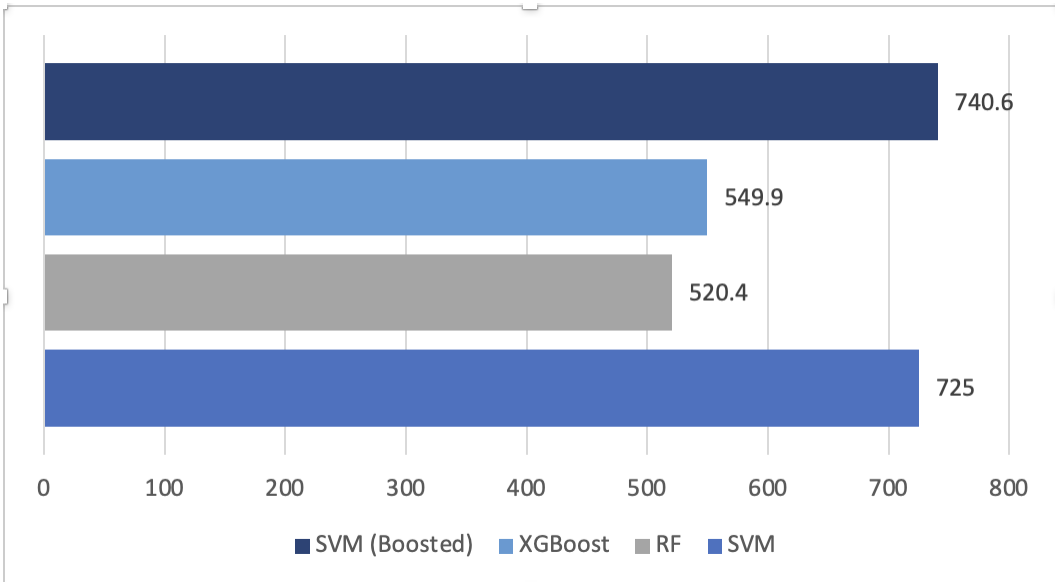

Figure 6.29. Training time when only using both the mouse and keystroke features

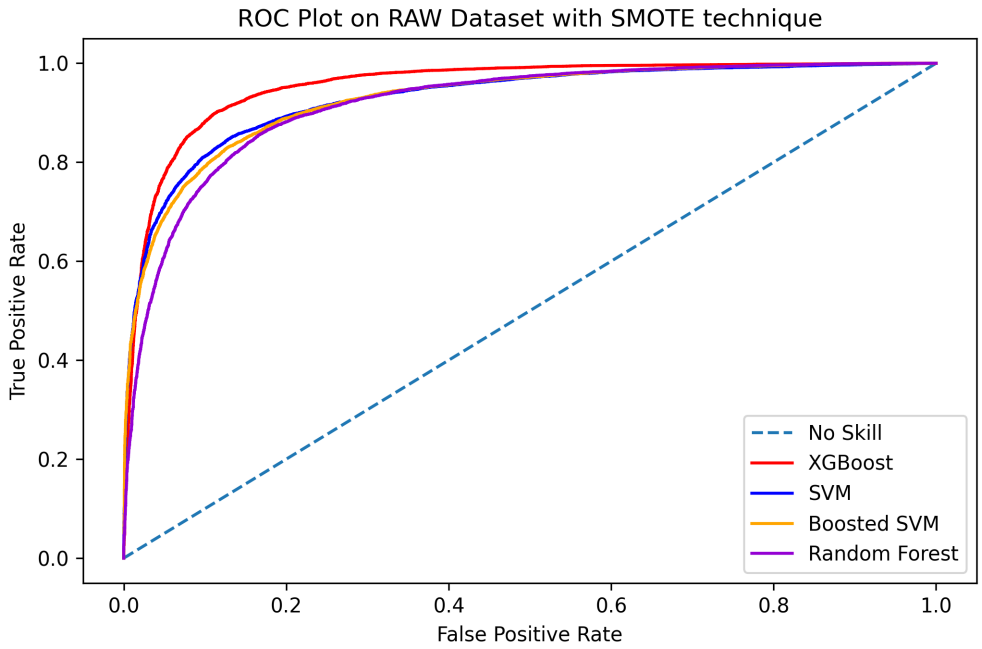

Figure 6.30. Receiver Operating Characteristics Curves for the raw dataset using the SMOTE technique

features, XGBoost performs better than Support Vector Machine (SVM) and boosted SVM, while having slightly longer training time than the model created using the Random Forest algorithm.

6.6.3 Using Area under the Receiver Operating Characteristics (ROC) Curves

As mentioned in Chapter 5, the calculated Area under the Receiver Operating Characteristics Curves (represented as AUROC or simply AUC) is a standard metric to estimate performance in field of biometrics. The curve is simply the false positive rate mapped against the false negative rate, and helps to visualize the trade-off between the two. A *good* ROC curve is one that's curved against the top and the left as seen in Figure 6.30 for analysis using the raw datasets and as seen in Figure 6.31 for the engineered dataset. The Area Under the Curve (AUC) gives us a more concrete metric to determine the performance of the curve.

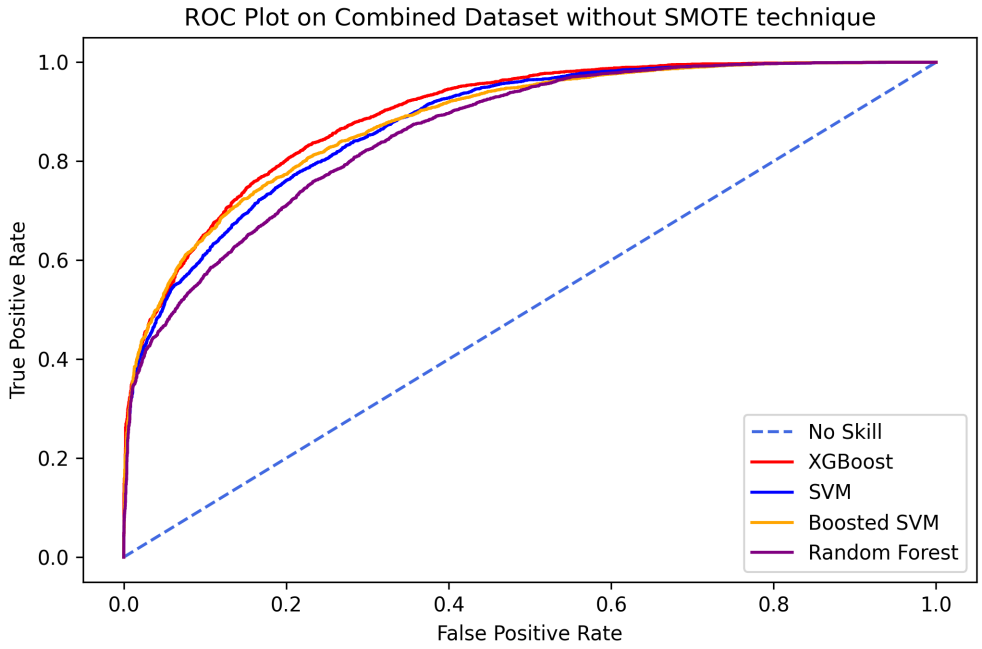

Figure 6.31. Receiver Operating Characteristics Curves for the engineered dataset

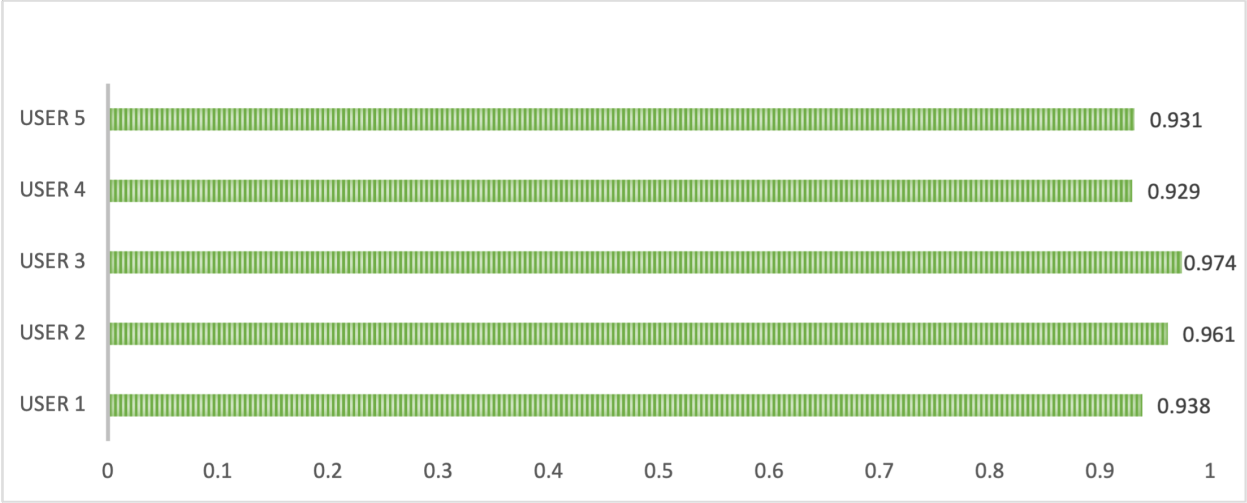

Figure 6.32. Area under the curve for five users, using the XGBoost algorithm

Table 6.4. Comparison of AUC scores for different algorithms

Algorithm	Raw Dataset		Engineered Dataset
	With SMOTE	Without SMOTE	
XGBoost	.954	.953	.899
SVM	.928	.925	.874
Boosted SVM	.926	.939	.881
Random Forest	.918	.921	.855

Figure 6.32 shows the AUC for five users using the XGBoost model with the raw dataset (without using the SMOTE technique). This can be interpreted as a measure of how well the model would classify events related to that specific user. The average of these values can be used to determine the AUC for the model, for each of the analyses performed. Table 6.4 show a comparative analyses of the AUC score obtained using the different algorithms for analysis.

As seen from the area under the curve and the F1 score, the results presented overwhelmingly supports the hypothesis that the XGBoost model proposed can be used for user attribution with an impressive success rate. It also indicates that XGBoost shows higher performance metrics, both using F1 scores and AUC than previously used models in this research area. The next chapter discussed the conclusions, some of the challenges with the research study, and suggested future directions.

7. CONCLUSIONS, DISCUSSION, AND RECOMMENDATIONS

As seen in the findings, the primary conclusion of this research study is that events on a computer can be reliably attributed to a specific user, using the XGBoost algorithm to model their keystroke and mouse usage, with an F1 score and Area Under the Curve (AUC) value of .95. In addition to this, XGBoost has better performance metrics than previously used algorithms for similar problems of user attribution. Table 7.1 shows how the results of this research compare with other studies that have explored user attribution for forensic purposes through modeling keystroke or mouse usage. As seen, the only study with comparable error rates has challenges with robustness due to a small sample size of 8 users [244]. It should be noted that Shen, Cai, Maxion, *et al.* [125] had slightly different goals than user attribution. The researchers used the keyboard and mouse data individually (so not a multi-modal approach) to infer the demographic traits of users. While this is not technically user attribution, given that they explored the use of keystroke and mouse data for forensic purposes, the research has been included in this comparative analysis.

Table 7.1. Previous research on user attribution in digital forensics using keystroke or mouse data

Study	Users	Classifier	Features	Accuracy %	Year
Bhukya and Banothu [244]	8	One-class SVM	Keystroke and Mouse	94.88	2011
Shen, Cai, Maxion, *et al.* [125]	58	Weighted RF	Keystroke and Mouse	82.11 - 87.32	2013
Mohlala, Ikuesan, and Venter [245]	42	J48, RF	Keystroke	70 - 100	2017
Ernsberger, Ikuesan, Venter, *et al.* [62]	3	RF	Mouse	78.1	2018
Ikuesan and Venter [11]	10	RF	Mouse	<80	2019

As seen, most of the previous studies have used a Random Forest (RF) classifier. Some studies explore multiple classifiers, but classifiers that align with the reported accuracy have

been included. It should also be noted that the *accuracy* is compared since that is the most consistently reported metric but both *accuracy* and *sensitivity* should be considered when evaluating classifier performance. It can also be seen that the earlier studies have a much smaller sample size, with the largest study employing 58 participants [125]. Small number of participants has been called out as a research challenge for past studies on mouse dynamics [62]. With a sample size of 117 participants, this research is the most comprehensive study using a multi-modal approach for behavioral biometrics, in the field of digital forensics. Also, this study uses a publicly available dataset for analysis, providing for enhanced opportunities for repeatability and comparative analysis. This research contributes towards originality in terms of the multi-modal approach, the sample size, and also in reporting the performance metrics of the model used.

7.1 Key Contributions

This section provides some discussions on the key contributions of this research as seen in the presentation of the results. Prior research in the area of user attribution in digital investigations usually attempts to discriminate between a small set of users [102], [125], [72], [108]. The results obtain show that events may be attributed to a unique user, even among a larger set of 100+ users. The results also show that XGBoost consistently performs better than previously used algorithms, irrespective of the specific nature of the features. While this is not surprising given that XGBoost is the current state-of-the-art algorithm, this study validates the improvement in algorithm performance towards user attribution.

7.1.1 Key contributions using the raw feature set

- With an F1 score of .95, this research provides an XGBoost model that performs really well in attributing events to the correct user, even with a large dataset of 117 users.

  The developed model also shows high accuracy, precision and recall with a value of .95 for each of those metrics. As a reminder, accuracy measures the ability of the model to correctly predict the user for an event. The model can accurately predict the right user 95% of the time. *precision* measures the number of true positives, or

in other terms, on an average 95% of the events attributed to a user, actually belong to user. *Recall* measures the ability to find all the events that belonged to a user. So on an average, 95% of all the events were correctly attributed, that should have been attributed to a user. From the lens of digital investigations, precision may become the most important metric, since inaccurately attributing a user activity to a specific user is an unacceptable scenario. However, if the analysis is more focused on goals such as establishing patterns, then accuracy and recall become equally significant.

- The results also suggest that XGBoost outperforms other algorithms such as Support Vector Machines, Boosted Support Vector Machines, and Random Forest in all areas such as the accuracy, precision, recall and as a result in its overall performance. This is a significant contribution since XGBoost has not been previously explored for user attribution in digital investigations.

- The model performs better with 5 selected raw features, and avoids the overfitting caused by using all 22 features. This provides a simpler model that can potentially reduce the storage and performance overhead.

- Once raw features are selected and the non-contributing features are discarded, the developed XGBoost model also shows reduced training times that are much faster than other algorithms, introducing the potential of reducing the time cost involved.

7.1.2 Key contributions using the engineered feature set

- The results illustrate that all algorithms perform better when using a multi-modal approach of keystroke and mouse features instead of using only keystroke features or only mouse features. To the knowledge of the authors, this comparison over the same dataset has not been previously done. While this may be intuitive, there is always the **curse of dimensionality** and risk of overfitting when using a larger feature set.

- The results also showed that for the specifically engineered set of keystroke and mouse features, the keystroke features outperformed the mouse features. This is not surprising since only 3 engineered mouse features are being used. In fact, the performance of

the 3 mouse features in attributing an event uniquely out of 117 users is quite impressive. The majority of previous research exploring use attribution using mouse features employs 10 users [97], [43], [98], [100].

Overall, a multi-modal approach using keystroke and mouse data is very promising in its applicability towards user attribution in digital investigation. It can provide *proactive evidence*, evidence that provides evidentiary weight and will contain contribute towards linking the crime to the criminal, assisting with successful prosecution [21]. As discussed previously, with the adoption of machine learning techniques, the shift from repeatability to validation necessitates research studies such as this one to establish published error rates. This is not only relevant discussing computer-assisted crimes but also when discussing information security in general. While information security and digital forensics are distinct fields, there is a relevant overlap. When security strategy for an organization is discussed, the focus is usually on:

- *Resistance* - the ability to withstand attacks through the use of tools such as user authentication, firewalls, etc.
- *Recognition* - the ability to detect attacks through the use of intrusion detection systems, event monitoring systems etc.
- *Recovery*- the ability to continue to operate critical services and restore all functionality after an attack through the use of disaster recovery and business continuity methods.

Endicott-Popovsky and Frincke [246] suggested that the security strategy also need to consider a fourth attribute - *Redress* or the ability to identify and hold intruders accountable in a court of law through the use of digital forensic and investigative techniques. Therefore, user attribution is an important security, as well as forensic problem to solve. As mentioned by Katz [22], a very contribution of such research is that it encourages and builds confidence in innovative methods in the field of digital forensics.

7.2 Challenges

While the approach has shown promise, this is still in very early stages with a lot of maturation required before this can be deployed for use. As an example, the European standard for commercial biometric technology requires a .001% false-acceptance rate and 1% false-rejection rate [247]. The obtained results are still miles from these required error rates. Considering future adoption, some of the other challenges of the study are discussed below:

1. *Study Limitations and Delimitations*: The limitations and delimitations of the study that were introduced in 1 discuss some known challenges in its adoption. In the context of the results obtained: some of the raw features that performed really well are related to the coordinates of the open window on the screen. The placement of the window can be seen as a preference. These patterns of preferences or habits can be used towards discriminating that specific user [14]. However, future work can be expanded (e.g., to collect data over multiple sessions or over multiple operating systems) for enhanced robustness.

2. *Data Integrity*: The reliance upon stored data for training and testing, introduces the possibility that the stored logs can be modified. In order to avert this, logs stored on each computer can be encrypted to ensure that these logs cannot be manipulated.

   There is also the unlikely possibility that the user on a computer purposely attempts to modify their behavior. In such a situation it would be very difficult for them to change their computer usage in such a way that it completely matches another user's profile. However, it may be possible that the user can modify their own behavior so that it does not match any stored profile. This can be seen as the equivalent of a bad print in fingerprinting or a damaged sample in DNA analysis which does not provide enough information to make a match.

3. *Data Availability*: Another concern is that the logs could be deleted from the system. Deleting the logs from a suspect computer can be seen as the equivalent of someone wiping fingerprints at the scene of the crime. While it can be seen as a setback to an

investigation, it would not have an adverse effect or misguide the investigation in any way. Also, the mechanism is suggested as an extensible part of the operating system. It would need a high level of computer expertise to locate the logs and attempt to remove them.

4. *Model Training*: A challenge that would need to be addressed is a way to create 'clean' profiles where it can be certain that the system learns the behavior from the actual user and not another user sharing the system. With the increase in the number of laptops enabled with physiological biometrics, it might be possible to consider sessions where biometric authentication is used, as 'clean' sessions. Another potential approach to flag clean sessions might be to use those sessions where a secondary authentication such as logging into an email account has occurred. This increases the chances that the correct user is on the computer.

5. *Over-reliance on Technology*: Machine learning techniques, like the one under discussion, are tools that provide more information in an investigation. However, the investigation still relies primarily on the abilities of the investigator. There are several logical decisions that must be made by an expert throughout the analysis. Even among researchers, the focus is often on the analysis without due diligence around supporting activities such as the feature determination, algorithm selection etc. [38]. Like traditional statistical approaches, machine learning techniques are only as good as the data presented and needs very specialized knowledge for accurate analysis and interpretation. With lack of regulated and formalized credentials that can validate the investigator's knowledge in the area, there is a danger of complete reliance on the machine learning algorithm, the data being misinterpreted, and eventually increasing the probability of a failed investigation [248]. This is exacerbated by the known gaps in digital forensics training for law enforcement, even for more established digital investigation techniques [249][250].

6. *Evolving Landscape*: A problem with all digital techniques is the ever-evolving technology landscape. The developed model is fairly agnostic to the operating environment,

making it decently robust to technology changes related to the operating system, networking technologies etc. However, the method suggested is still immature and there is an improbable yet possible chance that user interaction with the keyboard and mouse may evolve within the timeline of its maturation. In such an unlikely scenario, this adaptable model can be updated with different features or modalities, while still utilizing the underlying premise.

7.3 Discussion on the legal admissibility

The suggested technique involves using machine learning to train a system on keyboard and mouse usage while performing common tasks on a computer. This training profile can be stored on the local computer in the form of logs. If there is a situation in which there is a need to attribute a user event to a particular user on the computer, then the suggested model can provide a probabilistic measure of the likelihood of the occurrence. This section explores how the Daubert Criteria could be applied to the user attribution techniques defined through this research:

1. *Whether the technique has been tested*: The researchers have tested the hypothesis and published the error rates. While the testing was conducted in a controlled environment, which might not be reflective of a 'real-world' scenario that might be encountered in a court of law. However, this research is a foundation towards establishing techniques that can later be expanded into more generic situations. Therefore, while the proposed method in the current form might not be directly used in evidence analysis, it may form the basis of future research which can be used in court. In the current form, the proposed technique has been tested with vigorous scientific principles to ensure validity of the results.

2. *The error rate associated with the method*: As mentioned earlier, the error rates have been tested and published. The error rates include consideration of the false positive rates and the false negative rates. This dissertation also publishes a comparative analysis of the error rates with other established techniques used for profile analysis within the discipline. Weiss [153] has discussed the idea of relating the degrees of

scientific uncertainty with the 'standards of proof' used in the legal system. These are established definitions of the degree of proof required to maintain a balance between the rights of the different stakeholders according to the different situations. They propose 11 degrees of legal standards of proof, which are mostly obtained from the commonly accepted standards in the judicial system. They relate these legal standards to informal levels of scientific certainty and also relate them to Bayesian probabilities. Subscribing to their model, gives us a way to establish an informal legal standing of our method based on its scientific principles and probability measures. The testing criteria of this methodology defines success as XXX of user sessions being attributed correctly. A conservative adoption of Weiss [153]'s model would place this in the 'Clear Showing' legal standard of proof, which indicates a scientific level of 'very probable' with a Bayesian probability of 80-90 %.

3. *The publication in a peer-reviewed journal*: As mentioned earlier, the suggested criteria is not meant to be exhaustive. This dissertation is reviewed by a doctoral committee and will be published in an database. So while it fulfils the spirit of the intent behind this criteria, it may need further publication in a peer-reviewed journal before it fully satisfies this criteria.

4. *Whether the technique has gained widespread acceptance*: With a novel method, it is difficult to gauge widespread acceptance. Scientists understand that acceptance of new scientific assertions is a social process and some premise may be accepted earlier than the others [153]. A scientific assertion may go through many different stages when traversing the uneven path between acceptance and rejection. It is not common for scientists to hold opposing views on the same assertion based on their specific scientific discipline, their levels of skepticism, and possibly their political and economic interests. In fact, Woolgar and Latour [251] have commented that scientific research is a long and expensive effort to change the notion of *maybe* to *is*. With respect to this study, the research is based on established principles of machine learning where the acceptance within the scientific community is not disputed. The acceptance of the specific application of machine learning towards the usage of computer behavior

profiles in digital investigations is something that is soon gaining acceptance and has been proposed by several researchers. the next section discusses approaches based on similar principles that have gained widespread acceptance in scientific and legal communities.

The role of tools in digital forensics exists beyond the courtroom. Digital forensics tools are routinely used not simply to convict the perpetrator in court but to understand the mechanics of how a crime was committed and the role of the perpetrator in the crime [139]. The proposed technique provides a likelihood that a particular user was using the computer during an event. However, like for all scientific techniques, the limitations need to be understood for its effective use. The role played by the profile analysis will be a supportive role to guide the investigation. In this capacity, computer behavior profile analysis should pass the Daubert criteria. However, with a success rate close to 95%, it cannot be considered as a reliable measure by itself, to concretely differentiate the user on the system from other users. For this method to be 'rigorously proven' according to the scale suggested by Weiss [153], it would need to account for all possible explanations of alternative scenarios, which is not feasible at this time. However, until then, a profile match between users will provide a statistical correlation, instead of an absolute identification of the user. It would require additional evidence before the conclusion can be considered sufficient for a conviction. In spite of this, as a supportive tool, this technique should pass the Daubert standard. Garrie [12] comments that there are many unique situations that arise in the courtroom, which may require an innovative approach by the expert. The method proposed has the potential to be a scientifically sound new innovative approach in the courtroom.

7.4 Future Work

Some obvious future directions would be towards resolving the discussed challenges. The benchmark dataset used provides researchers with the ability of comparing different features, modalities, and classification algorithms on the same set of users. Further research can explore integration of modalities such as gait analysis, touchscreen biometrics etc., that may play a greater role in a mobile-centric society. In addition to that, future research can also

be further expanded to include inter-session data, such that temporally-spaced events are attributed to users. Such research studies exist [133] but have the limitation of small sample sizes.

As seen from the results of this study, a multi-modal approach combining keystroke and mouse data shows great promise. Future research can expand the nature of the feature set to include more discriminative features such as drag and drop events, double-click events etc. As mentioned by Ikuesan and Venter [11], future implementation and use of digital behavioral signatures will include updateable databases, potentially as part of forensic-friendly operating systems.

As previously discussed, the suggested behavioral attributes can only provide supporting information towards accepting or refuting hypotheses during investigations. While machine learning approaches present a shift from repeatability towards validation-focused techniques, there is a need for governance around the people, processes and technology of such evidence before it can be used and accepted [252]. Some examples can include identifying the skills required, the logical decisions around choosing the algorithm etc. and analysis tools. This standardization can help towards faster adoption, towards establishing a more robust method, and also towards building a consistent and cohesive training agenda for law enforcement agencies in the future. This leads to a related research direction focusing on updating existing digital investigation frameworks to include forensic readiness as a key component [20]. Such frameworks can explore the integration of behavioral biometrics into a 'user attribution' phase of the digital investigation life-cycle [11].

Individualization or the ability to specifically identify a user presents the greatest value to digital evidence analytics [252]. This research presents the concept of a tool that provides supporting information about a user. Combining such a tool with other investigative techniques can greatly support digital investigators of the future [14]. The several avenues of future research discussed present many exciting opportunities to advance digital investigation techniques towards greater preparedness for a complex, dynamic, and very data-centric forensic landscape.

REFERENCES

[1] N. Metropolis, *History of computing in the twentieth century*. Elsevier, 2014.

[2] M. Pollitt, "A history of digital forensics," in *IFIP International Conference on Digital Forensics*, Springer, 2010, pp. 3–15.

[3] C. M. Whitcomb, "An historical perspective of digital evidence: A forensic scientist's view," *International Journal of Digital Evidence*, vol. 1, no. 1, pp. 7–15, 2002.

[4] R. Saferstein and A. B. Hall, *Forensic Science Handbook, Volume I*. CRC Press, 2017.

[5] G. Gogolin, "The digital crime tsunami," *Digital Investigation*, vol. 7, no. 1, pp. 3–8, 2010.

[6] C. Ryan, *Computer and internet use in the united states: 2016. united states census bureau*, 2018.

[7] A. H. Dictionary, *The american heritage science dictionary*. Houghton Mifflin Harcourt, 2005.

[8] E. Huebner, D. Bem, and O. Bem, "Computer forensics–past, present and future," *Information security Technical report*, vol. 8, no. 2, pp. 32–46, 2007.

[9] D. W. Gresty, "Digital forensic system profiling using context analysis," Ph.D. dissertation, University of Greenwich, 2018.

[10] A. R. Ikuesan and H. S. Venter, "Digital forensic readiness framework based on behavioral-biometrics for user attribution," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, IEEE, 2017, pp. 54–59.

[11] A. R. Ikuesan and H. S. Venter, "Digital behavioral-fingerprint for user attribution in digital forensics: Are we there yet?" *Digital Investigation*, vol. 30, pp. 73–89, 2019.

[12] D. B. Garrie, "Digital forensic evidence in the courtroom: Understanding content and quality," *Nw. J. Tech. & Intell. Prop.*, vol. 12, p. i, 2014.

[13] A. Orebaugh and J. Allnutt, "Classification of instant messaging communications for forensics analysis," *The International Journal of Forensic Computer Science*, 2009.

[14] N. Eagle and A. S. Pentland, "Eigenbehaviors: Identifying structure in routine," *Behavioral Ecology and Sociobiology*, vol. 63, no. 7, pp. 1057–1066, 2009.

[15] W. L. Bryan and N. Harter, "Studies in the physiology and psychology of the telegraphic language.," *Psychological Review*, vol. 4, no. 1, p. 27, 1897.

[16] S. Gupta, M. Rogers, S. Elliot, and T. Hacker, "Using computer interaction as a soft biometric in digital investigations," Purdue University, 2012.

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[18] J. T. McDonald, Y. C. Kim, and A. Yasinsac, "Software issues in digital forensics," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 3, pp. 29–40, 2008.

[19] C. Taylor, B. Endicott-Popovsky, and D. A. Frincke, "Specifying digital forensics: A forensics policy approach," *Digital investigation*, vol. 4, pp. 101–104, 2007.

[20] A. Mouhtaropoulos, C.-T. Li, and M. Grobler, "Digital forensic readiness: Are we there yet," *J. Int't Com. L. & Tech.*, vol. 9, p. 173, 2014.

[21] C. P. Grobler and C. Louwrens, "Digital forensic readiness as a component of information security best practice," in *IFIP International Information Security Conference*, Springer, 2007, pp. 13–24.

[22] E. D. Katz, "Differentiating users based on changes in the underlying block space of their smartphones," Ph.D. dissertation, Purdue University Graduate School, 2020.

[23] T. Goldring, "Authenticating users by profiling behavior," in *Proceedings of the ICDM Workshop on Data Mining for Computer Security, ser. DMSEC*, vol. 3, 2003.

[24] B. E. Turvey, *Criminal profiling: An introduction to behavioral evidence analysis*. Academic press, 2011.

[25] M. K. Rogers, "Psychological profiling as an investigative tool for digital forensics," in *Digital Forensics*, Elsevier, 2016, pp. 45–58.

[26] E. Casey, "Cyberpatterns: Criminal behavior on the internet," *Criminal Profiling: An Introduction to Behavioral Evidence Analysis (Ed B Turvey)*, pp. 299–327, 1999.

[27] A. K. Belman, L. Wang, S. Iyengar, *et al.*, "Insights from bb-mas–a large dataset for typing, gait and swipes of the same person on desktop, tablet and phone," *arXiv preprint arXiv:1912.02736*, 2019.

[28] O. Mazhelis and S. Puuronen, "Comparing classifier combining techniques for mobile-masquerader detection," in *The Second International Conference on Availability, Reliability and Security (ARES'07)*, IEEE, 2007, pp. 465–472.

[29] J. Ryan, M.-J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," in *Advances in neural information processing systems*, 1998, pp. 943–949.

[30] D. Ariu, G. Giacinto, and F. Roli, "Machine learning in computer forensics (and the lessons learned from machine learning in computer security)," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 99–104.

[31] "Desktop vs mobile vs tablet market share worldwide." (), [Online]. Available: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet.

[32] "Share of active phones with enabled biometrics in north america, western europe and asia pacific from 2016 to 2020." (), [Online]. Available: https://www.statista.com/statistics/1226088/north-america-western-europe-biometric-enabled-phones/.

[33] *Us v. moreland*, 2011.

[34] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, "Benchmarking touchscreen biometrics for mobile authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2720–2733, 2018.

[35] P. Bhatt and P. H. Rughani, "Machine learning forensics: A new branch of digital forensics.," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 8, 2017.

[36] A. M. Qadir and A. Varol, "The role of machine learning in digital forensics," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, 2020, pp. 1–5.

[37] A. Guarino, "Digital forensics as a big data challenge," in *ISSE 2013 securing electronic business processes*, Springer, 2013, pp. 197–203.

[38] S. Qadir and B. Noor, "Applications of machine learning in digital forensics," in *2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, IEEE, 2021, pp. 1–8.

[39] E. S. Imsand, *Applications of GUI usage analysis*. ProQuest, 2008.

[40] A. I. Awad and A. E. Hassanien, "Impact of some biometric modalities on forensic science," *Studies in Computational Intelligence*, vol. 555, pp. 47–62, Jan. 2014. DOI: 10.1007/978-3-319-05885-6-3.

[41] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Statistical science*, pp. 58–74, 2001.

[42] T. Mitchell, "Machine learning," 1997.

[43] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 3, pp. 165–179, 2007.

[44] R. V. Yampolskiy, "Human computer interaction based intrusion detection," in *Information Technology, 2007. ITNG'07. Fourth International Conference on*, IEEE, 2007, pp. 837–842.

[45] R. W. Hammon and J. R. Young, *Method and apparatus for verifying an individual's identity*, US Patent 4,805,222, Feb. 1989.

[46] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, "Authentication by keystroke timing: Some preliminary results," DTIC Document, Tech. Rep., 1980.

[47] M. Obaidat and B. Sadoun, "Keystroke dynamics based authentication," in *Biometrics*, Springer, 1996, pp. 213–229.

[48] J. Zou, Y. Han, and S.-S. So, "Overview of artificial neural networks," *Artificial Neural Networks*, pp. 14–22, 2008.

[49] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[50] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*, 1. Springer, 2009, vol. 2.

[51] T. Qin, "Machine learning basics," in *Dual Learning*, Springer, 2020, pp. 11–23.

[52] I. Goodfellow, Y. Bengio, and A. Courville, "Machine learning basics," *Deep learning*, vol. 1, no. 7, pp. 98–164, 2016.

[53] P. Probst, A.-L. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1934–1965, 2019.

[54] T. O. Ayodele, "Machine learning overview," *New Advances in Machine Learning*, vol. 2, 2010.

[55] Q. Liu and Y. Wu, "Supervised learning," Jan. 2012. DOI: 10.1007/978-1-4419-1428-6_451.

[56] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[57] L. Rokach, "Ensemble-based classifiers," *Artificial intelligence review*, vol. 33, no. 1, pp. 1–39, 2010.

[58] F. Cohen, "Toward a science of digital forensic evidence examination," in *IFIP International Conference on Digital Forensics*, Springer, 2010, pp. 17–35.

[59] W. E. Boebert, "A survey of challenges in attribution," in *Proceedings of a workshop on Deterring CyberAttacks*, 2010, pp. 41–54.

[60] E. Casey, *Handbook of computer crime and investigation p116 london*, 2002.

[61] M. Rogers, "Computer forensics: Steps toward defining a common body of knowledge," in *Information Protection Association of Manitoba conference. Winnipeg Manitoba*, 2002.

[62] D. Ernsberger, R. A. Ikuesan, S. H. Venter, and A. Zugenmaier, "A web-based mouse dynamics visualization tool for user attribution in digital forensic readiness," in *International Conference on Digital Forensics and Cyber Crime*, Springer, 2017, pp. 64–79.

[63] J. T. Hackos and J. Redish, *User and task analysis for interface design*. Wiley New York, 1998, vol. 1.

[64] C. Steel, "Idiographic digital profiling: Behavioral analysis based on digital footprints," *Journal of Digital Forensics, Security and Law*, vol. 9, no. 1, pp. 7–18, 2014.

[65] T. Krone, "A typology of online child pornography offending.," *Trends and issues in crime and criminal justice*, no. 279, pp. 1–6, 2004.

[66] N. Nykodym, R. Taylor, and J. Vilela, "Criminal profiling and insider cyber crime," *Computer Law & Security Review*, vol. 21, no. 5, pp. 408–414, 2005.

[67] J. E. Douglas, R. K. Ressler, A. W. Burgess, and C. R. Hartman, "Criminal profiling from crime scene analysis," *Behavioral Sciences & the Law*, vol. 4, no. 4, pp. 401–421, 1986.

[68] N. Al Mutawa, J. Bryce, V. N. Franqueira, and A. Marrington, "Forensic investigation of cyberstalking cases using behavioural evidence analysis," *Digital investigation*, vol. 16, S96–S103, 2016.

[69] M. K. Rogers and K. C. Seigfried-Spellar, "Using internet artifacts to profile a child pornography suspect," 2014.

[70] R. J. Riding and E. Sadler-Smith, "Cognitive style and learning strategies: Some implications for training design," *International Journal of training and Development*, vol. 1, no. 3, pp. 199–208, 1997.

[71] A. Bandura, "Social cognitive theory of personality," *Handbook of personality*, vol. 2, pp. 154–96, 1999.

[72] I. R. Adeyemi, S. Abd Razak, M. Salleh, and H. S. Venter, "Leveraging human thinking style for user attribution in digital forensic process," 2017.

[73] U. K. Gaikwad and S. S. Sane, "Effective classifier for user's behavioral profile classification," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, 2014.

[74] T. F. Lunt, D. E. Denning, R. R. Schell, M. Heckman, and W. R. Shockley, "The seaview security model," *IEEE Transactions on software engineering*, vol. 16, no. 6, pp. 593–607, 1990.

[75] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proceedings of the 20th National Information Systems Security Conference*, Baltimore, USA, vol. 377, 1997, pp. 366–380.

[76] T. Rybak and R. Mosdorf, "Computer users activity analysis using recurrence plot," in *Biometrics and Kansei Engineering, 2009. ICBAKE 2009. International Conference on*, IEEE, 2009, pp. 189–194.

[77] L. Li and C. N. Manikopoulos, "Windows nt one-class masquerade detection," in *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC*, IEEE, 2004, pp. 82–87.

[78] Y. Song, M. B. Salem, S. Hershkop, and S. J. Stolfo, "System level user behavior biometrics using fisher features and gaussian mixture models," in *Security and Privacy Workshops (SPW), 2013 IEEE*, IEEE, 2013, pp. 52–59.

[79] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[80] W. Hu, Y. Liao, and V. R. Vemuri, "Robust support vector machines for anomaly detection in computer security.," in *ICMLA*, 2003, pp. 168–174.

[81] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of svm and ann for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.

[82] W. Wang, X. Zhang, and S. Gombault, "Constructing attribute weights from computer audit data for effective intrusion detection," *Journal of Systems and Software*, vol. 82, no. 12, pp. 1974–1981, 2009.

[83] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie, "Mining individual life pattern based on location history," in *2009 tenth international conference on mobile data management: Systems, services and middleware*, IEEE, 2009, pp. 1–10.

[84] A. Grillo, A. Lentini, G. Me, and M. Ottoni, "Fast user classifying to establish forensic analysis priorities," in *IT Security Incident Management and IT Forensics, 2009. IMF'09. Fifth International Conference on*, IEEE, 2009, pp. 69–77.

[85] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Behaviour profiling for transparent authentication for mobile devices," 2011.

[86] H. Zhang, Z. Yan, J. Yang, E. M. Tapia, and D. J. Crandall, "Mfingerprint: Privacy-preserving user modeling with multimodal mobile device footprints," in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, Springer, 2014, pp. 195–203.

[87] M. Singh, B. Mehtre, and S. Sangeetha, "User behavior profiling using ensemble approach for insider threat detection," in *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, IEEE, 2019, pp. 1–8.

[88] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, IEEE, 1996, pp. 120–128.

[89] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," in *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*, IEEE, 2002, pp. 219–228.

[90] K. H. Yung, "Using feedback to improve masquerade detection," in *Applied Cryptography and Network Security*, Springer, 2003, pp. 48–62.

[91] H.-C. Wu and S.-H. Huang, "User behavior analysis in masquerade detection using principal component analysis," in *Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on*, IEEE, vol. 1, 2008, pp. 201–206.

[92] C. Y. Shim, J. Y. Kim, and R. E. Gantenbein, "Practical user identification for masquerade detection," in *World Congress on Engineering and Computer Science 2008, WCECS'08. Advances in Electrical and Electronics Engineering-IAENG Special Edition of the*, IEEE, 2008, pp. 47–51.

[93] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing*, vol. 11, no. 2, pp. 1565–1573, 2011.

[94] D. Umphress and G. Williams, "Identity verification through keyboard characteristics," *International journal of man-machine studies*, vol. 23, no. 3, pp. 263–273, 1985.

[95] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Communications of the ACM*, vol. 33, no. 2, pp. 168–176, 1990.

[96] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation computer systems*, vol. 16, no. 4, pp. 351–359, 2000.

[97] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, ACM, 2004, pp. 1–8.

[98] A. Garg, R. Rahalkar, S. Upadhyaya, and K. Kwiaty, "Profiling users in gui based systems for masquerade detection," in *Information Assurance Workshop, 2006 IEEE*, IEEE, 2006, pp. 48–54.

[99] W. N. Bhukya, S. K. Kommuru, and A. Negi, "Masquerade detection based upon gui user profiling in linux systems," in *Advances in Computer Science–ASIAN 2007. Computer and Network Security*, Springer, 2007, pp. 228–239.

[100] A. Garg, S. Upadhyayal, and K. Kwiat, "A user behavior monitoring and profiling scheme for masquerade detection," *Handbook of Statistics: Machine Learning: Theory and Applications*, vol. 31, p. 353, 2013.

[101] A. Alzubaidi and J. Kalita, "Authentication of smartphone users using behavioral biometrics," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1998–2026, 2016.

[102] C. Shen, Z. Cai, X. Guan, Y. Du, and R. A. Maxion, "User authentication through mouse dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 16–30, 2012.

[103] E. S. Imsand and J. Hamilton, "Masquerade detection through guiid," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, IEEE, 2008, pp. 1–5.

[104] B. Camiña, R. Monroy, L. A. Trejo, and E. Sánchez, "Towards building a masquerade detection method based on user file system navigation," in *Mexican International Conference on Artificial Intelligence*, Springer, 2011, pp. 174–186.

[105] H. Saljooghinejad and W. N. Rathore, "Multi application user profiling for masquerade attack detection," in *Advances in Computing and Communications*, Springer, 2011, pp. 676–684.

[106] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[107] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE transactions on information forensics and security*, vol. 8, no. 1, pp. 136–148, 2012.

[108] H. Xu, Y. Zhou, and M. R. Lyu, "Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones," in *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*, 2014, pp. 187–198.

[109] M. Antal, Z. Bokor, and L. Z. Szabó, "Information revealed from scrolling interactions on mobile devices," *Pattern Recognition Letters*, vol. 56, pp. 7–13, 2015.

[110] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *2013 IEEE sixth international conference on biometrics: theory, applications and systems (BTAS)*, IEEE, 2013, pp. 1–8.

[111] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa, "Touch gesture-based active user authentication using dictionaries," in *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, 2015, pp. 207–214.

[112] S. Mondal and P. Bours, "Swipe gesture based continuous authentication for mobile devices," in *2015 International Conference on Biometrics (ICB)*, IEEE, 2015, pp. 458–465.

[113] R. Murmuria, A. Stavrou, D. Barbará, and D. Fleck, "Continuous authentication on mobile devices using power consumption, touch gestures and physical movement of users," in *International Symposium on Recent Advances in Intrusion Detection*, Springer, 2015, pp. 405–424.

[114] R. Kumar, V. V. Phoha, and A. Serwadda, "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns," in *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)*, IEEE, 2016, pp. 1–8.

[115] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, "Active user authentication for smartphones: A challenge data set and benchmark results," in *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)*, IEEE, 2016, pp. 1–8.

[116] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion, "Performance analysis of touch-interaction behavior for active smartphone authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 498–513, 2015.

[117] Z. Sitová, J. Šeděnka, Q. Yang, *et al.*, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2015.

[118] R. Kumar, P. P. Kundu, and V. V. Phoha, "Continuous authentication using one-class classifiers and their fusion," in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, IEEE, 2018, pp. 1–8.

[119] O. De Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *ACM Sigmod Record*, vol. 30, no. 4, pp. 55–64, 2001.

[120] C. E. Chaski, "Who's at the keyboard? authorship attribution in digital evidence investigations," *International Journal of Digital Evidence*, vol. 4, no. 1, pp. 1–13, 2005.

[121] T. Kucukyilmaz, B. B. Cambazoglu, C. Aykanat, and F. Can, "Chat mining: Predicting user and message attributes in computer-mediated communication," *Information Processing & Management*, vol. 44, no. 4, pp. 1448–1466, 2008.

[122] X. Cheng and J. Chen, "Modeling user interests based on cloud model for masquerade detection," in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, IEEE, 2009, pp. 1–4.

[123] G. Pannell and H. Ashman, "Anomaly detection over user profiles for intrusion detection," 2010.

[124] C. Colombini and A. Colella, "Digital profiling: A computer forensics approach," in *International Conference on Availability, Reliability, and Security*, Springer, 2011, pp. 330–343.

[125] C. Shen, Z. Cai, R. A. Maxion, and X. Guan, "On user interaction behavior as evidence for computer forensic analysis," in *International Workshop on Digital Watermarking*, Springer, 2013, pp. 221–231.

[126] J. Govindaraj, R. Verma, and G. Gupta, "Analyzing mobile device ads to identify users," in *IFIP International Conference on Digital Forensics*, Springer, 2016, pp. 107–126.

[127] N. Clarke, F. Li, and S. Furnell, "A novel privacy preserving user identification approach for network traffic," *computers & security*, vol. 70, pp. 335–350, 2017.

[128] T. Xue, Q. Li, P. Zhang, Z. Chen, P. Feng, and N. Luo, "Computer evidence analysis technology based on weighted frequent pattern growth algorithm," in *International Conference on Artificial Intelligence and Security*, Springer, 2019, pp. 430–441.

[129] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[130] J. Han and M. Kamber, *Pei. data mining concepts and techniques*, 2011.

[131] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," *Institute for Signal and information Processing*, 1998.

[132] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

[133] E. S. Imsand, D. Garrett, and J. Hamilton, "User identification using gui manipulation patterns and artificial neural networks," in *Computational Intelligence in Cyber Security, 2009. CICS'09. IEEE Symposium on*, IEEE, 2009, pp. 130–135.

[134] E. C. Ates, E. Bostanci, and M. S. Guzel, "Comparative performance of machine learning algorithms in cyberbullying detection: Using turkish language preprocessing techniques," *arXiv preprint arXiv:2101.12718*, 2021.

[135] R. Rowlingson *et al.*, "A ten step process for forensic readiness," *International Journal of Digital Evidence*, vol. 2, no. 3, pp. 1–28, 2004.

[136] S.-O. Act, "Sarbanes-oxley act," *Washington DC*, 2002.

[137] S. Quinn, "Examining the state of preparedness of information technology management in new zealand for events that may require forensic analysis," *Digital Investigation*, vol. 2, no. 4, pp. 276–280, 2005.

[138] *Merriam-Webster Dictionary*. Encyclopaedia Britannica, 2015, forensic. [Online]. Available: http://www.merriam-webster.com/dictionary/forensic.

[139] S. L. Garfinkel, "Digital forensics modern crime often leaves an electronic trail. finding and preserving that evidence requires careful methods as well as technical skill," *American Scientist*, vol. 101, no. 5, pp. 370–377, 2013.

[140] E. Casey, "Error, uncertainty, and loss in digital evidence," *International Journal of Digital Evidence*, vol. 1, no. 2, pp. 1–45, 2002.

[141] M. Kwan, K.-P. Chow, F. Law, and P. Lai, "Reasoning about evidence using bayesian networks," in *Advances in Digital Forensics IV*, Springer, 2008, pp. 275–289.

[142] P. Sommer, "Intrusion detection systems as evidence," *Computer Networks*, vol. 31, no. 23, pp. 2477–2487, 1999.

[143] D. J. Ryan and G. Shpantzer, "Legal aspects of digital forensics," in *Proceedings: Forensics Workshop*, 2002.

[144] P. C. Giannelli, "The admissibility of novel scientific evidence: Frye v. united states, a half-century later," *Columbia Law Review*, pp. 1197–1250, 1980.

[145] *Daubert v. merrell dow pharmaceuticals, inc.* 1993.

[146] *General electric co. v. joiner*, 1997.

[147] *Kumho tire co. v. carmichael*, 1999.

[148] M. H. Graham, *Handbook of Federal Evidence: Rules 701 to 801*. Thomson/West, 2011, vol. 3.

[149] D. Bosco, A. Zappalà, and P. Santtila, "The admissibility of offender profiling in courtroom: A review of legal issues and court opinions," *International journal of law and psychiatry*, vol. 33, no. 3, pp. 184–191, 2010.

[150] *Pennell v. state*, 1992.

[151] *State v. fortin*, 2000.

[152] C. Liu, A. Singhal, and D. Wijesekera, "Relating admissibility standards for digital evidence to attack scenario reconstruction," *Journal of Digital Forensics, Security and Law*, vol. 9, no. 2, pp. 181–196, 2014.

[153] C. Weiss, "Expressing scientific uncertainty," *Law, Probability and Risk*, vol. 2, no. 1, pp. 25–46, 2003.

[154] D. L. Faigman, D. Kaye, M. Saks, J. Sanders, and E. Cheng, *Modern Scientific Evidence*. Thomson/West, 2006.

[155] E. Zimmer, J. Lindemann, D. Herrmann, and H. Federrath, "Catching inside attackers: Balancing forensic detectability and privacy of employees," in *International Workshop on Open Problems in Network Security*, Springer, 2015, pp. 43–55.

[156] E. Lundin and E. Jonsson, "Anomaly-based intrusion detection: Privacy concerns and other problems," *Computer networks*, vol. 34, no. 4, pp. 623–640, 2000.

[157] K. Lakkaraju and A. Slagell, "Evaluating the utility of anonymized network traces for intrusion detection," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, 2008, pp. 1–8.

[158] S. Köpsell, R. Wendolsky, and H. Federrath, "Revocable anonymity," in *International Conference on Emerging Trends in Information and Communication Security*, Springer, 2006, pp. 206–220.

[159] D. W. Gresty, D. Gan, G. Loukas, and C. Ierotheou, "Facilitating forensic examinations of multi-user computer environments through session-to-session analysis of internet history," *Digital Investigation*, vol. 16, S124–S133, 2016.

[160] S. Dua and X. Du, *Data mining and machine learning in cybersecurity*. CRC press, 2011.

[161] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[162] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017.

[163] C. Machinery, "Computing machinery and intelligence-am turing," *Mind*, vol. 59, no. 236, p. 433, 1950.

[164] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[165] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[166] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, IEEE Computer Society, 1979, pp. 421–427.

[167] G. DeJong, "Generalizations based on explanations.," in *IJCAI*, vol. 81, 1981, pp. 67–69.

[168] T. J. Sejnowski and C. R. Rosenberg, "Nettalk: A parallel network that learns to read aloud. johns hopkins university electrical engineering and computer science technical report," JHU/EECS–86/01.[rTS], Tech. Rep., 1986.

[169] J. Han, Y. Cai, and N. Cercone, "Data-driven discovery of quantitative rules in relational databases," *IEEE transactions on Knowledge and Data Engineering*, vol. 5, no. 1, pp. 29–40, 1993.

[170] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[171] D. A. Ferrucci, "Introduction to "this is watson"," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 1–1, 2012.

[172] M. Helms, S. V. Ault, G. Mao, and J. Wang, "An overview of google brain and its applications," in *Proceedings of the 2018 International Conference on Big Data and Education*, 2018, pp. 72–75.

[173] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[174] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2017, pp. 3–18.

[175] M. Li, D. G. Andersen, J. W. Park, *et al.*, "Scaling distributed machine learning with the parameter server," in *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, 2014, pp. 583–598.

[176] "2.4.2. machine learning 101: General concepts," scikit-learn. (2021), [Online]. Available: https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/tutorial/text_analytics/general_concepts.html.

[177] C. Hayashi, "What is data science? fundamental concepts and a heuristic example," in *Data science, classification, and related methods*, Springer, 1998, pp. 40–51.

[178] M. A. Hernán, J. Hsu, and B. Healy, "A second chance to get causal inference right: A classification of data science tasks," *Chance*, vol. 32, no. 1, pp. 42–49, 2019.

[179] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised machine learning: A brief primer," *Behavior Therapy*, vol. 51, no. 5, pp. 675–687, 2020.

[180] R. B. Babu, G. Snehal, and P. A. S. Kiran, "Detection of crimes using unsupervised learning techniques," *Aptikom Journal on Computer Science and Information Technologies*, vol. 2, no. 1, pp. 8–11, 2017.

[181] K. Franke, E. Hjelmås, and S. D. Wolthusen, "Advancing digital forensics," in *Information Assurance and Security Education and Training*, Springer, 2013, pp. 288–295.

[182] S. Costantini, G. De Gasperis, and R. Olivieri, "Digital forensics and investigations meet artificial intelligence," *Annals of Mathematics and Artificial Intelligence*, vol. 86, no. 1, pp. 193–229, 2019.

[183] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview," in *Journal of physics: conference series*, IOP Publishing, vol. 1142, 2018, p. 012 012.

[184] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003.

[185] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 375–381, 2003.

[186] S. Markovitch and D. Rosenstein, "Feature generation using general constructor functions," *Machine Learning*, vol. 49, no. 1, pp. 59–98, 2002.

[187] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[188] M. A. Amin and H. Yan, "High speed detection of retinal blood vessels in fundus image using phase congruency," *Soft Comput.*, vol. 15, pp. 1217–1230, Jun. 2011. DOI: 10.1007/s00500-010-0574-2.

[189] R. Muhamedyev, "Machine learning methods: An overview," *Computer modelling & new technologies*, vol. 19, no. 6, pp. 14–29, 2015.

[190] V. Morde, *Xgboost algorithm: Long may she reign!* Apr. 2019. [Online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

[191] C. Mi, F. Huettmann, Y. Guo, X. Han, and L. Wen, "Why choose random forest to predict rare species distribution with few samples in large undersampled areas? three asian crane species models provide supporting evidence," *PeerJ*, vol. 5, e2849, 2017.

[192] J. P. Mueller and L. Massaron, *Machine learning for dummies*. John Wiley & Sons, 2021.

[193] S. Oppel, A. Meirinho, I. Ramírez, *et al.*, "Comparison of five modelling techniques to predict the spatial distribution and abundance of seabirds," *Biological Conservation*, vol. 156, pp. 94–104, 2012.

[194] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," 1984.

[195] J. F. Elder IV, "The generalization paradox of ensembles," *Journal of Computational and Graphical Statistics*, vol. 12, no. 4, pp. 853–864, 2003.

[196] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia*, Springer, 2008, pp. 21–49.

[197] F. Huettmann, "Boosting, bagging and ensembles in the real world: An overview, some explanations and a practical synthesis for holistic global wildlife conservation applications based on machine learning with decision trees," *Machine Learning for Ecology and Sustainable Natural Resource Management*, pp. 63–83, 2018.

[198] S. Malik, R. Harode, and A. Kunwar, "Xgboost: A deep dive into boosting (introduction documentation)," *Simon Fraser University: Burnaby, BC, Canada*, 2020.

[199] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[200] H. Drucker, "Improving regressors using boosting techniques," in *ICML*, Citeseer, vol. 97, 1997, pp. 107–115.

[201] A. Krogh, J. Vedelsby, *et al.*, "Neural network ensembles, cross validation, and active learning," *Advances in neural information processing systems*, vol. 7, pp. 231–238, 1995.

[202] T. Hastie, R. Tibshirani, and J. Friedman, "Ensemble learning," in *The elements of statistical learning*, Springer, 2009, pp. 605–624.

[203] J. H. Zar, "Biostatistical analysis pearson prentice-hall," *Upper Saddle River, NJ*, 2010.

[204] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[205] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[206] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble machine learning*, Springer, 2012, pp. 157–175.

[207] R. E. Schapire, "The boosting approach to machine learning: An overview," *Nonlinear estimation and classification*, pp. 149–171, 2003.

[208] M. Sewell, "Ensemble learning," *RN*, vol. 11, no. 02, pp. 1–34, 2008.

[209] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[210] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[211] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[212] "Amazon sagemaker developers guide," Amazon, Tech. Rep., 2021. [Online]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-dg.pdf#xgboost-HowItWorks.

[213] J. J. Li and X. Tong, "Statistical hypothesis testing versus machine learning binary classification: Distinctions and guidelines," *Patterns*, vol. 1, no. 7, p. 100 115, 2020, ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2020.100115. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389920301562.

[214] M. Law, "Reduce, reuse, recycle: Issues in the secondary use of research data," *IASSIST Quarterly*, vol. 29, no. 1, pp. 5–5, 2006.

[215] K. Bennett, *Https://www.ivoryresearch.com/blog/write-your-dissertation-using-only-secondary-research/*, Nov. 2020. [Online]. Available: https://www.ivoryresearch.com/blog/write-your-dissertation-using-only-secondary-research/.

[216] M. P. Johnston, "Secondary data analysis: A method of which the time has come," *Qualitative and quantitative methods in libraries*, vol. 3, no. 3, pp. 619–626, 2017.

[217] B. G. Glaser, "Retreading research materials: The use of secondary analysis by the independent researcher," *American Behavioral Scientist*, vol. 6, no. 10, pp. 11–14, 1963.

[218] T. Long-Sutehall, M. Sque, and J. Addington-Hall, "Secondary analysis of qualitative data: A valuable method for exploring sensitive issues with an elusive population?" *Journal of Research in Nursing*, vol. 16, no. 4, pp. 335–344, 2011.

[219] P. S. Hinds, R. J. Vogel, and L. Clarke-Steffen, "The possibilities and pitfalls of doing a secondary analysis of a qualitative data set," *Qualitative health research*, vol. 7, no. 3, pp. 408–424, 1997.

[220] J. Heaton, "Secondary analysis of qualitative data," 1998.

[221] K. E. Rudestam and R. R. Newton, "Surviving your dissertation: A comprehensive guide to content and process," 2014.

[222] C. Price, *An introduction to public use data sets*, Dec. 2015. [Online]. Available: https://research.phoenix.edu/blog/introduction-public-use-data-sets.

[223] T. Ritsema. "Free data: Using publicly available datasets for your dissertation or research project," PAEAonline. (2014), [Online]. Available: https://www.youtube.com/watch?v=2DYXvjqMexI.

[224] A. Dale, S. Arber, and M. Procter, *Doing secondary analysis*. Unwin Hyman, 1988.

[225] S. E. Fienberg, M. E. Martin, and M. L. Straf, *Sharing research data*. National Academy Press, 1985.

[226] A. K. Belman, L. Wang, S. S. Iyengar, *et al.*, *Su-ais bb-mas (syracuse university and assured information security - behavioral biometrics multi-device and multi-activity data from same users) dataset*, 2019. DOI: 10.21227/rpaz-0h66. [Online]. Available: https://dx.doi.org/10.21227/rpaz-0h66.

[227] F. Insa, "The admissibility of electronic evidence in court (aeec): Fighting against high-tech crime—results of a european study," *Journal of Digital Forensic Practice*, vol. 1, no. 4, pp. 285–289, 2007.

[228] E. S. Imsand, D. Garrett, and J. Hamilton, "User identification using gui manipulation patterns and artificial neural networks," in *Computational Intelligence in Cyber Security, 2009. CICS'09. IEEE Symposium on*, IEEE, 2009, pp. 130–135.

[229] N. O. F. Elssied, O. Ibrahim, and A. H. Osman, "A novel feature selection based on one-way anova f-test for e-mail spam classification," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 3, pp. 625–638, 2014.

[230] C.-J. Tu, L.-Y. Chuang, J.-Y. Chang, C.-H. Yang, *et al.*, "Feature selection using pso-svm," *International Journal of Computer Science*, 2007.

[231] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural computing and applications*, vol. 31, no. 1, pp. 171–188, 2019.

[232] S. Shakeela, N. S. Shankar, P. M. Reddy, T. K. Tulasi, and M. M. Sai, "Optimal ensemble learning based on distinctive feature selection by univariate anova-f statistics for ids," *International Journal of Electronics and Telecommunications*, vol. 67, no. 2, pp. 267–275, 2021.

[233] M. S. K. Inan, R. E. Ulfath, F. I. Alam, F. K. Bappee, and R. Hasan, "Improved sampling and feature selection to support extreme gradient boosting for pcos diagnosis," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2021, pp. 1046–1050.

[234] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, 2016.

[235] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[236] Ethen, *Model selection*. [Online]. Available: http://ethen8181.github.io/machine-learning/model_selection/model_selection.html.

[237] J. Brownlee, *Tune hyperparameters for classification machine learning algorithms*, Dec. 2019. [Online]. Available: https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/.

[238] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for nas," *arXiv preprint arXiv:1912.06059*, 2019.

[239] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated machine learning*, Springer, Cham, 2019, pp. 3–33.

[240] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[241] P. Banerjee, *A guide on xgboost hyperparameters tuning*, 2021. [Online]. Available: https://www.kaggle.com/prashant111/a-guide-on-xgboost-hyperparameters-tuning.

[242] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: Classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128–138, 2017.

[243] F. Deng, J. Huang, X. Yuan, C. Cheng, and L. Zhang, "Performance and efficiency of machine learning algorithms for analyzing rectangular biomedical data," *Laboratory Investigation*, vol. 101, no. 4, pp. 430–441, 2021.

[244] W. N. Bhukya and S. K. Banothu, "Investigative behavior profiling with one class svm for computer forensics," in *Multi-disciplinary Trends in Artificial Intelligence*, Springer, 2011, pp. 373–383.

[245] M. Mohlala, A. R. Ikuesan, and H. S. Venter, "User attribution based on keystroke dynamics in digital forensic readiness process," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, IEEE, 2017, pp. 124–129.

[246] B. E. Endicott-Popovsky and D. A. Frincke, "Adding the fourth" r": A systems approach to solving the hacker's arms race," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2006.

[247] CENELEC, "European standard en 50133-1: Alarm systems. access control systems for use in security applications. part 1: System requirements," 2002.

[248] A. Jarrett and K.-K. R. Choo, "The impact of automation and artificial intelligence on digital forensics," *Wiley Interdisciplinary Reviews: Forensic Science*, vol. 3, no. 6, e1418, 2021.

[249] D. Wilson-Kovacs, "Digital media investigators: Challenges and opportunities in the use of digital forensics in police investigations in england and wales," *Policing: An International Journal*, 2021.

[250] G. Humphries, R. Nordvik, H. Manifavas, P. Cobley, and M. Sorell, "Law enforcement educational challenges for mobile forensics," *Forensic Science International: Digital Investigation*, vol. 38, p. 301 129, 2021.

[251] S. Woolgar and B. Latour, *Laboratory of Life: The Construction of Scientific Facts*. Princenton University, 1986.

[252] D.-Y. Kao, Y.-T. Chao, F. Tsai, and C.-Y. Huang, "Digital evidence analytics applied in cybercrime investigations," in *2018 IEEE Conference on Application, Information and Network Security (AINS)*, IEEE, 2018, pp. 111–116.

[253] D. G. Brizan, A. Goodkind, P. Koch, K. Balagani, V. V. Phoha, and A. Rosenberg, "Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics," *International Journal of Human-Computer Studies*, vol. 82, pp. 57–68, 2015.

A. DETAILS OF PARTICIPANT ACTIVITIES

This appendix provides the details of the activities of each participant, as outlined in Belman, Wang, Iyengar, *et al.* [27, p. 12].

A.1 FIXED TEXT SENTENCES

- "this is a test to see if the words that i type are unique to me. there are two sentences in this data sample."
- "second session will have different set of lines. carefully selected not to overlap with the first collection phase."

A.2 COGNITIVE LOADS

Task	Level	Required activity
Remember	1	Retrieve knowledge from long-term memory to explain
Understand	2	Explain, summarize or interpret
Apply	3	Apply, execute or implement
Analyze	4	Organize or break material into constituent parts
Evaluate	5	Critique or make judgments based on criteria
Create	6	Generate, plan or put elements together

Figure A.1. Description of Cognitive Loads as described by Brizan, Goodkind, Koch, *et al.* [253] and taken from [27]

A.3 SHOPPING LIST

- Mountain Bike
- Plane tickets from Syracuse, New York to Los Angeles [1 week from today, Coach Seat]
- Bathing Suit (male or female)
- Converse All Stair Hiking Boots
- 24 Pack of Gatorade (24-oz)
- Ground Transportation (Train, taxi, Bus) from Los Angeles to San Diego [2 weeks from Today]

A.4 FREE TEXT QUESTIONS

1. List some of the things that you like about Syracuse University.
2. Which internet browser do you typically use (e.g, Google Chrome, Internet Explorer, Mozilla Firefox, etc.)?
3. What improvements would you like to see in that browser?
4. If you were to draw a picture of Syracuse University, what objects would you include in it?
5. What is your favorite vacation spot? Why do you like to visit there?
6. Give step-by-step driving directions to your favorite restaurant in the Syracuse Area, starting from your dorm room/ home.
7. What hobbies or activities are you involved with outside of school/work? Why?
8. Discuss step-by-step instructions for making your favorite type of sandwich. Write them so that the person who has never done this before can follow your instructions.

9. What television programs do you watch for the news and current events? Why? If you do not watch anything on TV, what is your primary source for news information? What do you like about it?

10. Give a brief, but sufficiently detailed plot description of your favorite book, story, or movie.

11. What social networking websites do you use? What do you like or dislike about these websites? If you do not use any social network, how do you stay in touch with your friends and acquaintances. Why do you not use social networking websites? Who is your favorite actor, actress, singer, comedian, or TV personality? What do you like about them?

A.5 FREE TEXT AND MULTIPLE CHOICE QUESTIONS ON PHONE

1. What type of Smartphone do you typically use?

    - Android
    - iPhone
    - Windows
    - None
    - Other

2. Which best describes you?

    - I have a very active imagination.
    - I take my civic duties, such as voting, seriously.
    - I crave excitement.
    - I would rather cooperate with people than compete with them.
    - I am a worrier.
    - I do not like to talk about myself

3. Of the courses you have taken in college, which was your favorite and why?

4. Think about a class that you did not enjoy. What improvements would you like to see to make the course better?

5. Re-read Question 2 and the responses. Which response do you feel is least applicable to you and why?

6. Do you intend to pursue an advanced degree (e.g., Master's or Ph.D.)? Why or why not?

7. Find a rule that makes four of the five options alike.

8. Select the option that does not follow this rule:

    - 11.28.45.62
    - 200.217.234.251
    - 192.209.226.243
    - 214.231.248.265
    - 111.127.140.165

9. (Horizontal swipes) Review Question 7 and the answer that you chose. Why was the rule you found/why did you select your answer?

10. What are the topics of Question 6 and Question 10?

11. Give step-by-step directions from this lab space to your dorm room, making specific notes of each time you would descend or ascend stairs.

A.6 FREE TEXT AND MULTIPLE CHOICE QUESTIONS ON TABLET

1. What type of Tablet do you typically use?

    - Android
    - iPad

- Windows
- Amazon Fire
- None
- Other

2. Which best describes you?
    - I don't mind bragging about my skills and accomplishments.
    - I often forget to or neglect to put things back where I found them.
    - I am dominant, forceful, and/or assertive.
    - I am easy-going and lackadaisical.
    - I am set in my ways.
    - I shy away from crowds.
3. What is your ideal job after graduation? Why?
4. Why did you decide to attend Syracuse University?
5. Re-read Question 2 and the responses. Which response do you feel is least applicable to you and why?
6. If all mangoes are golden in color and no golden colored things are cheap, which of the following is true?
7. A. All mangoes are cheap.
8. B. Golden-colored mangoes are not cheap.
9. Either A or B are true.
10. Both A and B are true.
11. Neither A or B are true.
12. Review Question 6 and the answer that you chose.

13. Why did you select your answer?

14. (Horizontal swipes) If Question 6 was changed to read "If some mangoes are golden in color and no golden colored things are cheap", which answer would be correct and why?

15. What are your thoughts on the current U.S. president?

16. Which policies, if any, would you like to see changed and how?

17. Discuss step-by-step the process for sending an email from your Syracuse email account. Write these instructions such that a person who has never done this before can follow your instructions.

18. Please provide any comments that you have about the survey or the experiment thus far.

Table A.1. Sample of combined engineered dataset

	key1	key2	time	key1.1	bf key2.1	time.1	key1.2	key2.2	time.2	key1.3	key2.3	time.3	key	keyhold	Distance	velocity	TimeBwClicks	user
0	0.0	0.0	546.0	0.0	0.0	718.0	0.0	0.0	702.0	0.0	0.0	874.0	11.0	141.0	1.077005	1.302620	1.277078	1
1	7.0	4.0	94.0	7.0	4.0	328.0	7.0	4.0	250.0	7.0	4.0	484.0	5.0	187.0	0.092515	0.476898	0.309678	1
2	6.0	1.0	358.0	6.0	1.0	514.0	6.0	1.0	468.0	6.0	1.0	624.0	0.0	156.0	0.667527	0.870704	0.776256	1
3	1.0	5.0	266.0	1.0	5.0	359.0	1.0	5.0	422.0	1.0	5.0	515.0	0.0	172.0	0.467144	0.542532	0.677804	1
4	6.0	1.0	78.0	6.0	1.0	171.0	6.0	1.0	156.0	6.0	1.0	249.0	11.0	156.0	0.057666	0.144492	0.108493	1