6-2014

User Identification and Authentication using Multi-Modal Behavioral Biometrics

Kyle O. Bailey
Air Force Institute of Technology

James S. Okolica
Air Force Institute of Technology

Gilbert L. Peterson
Air Force Institute of Technology

Recommended Citation

Bailey, K. O., Okolica, J. S., & Peterson, G. L. (2014). User identification and authentication using multi-modal behavioral biometrics. Computers & Security, 43, 77–89. https://doi.org/10.1016/j.cose.2014.03.005

User Identification and Authentication using Multi-Modal Behavioral Biometrics

Kyle O. Bailey, James S. Okolica, Gilbert L. Peterson

Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way, WPAFB, Ohio 45433
{*kyle.bailey, james.okolica, gilbert.peterson*}*@afit.edu*
Corresponding Author: Gilbert Peterson (937)-785-6565 x4281

Abstract

Biometric computer authentication has an advantage over password and access card authentication in that it is based on something you are, which is not easily copied or stolen. One way of performing biometric computer authentication is to use behavioral tendencies associated with how a user interacts with the computer. However, behavioral biometric authentication accuracy rates are worse than more traditional authentication methods. This article presents a behavioral biometric system that fuses user data from keyboard, mouse, and Graphical User Interface (GUI) interactions. Combining the modalities results in a more accurate authentication decision based on a broader view of the user's computer activity while requiring less user interaction to train the system than previous work. Testing over 31 users shows that fusion techniques significantly improve behavioral biometric authentication accuracy over single modalities on their own. Between the two fusion techniques presented, feature fusion and an ensemble based classification method, the ensemble method performs the best with a False Acceptance Rate (FAR) of 2.10% and a False Rejection Rate (FRR) 2.24%.

Keywords: Computer security, Behavioral Biometrics, Multi-modal fusion, Insider threat, Active Authentication

1. Introduction

Traditionally authentication is based on something you know and/or something you have. An example would be using a Common Access Card (CAC) and pin number or a username and password (Matyas and Zdenek, 2003). One downside however is that this type of authentication can be lost, stolen, or disclosed. It also does not truly identify the user as themselves, but instead by something they know or have. Biometric authentication is an emerging method of authentication that is based on something you are (Ahmed and Traore, 2007). There are two subsets of biometric authentication, physiological and behavioral. These authentication methods identify the user as themselves based on measurable physical or behavioral characteristics.

Physiological biometric authentication involves measuring physical characteristics of a person's body that make them unique. Physiological methods include fingerprint scanning, facial recognition, hand geometry recognition or retinal scans (Bhattacharyya et al., 2009). These methods have generally been more reliable and currently have a more successful implementation in the real world than behavioral techniques (Ahmed and Traore, 2007). One drawback of physical biometrics is that they require hardware to perform the biometric data collection. This hardware adds cost and another layer of complexity to the login process for the user. Another drawback is that all of the physical biometric methods still contain some type of error. Comparison testing by Bhattacharyya, et al. (Bhattacharyya et al., 2009), found that the iris scanner, with an Equal Error Rate (EER) of 0.01%, performed the best.

Behavioral biometric authentication is the process of measuring behavioral tendencies of a user resulting from both psychological and physiological differences from person to person. Behavioral methods include keystroke dynamics (Joyce and Gupta, 1990; Brown and Rogers, 1993; Monrose and Rubin, 1997; Gunetti and Picardi, 2005; Marsters, 2009), mouse dynamics (Ahmed and Traore, 2007; Shen et al., 2010; Zheng et al., 2011), voice recognition (Bhattacharyya et al., 2009), signature verification (Bhattacharyya et al., 2009) and Graphical User Interface (GUI) usage analysis (Gamboa and Fred, 2004; Imsand, 2008). Due to the variability of the human body and mind, the adoption of this type of biometrics has lagged behind physiological biometrics. However the use of keystrokes, mouse dynamics and Graphical User Interface (GUI) interaction for biometrics does not require extra hardware. The data collection uses software that gathers information from the existing keyboard, mouse and GUI messages sent by the installed operating system. A second benefit to usage based biometrics is that authentication can occur actively throughout a user's session as opposed to once during initial logon. This can prevent a user's session from being hijacked after the initial logon has occurred.

This article presents a behavioral biometric system that fuses user data from keyboard, mouse, and GUI interactions. The system collects user characteristics relating to the way a particular user interacts with the computer. This is done by monitoring a user's keystrokes, mouse movements, and GUI usage

patterns. Features are calculated on these actions. The features are then fused together by combining feature vectors.

Classification occurs on the data in both an identification (multi-class) and authentication (binary class) situation to simulate an active authentication scenario. Identification is the process of determining who the user is, while authentication is used to confirm the validity of that identity. Additionally, an Ensemble Based Decision Level (EBDL) fusion method is analyzed that first classifies on each modality alone and then generates a fusion of those results. EBDL fusion is a two step process. In the first step, multiple classifiers arrive at what each thinks is the correct answer. This may take the form of a single value or its score for each possible value. Then, in the second step, a second classifier fuses the outputs from each of these classifiers into a single answer. EBDL provides better generalizability and can handle the problems of too much or too little data (Polikar, 2006). From the experiments that follow, it is found that by using EBDL fusion, significant identification and active authentication improvements are achieved over each of the individual modalities on their own, and feature fusion.

2. Related Work

The idea of using keystroke and mouse dynamics as means to perform active authentication has been around for several decades. (Gaines et al., 1980), While both of these methods have seen a large amount of research, there has been less work done on combining these two techniques into one system. GUI usage analysis is a relatively young technique (Pusara, 2007) which analyzes how the user accomplishes a certain task within the operating system interface. For instance, GUI usage analysis differentiates between a user who click on the menu bars from one who uses hotkeys. This article considers the fusing of keystroke dynamics, mouse dynamics, and GUI usage to create a better method for actively authenticating users.

2.1. Testing Behavior Biometrics

Being able to quantify the effectiveness of the authentication technique is important. Techniques used to create the samples used for training and testing consists of calculating modality features over some range. Examples of some of the ranges used are the number of seconds of interaction (Gamboa and Fred, 2004), the number of mouse events (Zheng et al., 2011), the number of GUI events (Pusara, 2007), a set task (e.g., user name and password), and the number of keystrokes (Marsters, 2009).

The preponderance of previous work in active authentication has measured performance using the metrics of False Acceptance Rate (FAR), False Rejection Rate (FRR) and the Equal Error Rate (EER) on the ranged samples. Both FAR and FRR are reported as a percentage, and signify the percentage of times an impostor user is authenticated (FAR) or the percentage of time a legitimate user is denied access (FRR). The EER is the value where the FAR and FRR are equal. This point is determined by creating a curve for both FAR and FRR based on the Receiver Operating Characteristic (ROC) for the classification algorithm (Bhattacharyya et al., 2009).

There are signs this is changing. Bours (Bours, 2012) and Mondal (Mondal and Bours, 2013) have recently used a different method for measuring authentication performance. Rather than considering both FAR and FRR, Bours and Mondal have focused only on how long it takes for an impostor user to be discovered. After using training data to create the legitimate user templates, they use the legitimate user test data to set the thresholds such that the legitimate users are never falsely rejected. Then, they can measure how many events (e.g., keystrokes, mouse movements, etc.) it takes for the impostor's to be discovered.

Lastly, identification focused work (Garg et al., 2006; Monrose and Rubin, 1997) uses the accuracy of identification or detection as a measure for performance. Identification accuracy is recorded as the percentage of time the system can make a correct decision on the identity of the user.

2.2. Keystroke Dynamics

Gaines, et al. (Gaines et al., 1980), introduced the idea of using behavioral biometrics as a supplement to traditional authentication. Initially, keystroke timing data was used to supplement password entry (Gaines et al., 1980; Joyce and Gupta, 1990; Bleha et al., 1990; Brown and Rogers, 1993; Haider et al., 2000). This evolved into being able to analyze long structured text as a basis for authentication (Monrose and Rubin, 1997; Pusara, 2007), and finally long free text samples (Bergadano et al., 2002; Gunetti and Picardi, 2005; Marsters, 2009). Although the long free text better imitates free use, interest in keystroke timing to supplement password entry has remained (Bartlow, 2006; Hu et al., 2008). Each use a similar set of features for classification which include intra-key timing, or the latency between the depress of one key to the next, and key hold duration, or the average time between when a key is depressed and released. Research has been done using several statistical classifiers that attain similar results in terms of classification accuracy (Marsters, 2009).

Early work was done using short amounts of fixed text by Joyce, et al. (Joyce and Gupta, 1990) who used a custom built distance measure and statistical classifier to monitor the dynamics of password entry. This work was complemented by Brown, et al. (Brown and Rogers, 1993) who used a neural network to identify a user who had typed in a short string such as their name. Monrose, et al. (Monrose and Rubin, 1997) focused on improving long structured text (100-200 words) results using a Bayesian likelihood model as the classifier. This was followed by Bergadano, et al. (Bergadano et al., 2002) and Gunetti, et al. (Gunetti and Picardi, 2005) who both focused on the analysis of long free text (700-900 characters) using a custom built distance measure for classification. Two distance measures developed were R measures and A measures. Implementing the R measure on digraphs, trigraphs and four-graphs as well as the A measure on digraphs Gunetti, et al. (Gunetti and Picardi, 2005), were able to achieve a FAR of 0.005% and a FRR of 5.0%. Finally, Marsters (Marsters, 2009) similarly looked at keystroke dynamics on long free text but used a Bayesian Network classifier to achieve an EER of 0.27% when performing

10-fold cross-validation. Recently, Bours (Bours, 2012) has approached active authentication with keystroke dynamics in an innovative way. Rather than considering both how often a system wrongly accepts an impostor user (FAR) and rejects a legitimate user (FRR), Bours uses the legitimate user test data to set a threshold such that the FRR is 0. He is then able to look exclusive at how long it takes to discover impostor users. Results from this work are promising, with initial results showing all of the impostor users in the test data were identified within on average 182 keystrokes. In this article, we use FAR and FRR in accordance with the majority of work in active authentication.

2.3. Mouse Dynamics

Biometrics based on mouse dynamics involve monitoring the way a user moves the mouse in order to use that data as a means for authentication (Gamboa and Fred, 2004; Pusara and Brodley, 2004; Hashia et al., 2005; Ahmed and Traore, 2007; Pusara, 2007; Shen et al., 2010; Fehrer et al., 2012). The features calculated on this type of data include average speed per movement direction, click based interval times, action histogram, and average movement speed per travel distance. A full list of the features used in this article appears in Table 1.

Gamboa, et al. (Gamboa and Fred, 2004) collected mouse movements from users playing a memory game and attempted to identify the users from this data. Following this Ahmed et al. (Ahmed and Traore, 2007) focused on using data collected from a user's normal day to day computer use as a means for authentication. Using a neural network an EER of 2.46% was achieved. Shen, et al. (Shen et al., 2010) adjusted the features calculated and used feature selection with a comparison between an Artificial Neural Network (ANN) and a Support Vector Machine (SVM) to achieve a FAR of 1.86% and a FRR of 3.46% with the SVM. Lastly, Zheng, et al. (Zheng et al., 2011) searched for a method that requires fewer mouse strokes per user but achieves similar classification results. This was done by calculating angle based features on the points that the user clicked or hovered at with the mouse. By doing this an EER of 1.3% was achieved.

Recently, Mondal (Mondal and Bours, 2013) has approached active authentication with mouse dynamics in an innovative way. Rather than considering both how often a system wrongly accepts an impostor user (FAR) and rejects a legitimate user (FRR), Mondal uses the legitimate user test data to set a threshold such that the FRR is 0. He is then able to look exclusive at how long it takes to discover impostor users. Results from this work are promising, with initial results showing all of the impostor users in the test data were identified within 344 average number of impostor user actions (ANIA) and the average ANIA was 96 with a standard deviation of 79.

This article uses Ahmed, et al. (Ahmed and Traore, 2007) and Shen, et al. (Shen et al., 2010) as a basis of reasoning for the features calculated over the mouse movements of individuals. It then uses FAR and FRR for measuring results in accordance with the majority of work in active authentication.

Table 1: Mouse Modality Features.

Feature Description	**Calculation Details (# of Features)**
Average Speed per Movement Direction (Ahmed and Traore, 2007)	Average velocity in pixels/sec (8)
Movement Direction Histogram (Ahmed and Traore, 2007)	% of movement in each of the 8 directions (8)
Travel Distance Histogram (Ahmed and Traore, 2007)	% of movements occurring in each of 3 distance ranges (3)
Distribution of Actions on Screen (Shen et al., 2010)	% of actions ending in each of the 9 screen regions (9)
L/R Single Click Interval Times (Shen et al., 2010)	Avg and St Dev for L/R button click duration (4)
Left Double Click Interval Times (Shen et al., 2010)	Avg and St Dev for all consecutive presses and releases (8)
Pause and Click Time (Zheng et al., 2011)	Avg and St Dev between when cursor stops and click occurs (4)
Action Histogram (Ahmed and Traore, 2007)	Avg and St Dev % of time each of the 5 core actions occur (5)
Extreme Movement Speed (Shen et al., 2010)	Largest recorded velocity (pixels/sec) for each of the 3 distance ranges (3)
Movement Elapsed Time Histogram (Ahmed and Traore, 2007)	Histogram of movements based on elapsed movement time (9)
Average Movement Speed Relative to Travel Distance (Ahmed and Traore, 2007)	Avg movement velocity seen in each of three travel distances (3)

2.4. User Interface Interaction Analysis

The core concept behind using a user's Graphical User Interface (GUI) interaction style for biometrics has its roots in command line profiling (Imsand, 2008). Command line profiling (Schonlau et al., 2001; Maxion and Townsend, 2002), monitors the commands a user inputs into a command line based system, such as UNIX, in order to create an Intrusion Detection System (IDS). The idea behind the concept was that different people use different sets of commands to perform the same core task. When a user wants to accomplish a task on the system, there are often many different modalities that can be used. This includes entirely different programs that perform the same end task, using keyboard shortcuts versus GUI buttons, etc. When thinking of interacting with the GUI by sending "commands" one can draw parallels between command line profiling and GUI interaction in terms of their use as a biometric technique.

GUI usage authentication focuses on differentiating between what a user is doing and how they are doing it, by monitoring GUI messages sent internally to the Windows operating system (Pusara, 2007; Imsand, 2008). The features cal-

culated on the data are all count based. This means that the number of times certain user actions (key presses or button clicks), control types (using the scroll bar, clicking a GUI button) and the processes these actions originated from were observed and counted to generate a set of features. To generate a dataset, participants in Pusara's (Pusara, 2007) and Imsand's (Imsand, 2008) research were given a list of tasks to perform. These included word processing, web browsing, searching, and file/folder manipulation within the operating system. By doing this it allowed the researchers to take the actions of the user out of the equation and focus on their GUI interaction style. Pusara (Pusara, 2007) used a Decision Tree to achieve a FAR of 33.36% and a FRR of 1.49% while Imsand achieved a FAR of 8.66% and FRR of 0.0% using term frequency-inverse document frequency (TF-IDF) analysis. Imsand (Imsand, 2008), also experimented with an ANN which achieved a successful identification rate of 77.1%.

2.5. Multimodal Biometric Techniques

Several instances of research have attempted to combine multiple forms of biometric based authentication to improve the accuracy of the overall system. Asha, et al. (Asha and Chellappan, 2008) combined fingerprint biometrics with mouse dynamics in order to identify the users enrolled in an e-learning class. Rabuzin, et al. (Rabuzin et al., 2006) also make the case that combining multiple biometric techniques would be beneficial in creating a more robust authentication method for e-learning platforms. Other combinations include voice and facial recognition (Soltane et al., 2010); facial recognition and fingerprint (Azzini and Marrara, 2008), voice, facial recognition and fingerprint (Altinock and Turk, 2003); and iris and retinal features (Singhal et al., 2012).

2.5.1. Fusion Methods

Fusion of biometric modalities can occur in different ways. According to Ross, et al. (Ross and Jain, 2003), in biometric systems fusion can occur by fusing features together, fusing matching scores together, or a fusion of the decisions made by each individual modality. Fusion of features is the simple concatenation of feature vectors from multiple modalities to be input into the classifier (Ross and Jain, 2003), while decision level fusion uses the results from each individual modalities classifier in order to make a final decision (Ross and Jain, 2003). In this article, both feature fusion and decision level fusion are considered, but matching score fusion is not.

2.5.2. Fusion of Behavioral Biometrics

Ahmed, et al. (Ahmed and Traore, 2005) integrated keyboard and mouse dynamics into a single architecture that could act as an intrusion detection system. Twenty two subjects were asked to install a monitoring system on their workstations that collected keystrokes and mouse information. They ran the software for nine weeks. For the mouse movements they calculated a subset of features from (Ahmed and Traore, 2007) which appear in Table 1. A neural network was created and trained for each user. Doing this for all 22 users Ahmed, et al. were able to achieve a FAR of 0.651% and a FRR of 1.312%.

Pusara (Pusara, 2007) integrated keyboard, mouse dynamics and graphical user interface information into an integrated architecture that could also act as an intrusion detection system. Pusara enlisted 61 volunteers from undergraduate and graduate students to use a Windows machine and behave normally as they reviewed a reading assignment and then answered a set of twenty questions. They had ten days to complete the assignment and some of them did work on it over multiple days. Pusara calculated latencies and durations for digraphs as well as the mean, standard deviation and skewness as well as the number of occurrences of each alphabet letter and numeral. For mouse events, Pusara calculated the number of mouse movements as well as the mean, standard deviation, and skewness of distance, speed, angle of orientation, X-coordinates, Y-coordinates, and duration between movements. Finally, Pusara collected spatial and temporal GUI events that included items like minimizing, maximizing, restoring, moving windows, opening and closing processes, and selecting menus and buttons. Pusara then performed some smoothing to improve the results. The final results were a FAR 23.37% and a FRR of 1.50%.

3. Modality, Measurement and Features

This study fuses data from three modalities, the keyboard, mouse and GUI to determine if the fused features generated from the keyboard, mouse and GUI could increase the performance of a system designed for active authentication. The following subsections present the data collection method, the collection environment and participant tasks, followed by the features generated from the keyboard, mouse and GUI.

3.1. Data Collection Software

Windows 7 applications receive kernel and user input via message passing. Specifically, when a user input device is activated, it generates a message that is passed through a "hook chain". The Windows operating system maintains a hook chain for each different type of application level hook that can be made. When a message is generated that is associated with one of the hook chains it is passed down the chain so that all applications receive the message appropriately (Microsoft, 2012). For example, when the delete key is pressed, a "KeyDown" message is generated that includes a code for the delete key. In general, the active application will then perform the appropriate functionality. However, if the two previous keys where "Ctrl" and "Alt" and neither key has been released (as would be seen by a "KeyUp" message), then other processes may act on the delete keystroke prior to the application.

The authors developed software that runs with administrator privileges and connects to the hook chain for the WH_CALLWNDPROC, WH_KEYBOARD_LL, WH_MOUSE, and WH_GETMESSAGE hook types. The software receives all messages for those hook types before the applications. It collects these messages and stores them in a file for later processing.

The operating system dictates the resolution at which mouse movement events are recorded. In a typical recorded movement, mouse move events are registered about every 20 milliseconds. Key presses, releases, and mouse button clicks are recorded when they are registered by the operating system in both the up and down direction. For GUI usage analysis, control types are captured as well. Control types include buttons and menus the user accesses through the GUI. These control types are monitored by their window class name. Unfortunately, these window classes have general names making it impossible to consistently capture fine grained information such as the name of the control used. Therefore, controls such as buttons are all counted as the same.

3.2. Collection Environment and Participant Selection

The data collection was performed on a standard desktop configured with Windows 7 Service Pack 1, Microsoft Office Professional Plus 2010 and three popular internet browsers, Internet Explorer 9, Firefox 15.0.1 and Google Chrome 23. All of the participants were asked to perform three separate but similar internet based research tasks. The three tasks asked users to research the pros and cons of installing wind power, solar power and solar water heating at the Air Force Institute of Technology (AFIT) and write a 400-500 word report on each, to include pictures and/or charts to their liking. The topic of the task was not essential to the experiment as the main goal was to have the users interact with the machine by doing tasks like searching for text, switching between applications, scrolling documents, choosing the type of applications to use, etc.

Thirty one participants came from the general population of AFIT. The participants took part in the study during normal work hours, and completed the three tasks during a single session. The majority of the participants were graduate students but there were also instructors and other administrative personnel involved. Since the subjects were all some type of government employee we were able to assume that they had basic computer skills with the Windows operating system to include performing internet searches and the use of a Microsoft Office application for composition. For this reason, no time was allotted for the user to get comfortable with the system.

Demographic information was taken on each of the 31 participants to include, age range, gender, education level, dominant hand, self ranking of computer skills, etc. Each of these were analyzed using an ANOVA test to determine if there were any significant difference between different demographic categories that made certain users more or less feasible for behavioral biometric active authentication. It was determined however, that none of the demographic groups had tendencies that were any better or worse in using this type of authentication.

3.3. Analysis Method

After data collection, the raw data is processed to create features for identification and active authentication testing. The features calculated are selected from prior works because of their recurrence or due to their promising results. Some of the ranges used are the number of seconds of interaction (Gamboa and Fred, 2004), the number of mouse events (Zheng et al., 2011), the number of GUI events (Pusara, 2007), and the number of keystrokes (Marsters, 2009).

For this experiment, a sample consists of the keyboard, mouse, and GUI features calculated over a 10 minute sliding window with a two minute sampling interval. For instance, the first sample has data from 0 to 10 minutes. The second sample has data from 2 to 12 minutes, and the last sample has data from 58 to 68 minutes. This method is selected to simulate an environment where the system polls for user authentication every two minutes while still providing enough user activity to the biometric system in order to allow it to make a consistent decision. This method creates a scenario in which the classifier would notice an increasing deviation between a legitimate user's template and the observed test data every two minutes. This occurs until the user is determined to be not genuine and then locked out from the computer.

3.4. Keystroke Features

Keystroke features were based off of the work of Monrose (Monrose and Rubin, 1997). Two different types of features were calculated, durations and latencies. Durations include the mean time that each key is held down also described as the average difference in time between the depress and release of each key. Keystroke latency is the average time it takes for someone to transition between two keys. For example when typing "in" the time between when the user depresses "i" and depresses "n". With a 104 key keyboard this results in 10,816 possible digraph combinations, most of which will never occur. Due to this, any features that never get assigned a value for any user are removed as they do not add value for the classification algorithm.

3.5. Mouse Features

The mouse features were derived from Ahmed, et al. (Ahmed and Traore, 2007), Zheng, et al. (Zheng et al., 2011), and Shen, et al. (Shen et al., 2010). The calculation of each feature type is discussed below and listed in Table 1. Some of the features are movement based and require a movement to be defined in order for the features to be calculated. It was determined that there are two things that can start a movement for the mouse cursor. The first is a period of silence where there is no movement. If the cursor registers no movement for one second, it is deemed to be a period of silence. The second is a left button release. This is necessary since a user can click and drag an item and then release it without stopping movement. These two events, the dragging of an item followed by movement that is not dragging should be considered two separate events. Furthermore, a mouse movement was also required to have a pixel movement distance of 30 in order to actually be processed as a movement in order to eliminate the scenario where the user clicks and then does not move the mouse. Since all events recorded by the driver are in chronological order, movements are discovered by iterating until a movement starter is found. Next the nearest movement ender is located. There are three items that classify as a movement ender. They are a button press, mouse wheel scroll or mouse silence (one second).

3.5.1. Average Speed per Movement Direction

The average speed per movement direction records the user's mouse movement speed in eight different directions along the screen which are represented in Figure 1 (Ahmed and Traore, 2007). In order to determine the direction of movement, the angle between the coordinates of the movement starter and movement ender is calculated. This is followed by the speed of the movement using the distance formula and the time stamps associated with the beginning and end.

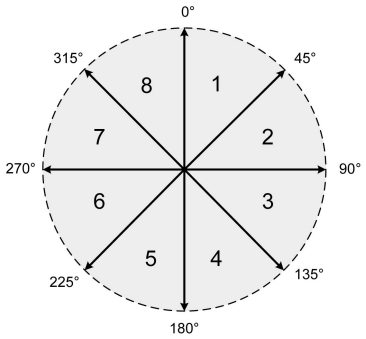


Figure 1: Direction sectors of Mouse Movements (Ahmed and Traore, 2007).

3.5.2. Movement Direction Histogram

The movement direction histogram is a histogram containing the percentage of movements that the user makes in each of the eight directions (Ahmed and Traore, 2007).

3.5.3. Travel Distance Histogram

The travel distance histogram contains percentages of the movements that a user makes in certain distance ranges (Ahmed and Traore, 2007). All of the distance ranges are measured in pixels. The histogram contains 3 values: short (0-300 pixels), medium (301-600 pixels) and long (601+ pixels). These ranges are from Shen, et al. (Shen et al., 2010) and due to the resolution of the screen in the testing environment being 1024x768 in their experiment as well.

3.5.4. Distribution of Actions on the Screen

The distribution of actions made on the screen results in a histogram containing information with the percentage of movements that end in nine different regions of the screen as seen in Figure 2 (Shen et al., 2010).

3.5.5. Single Click Interval Times

The click interval times are calculated for left and right button single clicks. The single click interval times were calculated by subtracting the time of the down click from the time of the up click, establishing an interval. The average and standard deviation of the intervals for left and right single clicks are calculated and turned into four features (Shen et al., 2010).

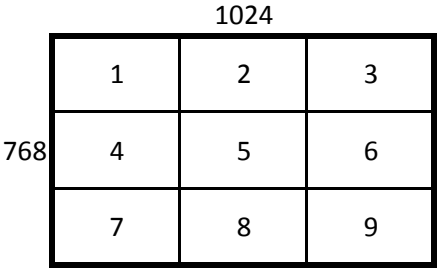


Figure 2: The nine screen regions.

3.5.6. Left Double Click Interval Times

Double click interval times were calculated by determining the interval times between all consecutive button down and button ups in the four event sequence. This lead to three different intervals, and the total time is also used, creating four intervals. The intervals are turned into eight features by calculating the average and standard deviation for each (Shen et al., 2010).

3.5.7. Pause and Click Time

The pause and click time is the amount of time it takes for the user to click the mouse button after they have stopped moving the cursor. This was shown to be a discriminating feature by Zheng, et al. (Zheng et al., 2011).

3.5.8. Action Histogram

The action histogram contains the percentage of actions of a given type made by the user (Ahmed and Traore, 2007). It is made up of five different action types: the number of left, right and double clicks, the number of mouse wheel events, and the number of click and drag actions in where the user holds down the left button while moving the cursor.

3.5.9. Extreme Movement Speed Relative to Travel Distance

The extreme movement speed made by a user in relation to travel distance is similar to the travel distance histogram but instead looks for the largest recorded speed for a given distance range (Shen et al., 2010). The same three range lengths are used from the travel distance histogram but with the units of pixels per second.

3.5.10. Movement Elapsed Time Histogram

The time it takes to complete each movement is calculated and stored. Using this stored data, a histogram is created that has information about the number of movements that fall into each histogram time window interval. Ahmed, et al. (Ahmed and Traore, 2007) set the histogram time window bin size to half second intervals from 0-4 seconds. We have also set each column in the histogram to a size of a half second. For instance, if a movement took 1.78 seconds it would fall into the fourth interval of [1.5 seconds, 2.0 seconds) and if it took exactly 6 seconds it would fall into the thirteenth interval of [6.0 seconds, 6.5 seconds).

3.5.11. Average Movement Speed Relative to Travel Distance

The average movement speed for each travel distance (Shen et al., 2010) is calculated using previously stored distance and speed calculations about each movement. The same travel distances are used again from the travel distance histogram in order to determine the average speed for short, medium and long movements.

3.6. GUI Usage Features

The features for the GUI usage analysis are calculated by determining the number of times each message occurs. This method follows Imsand's process (Imsand, 2008), and enumerates differences between the usage styles of different individuals. A counting method is used to translate the text output from the driver into numerical values that the machine learning algorithms can utilize. In order to do this, three different classes of items are monitored: user actions, control types and executing processes.

3.6.1. User Actions

This can be any type of user initiated action such as keystroke or mouse event. The counts of each of these separate events are used as the feature values.

3.6.2. Control Types

This is represented by a count of each unique type of window class name, which gives a general idea for the GUI buttons and controls that a person uses.

3.6.3. Processes Executed

A count of the number of times each process is seen. This captures what process/application the participant is using, as well as a rough estimate on the number of actions that process is used for.

4. Fusion System Design

To determine if the fusion of features from all of the modalities (keyboard, mouse and the GUI) provides better results than the individual modalities by themselves, comparison testing of each of the modalities individually, along with two fusion approaches, is performed for the identification (multi-class) and authentication (binary class) problems. All possible paired modality combinations were tested but none produced significantly improved results over the fusion of all three. Therefore only the fusion of all three results is discussed further.

The first fusion approach, seen in Figure 3 (a), involves combining all of the features into one sample that then has feature selection and classification performed on it to produce results. Figure 3 (b) shows decision level fusion in which each modality is classified individually, with the results of those classifications sent to a final classifier that produces a decision. Both of these experiments use all 31 participants.

Sliding windows samples (10 minutes) were completed for each participant and ranged, for each task, from a low of 8 windows to a high of 27 windows with the average being 10.6 windows. Each sample contains all of the keystroke, mouse, and GUI dynamic features described previously. There were an average of 14,552 keystroke dynamics per user with a standard deviation of 353, an average of 465 mouse dynamics per user with a standard deviation of 10 and average of 85 window class names with a standard deviation of 13. There were only 1 or 2 outliers for keystroke, mouse, and GUI dynamics and they were all above the average. Since two experiments are being performed, one for identification and one for authentication, the dataset is duplicated.

All feature selection and machine learning classification is done using the Weka data mining toolkit (M. Hall and Pfahringer, 2009). Three different classification algorithms are tested; BayesNet, LibSVM and the J48 (C4.5) decision tree. BayesNet was used successfully in keystroke identification by Marsters (Marsters, 2009), LibSVM was successfully used by Shen, et al. (Shen et al., 2010) for classifying mouse dynamics and a variant of the C4.5 decision tree was used by Pusara, et al. (Pusara and Brodley, 2004). J48 (C4.5) was determined experimentally to be the best fusion classifier.

Before discussing the fusion techniques in detail, it is necessary to distinguish between the two datasets that are tested. Both a multi-class (identification) and binary-class (authentication) dataset are tested for each individual modality and the two methods of feature fusion.

4.1. Identification

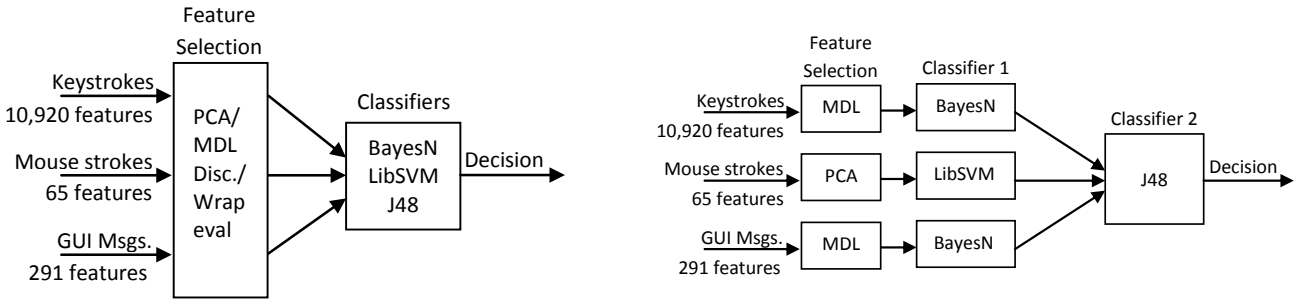
Identification is a multi-class classification problem. From the data, an ideal classifier distinguishes and identifies the user that generated a given feature sample, returning the user ID number that it thinks that feature sample belongs to. Identification classification testing is done with 10-fold cross-validation, using all of the data, and performance is assessed using the accuracy of identification across the ten folds.

4.2. Authentication

Authentication is a binary classification problem. When using this method a classifier is trained for each individual user. The classifier is given a set of samples from the legitimate user, and also a sampling of samples from users which are not the legitimate user. The remaining legitimate user and impostor user samples are then used for testing on whether each sample belongs to the legitimate user or not. Authentication testing is done with 3-fold cross-validation and performance is assessed using the accuracy of authentication across the three folds. This type of experiment can lead to two different types of error. Type I error, in which a user who should be authenticated is not, and Type II error in which a user who should not be authenticated is. These errors are represented as the False Rejection Rate (FRR) and False Acceptance Rate (FAR) respectively. The following is broken up into two sections, information on how the training set is created for the classifier and what type of data the classifier is tested with.

4.2.1. Training Data

Two-thirds of the available samples for each participant were used for training and the remaining one-third was used for testing. As mentioned above, the number of samples for a user



(a) Feature Level Fusion

(b) Ensemble Based, Decision Level Fusion

Figure 3: Graphical representation of the architecture used for two types of multi-modal fusion.

ranged from 8 to 27 with an average of 10.6. Due to the small number of training samples that are available per user when compared to the entire dataset as a whole, using all of the data for the impostor user creates an imbalance in the training data (for each legitimate user sample, there are about 30 samples from impostor users). Thus, to reduce this imbalance less impostor user samples were used. As seen in Table 2, six impostor user samples produced the best results when taking both FAR and FRR are taken into account. Thus, a random sampling of six impostor user samples are used. For the user with the fewest samples, this results in approximately a 1:1 legitimate user to impostor user ratio while for the user with the most samples, this results in approximately a 3:1 legitimate user to impostor user ratio. On average, the ratio is approximately 7:6 legitimate user to impostor user ratio.

Table 2: Relationship with the number of impostor user training instances and FAR/FRR for Feature Fusion Authentication.

# of Impostor Instances	FAR (%)	FRR (%)
4	8.12	0.53
5	4.94	1.19
6	3.76	2.51
7	2.57	7.34

4.2.2. Testing Data

The testing data is selected from datasets that have not been used for training. The remaining samples from the legitimate user are added along with one feature vector from each of the remaining impostor users. Thus, on average, there are approximately 3.6 legitimate user samples compared to 30 impostor user samples (1 for each of the other participants). Using a test set from each user as opposed to just users who have been trained on, allows for many more tests, but more importantly provides a more realistic scenario where the classifier may be encountering testing data for an impostor user that it has not been trained on. Recall that each sample contains summary information described in Section 3 for the keystroke, mouse and GUI dynamics that occurred within that sample's ten minute time window.

4.2.3. Replications

Each of these testing and training cycles are performed three times per user to ensure that several combinations of testing and training data are achieved with the legitimate user samples. Each time the samples from two tasks are used for training and the samples from the third task are used for testing. Due to the fact that the six impostor user training samples are selected randomly, it is necessary to run the test multiple times in order to achieve statistical normality. Each test is run 30 times selecting different impostor user samples each time. In order to achieve the final FAR, and FRR, the average is taken over each of the three legitimate user combinations and all 30 replications (i.e., 90 tests per user).

4.3. Feature Level Fusion

To test the feature fusion method represented by Figure 3 (a), all features are combined and feature selection is performed on the data as a whole. Classification is performed using the BayesNet, LibSVM and J48 machine learning algorithms. Feature selection is performed on a per classifier basis. LibSVM requires the data to be run through a Principle Components Analysis (PCA) prior to being classified. BayesNet requires discretized data so each dataset is classified through BayesNet, is passed through a supervised discretizing filter based on Fayyad and Irani's Minimum Description Length (MDL) method (Fayyad and Irani, 1993). Finally, since J48 can handle a wide variety of data, the best attribute selection method was found to be using a wrapper evaluator with a best first search method.

4.4. Ensemble Based Decision Level Fusion

In ensemble learning multiple classifiers make decisions on smaller pieces of a larger dataset. These predictions are then combined into a single predictive model which generally will have better performance than the individual classifiers alone (Opitz and Maclin, 1999).

In Figure 3 (b) the features from each modality are passed through their individual classifier before they are fused together. The type of classifier used for each modality is based on the results from the identification section in Table 4. The decisions from each of these classifiers are then classified using J48 with bagging to generate the final decision from the ensemble classifier.

Three sets of testing and training data are generated for each of the 31 users, and once again they are generated 30 times to

achieve statistical normality. This is done for all three modalities. Each of the testing and training set pairs, are then run through their respective feature selection methods and classifier. Each individual modality classifier outputs the class predicted by the classifier for that sample, and the classifiers confidence in it's decision represented as a probability. The confidence probability from the individual modality classifiers are then sent to the ensemble classifier. The confidence probability is expressed with respect to the legitimate user (a probability of 1.0). This means that if the predicted class was an impostor user the target probability is 0.0.

It should be noted, that it is possible to have a mix of decisions from the initial classifiers. For example, one modality could predict the data is from the legitimate user, while the other two predict it is from an impostor user meaning the initial classifiers have made contradicting decisions. Ideally this allows the ensemble classifier to decide which modality should be allotted more significance in the model.

5. Results

The 31 test subjects worked on three separate tasks allowing the data collection to include an average of 14,552 keystrokes, 673 digraphs logged, 77 of the 104 keys being pressed, 465 mouse movements, 23 different processes used, and 85 window class names being registered. Based on this, over nine thousand keyboard features were eliminated due to the fact that no feature values were generated by any of the users.

Each of the classifiers and feature selection methods were tuned to provide the highest active authentication accuracy, with the final parameters shown in Table 3. BayesNet was left in its default configuration as provided by Weka. Different estimators and search algorithms were tested but none outperformed the *SimpleEstimator* or the *K2* search algorithm. LibSVM allows for different kernel functions as well as the manipulation of several parameters for each. The sigmoid kernel consistently generated the best results. An experiment was run inside of Weka on the γ parameter and it was determined that setting it to 0.01 yielded the highest classification accuracy. The J48 decision tree was tried with several feature selection methods to include ReliefF and a discretization filter however the wrapper evaluator produced the best results. Parameters were also adjusted to include, using and not using pruning, and adjusting the confidence factor however, none improved the results over the Weka defaults.

Table 3: Final parameters used for the selected algorithms.

Classifier	Final Parameters Selected
BayesNet	MDL discretization (Fayyad and Irani, 1993) Weka defaults
LibSVM	Principle component analysis Sigmoid kernel, $\gamma = 0.01$
J48	Wrapper evaluator with Weka defaults

5.1. Feature Level Fusion Results

5.1.1. Identification (Multi-class Dataset)

The identification results presented in Table 4 show that the fusion of features, using the method shown in Figure 3 (a) with a BayesNet Classifier, performed better than any of the individual modalities on their own. An identification percentage of 99.39% was achieved using BayesNet which outperforms the keystroke, mouse and GUI modalities when classified on their own. The high fusion percentages validate our hypothesis that by combining features from multiple modalities, classification accuracy can be improved. As can be seen, the keystroke features consistently performed better than the other two modalities which is discussed in Section 5.1.3.

Table 4: Identification multi-class classification comparison results.

	Identification (10-fold CV) (%)			
	Keyboard	Mouse	GUI	Fusion
BayesNet	**97.05 ±3.03**	82.77±2.96	**86.57±2.69**	**99.39±1.11**
LibSVM	96.86±2.38	**85.53±4.26**	69.74±5.00	96.66±2.23
J48	85.68±4.37	74.26±5.55	81.72±5.45	86.64±4.62

5.1.2. Authentication (Binary-class Dataset)

Being able to authenticate a user while they perform their daily work is the primary goal behind this system. By having the participants research the pros and cons on the Internet and then write a report about their findings, we have given them tasks that closely resemble the work that many of them do on a regular basis. The results achieved when performing the authentication experiment show similar trends with the multi-class dataset, as seen in Table 5. BayesNet outperforms both LibSVM and J48 with a full fusion False Acceptance rate (FAR) of 3.76% and False Rejection Rate (FRR) of 2.51%. Correcting the imbalance of data when performing the binary class experiment was necessary in order to improve classification performance of the system.

Table 5: Authentication binary-class classification comparison results.

	Authentication FAR (top) & FRR (bottom) (%)			
	Keyboard	Mouse	GUI	Fusion
BayesNet	5.21±0.61 6.99±0.90	10.15±0.56 11.28±1.09	17.87±1.01 5.34±1.15	**3.76±0.48** **2.51±0.57**
LibSVM	7.46±0.58 14.99±1.52	3.88±0.49 51.90±2.65	10.88±1.00 34.71±1.68	11.67±0.77 18.80±1.50
J48	13.26±0.98 17.21±2.31	15.07±1.05 32.61±2.96	14.33±1.34 23.91±2.48	16.29±1.04 21.37±2.63

In order to ensure that the fusion results show significant classification improvement over any of the modalities on their own, significance testing using the Welch two sample t-test (Welch, 1947) is performed to ensure that the fusion results show significant classification improvement over each of the individual modalities. The Shapiro-Wilk normality test (Shapiro and Wilk, 1965) is used to confirm that the data is normally distributed. A p-value of 0.78 was achieved, implying that the null hypothesis is rejected (the data is not normally distributed), and accepting the alternative hypothesis that the data is normally distributed. The Welch t-test was selected because it is designed

to determine whether a difference in two datasets occurred simply due to chance or not. A standard significance level of 0.05 was selected for the test.

The p-values in Table 6 are much smaller than the significance level that was set. This means there is convincing evidence that each of the outcomes recorded in Table 5 did not occur due to chance. The feature fusion results in Table 5 are displayed in bold to represent that they are significantly better than any other results in the table. In Table 6 all results are recorded with respect to the individual modality data. This means that a confidence interval range of {1.17%, 1.73%} for fusion versus the keystroke modality, means there is 95% confidence that the FAR of the keystrokes will be 1.17% to 1.73% higher than the fusion FAR. Table 6 shows that the fusion technique is statistically significantly more effective for authentication over any individual modality by itself. The FAR value for mouse data produces the only confidence interval containing zero, however this can be discounted because of its extremely high FRR values.

Table 6: Significance of fusion FAR/FRR vs individual modalities FAR/FRR.

Comparison		p-Value	95% Confidence Interval
Fusion vs. Key	FAR	<0.001	{1.17%, 1.73%}
	FRR	< 0.001	{4.10%, 4.86%}
Fusion vs. Mouse	FAR	0.334	{-0.13%, 0.36%}
	FRR	< 0.001	{48.39%, 50.38%}
Fusion vs. GUI	FAR	< 0.001	{13.70%, 14.52%}
	FRR	< 0.001	{2.37%, 3.29%}

5.1.3. Individual Modality Performance

In terms of the individual modalities, keystroke features performed the best across all of the identification and active authentication classification algorithms mainly because of the large number captured during data collection. Bours (Bours, 2012) was able to detect impostor users within on average 182 keystrokes. Further supporting this, Marsters (Marsters, 2009) determined that a training block could be calculated effectively with as few as 300 keystrokes. On average our participants generated 987 keystrokes per 10 minute sliding window. This was the only modality that exceeded the number of user events needed for generating consistent features.

The highest identification rate seen for the mouse dataset was 85.53% using LibSVM. The decreased performance in comparison with prior work, is attributed to the lack of movements during subject testing. Previous mouse dynamics work (Ahmed and Traore, 2007; Shen et al., 2010), required 2,000 mouse actions per feature sample to achieve their EER of around 1-3 percent. When our users performed the tasks, they generated an average of 28 mouse movements per 10 minute sliding window. This does not meet the requirements from Ahmed, et al. (Ahmed and Traore, 2007) and Shen, et al. (Shen et al., 2010) in order to achieve their level of performance and thus resulted in the mouse features under performing.

The point to point mouse features derived by Zheng, et al. (Zheng et al., 2011) were also included in order to gauge their effectiveness. According to Zheng they needed far less testing data than the features derived by Ahmed, et al. (Ahmed and Traore, 2007) and Shen, et al. (Shen et al., 2010). Zheng's work achieved an EER of 1.30% using only 25 mouse movements in the test set. After implementing these features, the feature fusion identification results were unchanged due to feature selection eliminating Zheng's mouse features. Testing these features on their own produced an identification rate of 15.61% using LibSVM with principle component analysis. For this reason these features were dropped from the dataset. One reason for the features poor performance could be that the point to point angle based calculations vary based on the activity the user is performing. Given that the task here is short and free-use for the mouse, the needed repeated motions for strong classification rarely occurs.

The GUI features performed well given the unstructured nature of the task. Using a BayesNet in this experiment a 86.57% identification rate was achieved. It is thought that the broader task we selected for the participants accentuated the preferences and tendencies that a user has inside of the GUI. It is also feasible that allowing a user to perform free computer use could further improve these results; however this would need to be tested.

5.2. Ensemble Based, Decision Level Fusion Results

Ensemble based classification, Figure 3(b), provides another method for generating the fusion of features for active authentication. By combining the modalities together once they have been individually feature selected and classified, it provides increased accuracy compared to what each of the modalities could provide on their own, and over feature fusion. The classifiers used for each individual modality was determined by the performance listed in Table 4. BayesNet was selected for both the keystrokes and GUI messages while LibSVM was chosen for the mouse.

Table 7: EBDL fusion authentication classification per machine learning algorithm

Ensemble Classifier		Feature Fusion (%)	EBDL Fusion (%)
BayesNet	FAR	3.76 ± 0.48	2.47 ± 0.40
	FRR	2.51 ± 0.57	2.53 ± 0.37
LibSVM	FAR	11.67 ± 0.77	2.61 ± 0.01
	FRR	18.80 ± 1.50	2.51 ± 0.01
J48 with Bagging	FAR	16.29 ± 1.04	**2.24 ± 0.45**
	FRR	21.37 ± 2.63	**2.10 ± 0.30**

Table 8: Relationship with the number of Impostor training instances and FAR/FRR for EBDL Fusion.

Ratio of Impostor to Legitimate User Instances	FAR (%)	FRR (%)
30:1	3.60	4.17
15:1	3.17	3.88
2:1	2.91	2.32
1:1	2.24	2.10

The classifier that performed the best as the ensemble classifier was J48 with bagging (Table 7). Bagging, also known as Bootstrap aggregating, generates multiple versions of a classifier and uses a majority voting scheme to make its decision

(Breiman, 1996). As with previous authentication tests, due to the data imbalance per class only one impostor user was randomly selected for training against the legitimate user. As shown in Table 8, the one-to-one ratio of impostor users to legitimate users performed the best.

Significance testing using the Welsh t-test (Welch, 1947) is performed comparing EBDL fusion method to feature fusion. This test was performed with a significance level set to 0.05. Table 9 shows that the ensemble based method is significantly more effective than feature fusion when comparing FAR and FRR.

Table 9: Significance of feature fusion vs. EBDL fusion.

Comparison	**p-Value**	**95% C.I.**
False Acceptance Rate (FAR)	< 0.001	{1.28%, 1.76%}
False Rejection Rate (FRR)	<0.001	{0.18%, 0.64%}

5.3. Prior Work Results

Table 10 presents results from prior work in active authentication on each of the other individual modalities and the EBDL fusion method. Table 10 shows the number of actions required from the legitimate user in the testing and training set used by previous work along with their best performance classification accuracy. In Table 10 KS stands for keystrokes, MM for mouse movements, and SW for sliding windows. The ratio of training actions to testing actions for EBDL in Table 10 does not show a ratio of 2:1 in accordance with the 2:1 ratio of training to testing samples due to the low number of samples coupled with the high standard deviations in user activity between samples.

It needs to be noted that the best results on the individual modalities have better FARs and FRRs than appear here, however it is difficult to make a direct comparison based on the differences in experimental techniques. These differences include user task (structured task (Ahmed and Traore, 2007; Imsand, 2008), simulated free use (Ahmed and Traore, 2007; Marsters, 2009; Zheng et al., 2011)), the amount of data in training and testing, and incorporating environmental effects (single collection time, multiple collection times, device flexibility). Because of these discrepancies a direct comparison between the individual modalities and the fusion methods should not use Table 10 in favor of the results in Table 5 and Table 7 which use the same data source.

One of the commonly reoccurring issues in the area of behavioral biometric authentication is the amount of time required to detect a malicious user. An experienced malicious user needs only a few minutes on a internal computer to impact a network. In an active authentication system, designed to detect and deter impostor users, there needs to be a high accuracy using a small number of actions or over a short time period. Some of the previous works require less testing data than our fusion system but this is potentially offset by the large amount of training data needed or other experimental design differences. However, their fully trained modality could easily be included into the EBDL. Although prior work shows better results, much of it is focused on actions rather than time. By focusing on time, our research demonstrates a more continuous method of authentication enabling detection during times of reduced actions. Such a technique is particularly useful for authentication individuals who only use their computers sporadically and without the high amount of data entry that previous research assumes. An additional benefit to the fusion system over previous systems is that it is able to capture a malicious user's actions regardless of whether they are using the keyboard or mouse to accomplish their goal.

6. Conclusions and Future Work

There are thousands of minute differences between how two different users interact with a computer system. Analyzing the entire picture of a users interaction is shown to improve the accuracy and reliability of a behavioral biometric system designed for active authentication over using a singular modality. Multimodal fusion also required far less user interaction to achieve similar classification accuracy as systems that used an individual modality. EBDL fusion significantly outperformed each individual modality as well as feature fusion, producing a FAR of 2.24% and FRR of 2.10%. These results are in line with previous singular modality work but more closely simulate an active authentication scenario by using a sliding window technique to perform user authentication on 10 minutes of user input every two minutes.

Future work in the area of active authentication will include the collection of data in a free use environment over a longer period of time in order to asses the feasibility of this system performing active authentication in a real world environment. Larger amounts of data on each user will allow for slicing of the data on smaller intervals or by a threshold of their activity allowing for the development of a means to detect the impostor user in real-time. Finally the accuracy of these systems must continue to be improved. In this experiment an FRR of 2.10% means that there is a 50% chance the user will be locked out after an hour of work, using the 10 minute sliding window on a 2 minute interval. For this reason the False Rejection Rate (FRR), as well as the training and testing time must be improved if there is ever a hope for real world deployment.

Acknowledgments We would like to thank everyone who committed their time to being a test subject, and also Alanna Keith for acting as the testing proctor. The views expressed in this work are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

References

Ahmed A, Traore I. Anomaly intrusion detection based on biometrics. In: IEEE Workshop on Information Assurance. 2005. p. 1–7.

Ahmed A, Traore I. A new biometric technology based on mouse dynamics. IEEE Transactions on Dependable and Secure Computing 2007;4:165–79.

Altinock A, Turk M. Temporal integration for continuous multimodal biometrics. In: Multimodal User Authentication 03. 2003. p. 11–7.

Asha S, Chellappan C. Authentication of e-learners using multimodal biometric technology. In: International Symposium on Biometrics and Security Technologies. 2008. p. 1–6.

Table 10: Required number of testing and training actions (avg) per previous active authentication work.

Previous Work	Training Actions	Testing Actions	Results	Number of Users in Study
Marsters (Marsters, 2009)	>85,000 KS	>300 KS, 3 Hrs	EER 0.27%	10
Ahmed, et al. (Ahmed and Traore, 2007)	10,000 MM	2,000 MM	EER 2.46%	22
Zheng, et al. (Zheng et al., 2011)	12,500 MM	25 MM	EER 1.30%	30
Imsand, et al. (Imsand, 2008)	unspecified	unspecified	FAR 8.66% FRR 0.0%	31
Pusara (Pusara, 2007)	82,861 KS,MM,GUI	9,207 KS,MM,GUI	FAR 23.33% FRR 1.50%	61
EBDL Fusion	10,446 KS, 335 MM 147 GUI	987 KS, 28 MM 85 GUI	FAR 2.24% FRR 2.10%	31

Azzini A, Marrara S. Imposter users discovery using a multimodal biometric continuous authentication fuzzy system. Knowledge-Based Intelligence Information and Engineering Systems Lecture Notes in Computer Science 2008;5178:371–8.

Bartlow N. Evaluating the reliability of credential hardening through keystroke dynamics. In: International Symposium on Software Reliability Engineering, 2006. 2006. p. 117–26.

Bergadano F, Gunetti D, Picardi C. User authentication through keystroke dynamics. ACM Transactions on Information and System Security 2002;5:367–97.

Bhattacharyya D, Ranjan R, Alisherov F, Choi M. Biometric authentic: A review. International Journal of Service, Science and Technology 2009;2:13–28.

Bleha S, Slivinsky B, Hussein B. Computer access security systems using keystroke dynamics. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990;12:1217–22.

Bours P. Continuous keystroke dynamics: A different perspective towards biometric evaluation. Information Security Technical Report 2012;.

Breiman L. Bagging predictors. Machine Learning 1996;24:123–40.

Brown M, Rogers S. User identification via keystroke characteristics of typed names using neural networks. International Journal of Man-Machine Studies 1993;39:999–1014.

Fayyad U, Irani K. Multi-interval discretization of continuous-valued attributes for classification learning. In: Thirteenth International Joint Conference on Artificial Intelligence. 1993. p. 1–6.

Fehrer C, Elovici Y, Moskovitch R, Rokach L, Schclar A. User identity verification via mouse dynamics. Information Sciences 2012;201:19–36.

Gaines R, Lisowski W, Press S, Shapiro N. Authentication by Keystroke Timing: Some Preliminary Results. Technical Report; Rand Corporation; 1980.

Gamboa H, Fred A. A behavioural biometric system based on human computer interaction. Proceedings of SPIE, Biometric Technology for Human Identification 2004;5404:381–92.

Garg A, Rahalkar R, Upadhyaya S, Kwiat K. Profiling users in gui based systems for masquerade detection. In: IEEE Workshop on Information Assurance. 2006. p. 1–7.

Gunetti D, Picardi C. Keystroke analysis of free text. ACM Transactions on Information and System Security 2005;8:312–47.

Haider S, Abbas A, Zaidi A. A multi-technique approach for user identification through keystroke dynamics. IEEE International Conference on Systems, Man, and Cybernetics 2000;2:1336–41.

Hashia S, Pollett C, Stamp M. On using mouse movement as a biometric. In: Proceedings in the International Conference on Computer Science and its Applications. volume 1; 2005. .

Hu J, Gingrich D, Sentosa A. A k-nearest neighbor approach to usr authentication through biometric keystroke dynamics. In: IEEE Conference on Communications, 2008. 2008. p. 1556–60.

Imsand E. Applications of GUI Usage Analysis. Ph.D. thesis; Auburn University; 2008.

Joyce R, Gupta G. Identity authentication based on keystroke latencies. Communications of the ACM 1990;33:168–76.

M. Hall E. Frank GH, Pfahringer B. The weka data mining software: An update. SIGKDD Explor Newsletter 2009;11(1):10–8.

Marsters J. Keystroke Dynamics as a Biometric. Ph.D. thesis; University of Southampton; 2009.

Matyas V, Zdenek R. Toward reliable user authentication through biometrics. IEEE Security and Privacy 2003;May/June:45–9.

Maxion R, Townsend T. Masquerade detection using truncated command lines. In: IEEE International Conference on Dependable Systems and Networks. 2002. p. 1–10.

Microsoft . Hooks overview. Microsoft Developer Network; 2012.

Mondal S, Bours P. Continuous authentication using mouse dynamics. In: Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG). 2013. p. 1–12.

Monrose F, Rubin A. Authentication via keystroke dynamics. In: ACM Conference on Computer and Communications Security. 1997. p. 48–56.

Opitz D, Maclin R. Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 1999;11:169–98.

Polikar R. Ensemble based systems in decision making. Circuits and Systems Magazines, IEEE 2006;6(3):21–45.

Pusara M. An Examination of User Behavior for User Re-Authentication. Ph.D. thesis; Purdue University; 2007.

Pusara M, Brodley C. User re-authentication via mouse movements. In: ACM Workshop on Visualization and Data Mining for Computer Security. 2004. p. 1–8.

Rabuzin K, Baca M, Sajko M. E-learning: Biometrics as a security factor. In: International Multi-Conference on Computing in the Global Information Technology. 2006. p. 1–6.

Ross A, Jain A. Information fusion in biometrics. Pattern Recognition Letters 2003;24:2115–25.

Schonlau M, DuMouchel W, Ju W, Karr A, Theus M, Vardi Y. Computer intrusion: Detecting masquerades. Statistical Science 2001;16:1–16.

Shapiro S, Wilk M. An analysis of variance test for normality (complete samples). Biometrika 1965;52:591–611.

Shen C, Guan X, Cai J. A hypo-optimum feature selection strategy for mouse dynamics in continuous identity authentication and monitoring. In: IEEE International Conference on Information Theory and Information Security. 2010. p. 349–53.

Singhal R, Singh N, Jain P. Towards an integrated biometric technique. International Journal of Computer Application 2012;42:20–3.

Soltane M, Doghmane N, Guersi N. Face and speech based multi-modal biometric authentication. International Journal of Advanced Science and Technology 2010;21:41–56.

Welch B. The generalization of student's problem when several different population variances are involved. Biometrika 1947;34:28–35.

Zheng N, Paloski A, Wang H. An efficient user verification system via mouse movements. In: ACM Conference on Computer and Communications Security. 2011. p. 1–12.

Kyle O. Bailey is a Cyberspace Operations Officer for the United States Air Force. He received a M.S. in Cyberspace Operations from the Air Force Institute of Technology and a B.S. in Computer Science from the United States Air Force Academy.

James S. Okolica is a PhD candidate at the Air Force Institute of Technology. He received a BA in Computer and Applied Mathematics from Drew University and an MS in Computer Science from the Air Force Institute of Technology.He is currently working as a research engineer at the Center for Cyberspace Research. His research interests include text mining and memory forensics.

Gilbert L. Peterson is an Associate Professor of Computer

Science at the Air Force Institute of Technology, and Vice-Chair of the IFIP Working Group 11.9 Digital Forensics. Dr. Peterson received a BS degree in Architecture, and an M.S and Ph.D in Computer Science at the University of Texas at Arlington. He teaches and conducts research in digital forensics, statistical machine learning, and autonomous robots.