Comparison of Biometric Authentication Software Techniques: GEFE vs. Angle Based Metrics

Robert Stokes, Angelica Willis, Kelvin Bryant, Zanetta Tyler, and Anthony Dobson
Department of Computer Science
North Carolina A&T State University
1601 E Market St. Greensboro, NC 27411
kingstokes@gmail.com, awillis@aggies.ncat.edu, ksbryant@ncat.edu, zrtyler@aggies.ncat.edu, amdobson@aggies.ncat.edu

Abstract

In this paper, we explore three alternatives for developing a biometric authentication software system. The first approach we will consider is a computer vision technique optimized by Genetic and Evolutionary Feature Extraction (GEFE); the second is Angle Based Metrics (ABM); and the third is Angle Based Metrics combined with Genetic and Evolutionary Computation (ABM + GEC). Each of these techniques are research areas which show promise in regards to being able to authenticate users based on their natural mouse movements. When applied to the same data set, the results of our experimentation indicate that both the ABM and ABM + GEC techniques are more accurate than GEFE in correctly verifying genuine users, as well as correctly rejecting impostors.

Keywords – Biometrics, genetic and evolutionary feature extraction (GEFE), angle based metrics

Introduction

Biometric systems are able to authenticate or identify people based on physiological or behavioral characteristics which are unique for each person [5]. As biometric systems become increasingly accurate, they will be selected more often as the option of choice for authentication, intrusion detection, or access control within software systems. One of the most useful applications for biometrics is user authentication. Authentication is a way to prove that a user is who they claim to be. In most systems, authentication involves asking a person to prove who they are by what they know – such as a username and password combination [9]. Biometric authentication attempts to carry out the verification process based on analysis of characteristics that are unique to a given individual. Physiological biometrics include analysis of characteristics such as fingerprint, iris, or facial features. Behavioral biometrics focus on the way in which users interact with their computer device. Some examples are mouse movements [8], keystroke rhythm, and touch screen interaction. The main benefits of biometrics is that they are difficult to mimic and they have an advantage over password authentication in that they are not susceptible to being cracked (via dictionary attacks or brute force attacks), lost, or stolen [11].

An emerging application of biometrics is active authentication (AA). Active authentication is a way of continuously authenticating or verifying a user's identity during a session. Typically, a user is only authenticated at the beginning of a session. If the user steps away from the computer or if the session is hijacked then the secured assets are vulnerable to exploitation. Active authentication attempts to continually verify that a user's biometric patterns (human to computer interactions) are consistent with those demonstrated during their previous sessions [3]. The goal is to determine whether or not the current user is an imposter or the original authenticated user.

In this paper, we compare three different approaches to implementing biometric authentication using mouse movement. The first approach uses Genetic and Evolutionary Feature Extraction (GEFE) [1] to optimize computer vision and evolutionary computation techniques. The second approach, called Angle Based Metrics (ABM) [15], uses angle analysis in order to extract features and distinguish between valid users and impostors. And the third approach, called ABM+GEC is an enhanced version of ABM which utilizes a genetic and evolutionary computation (GEC) technique in order to reduce the size of the extracted feature set. Though both GEFE and ABM+GEC use evolutionary computation as a method of improving the efficiency and success of their root techniques, they are completely independent approaches.

In addition to exploring how these three approaches compare, we also present evidence that GEC is a valuable method of reducing the complexity of systems like ABM, by eliminating irrelevant data from consideration, thus increasing the efficiency and feasibility of Active Authentication. The true acceptance rate (TAR) and false ac-

ceptance rate (FAR) results for all three techniques were computed using the same data set. The rest of the paper is as follows. The next section describes GEFE. Following the GEFE section, ABM is introduced. Next, a discussion of how the GEC was combined with ABM is presented, followed by a section that presents the advantages and disadvantages of AMB and GEFE. The last three sections describe how the experiment was conducted, present a comparison of the results and, finally, present conclusions and future work.

GEFE

The GEFE technique involves the use of algorithms which have been adopted from the fields of Evolutionary Computation and Computer Vision in order to be able to classify images [4]. The path of each mouse movement is recorded using the (x, y) screen coordinates and then saved as an image file. The image is then analyzed in a similar biometric manner as a facial image. Images are compared by using image processing techniques to extract features. It is important that the features extracted are useful in distinguishing one image from another. GEFE uses Local Binary Pattern (LBP) [10] for extracting features from the images and storing them into feature vectors/templates. These feature vectors allow images to be mathematically compared to one another to determine how similar they are. Traditionally, the comparison is accomplished by utilizing a distance metric (e.g. Euclidean Distance or Manhattan Distance) to determine how close the images are to each other [7].

LBP works by dividing an image canvas into rectangular regions called patches. Within each patch, the LBP algorithm will iteratively select each interior pixel as a center pixel. Next, the intensity value of the center pixel is compared with its neighboring pixels in order to generate a texture pattern (bit string) for a given pixel. For each neighboring pixel, if the grayscale value is greater than the center pixel's grayscale value then a 0 bit is generated; otherwise, a 1 bit is generated. For each center pixel, an 8 bit binary string is generated that denotes the relationship between the center pixel's grayscale value and that of the 8 neighbors (top, top right, right, bottom right, bottom, bottom left, left, top left). Each patch is then treated as a histogram where the different bins consist of all the texture patterns or bit strings that are possible. The strings for each patch are concatenated in order to form feature sets or feature vectors.

It is possible to designate the number of features that are included in the extracted feature set of a given mouse movement session. For example, GEFE-56 uses feature sets of size 56 (per patch) while GEFE-256 uses feature sets of size 256 for each patch.

GEFE uses a genetic algorithm in order to select the best feature extractor no matter how many features are designated per patch [12]. This means that the genetic algorithm will be able to optimize the feature set to ensure that only the significant features are included in the feature vector. The size of the patches, the center of each patch, and which patches should be included in the feature vector are all decided by the genetic algorithm which evolves the feature extractor as the algorithm is run repeatedly. In contrast, the generic LBP method uses non-overlapping, uniform sized patches for matching.

The process of "evolving" a feature extractor is accomplished via the Estimation of Distribution Algorithm (EDA). An EDA will select a specified number of elites (candidate solutions with the best fitness) to be automatically included in the next population iteration. The remaining offspring in the population will be generated by choosing a subset of the current population to be used to create a probability distribution function (PDF). The PDF is then sampled to generate the remaining offspring for the next population.

The feature vectors for each mouse movement session of a given user will be stored in a profile, and new movements can be compared to the profile of a user to determine if the distance is within a certain threshold. This technique allows users to be authenticated (based on their mouse movements) with a fairly high accuracy rate.

ABM

Angle Based Metrics [15] is an approach to designing a biometric system that focuses on the angles that are generated by the mouse movements of a user. The angles are used to derive useful features or metrics which may be used to distinguish one user from another. The main advantage of this approach is that it works well even if the user's hardware or computing environment changes from one session to the next.

As with most biometric systems, the Angle Based Metrics approach is comprised of four different components: Recorder, Preprocessor (feature extractor), Classifier, and Decision Maker. The Recorder is the simplest of these components and is positioned on the client side of an application to capture user mouse movement events and send that data to the Pre-processor. The Preprocessor executes on the server side and is responsible for translating the data it receives from the Recorder into valuable metrics. There are 3 metrics which our Pre-processor calculates from the mouse coordinates and mouse clicks: the direction angle, curvature angle, and the curvature distance ratio. These metrics are calculated by examining groupings of 3 points -- in the order in which those points were visited by the user's mouse movement. Thus point A is visited before

point B, and point B is visited before point C (See figure 1).

- The direction angle (1) is the angle measured from a horizontal line to the line AB. Line AB is formed by traveling from the first point in the group of 3 to the second point.
- The curvature angle (2) is the angle ABC where A, B, and C are consecutive points read into the Preprocessor from the Recorder.
- The curvature distance (r) is as follows: for a line AC, let point Z be the point located from B to AC that is perpendicular to AC. Then the curvature distance is the ratio BZ/AC.

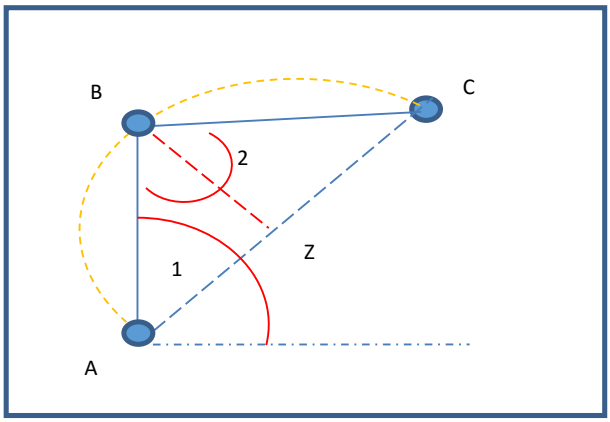

Figure 1: Illustration of Angle Based Metrics

The metrics calculated in the Preprocessor are organized as a cumulative distribution function (CDF), with intervals of direction angle (x), curvature angle (y), and the curvature distance/ratio (r). The CDF is a mathematical model that illustrates which percentage of a user's metrics fall within a given range of values. The percentage values within each CDF bin (interval) are what help to distinguish one user from another and are referred to as "features". The collection of all the features for a given session of user action is referred to as a feature set or template. The feature sets are used as input to the Classifier component of the ABM system.

The main task of the Classifier is to be able to tell whether or not a feature set or group of feature sets belong to a given user or not. There is more than one way to implement the Classifier. One way is to utilize a support vector machine (SVM). A support vector machine is a machine learning component often used for classification [14]. A SVM will take in a group of feature sets derived from a user and utilize them to create a training model of the user's mouse movement characteristics. Then, whenever new mouse data arrives, the SVM can compare the features of that data in order to determine whether the movements belong in the same grouping/classification with the other movements in the user's training model/profile.

Another classification technique utilizes the Normalized Manhattan Distance (NMD). NMD is calculated by taking the sum of the differences between two feature-sets (where each feature set is simply a list of percentages or floating point numbers) divided by the total number of features. For the purposes of our own analysis, NMD was the chosen method for comparison and classification. The NMD value represents how close mathematically a template is to those in a user's profile/training set. That value is sent to the Decision Maker component.

The Decision Maker is the component that is tasked with deciding whether the actions being generated by a user's session are similar enough to those movements saved under the user's profile to be considered a match. One way to do this is to establish a threshold value in order to be able to accept or reject a feature set based on the NMD value. Another approach is to utilize a SVM to determine whether or not a feature set may be classified with the other feature sets known to belong to a given user. The SVM will output a decision value to accept or reject, and that information may be utilized by the security mechanisms within a larger system in order to determine if a user needs to be prompted to re-authenticate or not.

ABM + GEC

All of the main components of the ABM + GEC approach are consistent with that of ABM. In fact, ABM + GEC can be considered an optimized version of ABM. Upon the initial implementation of the ABM system, it was observed that the greatest experimental results were achieved when the CDF bin sizes for the x, y, and r metrics were set to very small values. However, this presented a practicality problem because decreasing the bin sizes results in an increase in the number of features. This is due to an idea known as the curse of dimensionality, where it can be said that, as the number of dimensions in a vector problem increases, so does the complexity of the problem, and therefore, the time devoted to solve the problem increases as well. The natural relationship between the interval sizes and the magnitude of the feature set is an inversely proportional relationship, and so, as the size of the intervals decreased, the size of the feature set grew profoundly. For example, when using x and y intervals of .05, the feature set contained 2683 features. Because features represent the vector dimensionality of the authentication problem, this meant the system incorporated 2683 dimensions, and created an authentication environment that was very slow and difficult to manage. To solve this problem, a genetic algorithm toolset called X-TOOLSS [13] was used. The objec-

tive for using X-TOOLSS was to optimize the system by evolving new, smaller feature sets with larger intervals that could produce similar results -- in terms of authentication accuracy -- as the .05 intervals. In addition, X-TOOLSS eliminated redundant features which were non-essential to authentication. This process is called feature masking.

X-TOOLSS uses genetic algorithms (GAs) that, based on the "survival of the fitness" concept, develop optimal solutions for many types of parametric software systems. In this case, the feature masks and interval (bin size) combinations were designated as candidates. The GA evolves a population of candidate solutions by first generating random candidates and assigning fitness values to feature extractors implementing different versions of those candidates. Depending on the type of genetic algorithm being used, different methods are employed to create offspring from high-fitness "parent" candidates, and introduce those offspring into the next generation of the candidate population as a whole. Fitness values were calculated using the authentication accuracy of the candidate system (explained further as the Cumulative Match Curve (CMC) in the Comparisons and Results section). For the ABM + GEC system, a Steady-State GA was used, which stipulates that adding the offspring candidates to the population can only occur when those children have a higher fitness value than their parents. Therefore, the population size remains constant, or steady, throughout the evolution process.

The x, y and r intervals were evolved using double-precision 64-bit floating point values, between a range .5 and (large enough intervals to produce a more manageable volume of features), a population size of 20 individuals, a Crossover Usage Rate of 1.0, a Mutation Usage Rate of 1.0, a Mutation Range of .2, with 1000 total evaluations. These settings evolved new x, y, and r intervals of 6.024, 1.0, and 20.0 respectively. As for the feature mask evolution, the range was limited to the integers 0 and 1, and was applied to each feature in the template, representing either "on" (1) or "off" (0) for that corresponding feature. All other parameters for the Steady-State GA were the same as the interval optimization, save the number of total evaluations, which was 1000. The average results are based off of 10 runs of the GEC.

The evolution of the ABM system produced a remarkable complexity reduction from a 2683-dimensional system to a 283-dimensional system, using interval evolution, and then even further to a 150-dimensional one using the evolved feature mask. This resulted in an overall decrease in complexity of about 94.4%. The evolved system is far faster and more practical for real-world implementation; not only did the efficiency of the authentication system show improvement, the overall accuracy of the authentication improved as well (See Comparisons and Results section).

Pros and Cons of ABM and GEFE

One of the major benefits of both the ABM (including ABM+GEC) and GEFE approach to software biometrics and active authentication is that these techniques are able to effectively verify a user's mouse movements across different platforms without losing a significant amount of accuracy due to differences in hardware devices [15]. This is a major benefit over other metric approaches, such as speed and acceleration that are affected by the user's operating system as well as the mouse or the screen resolution [6]. Speed and acceleration are also poor metric choices due to the endless possibilities of situational diversity. For example, a user may quickly make a decision to advance toward and click a submit button, yet the same user may slowly advance and then pause before clicking a hyperlink on a text-rich web page such as a wiki article.

Another benefit of the ABM authentication approach over other authentication techniques lies in its generated data's minimal impact on user privacy. In the hands of a malicious culprit, mouse movement data would be of little use, as such data would not lend itself to reproduction. The mouse dynamics of a user can be compared to a signature; however, unlike the forging of a signature, where authentication is carried out once, an impostor would be required to continuously mimic the genuine user's biometric behavior throughout the duration of the session [2].

One possible hindrance that could be encountered by ABM authentication involves genuine users who undergo sudden biometric behavioral changes that render them unable to match up to their former biometric profiles. For example, a user could sustain a wrist fracture, causing a sudden change in mouse movement dynamics. Such occurrences, though rare, would possibly require intervention by system administrators to ensure the user is not falsely rejected from the system.

Experiment

The experiment that we developed was closely related to the experiment performed by J. Shelton et al. [12]. The mouse pointer was automatically centered on the screen and users were instructed to move the mouse in order to bring up the login box. The subjects were unaware of the purpose of the experiment.

We obtained and utilized the same data set used by Shelton. The data consisted of mouse movements collected for 16 unique subjects. Each subject had a "profile" comprised of 10 different sessions or sequences of mouse movements. Our experiment was to take a sequence (template) from any user and compare it with the profiles of all other users including the "self" profile to see if we could authenticate or verify a user based solely on their move-

ment pattern. The comparison was based on calculating the NMD between a single sequence and all of the other sequences in each profile. And based on a certain threshold value that we set for the NMD we were able to accept or reject each sequence as belonging to the owner of a certain profile or not. We were able to analyze the TAR and the FAR for ABM, ABM+GEC, and GEFE.

Comparison and Results

Our experimental results consist of the following categories: FAR, FRR, TAR, and the threshold. Note that the threshold is the independent variable but the results are also influenced by the interval that we utilized for the x, y, and r bins (representing the direction angle, curvature angle, and curvature distances respectively) in the CDF that generates the feature vectors. We selected a single template which we designated as a probe and we used all the remaining templates as our gallery set. The probe was then compared to every template in the gallery and if the NMD for probe and gallery member was less than or equal to the threshold value then this would count as an acceptance. True acceptances were those cases where both templates being compared belonged to the same subject and the NMD was below the threshold. A false acceptance occurred if the NMD for probe and gallery template was below the threshold but the templates did not belong to the same subject. And a false reject occurred if the NMD value was above the threshold but the templates were both from the same subject. We iterated through and allowed each of the 160 templates in our data set to have their chance to act as the probe and then designated the remaining 159 templates as our gallery set for each iteration. As we increased the threshold, the TAR value continued to increase towards 100%. Our best results were the ones that minimized FAR and FRR while maximizing TAR. When we set the threshold at .081, it yielded a TAR of approximately 70%, a FAR of approximately 42% and FRR of 30%. Likewise, while using a threshold of .0161 we calculated TAR of 90%, a FAR of 74% and a FRR of 10% (See Figure 2). These results are significantly better than what was achieved with GEFE. When the TAR for GEFE (specifically GEFE-256) approaches 80%, it yields a FAR 76%, and when the TAR reaches 90% it yields a FAR which is close to 90% as well.

We also computed a Cumulative Match Characteristic (CMC) in order to analyze the ABM technique. The CMC uses a single template as a probe and the remainder of the templates from all subjects (including self) in the population as the gallery. The CMC applies a rank for each probe to determine the percentage of templates which are able to find a match which belongs to the same subject on the first probe (rank 1), second probe (rank 2), third probe (rank 3), etc. The percentages on the CMC chart were calculated by letting every template in the population serve as the probe exactly one time. For a given rank, the percentage includes all the matches which were produced using x number of probes where x is less than or equal to the rank number. So rank 3, for example, includes the percentage of probes that found a match within 1, 2, or 3 attempts. A match occurs when a probe is compared with the population gallery and the template discovered to be closest in distance from the probe belongs to the same subject as the probe. If any attempt to find a match results in discovering a template that is closest in distance to the probe but belonging to a different subject, this is a "miss". After any miss, we removed all the templates from the population which belong to the subject which caused the miss.

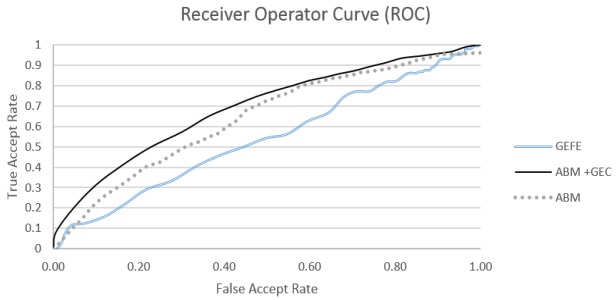

Figure 2: ROC results for ABM, GEFE, and ABM +GEC

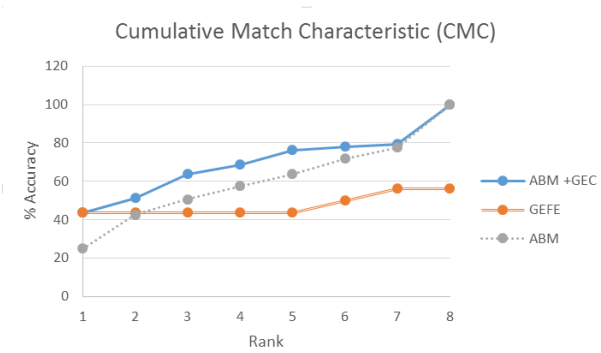

Figure 3: CMC results for ABM, GEFE, and ABM +GEC

The CMC results show that though GEFE has a considerably higher rank 1 accuracy of 43.75%, compared to ABM's 25.0% rank 1 accuracy, ABM begins to substantially outrank GEFE from rank 3, and beyond, including double digit differences in accuracy beginning with rank 4. (See Figure 3 CMC Chart). ABM + GEC further widens the accuracy gap, by matching GEFE's 43.75% rank 1 accuracy and greatly outperforming every other rank for GEFE, including double digit percentage leading from rank 3 and on.

Conclusions and Future Work

Based on the results we have tabulated and displayed in the ROC and CMC curves, it appears that the ABM + GEC technique is more accurate and more effective as a software biometric approach when compared to the GEFE. In addition, ABM+GEC is able to accomplish higher accuracies than standard ABM although using a significantly lower number of features.

Future work needs to be done in order to improve both the GEFE and ABM + GEC techniques if either strategy is going to become applicable to the mainstream authentication. Each approach will have to decrease the FAR while maintaining a high TAR. Also, the entire system needs to be modified and tested in a real time environment in order to better evaluate the feasibility of the technique for deployment in a production setting. The evolutionary computation that GEFE and ABM+GEC undergo can both take hours to run depending on the algorithm parameters. However, each system can be viewed as a feature "update" algorithm which would run as a background component to an AA system, as new data becomes available, to maintain optimal accuracy. Therefore, there should be little impact on user experience due to the speed of completion.

Furthermore, we would like to test the system on a larger pool of users in order to see how that affects the accuracy measurements. Some things to consider in a real time active authentication (AA) system also include: how many templates should be stored in a user's profile during training phase; and how long should each template remain in profile before being "aged out" by new templates.

Acknowledgements

We would like to thank Dr. Gerry Dozier and Joseph Shelton for their consultation on the technical methodology behind prior GEFE research at North Carolina A&T State University.

References

1. J. Adams, D. L. Woodard, G. Dozier, P. Miller, G. Glenn, K. Bryant. "GEFE: Genetic & Evolutionary Feature Extraction for Periocular Based Biometric Recognition," Proceedings 2010 ACM Southeast Conference, April 15-17, 2010, Oxford, MS.
2. A. A. E. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. IEEE Transactions on Dependable and Secure Computing, 4(3):165–179, 2007.
3. Ingo Deutschmann and Johan Lindholm. "Behavioral biometrics for DARPA's active authentication program". BIOSIG 2013: 225-232.
4. Dozier, G., Homaifar, A., Tunstel, E., and Battle, D. (2001). "An Introduction to Evolutionary Computation" (Chapter 17), Intelligent Control Systems Using Soft Computing Methodologies, A. Zilouchian & M. Jamshidi (Eds.), pp. 365-380, CRC press.
5. K. Jain, L. Hong, and S. Pankanti, "Biometric Identification" Commun. ACM 43, 90-98, 2, 2000.
6. Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS '11, pages 476–482, 2011.
7. Malkauthekar, M. D. "Analysis of euclidean distance and Manhattan Distance measure in face recognition", International Journal of Computer Science and Engineering (IJCSE) ISSN(P): 2278-9960; ISSN(E): 2278-9979 Vol. 3, Issue 4, July 2014, 89- 98.
8. Nazirah Abd Hamid; Suhailan Safei; Siti Dhalila and Mohd Satar. "Mouse Movement Behavioral Biometric Systems"; Kuala Terengganu, Malaysia.
9. L. O'Gorman, "Comparing Passwords, Tokens, and Biometrics for User Authentication," Proc. IEEE, Vol. 91, No. 12, 2019-2040, 2003.
10. Timo Ojala; Matti Pietikainen; Topi Maenpaa. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns".
11. Douglas A. Schulz, MOUSE CURVE BIOMETRICS, Pacific Northwest National Laboratory, U.S. Department of Energy
12. Joseph Shelton; Joshua Adams; Derrick Leflore and Gerry Dozier. "Mouse Tracking, Behavioral Biometrics, and GEFE"; In Proceedings of IEEE Southeastcon; 2013, p1-6, 6p.
13. Tinker, M. L., Dozier, G. & Garrett, A. (2010). The exploratory toolset for the optimization of launch and space systems (X-TOOLSS). *Available online: http://nxt.ncat.edu/*.
14. S. Tong. Support vector machine active learning for image retrieval. In Proceedings of the ninth ACM international conference on Multimedia, 2001.
15. Nan Zheng; Aaron Paloski; Haining Wang. "An Efficient User Verification System via Mouse Movements"; Williamsburg, VA, USA.