Statistical Analysis of Hand Biometrics for Structure-Borne Sound Authentication in Mobile Devices

Sahil Chatiwala
sahilchatiwala@gmail.com

Reva Hajarnis
reva.hajarnis@gmail.com

Alexei Korolev
alexeikorolev7@gmail.com

Danielle Park
danielle.kim.park@gmail.com

David Shenkerman
shenkermandavid@gmail.com

***Dr. Yingying Chen**
yingche@scarletmail.rutgers.edu

***Yilin Yang**
yy450@scarletmail.rutgers.edu

Governor's School of New Jersey Program in Engineering & Technology
July 22, 2022

*Corresponding Author

***Abstract*—Methods to protect information on personal devices are shifting towards biometric user authentication as technology advances. While facial recognition and fingerprint scanning are efficient and secure, they use personally identifiable data. Conversely, software for secure non-biometric authentication requires complex physical and memory-based tasks. Using structure-borne signals (sound compressions traveling through a solid medium) to recognize a user in contact with the device is a viable solution to the problem. Hand information about users possesses less of a privacy risk than that from the aforementioned authentication methods. The proposed experimental procedure leverages hand biometrics and machine learning from data collected on a mobile device to compare and explore the K-Nearest Neighbors, Support Vector Machines, and Bagged Decision Trees algorithms for authentication. This study aims to examine and elaborate on previous research with the goal of improving classification of distinct hand positions as well as investigating the strengths and weaknesses of each tested algorithm. The results show that the structure-borne signals of different users and hand positions both have statistically significant properties that enable accurate classification. Bagged Decision Trees provided the highest accuracy in both parts of the research with 93.6% accuracy in identifying the true user and 84.8% accuracy in identifying the hand positions. The algorithm excelled in security, rejecting false users with a 96.1% accuracy. This research demonstrates that acoustic sensing is an effective method of authentication that harnesses unique features of users' hands and positions with an emphasis on security.**

I. Introduction

Overview: Current approaches to authentication have raised personal data concerns as technology becomes more pivotal to validating a user's credentials [1]. Whereas over 60% of people use smartphones worldwide, knowledge-based verification such as passwords put users at risk due to their susceptibility to spoofing. These risks can be reduced with the implementation of artificial intelligence and active participation from the user [2]. With these challenges in mind, the advantages of hand biometrics in authentication become apparent due to their ability to provide security while maintaining accuracy and reliability. Acoustic sensing utilizes the structure-borne vibrations (noise created by impact) that propagate through users' hands to perceive details that make them unique. [3].

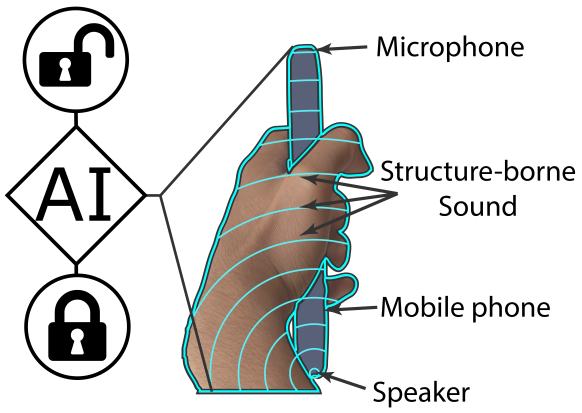

Fig. 1: An overview of using structure-borne sound to authenticate users. High frequency sound is emitted from the phone's speaker, travelling through and being modified by the user's hand. The resulting input from the microphone is fed into a machine learning algorithm that determines whether or not to authenticate the user.

Motivation: Widespread techniques of biometric user authentication in mobile devices do not currently include hand structure-based authentication. However, past studies have re-

vealed that artificial intelligence can effectively identify users from structure-borne vibrations [4]–[8].

The methodology being examined suggests that hand biometrics are feasible since structure-borne signals propagate predictably even with variations in light or background noise [9]. An advantage of this approach over the others aforementioned is its increased security through lower personal identifiability. In other words, even if a malicious entity is able to acquire user profiles of hand structure data, they would not be able to deduce information about a user's appearance or voice due to the feature extraction process. The only drawback is a situation in which attackers employ replay attacks to impersonate users [10]. These bypass the need to physically recreate topological characteristics of a user's hand by resending the authentication request of a previous instance. However, the ability to execute these types of attacks is diminishing with growing network security [3] [6]. Thus, this method of authentication also poses a minimal privacy risk to users.

Contribution: The research demonstrates the benefits of hand-based authentication that integrates artificial intelligence. To make the design resilient to various conditions and environments, we investigate differences in hands and positions to develop models that will perceive measurable differences in the transmissions. Overall, the proposed technique aims to simulate a high fidelity authentication system in which successful authentication of false users is minimized.

II. Background

A. Digital Sound Production

Mobile devices use speakers to translate digital data into mechanical sound waves. The digital sound is converted into an electrical signal with its voltage representing the sound waves that were captured by the microphone. The speaker on the device consists of a diaphragm with a coil attached to its inner side beside a permanent magnet. When the generated electrical signal runs through the coil, the fluctuating magnetic fields resulting from the flow of current interact with the magnetic field of the magnet. As a result, the diaphragm vibrates in a way that accurately represents the digital audio representation. This method is also used to play a single-frequency tone. High-pitched single-frequency tones known as chirps are a crucial part of data collection in this experiment.

To account for the physical limitations of the speaker, the diaphragm cannot be instantaneously accelerated to high frequencies. Chirp signals must be enveloped in a Hanning Window using weighted cosine curves in order to accelerate the diaphragm based on hardware specifications. Since the diaphragm isn't decelerated instantaneously at the end of the chirp either, the Hanning window is used to envelope portions of the end of the chirp so that there is no frequency leakage to cause disturbance in subsequent chirps.

Fig. 2: Diagram of the Samsung Galaxy S20 FE 5G microphone and speaker location, the AVR application used in data collection, and a sample hand grip. The structure-borne sound is depicted as cyan waves being emitted from the speaker.

B. Stereo Microphone Reading

A microphone is able to record audio by allowing a diaphragm to vibrate with the medium (often air) around it. Therefore, the microphone is also able to pick up sounds travelling through the phone itself (see Figure 2). The diaphragm synchronizes with sound waves, allowing a coil that is attached to the inner side of the diaphragm to vibrate beside a permanent magnet with the same frequency. Because the coil moves through the magnetic field, a voltage is induced in the coil with a magnitude that corresponds to the change in position of the diaphragm. Oftentimes, a microphone will record with separate left and right channels in a process called stereo microphone recording.

C. Biometric Data and Authentication

Biometric data is collected from parts of a person's body. These areas of the body are sensitive to the conditions of early development and have a large number of possible variations, making them effective identification factors of an individual. Biometric authentication refers to authentication techniques based on biometric data and includes fingerprint scans, retina/iris scans, facial recognition, and voice recognition.

Hand biometrics are already incorporated into mobile authentication through ultrasonic fingerprint scanners that create an accurate image of a user's fingerprint. Fingerprint scanning is a secure method of identity verification and normally takes around a second to complete [11]. However, these scanners require specialized hardware for authentication as well as active participation from the user. An individual must actively

press and hold the correct finger on the designated sensor area, increasing authentication time and inconvenience.

Other hand structure traits such as size, palm ridges, wrinkles, grip strength, hand shape, and contours are also candidates for use in biometric authentication. Due to all of these traits, the structure of an individual's hand is unique, making it a prime candidate for authentication [3]. Hand structure-based authentication using acousting sensing does not require active participation from users or special hardware due to a data collection process that makes use of built-in microphone/speaker functionality [4]. The lack of additional steps makes this form of authentication both user-friendly and secure [3].

D. Artificial Intelligence

Artificial intelligence (AI) is a term associated with software that can simulate human decision-making such as classification and prediction. In order to emulate these decisions, AIs are programmed to learn in methods specific to an algorithm. AI's role in mobile authentication is to create more secure and efficient methods of authenticating users, most commonly through the use of biometric data. Specifically, machine learning can be used to analyze features of users' structure-borne signals that allow the model to classify and authenticate effectively. The classification algorithms used in the experiment are the supervised learning algorithms: K-Nearest Neighbors (KNNs), Support Vector Machines (SVMs), and Bagged Decision Trees (BDTs). Supervised learning refers to the category of algorithms that are used when data has the input variables as features and classifications as labels. The software determines if the result was achieved accurately after each epoch, or iteration, of training.

The crux of supervised learning is in the loss function which minimizes the error of a model's predictions. To minimize this loss function effectively, the learning rate, or the rate at which the model reduces it's loss function, must be calibrated for different scenarios. Before the models are trained, certain numerical "costs" are assigned to each false predictions of the AI in MATLAB. These costs help inform how the learning rate will change based on the predictions made my the model at each epoch (iteration). All three algorithms improve their accuracy by trying to minimize the loss function and costs through a process called gradient descent. Gradient descent modifies the weights and coefficients of a model while attempting to reach the point of minimum cost. The location of minimum cost indicates that the model is globally optimized.

KNN classifies data points by first finding the Euclidean distance (see equation (1)) from each point in a data set to the test entry that is being classified. Euclidean distance determines the shortest distance to the nearest data points by finding the sum of the squared distances between coordinates in each dimension. In this case, equation (1) assumes that there are n dimensions, or features, in each data entry [12]. After determining these distances, the algorithm determines the *k* data points associated with the *k* lowest distances and classifies the desired entry accordingly. Since the distance to every data entry must be calculated for each entry that needs to be classified, KNN is computationally expensive algorithm for large data sets [13].

$$d_{\text{Euclidean}} = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \tag{1}$$

On the other hand, SVM is a method that uses training data to create the best separation of classes using a hyperplane. Unlike KNN, SVM has a training period but does not query the entire data set while making predictions. During training, the model begins by choosing an arbitrary coefficient for each feature in the data set and creates a linear combination of them, resulting in a hyperplane. Next, the Euclidean distance (1) from each data point to the hyperplane is computed, as well as the consistency of classification between data in the same class. After each epoch of training, coefficients of the hyperplane are updated and the same process is repeated with those coefficients. The model is trained until a reasonable level of accuracy is obtained. In SVMs, the confidence of the model can be evaluated by determining the margin, or distance of the closest point to the hyperplane. The higher the margin, the more confident the model is because the model finds larger distinctions between the different output classes.

Bagged Decision Trees (BDTs) refer to the bootstrap aggregation of multiple decision trees that are trained on separate subsets of the training data. Decision trees are machine learning models that make use of conditional statements to narrow input features into an output classification. These conditions are decided based on their ability to find qualities of features that directly affect their classification, and can be statistically optimized using methods such as Gini indexing, entropy analysis, etc. During training, these conditions are modified after each epoch on the training data to correct for inaccuracies. Although effective at classifying information that is similar to the training data, decision trees often overfit during training and subsequently perform poorly in practice. Bootstrap aggregation, or "bagging", of the trees fixes this issue by segmenting the data set into a specified number of subsets. Bootstrapping refers to performing calculations on multiple subsets of a data set instead of on the entire data set. Aggregation refers to the combination of the values calculated through bootstrapped subsets. Thus, BDTs actually refers to multiple decision trees trained on different combinations of features whose outputs are used collectively for classification. This reduces the effect of skews in data and are especially helpful for decision trees which often overfit. In this specific research, the mode of each tree's output is taken to provide the model's final prediction. Segmenting the data set in such a way and aggregating the results makes it so that overfitting of individual decision trees does not cause overfitting over the entire data set.

To enable the models to learn and classify data, it is important to have all collected data compiled into an array format. Feature extraction is the method by which features, or statistical properties of data points (mean, skew, etc.), are

compiled from the discretized microphone recordings into a user profile. During feature extraction, features that do not contribute greatly to the accuracy of the model can also be assessed and removed. Learning about these different statistical properties and finding patterns to determine the uniqueness of each person's structure-borne hand signals allows the algorithms to recognize and authenticate the user [14].

E. Cross-Correlation

Cross-correlation quantifies the similarity between two discrete signals by comparing each data point in one signal to each data point in the other. The cross-correlation calculation provides two outputs: firstly, the correlation coefficients calculated for each comparison, and secondly, the lag in time between points being compared. Higher correlations imply that the two points being compared are similar, while a lower correlation implies the opposite. Lag refers to the difference in time for when a data point in one signal occurs from a data point in the other signal. Using a comparison of both correlation and lag values, the points of maximum similarity between two signals can be found by identifying the highest calculated correlations. Then, using the lag values, the exact time that a similarity occurs is located [15].

F. Fast-Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is an optimized way to calculate the Discrete Fourier Transform (DFT), a function that converts a discrete signal from the time domain to the frequency domain. This means that taking the DFT of a signal and graphing the result produces a graph of power (energy of compressions) in terms of frequency [16]. This plot is called the power spectrum or the periodogram of the signal, and the power of a certain frequency is called the spectral density of the signal at that frequency. The equation for the Fourier Transform of a signal $f(x)$ in the time domain is given by its frequency content $F(\xi)$ in the equation

$$F(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx \tag{2}$$

Specifically, the FFT makes this conversion using the constructive/destructive interference of sinusoidal waves of various frequencies that sum to the original signal [16]. Peaks in the power spectrum of a signal represent the frequencies of the sinusoidal waves used in the signal's construction. If a certain frequency sinusoidal wave is used to construct portions of the signal during the FFT, its frequency is marked to have a high power, or peak, in the transformed graph of the signal. Conversely, if a certain frequency has a power value of zero in a signal, the sine/cosine wave with its frequency was not used during transformation at all (see Figure 2).

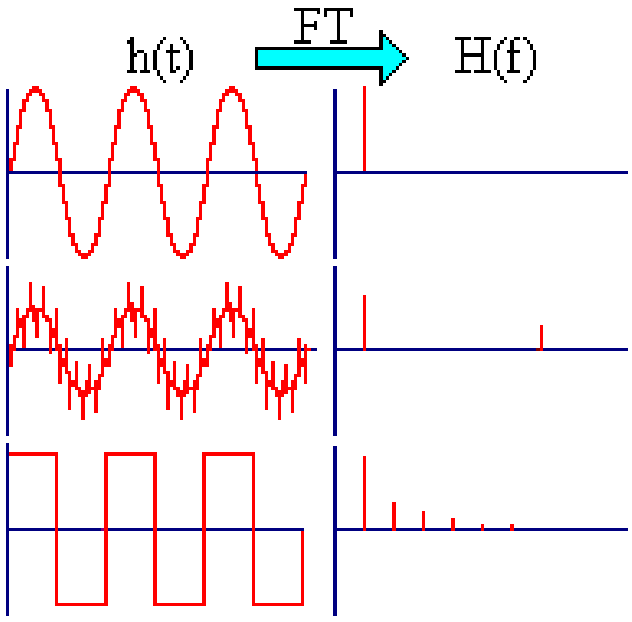

Fig. 3: The FFT maps waves from the time domain to the frequency domain. Peaks shown on the graphs on the right represent the frequencies of signals that interfere to form the signals on the left [17].

The FFT is an optimization of the DFT that runs significantly faster, especially for data sets with millions or billions of samples. More specifically, the FFT algorithm runs in less steps than the DFT algorithm for the same input. The FFT enables the calculation of the Mel Frequency Cepstrum Coefficients of a signal.

G. Mel Frequency Cepstrum Coefficients

Mel Frequency Cepstrum Coefficients (MFCCs) refers to the set of coefficients that describe the Mel Frequency Cepstrum of a signal. Both of these statistics are dependent on the Mel scale, which is a modification of the frequency scale that is based on the listeners' perception on the difference between frequencies. The Mel scale is based on the notion that the difference between lower frequencies can be perceived by humas more easily than the difference in higher frequencies [18]. For example, if a person perceives the difference between 1 Hz and 100 Hz to be the same as the difference between 1000 Hz and 2000 Hz, the space between 1 Hz and 100 Hz is the same as the space between 1000 Hz and 2000 Hz on the Mel scale.

After the FFT is applied to a signal, the resulting signal is converted from the frequency domain into the Mel scale domain (equation (3)). This is completed by means of the Mel-filter bank, a group of triangular filters, or filters whose weights vary linearly over the filtering interval instead of remaining constant such as in rectangular filtering. Specifically, the Mel-filter banks are bandpass filters that are effective at interpreting the energy at each frequency present in a signal as well as

estimating the powers of the frequency similar to the FFT.

$$\text{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right) \tag{3}$$

This denoises data so that analytic models can focus on more important audio features. After these filters have been applied, the Discrete Cosine Transform (DCT) of the resulting signal is computed; this calculation is similar to the DFT of a signal except that it only uses cosine waves for reconstruction of the signal. This transformation creates the graph of a signal's Mel frequencies over time, or the Mel Frequency Cepstrum. In this case, since the frequency unit is inverted through computation of the DCT, Mel Frequency Cepstrum is used to refer to the domain that is the inverse of the spectral domain (frequency).

While the Mel Frequency Cepstrum of a signal is useful for deep learning models, MFCC's, in the form of vectors, can be extracted from the Cepstrum in order to enable their use for machine learning models. These Mel Frequency Cepstrum Coefficients can be used to uniquely identify the Mel Frequency Cepstrum, and are useful to machine learning models in denoising data to more-clearly interpret features of sound signals.

III. Experimental Procedure

The authentication system utilized hand biometrics and artificial intelligence in four phases: signal design, data collection, data analysis, and training/testing the AI. These phases consisted of creating/recording chirp signals, cross-correlation, feature extraction, and AI learning/classification. Throughout this section, the four MATLAB scripts chirp.m, cross-correlate.m, feature extraction.m, and learning.m were used to describe processes that complete the first two aforementioned phases while the MATLAB Classification Learners Application was used to devise, test, and train models.

A. Signal Design

Using the above range of frequencies along with the hyperparameters of *chirps_per_signal* and *reps_per_freq*, the transmissions were created. For the data collected in this experiment, the signals consisted of 10 chirps (*chirps_per_signal* = 2), with two at each frequency ranging from 18 kHz to 22kHz, incrementing by 1 kHz. Each frequency was also repeated four times (*reps_per_signal* = 4) meaning that each signal consisted of 40 chirps, with eight at each frequency. Finally, in the signal used to gather microphone data, each signal was repeated about 20 times per transmission. During the recording, a user gripped the phone firmly, maximizing the surface area that was in contact with the phone as this would maximize the ability to distinguish of the structure-borne waves (see Figure 4). The transmission was then played, and the resulting structure-borne acoustic waves were recorded and plotted. This process was repeated 10-13 times each per user and saved for analysis.

Fig. 4: The pictures detail examples of differing hand positions that were used while recording audio files. The top left shows Subject 2, the top right Subject 1, the bottom left Subject 4, and the bottom right Subject 3. The phone pictured was used when recording data.

B. Data Collection

The Android mobile application, Amazing Voice Recorder (AVR), was used to record data. First, parameters were tuned to generate the transmission and set the sampling frequency; the minimum and maximum transmitted frequencies were 18 kHz and 22 kHz, respectively, while the sampling frequency was 48 kHz. The sampling frequency was determined using the Nyquist-Shannon Theorem (see equation(4)) which states that an accurate discretization of a continuous audio signal can be achieved by sampling (f_{sample}) at a frequency that is at least twice the greatest frequency (f_{max}) in the signal [19].

$$f_{\text{sample}} \geq 2 f_{\text{max}} \tag{4}$$

In order to maximize the frequency that the audio signals could be transmitted, the sampling frequency was set to 48 KHz, which was the maximum sampling capability of the Android devices used. However, it was determined that the optimal frequency to transmit chirps would be between 18 kHz and 22 kHz as these are inaudible frequencies for humans and also fit comfortably within the limitations of the theorem.

C. Data Analysis

Given the audio file containing the microphone recording of a user's structure-borne sound transmission, we used a series of statistical processes to extract the desired sequence. First, the left channel file for the recording of each structure-borne signal was loaded into the MATLAB workspace. Next, the bandpass filter, which is a device that passes frequencies within a certain range and rejects frequencies outside that range, was used to remove frequencies out of the microphone recording that was outside of the range of 18kHz to 22kHz. This did not affect the structure-borne signals because the medium of transmission does not affect the frequency of the transmission. [20]. Thus, the structure-borne signals still shared the same frequencies as the chirps and only differed in amplitude. The sequence was then divided by the number of signals, chirps

frequencies, and individual chirps. After this post-recording segmentation, the structure-borne transmissions were cross-correlated with the original transmitted audio to determine the point in the recorded audio where the structure-borne waves began to enter the microphone. From these cross-correlation coefficients, the point at which the structure-borne signal began was determined. If cross-correlation overestimated the start point, the next highest coefficient would have been found; if found too late, the system would have executed a catch function.

After the start of the signal was found in the array of microphone data, the transmission was extracted using the known length of the transmission and its correspondence to indices in the recording. Next, using the matrix of values for the structure-borne signals, statistics such as mean, spectral coherence, kurtosis, etc. were calculated. These statistics were used when creating a unique user profile that helped train the AI model. Additionally, using MATLAB's plot interface, two plots were created to get a better visualization of the chirps, signals, and transmissions. In the first, the amplitude was graphed against time while in the second graph, the amplitude was graphed against frequency.

D. Training/Testing the AI Model

Models were designed, trained, and tested in the MATLAB Classification Learners application by importing the compilation of all user profiles and their corresponding labels. The raw data was interpreted and 315 different statistics were extracted from the file. Some of these included standard deviation, skew, and FFT variations (see Figure 4).

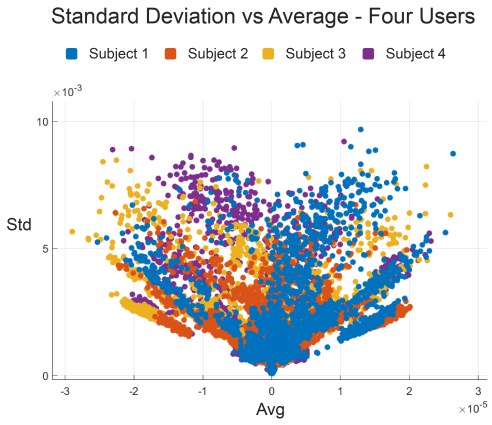

Fig. 5: The scatter plot of the standard deviation and average data points for four of the subjects.

These features aided the machine learning process and created clear numerical signatures for different transmissions. The size of the data for user samples is the product of the number of chirps in the training data and the number of users involved in classification. The Classification Learner application offers several base model architectures along with multiple variations of each model. Some examples include linear, quadratic, and cubic SVMs which were all tired for the research. Of the available options, such as Neural Networks and Naive Bayesian Classifier, the data was tested using SVM, KNN, and Bagged Decision Trees because they have proved to be the most accurate and efficient algorithms. Different combinations of data were used - factors such as the number of people/hand positions for the AI to differentiate between as well as the number of recordings to analyze were all varied in each test. Additionally, to prevent overfitting of the model, k-Folds Cross-Validation, in which the training data is divided into a specified number of equally-sized subsets, was used with a certain amount of the subsets used for training and the others used for validation during each epoch. This guarantees that the model does not see the same data for the entire training phase, and also provides a metric for the model to evaluate with "new" data periodically.

IV. Results

A. Subjects

The authors of this paper participated as the five subjects in this study; their hand biometric data was all sampled on the same Samsung Galaxy S20 FE 5G smartphone. The subjects had varying hand shapes, sizes, and positions which were compared to train the model and test its accuracy.

B. Training and Testing the Models

The SVM, KNN, and BDT models were trained on data divided into two parts: an authorized user's recordings and those of the other three subjects. In this test, seven unique recordings were used per subject. Then, the dataset was divided with a training-test ratio of 90% to 10% for the model to use. We found that the BDT algorithm was the most accurate with an average testing accuracy of 91.2% and an average training time of around 150 seconds for 28 individual recordings. The Weighted KNN algorithm had an average accuracy of 85.8% and consistently took 37 seconds to complete. SVM algorithms had an average accuracy of 85.3% with Fine and Medium Gaussian variations performing the best (see Figure 6).

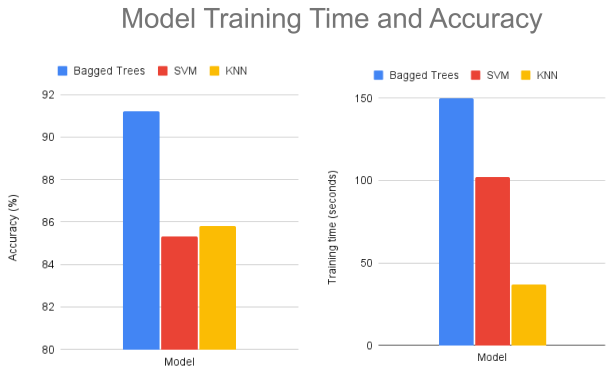

Fig. 6: The three main models used (Bagged Decision Trees, SVM, and KNN) analyzed based on their training time and accuracy. Accuracy and time averages were computed from four different trials. The model attempted to differentiate a valid user from all others combined

with the valid user changing with each trial.

The validation and testing accuracies revealed key data about which algorithms were most effective. Of the three algorithms used, Bagged Decision Trees were the slowest but most accurate across all domains. SVM was also able to analyze the data relatively quickly and provided accurate predictions about identifying users with particular efficacy in denying authentication to false users. KNN was both slower and less effective with accuracy decreasing as the structure-borne data set was expanded. KNN's prediction time as well as its lower accuracy made it the least successful model of the three examined extensively.

The average training time for the Medium Gaussian algorithms was 103 seconds, with the fine Gaussian algorithm taking around 253 seconds to complete. These measurements demonstrated the trade-off that exists between accurate and more generalized algorithms. An algorithm like BDT is more accurate, but has a higher training time compared to SVM and KNN.

When comparing a single user to all other users, all algorithms had a pattern of high security, or a low false validation rate, confirming the security of this method of authentication. Although the algorithm may occasionally deny access to the true user, it is highly unlikely to falsely validate an unauthorized user. While the models had an accuracy within the upper 80% or low 90% range, their accuracy for detecting an invalid user was consistently within the middle to upper 90%, indicating strong security. For example, in Figure 7, the BDT algorithm could only identify the valid user's hand signature correctly 79.2% of the time. Nonetheless, it was able to reject an invalid hand signature in 96.1% of test cases.

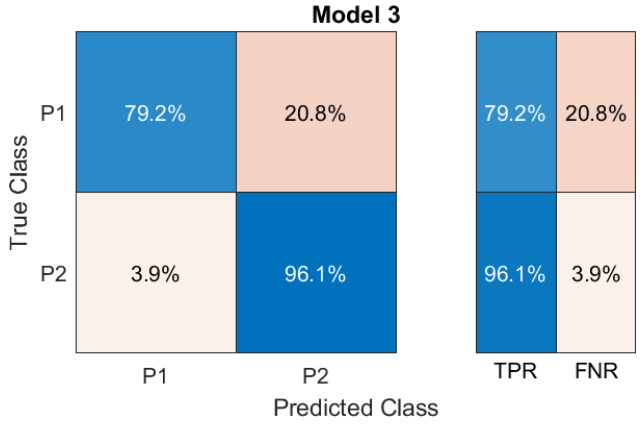

Fig. 7: The confusion matrix for the Bagged Decision Trees model which analyzed Subject 3 in comparison to the subjects.

The accurancy and security of a model can be adjusted by modifying the learning rates of the algorithms for each model and changing the target loss values. Specifically, by adjusting the "cost" of each mistake in the Classification Learners Application, the model was made to reflect the emphasis on security or accuracy. Focusing on the BDT algorithm with Subject 3 as the authorized user, we studied the different costs of a false validation result with the other constant. By adjusting the misclassification costs, the security and reliability of the model could be altered (see Figure 8).

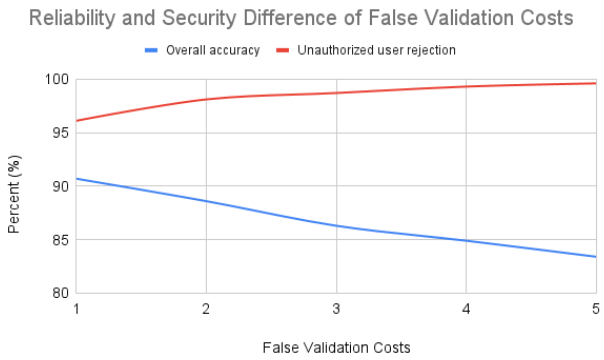

Fig. 8: As the false validation cost linearly increases, the security of the algorithm increases at what appears to be a logarithmic scale while total accuracy begins to decrease rapidly.

When the cost proportion was skewed to more heavily penalize a false validation, the algorithm was able to improve its security. Its false validation percent fell from 3.9% to 0.4% when the false validation cost was scaled five-fold. However, the algorithm began to misclassify an authorized user more often, with the false negative rate increasing from 25.4% to 65.4%. When prioritizing accuracy by adjusting the costs, the reliability of a model decreased. In other terms, both accuracies are inversely proportional to each other. While the security of a model increases at a seemingly logarithmic rate, its accuracy drops linearly.

Throughout the entire experiment, BDT stood out as an incredibly reliable method for determining the complex relationships between hand biometrics and the extracted signal features. For one trial, a single user, Subject 5, (not involved with other trials) recorded data with three contrasting hand positions (see Figure 9).

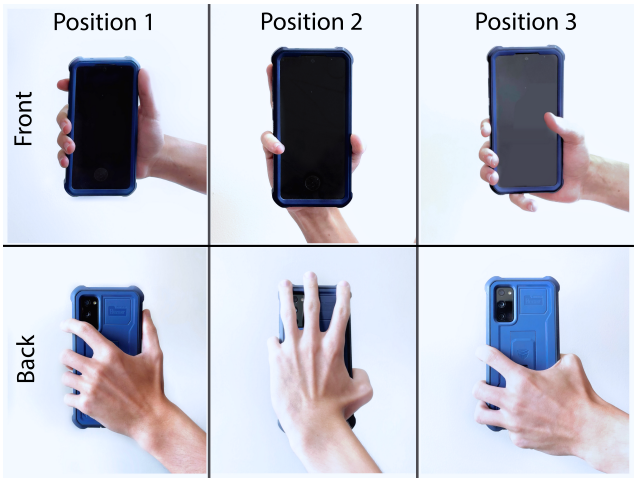

Fig. 9: Subject 5's varying hand positions.

When training the SVM, KNN, and BDT algorithms to differentiate between the different positions, both SVM and KNN lost a significant amount of accuracy. The SVM algorithm's validation accuracy was initially 65.3%, dropping to an accuracy of 63.7% when tested. KNN had a similar result, with a validation accuracy of 71.7% and a test accuracy of 69.0%. The low overall accuracy as well as the drop in accuracy when testing indicated that the SVM and KNN algorithms struggled to identify the correlation between different hand positions and the set of statistical features. Also, these algorithms were overfitting, which caused the accuracy of the models to decrease when they were tested on data that they were not exposed to before. Nonetheless, the BDT algorithm had a validation accuracy of 84.0% which only rose to 84.8% when tested. In addition to being more accurate, the BDT model exhibited less overfitting than the SVM and KNN models. Even so, BDT took the longest to train, with a time of 48.3 seconds. SVM took 25.3 seconds, and KNN took 10.5 seconds to train. Nonetheless, BDT demonstrated their ability to be most able to identify common patterns present within similar hand positions as well as being able to distinguish between multiple hand positions accurately.

To test the ability of these algorithms to differentiate between multiple users, we tested a multiclass BDT model with each class representing a unique user. The BDT algorithm was the most effective at identifying users, with an 82.7% test accuracy. The algorithm had its highest accuracy with Subject 2 at 87.1%, and its lowest accuracy with Subject 3 at 77.3% (see Figure 10).

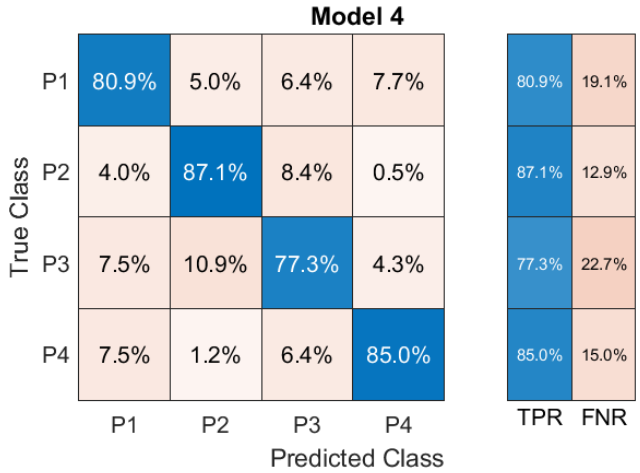

Fig. 10: The confusion matrix for the Bagged Decision Trees model which classified users into their distinct user profiles.

The Gaussian SVM algorithm performed similarly, with an overall accuracy of 81.5%, with Subject 2's data yielding the most accurate predictions at 82.4% testing accuracy, and Subject 3 yielding the lowest at 74.0% (see Figure 11).

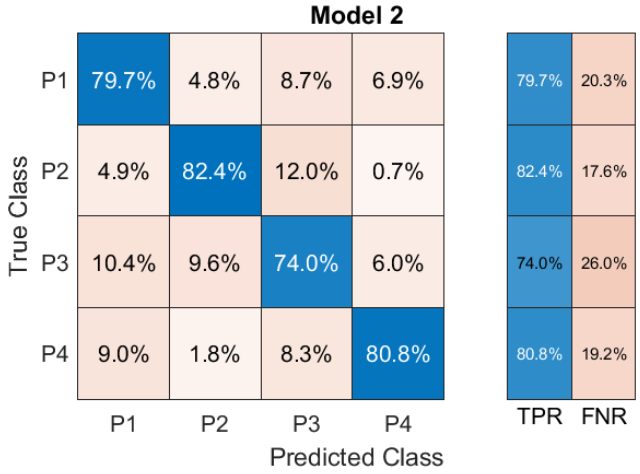

Fig. 11: A confusion matrix for SVM when attempting to identify each of the four subjects.

While the KNN model had a maximum accuracy of 82.1% for Subject 2, its overall accuracy was only 76.6%. The main reason for the reduced accuracy was the model's confusion between Subjects 1 and 2, which it was able to identify with an accuracy of 65.6% and 62.7% respectively. These findings suggest that KNN may not be able to distinguish between fine details and multiple classes (see Figure 12).

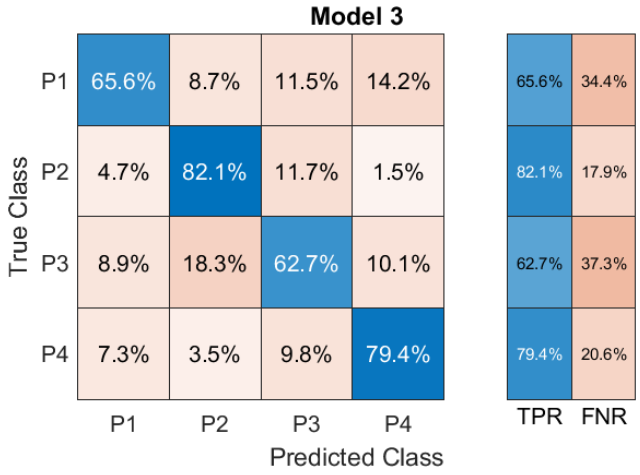

Fig. 12: A confusion matrix for KNN when attempting to identify each of the four subjects.

Eliminating 6 recordings from the data and retraining the models dropped the SVM accuracy to 80.0% while increasing the accuracy of both the BDT and Weighted KNN algorithms to 93.8% and 88.8% respectively (see Figure 13). The increase in accuracy due to a reduction in data is a sign that an algorithm was overfitting. Therefore, SVM may be more suitable for applications where there is a limitation on the quantity of data gathered, due to its lack of overfitting. KNN and BDT have a higher accuracy, but are more susceptible to overfitting and should be implemented in cases where there is a substantial amount of diverse data.

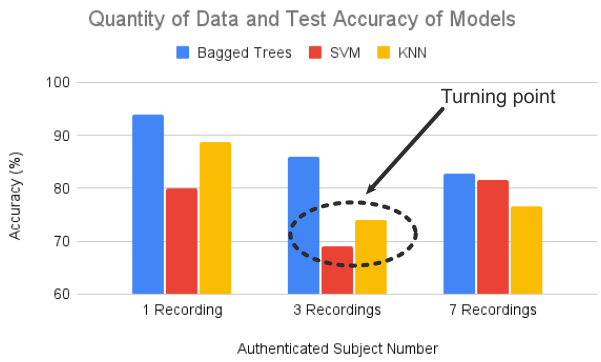

Fig. 13: The accuracy of the SVM, KNN, and Bagged Decision Trees models depending on the quantity of recordings.

As the amount of data increased, the algorithm was forced to become more generalized. The sharp reduction in accuracy for the SVM and KNN models when increasing the number of recordings per person from one to three was likely caused by the models not having enough data to generalize their predictions and the amount of overfitting decreasing from the diversification of data. After that point, the accuracy of both models increased, indicating that they were identifying a reliable and general pattern to distinguish data. However, the BDT saw a constant decrease in its accuracy as the amount of data increased, a sign that the model may be susceptible to overfitting with low amounts of data.

V. CONCLUSIONS

A. Significance of Findings

The research resulted in statistically significant evidence that showed hand biometric authentication to be both accurate and secure. Through analysis of our data, we confirmed that there was a noticeable difference between the structure-borne signals produced by different hands and positions. Keeping other variables constant, only the user's hand or their grip was a factor that changed in the two parts of the experiment. The resulting data had significant difference in its extracted features which we were able to isolate statistically using the machine learning algorithms BDTs and Weighted KNNs. The BDT model took the longest to train and resulted in the most accurate predictions for both parts of the experiment. Specifically, BDTs provided high security against the authentication of false users in both the different users and hand position trials. In contrast, weighted KNN was a faster-performing algorithm that achieved high reliability, but declined in accuracy when a bigger data set was used. Additionally, the model's learning rate was manipulated by assigning different training cost combinations. This helped to tune the model specifically towards security or reliability; for the security-orientated goal of the research, the cost combinations were adjusted so that true user authentication accuracy was not increased to a point that compromised model security. Therefore, for mobile device authentication, we conclude that BDTs would be most effective for preserving security of the system. A lower security version of the model is useful for applications where there is a smaller privacy risk such as a personally-managed system. The software repository for the collection, preprocessing, analysis, and machine learning model of the data is available at https://github.com/sahil485/TouchBasedAuth. The method of using structure-borne sound for user authentication can revolutionize the field of mobile security by making user authentication both seamless and secure.

B. Limitations

A major limitation faced was the amount of subjects available for data collection. Four subjects were available for the proposed experimental procedure, two of which had a similar hand size. This could have slightly skewed user profiles and confounded the performance of the models, especially for the classfication of the four users in the first part of research.

Another limitation of the study that prevents generalization of the results is the maximum sampling frequency of the device used in testing. Mobile devices are not built to universal standards meaning that they have different microphone qualities with different maximum sampling frequencies. As a result, the clarity of structure-borne waves recorded in this experiment may not be consistent among users of all mobile devices. If the microphones recorded audio with a sampling frequency of above 48 kHz, the AI models could been able to differentiate between samples more effectively with higher frequency chirps. Lower quality microphones on the other hand might have required a different approach for data collection or preprocessing to extract the necessary contrasts for authentication.

C. Further Improvements

To improve the applicability of this solution, a variety of mobile devices with differing recording capabilities should be considered to simulate different user experiences. Better feature extraction techniques and denoising transformations would also be useful in improving this model for environments with higher ambient sound levels. A greater number of subjects are desirable to create a more statistically relevant and diverse sample. With such a sample, there could have been more statistical certainty in model's ability to accurately and securely authenticate users. In addition to conducting trials with a greater number of participants, every participant should collect data from multiple hand positions. While SVM, Bagged Decision Trees, and KNN were found to have the best performance, further research into an ensemble model to incorporate both user identification and hand positions adds another layer of security for authentication. By using the proposed hand biometrics solution, we are mitigating modern cybersecurity risks and inconveniences by proposing a model to secure our future devices.

ACKNOWLEDGMENT

The authors of this paper gratefully acknowledge the following: Rutgers School of Engineering; Rutgers University; The

NJ Office of the Secretary of Higher Education; Governor's School Alumni for their continued participation and support; Dean Jean Patrick Antoine, the Director of the Governor's School of New Jersey Program in Engineering and Technology 2022 (GSET) for his management and guidance; project mentors Dr. Yingying Chen and Teaching Assistant Yilin Yang for their direction and expertise; Residential Teaching Assistant Benson Liu for his invaluable assistance; and Research Coordinator June Lee for her advice on conducting proper research.

REFERENCES

[1] S. Arora and M. P. S. Bhatia, "Fingerprint Spoofing Detection to Improve Customer Security in Mobile Financial Applications Using Deep Learning," Arabian Journal for Science and Engineering, vol. 45, no. 4, pp. 2847–2863, Oct. 2019, doi: 10.1007/s13369-019-04190-1.

[2] S.-J. Kim, J.-M. Kim, and I.-J. Jo, "Multimedia image data processing on smartphone for authentication," Multimedia Tools and Applications, vol. 78, no. 5, pp. 5287–5303, Feb. 2018, doi: 10.1007/s11042-017-5600-2.

[3] R. K. Rowe, U. Uludag, M. Demirkus, S. Parthasaradhi, and A. K. Jain, "A Multispectral Whole-Hand Biometric Authentication System," Oct. 2007, pp. 1–6. Accessed: Jul. 10, 2022. [Online]. Available: https://www.researchgate.net/publication/4311442_A_Multispectral_Whole-Hand_Biometric_Authentication_System

[4] . Yang, Y. Chen, Y. Wang, and C. Wang, "Echolock: Towards Low-effort Mobile User-Identification Leveraging Structure-born Echos," in ASIA CCS '20: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, Oct. 2020, pp. 772–783. Accessed: Jul. 04, 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3320269.3384741

[5] C. Qiu and M. W. Mutka, "Silent Whistle: Effective Indoor Positioning with Assistance from Acoustic Sensing on Smartphones," in IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Macao, China, Jun. 2017, pp. 1–6. doi: 10.1109/WoWMoM.2017.7974312.

[6] L. Lu et al., "Lip Pass: Lip Reading-based User Authentication on Smartphones Leveraging Acoustic Signals," in IEEE INFOCOM 2018, Honolulu, HI, USA, Oct. 2018, pp. 1466–1474.

[7] M. Goel et al., "SurfaceLink: Using Inertial and Acoustic Sensing to Enable Multi-Device Interaction on a Surface," in CHI '14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Ontario, Toronto, Canada, Apr. 2014, pp. 1387–1396. Accessed: Jul. 10, 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/2556288.2557120

[8] L. Lu, J. Yu, Y. Chen, and Y. Wang, "VocalLock," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 4, no. 2, pp. 1–24, Jun. 2020, doi: 10.1145/3397320.

[9] L. Lu et al., "Lip Reading-Based User Authentication Through Acoustic Sensing on Smartphones," IEEE/ACM Transactions on Networking, vol. 27, no. 1, pp. 447–460, Feb. 2019, doi: 10.1109/tnet.2019.2891733.

[10] Y. Lee, J. Li, and Y. Kim, "MicPrint: Acoustic Sensor Fingerprinting for Spoof-Resistant Mobile Device Authentication," in MobiQuitous '19: Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Texas, Houston, USA, Nov. 2019, pp. 248–257. Accessed: Jul. 01, 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3360774.3360801

[11] C. Sousedik and C. Busch, "Presentation attack detection methods for fingerprint recognition systems: a survey," IET Biometrics, vol. 3, no. 4, pp. 219–233, Dec. 2014, doi: 10.1049/iet-bmt.2013.0020.

[12] G. Pepe, L. Gabrielli, S. Squartini, and L. Cattani, "Designing Audio Equalization Filters by Deep Neural Networks," Applied Sciences, vol. 10, no. 7, pp. 2483–2504, Apr. 2020, doi: 10.3390/app10072483.

[13] J. Liu, C. Wang, Z. Tu, X. A. Wang, C. Lin, and Z. Li, "Secure KNN Classification Scheme Based on Homomorphic Encryption for Cyberspace," Security and Communication Networks, vol. 2021, pp. 1–12, Nov. 2021, doi: 10.1155/2021/8759922.

[15] M. Krumin and S. Shoham, "Generation of Spike Trains with Controlled Auto- and Cross-Correlation Functions," Neural Computation, vol. 21, no. 6, pp. 1642–1664, Jun. 2009, doi: 10.1162/neco.2009.08-08-847.

[16] S. Bakheet, A. Al-Hamadi, and R. Youssef, "A Fingerprint-Based Verification Framework Using Harris and SURF Feature Detection Algorithms," Applied Sciences, vol. 12, no. 4, p. 2028, Feb. 2022, doi: 10.3390/app12042028.

[17] C. E. Efstathiou, Fast-Fourier Transform. Accessed: Jul. 16, 2022. [Online]. Available: http://195.134.76.37/applets/AppletFourAnal/Appl_FourAnal2.html

[18] V. Tiwari, "MFCC and Its Applications in Speaker Recognition," International Journal on Emerging Technologies, pp. 19–22, Feb. 2010.

[19] Z. Song, B. Liu, Y. Pang, C9 Hou, and X. Li, "An Improved Nyquist–Shannon Irregular Sampling Theorem From Local Averages," IEEE Transactions on Information Theory, vol. 58, no. 9, pp. 6093–6100, Sep. 2012, doi: 10.1109/tit.2012.2199959.

[20] A. Abadleh, B. M. Al-Mahadeen, R. M. AlNaimat, and O. Lasassmeh, "Noise segmentation for step detection and distance estimation using smartphone sensor data," Wireless Networks, vol. 27, no. 4, pp. 2337–2346, Mar. 2021, doi: 10.1007/s11276-021-02588-0.