USER SYNDICATION SYSTEM USING SPEECH RHYTHM

by

Faisal Alnahhas

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science at The University of Texas at Arlington

August 2019

Arlington, Texas

Supervising Committee:

Ming Li, Supervising Professor

Dajiang Zhu

Changkai Li

ABSTRACT

USER SYNDICATION SYSTEM USING SPEECH RHYTHM

Faisal Alnahhas, M.S.

The University of Texas at Arlington, 2019

Supervising Professor: Ming Li

In recent years we have seen a variety of approaches to increase security on computers and mobile devices including fingerprint, and facial recognition. Such techniques while effective are very expensive. Voice biometrics, specifically speech rhythm, is a method that has been drawing attention and growing in recent years. Unlike other methods, it requires little to no additional hardware installed on a device for it to work accurately. Speech rhythm utilizes the device's built-in microphone, and analyzes speakers based on features of their speech. In this work we leverage the existing hardware and simply add an efficient layer of software to achieve user authentication. When the user speaks a passphrase, voice features are extracted and passed on to a neural network that analyzes those features and classifies whether the speaker is a recognized user or not. The reduced cost, coupled with the efficiency of speech rhythm makes it appealing to a variety of devices, as well as large base of users. 13 users participated in this study and yielded 93.3% accuracy. The results are robust and show a lot of promise for future work.

ACKNOWLEDGEMENTS

I would like to thank Dr. Ming Li, my supervising professor, for her guidance and support in this research project. Dr. Li provided tremendous guidance and direction in my thesis work, from getting started with reading relevant papers, to set up lab equipment and space for experimentation, to the final stages of testing and writing my thesis.

I also would like to thank the rest of the committee members, everyone who participated in the experiments for the thesis, as well as the professors and PhD candidates from the linguistics department who gave me insight and direction on how to approach speech rhythm analysis.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

API – Application Programming Interface

MLP – Multilayer Perceptron

VOT – Voice Onset Time

ANN – Artificial Neural Network

ARFF – Attribute Relation File Format

F – Formant

α - Learning rate

e – epochs

H – Number of hidden layers

FAR – False Acceptance Rate

FRR – False Rejection Rate

TPR – True Positive Rate

TNR – True Negative Rate

CHAPTER 1

INTRODUCTION

Over the past two decades mobile technology has transformed our daily life. As more personal and sensitive information is stored and carried on mobile devices, a greater need for security arises. Classically authentication methods for mobile users relied on pass codes, passphrases, or patterns. More recently we have seen evolution of fingerprint recognition, and FaceID. Such metrics, while powerful and effective, can be costly. The added layer of hardware comes with increased design, material, manufacturing, and software costs. In addition, all these methods require the user to be in direct contact with the device. On the other hand, relying on voice as a biometric for user authentication is practical because it requires no additional hardware, other than the simple built-in microphone in the device. Moreover, it does not require the user to be in direct contact with the device.

Unlike any other form of authentication, speech rhythm (and all voice biometrics) has a special advantage of being naturally integrated with capable devices. The first cellular phone call was made in 1973 (Dyroff, 2018). Since then, mobile devices have evolved to include more powerful technology and processing power. We utilize the very foundation of mobile devices in this research project. This way we maintain the low-cost and supply high reliability of speech rhythm. Recently, using voice biometrics for authentication has gained popularity and we are starting to see it in a lot of applications, those include access control, forensics, and banking (Si Chen, 2017).

Particularly, with the advances of mobile technologies, voice authentication is becoming increasingly popular in a growing range of mobile applications. For instance,

voice biometrics have been integrated with smartphone operating systems and mobile apps for secure access and login, this includes Google's "Trusted Voice" for Android devices (Google, 2019), Lenovo's voice unlock feature for smartphones (Millward, 2012), Tencent's "Voiceprint" feature in WeChat for voice based app login (Voiceprint, 2015), and Twilio's VoiceIt voice biometric authentication (Twilio, 2019). We have also seen voice biometrics used in e-commerce and mobile banking. Saypay, a company that provides biometric authentication solutions, provides voice authentication services for e-commerce online transactions (Saypay Technologies, 2017). While banks like HSCBC and Barclays introduced voice authentication for their mobile users (Tode, 2017; HSBC, 2019). These examples show us how voice biometrics authentication systems are gaining momentum in mass-markets. The predicted market share value is up to $21.4 billion by 2024 (MarketsAndMarkets, n.d.).

Modern voice authentication systems are progressing drastically to achieve higher accuracy and reliability. For example, current service providers such as Nuance (VocalPassword, 2016), and VoiceVault (VoiceVault, 2019) provide highly successful challenge-response based voice authentication systems. Just like all other security systems, voice authentication has its drawbacks. Current systems are designed in a way that is cumbersome for the user. Modern service providers define a set of phrases that the user has to repeat, on top of the user chosen phrases (Almog Aley-Raz, 2017). This approach, while increases reliability, does come with increased overhead. In addition, voice-based systems require reasonable physical proximity between the user and the device to successfully authenticate users. As we will see later, there's a significant drop in accuracy when users are not close to the device.

We have also seen more novel approaches; Zhang et al. calculate the time-difference-of-arrival between the two built-in microphones of the phone (Linghan Zhang, 2016). The system requires the device to be held at a specific location for the maximal effectiveness. This also might be bothersome for the user. Moreover, Chen et.al developed a system that measures the magnetic field emitted from loudspeakers in smartphones. Their system requires moving the phone in a pre-defined path (Si Chen, 2017). The above solutions, while novel and cheap require extra work from the user. Constraining users by a specific location, or motion can be cumbersome, and therefore impractical. Successful modern technology takes little effort from the user, while maintaining reliability and accuracy.

In this paper we propose a rhythm-based system that provides reliable and flexible authentication system. We aim to provide the users with a hassle-free method that delivers high quality security, as well as success in a variety of scenarios. One distinct advantage of using rhythm, over traditional authentication methods and voice-based systems, is authentication from distance. Our system presents a solution for an everyday problem for users worldwide. The practicality of speaking to a device from a distance (different room for example) comes in handy in a lot of aspects of daily life. Mostly, for users with central home units like Alexa (Boyd, 2018), or Apple's Siri (Bell, 2015) where the device is not always within close physical proximity from the speaker.

CHAPTER 2

RELATED WORK

In recent years there has been quite a few attempts to come up with smart alternatives to traditional user authentication methods. While voice biometrics have taken a step forward and succeeded, studies also show a rapid increase in spoofing attacks (Arthur Janicki, 2016; Andreas Kipp; Kaavya Sriskandaraja; Zhinzheng Wu, 2014). Current systems struggle to defend against replay attacks (Philip L. De Leon, 2012; Rosa Gonzalez Huatamaki, 2014; Zhi-Feng Wang, 2011). A recent study showed a major increase in EER (from 1.76% to 30.71%) under replay attacks. The aforementioned commercial systems rely on challenge-response method to authenticate users. Such methods require certain steps from the user to work, which can be considered cumbersome.

We have also seen other forms of rhythm-based authentication systems that are not speech related. A team of researchers from University of Nevada, Reno introduced *Beat-Pin: A User Authentication Mechanism for Wearable Devices Through Secret Beats*. In their paper they introduce an authentication system for wearable devices using timing of beat sequences for direct authentication to wearable devices. The work shows accurate authentication, low processing overhead, and is fairly convenient for users. (University of Nevada, Reno, 2018).

Other rhythm-based systems that have surfaced lately including Wobbrock's Tapsongs, a system which authenticates users on a single binary sensor. Their system relies on matching rhythmic taps to a jingle timing model that the user creates. The withdraw on this system is the high rate of false rejections at 16.8% (Wobbrock). We

have also seen, device pairing systems that rely on rhythm, such as RhythmLink by Lin et al. Their system securely pairs a host with a secondary device via rhythmic taps (Felix Xiaozhu Lin). However, their system cannot be used for authentication of devices.

In another paper, *Hearing Your Voice is Not Enough: An Articulatory Gesture Based Liveness Detection for Voice Authentication*, a team from Florida State University introduce *VoiceGesture*, an anti-spoofing system that accounts for replay attacks from recording devices. The system relies on extracting features in the Dopper shifts that are caused by articulatory gestures associated with a passphrase (FSU, 2017).

Moreover, other works have shown some impersonation attacks, which solely rely on physical accessibility to the device. Such methods are elementary and do not rely on any technology to replicate a person's voice. Wu et. Al in their work suggest that an imposter may be able to mimic the F0 pattern, but nearly impossible to replicate all formants themselves to authenticate a device that relies on voice biometrics for authentication (Zhizheng Wu N. E., 2015). Recent works have shown that even expert mimicry artists or linguists cannot get past voice authentication systems (Rosa Gonzalez Huatamaki, 2014; Prakash, 2014).

An interesting paper, presented a voice based system for house emergency of elderly people. They studied the distortion in voice when a person falls and how is it received by the home central emergency assistance unit, as well as accuracy of device as it compares to distance.. Their foundings showed that the average device drops in accuracy to about 70% when the device is more than 15 feet away from the speaker (Quan Zhang, 2013)

CHAPTER 3

MOTIVATION

In chapter 2 we discussed related works that introduced clever and effective authentication systems. In this research project we aim to introduce a system that has users' interest at heart. Just as technology advances, attacks and security breaches advance as well. Speech rhythm provides the users with a flexible and practical system that authenticates with high accuracy. In addition, unlike traditional voice-based authentication systems that can be replicated using voiceover techniques and modern technology, rhythm is unique to each individual (Dafyd Gibbon, 2001). Speech rhythm relies on the structure of the vocal tract of each individual making it as unique as fingerprint (The Vocal Tract, n.d.). Moreover, rhythm-based authentication works in scenarios where classic authentication systems fail. For example, passwords, Fingerprint, FaceID, even voice-based systems require users to be within arm's reach of the device which can be impractical in a lot of situations. Speech rhythm-based authentication works when users cannot be in direct contact with a device, or when users can only rely on their voice.

The second aspect of this research project is the cost effectiveness of the solution. The system can be deployed on a variety of devices and only requires a layer of software, unlike other authentication systems that require extra layers of hardware. Keeping the cost to a minimum is an appealing aspect to millions of people around the world, who pay a lot for expensive devices to protect their private data. From the early 1980's until mid 2000's there was a significant drop in prices of cell phones. However, from that point forward the prices only increased. The added layers of hardware that

are used for authentication (facial recognition projectors, fingerprint sensors, etc..) play a major role in the increased price (Gustke, 2019). Minimizing cost while maintaining high accuracy and efficiency in a user-friendly fashion is a great step towards fully embracing voice biometrics at the core of modern security. We have recently seen how companies, like Amazon, IBM, Google and Apple introduced smart home devices that work with voice commands such as Amazon's Alexa, Apple's Siri, and IBM's Watson (Boyd, 2018). With a rapidly growing market, the need for an effective, cheap and secure authentication system becomes of great importance. There are more than 100 million Alexa devices around the world (Castro, 2019), about 3 million Apple HomePods units worldwide (MacRumors, 2018), and -as of 2018- about a billion direct and indirect users of IBM's Watson (Clark, 2016). All these devices, and other home units, can easily deploy a rhythm-based system to help protect the data of millions of people. Our novel approach definitely speaks to a major need in modern systems that impact the lives of millions around the world. We provide a system that surpasses voice-based authentication in flexibility while maintains the highly reliable security of voice biometrics.

CHAPTER 4

VOICE FEATURES

Human voice has a lot of unique features: number of words in one breath, pitch, loudness, ease of breathing, rhythm among others (Boone, 2016). In this paper we focused on speech rhythm, as it is one of the most dominant features of speech that has been researched quite a bit and has concrete scientific support in security (FSU, 2017), linguistics (Dafyd Gibbon, 2001) and other fields.

Speech rhythm comprises of a number of features that can be extracted and used for user classification. In this research project we attempted a number of combinations of voice features to find the combination that yields the most accurate and efficient system. The tools at our disposal were capable of extracting Voice Onset Time (VOT), salience, time elapsed between VOT values, intensity, beat, pitch, and probability of a voiced syllable in speech. The algorithm yielded the best accuracy with four features: VOT, time elapsed between VOT values, salience, and probability of a voiced syllable. Details about each feature are described below. Further details of the algorithms, and results are in chapter 6.

To gain a better understanding of speech rhythm and its features, we need to establish an understanding of a few concepts:

I. Syllables: English alphabet is divided into vowels and consonants. Namely the letters A, E, I, O, U are vowels, and the 21 other letters are considered consonants, with some exception of the letter Y depending on utterance (Dictionary, n.d.). The combination of vowels and consonants produces syllables, which are defined as "an uninterrupted segment of speech consisting of a vowel sound, a diphthong, or

a syllabic consonant, with or without preceding or following consonant sounds." (dictionary.com, n.d.) In this project we leverage this core concept to extract the features mentioned above to classify users.

II. Rhythm: is defined as "the recurrence of a perceivable temporal patterning of strongly marked (focal) values and weakly marked (non-focal) values of some parameter as constituents of a tendentially constant temporal domain (environment)." (Dafyd Gibbon, 2001) These focal and non-focal points in rhythm are also present in speech, also known as voiced and voiceless sounds. Figure 1 below shows the International Phonetic Alphabet stating the voiced and voiceless sounds of the world, which constitute all speech (including English).

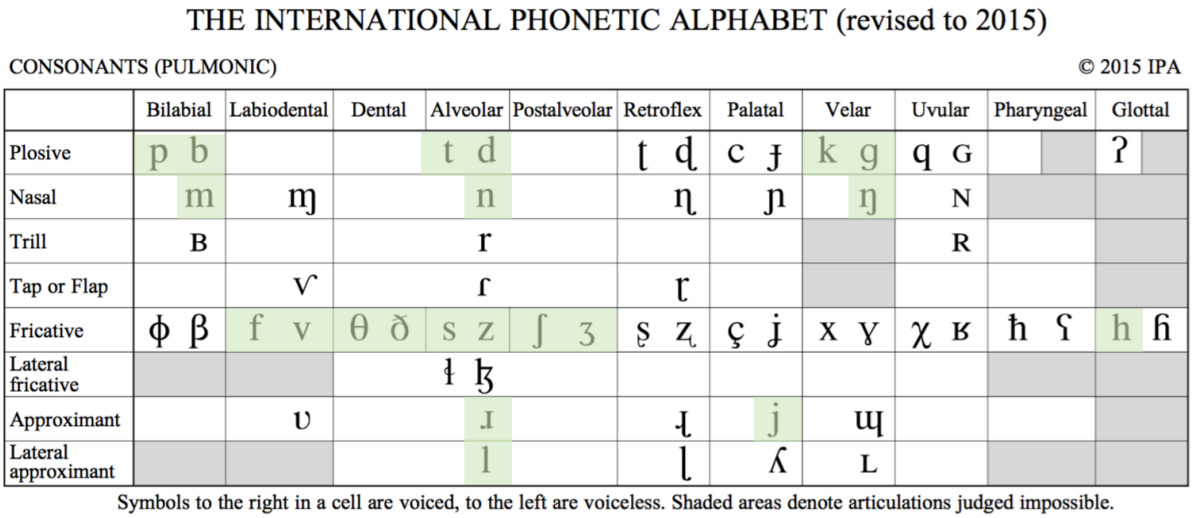


Figure 1: shows the International Phonetic Alphabet stating the voiced and voiceless sounds of speech (IPA, 2015).

III. Spectrogram: As cameras capture an image of motion, a sound spectrogram produces an image of a sound. A spectrogram is defined as: a record produced by a sound spectrograph with time shown along the horizontal axis, frequency shown

along the vertical axis, and intensity indicated by varying shades of darkness of the pattern (Merrian-Webster, n.d.). In this project we use PRAAT software to produce spectrograms to demonstrate and explain the features we are leveraging. PRAAT offers the user options to show more features on a sound spectrogram, not just frequency vs. time. Depending on settings, other relevant features can be displayed in a spectrogram including pulses, formants, intensity, pitch, and the spectrogram itself.

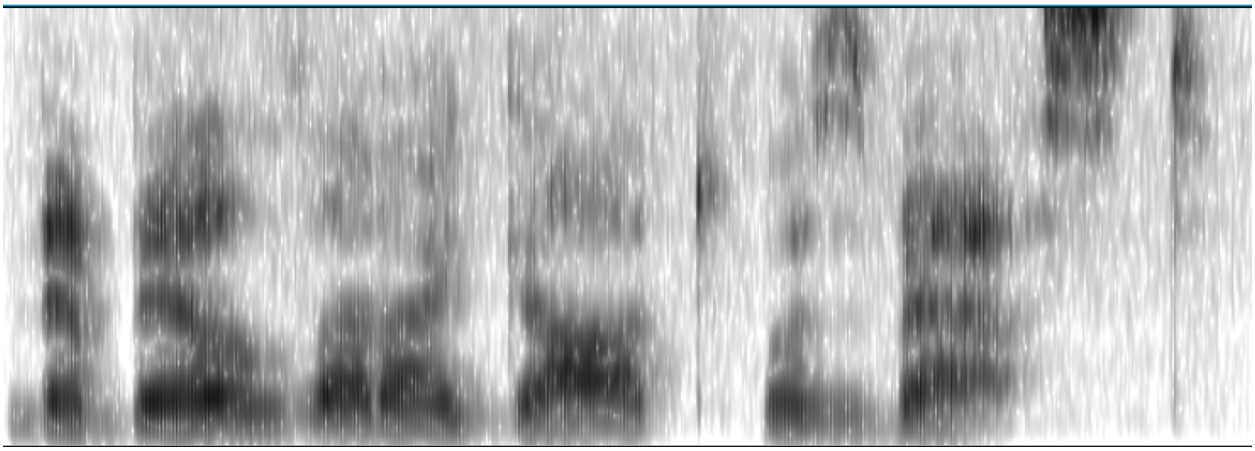


Figure 2 shows the simplest form of sound spectrogram produced by PRAAT.

IV. Formants: As described above, rhythm in speech consists of voiced and voiceless sounds. Voiced sounds require vibration of vocal cords, while voiceless sounds do not (Beare, 2019). Figures 3.1-3.2 show a bunch of red marks on a voice spectrogram. The combination of red marks is known as a formant. Formants are associated with voiced sounds (Wood, n.d.). For example, when a speaker says "banana" the b requires a vibration of the vocal cords, which travels through the vocal tract until it comes out of the mouth. As the wave travels it produces resonant frequencies, such frequencies are known as formants (Ladeforged, 2014). The F0 is the lowest frequency formant, that comes directly from the vocal cords, while F1,

F2, and in some cases F3 are produced from resonating frequencies. The scattered dots towards the end of the vertical axis represent an utterance as it nears existing the vocal tract, which is a lot more difficult to quantify unlike F0, F1, and F2. As we can see in figure 3.1, spectrogram of the sentence “the bill that I got was vast” (which includes seven vowels and a starting frequency close to 465Hz) shows that as time increases, we can see continuous lines of red dots. The lines represent the various formants on the time vs. frequency plot. On the other hand, Figure 3.2., shows the sound spectrogram of a voiceless utterance “sssssssss”. Voiceless sounds require no vibration of vocal cords (Wood, n.d.), and hence have no F0. Figure 3.2 shows the scattered dots of a voiceless sound that does not produce F0 formant. It is also worth noting that voiceless sounds start at a higher frequency range (around 1500Hz in this case) since they do not require vocal cord vibration. Since voiceless sounds start at a higher (physical) location in the vocal tract means they produce less resonant frequencies than voiced sounds. This distinction between voiced and voiceless sounds is essential for speech rhythm. The time it takes for an utterance to travel through a person's vocal tract is unique (Boone, 2016) and can be leveraged for the purposes of this research project.

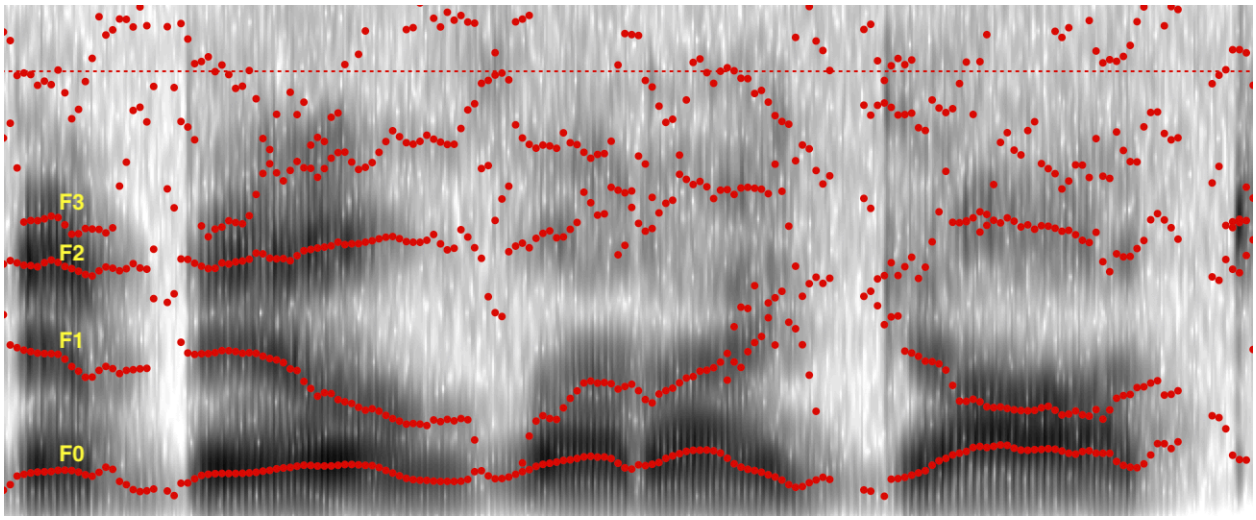


Figure 3.1 shows F0, F1, F2, and F3 formants in a sentence. There are some scattered points showing what can be considered as F4.

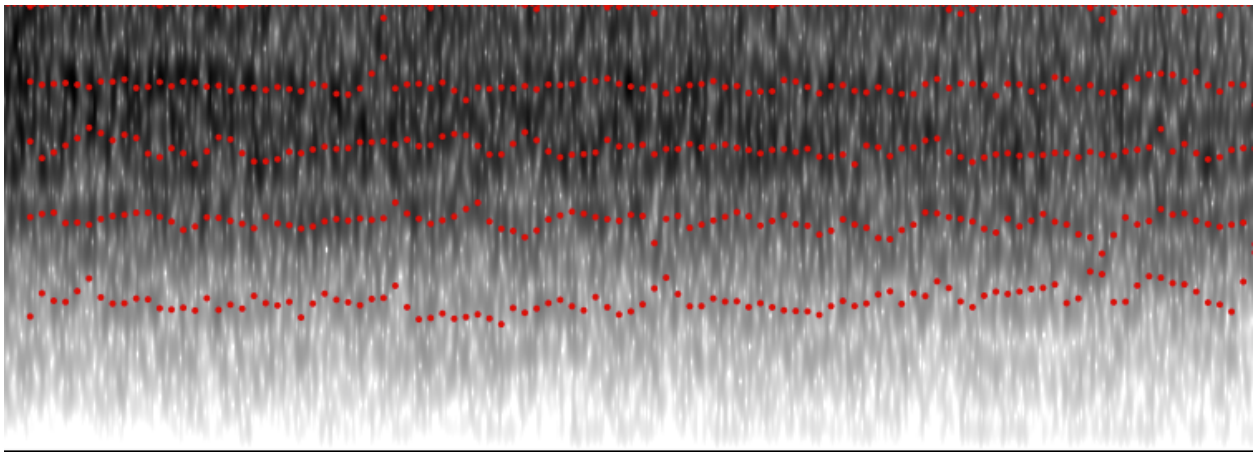


Figure 3.2 shows the scattered values of a voiceless utterance, incapable of producing F0 formants.

Now that we have established a clear understanding of the main concepts, we can discuss the four features used for user classification.

I. VOT: In any given English sentence there are a number of syllables. Extracting the length of both voiced and voiceless sounds is one of the parameters of choice in this paper. The Voice Onset Time (VOT) is defined as "the time that elapses between the release of the articulators for a stop and the onset of vocal cord

vibration of the following segment. This period is usually measured in milliseconds (ms)." (Kaur, 2015)

II. Time elapsed between VOT: since the feature extraction tool yields timestamps of VOT for each person, we added another dimension of data by finding the time between consecutive VOT instances. This metric adapts to changes in people's rhythm in a variety of situations. For example, when a person is running their VOT might be different than normal, but the difference in the VOT will stay consistent.

III. Probability of a voiced sound: speech rhythm is centered around the idea of voiced and voiceless sounds. The features extracted are marked with timestamps. This feature calculates the probability that the utterance associated with a timestamp is a voiced sound. The values are calculated by taking into consideration the formant values associated with the utterance (Joren Six, 2014). Knowing whether an utterance is voiced or voiceless is a great indicator of how its timestamp contributes to the rhythm of the speaker. Having high probability for a voiced syllable means detecting F0 formant is also highly likely, and therefore we have a start pointing for the next VOT.

IV. Salience: "quality which determines how semantic material is distributed within a sentence or discourse, in terms of the relative emphasis which is placed on its various parts." (Flowerdew, 1992) In other words, salience defines on which part of an utterance the speaker emphasizes in his/her speech. Just like probability of a voiced syllable, salience also indicates where the next VOT starting point is, and therefore making a difference in the rhythm of the speaker.

CHAPTER 4.1

OVERVIEW OF THE APPROACH

This chapter gives an overview of the process of user identification in this project. The users are prompted to read a paragraph. The feature extractor then passes on the user's numbers to train the neural network. User's name is used as a label for the trained data. When a new (test) input comes in, the network then extracts the features (just like it did in training) and the classification algorithm then decides whether the user is in the list of users, or an unknown new user. The algorithm stores a list of all existing users with their names as keys and keeps an "unknown" user as a possible entry for new input that does not match any of the existing users.

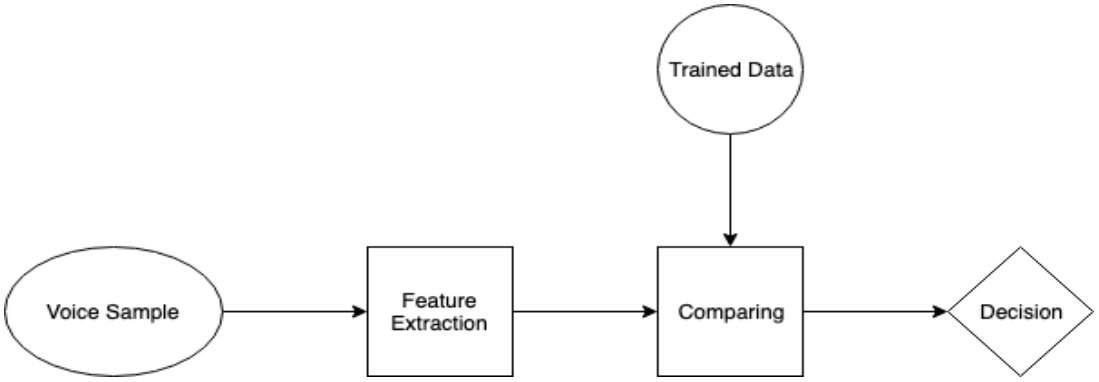


Figure 4 is a diagram highlighting the overall process of features analysis and user classification.

CHAPTER 4.2

CLASSIFICATION ALGORITHM

To achieve highest levels of accuracy possible while maintaining efficiency, we tested authentication using speech rhythm with a variety of well-known supervised learning algorithms that are used for classifying input. We tested this concept with a decision tree, random forest, naive Bayes classifier, and logistic regression. However, the algorithm that yielded the best result was multilayer perceptron (MLP). Figure 6 below shows a schematic of a generic multilayer perceptron. MLP does a forward pass to compute the values, and a backward pass to calculate errors. Based on the numbers from both passes it classifies users (Howard B. Demuth, 2014). MLP computes error and corrects the weights of the network from the computed error. The ability to improve values as it learns until the network reaches a stable solution provides a path for accurate results.

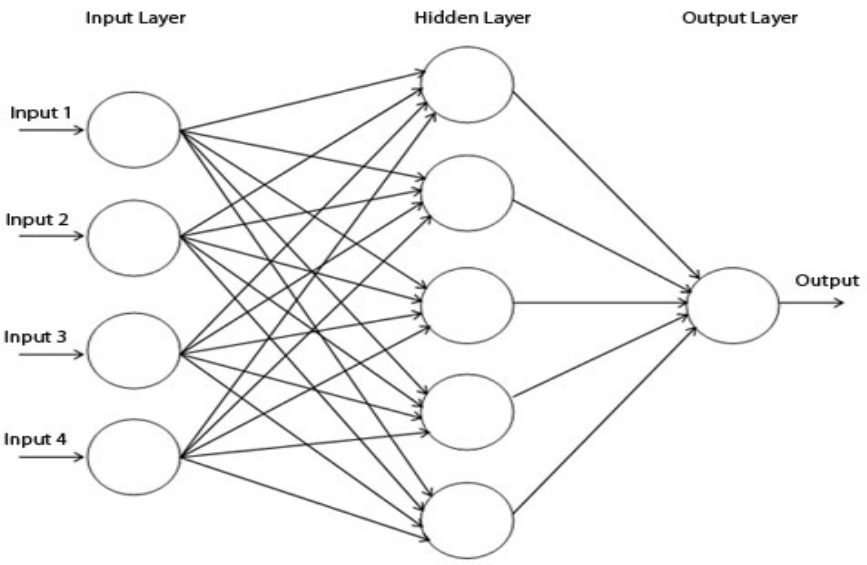


Figure 5 shows a generic MLP with one hidden layer, from which an output is predicted to do classification (Zahran, 2015).

We leveraged some existing libraries to develop a system that is tailored for the specifics of User Syndication Using Speech Rhythm. The feature extraction was done through an open-source Java voice library called TarsosDSP. We utilized its functionality to turn user input into acceptable format for the library, which in turn produced the desired values for the chosen features. With additional code we were able to do some preprocessing to expedite the process of training and testing, and therefore reduce the overall run time of the algorithm. In addition, we used Weka's API to help in writing the code for all the MLP and all other classification algorithms discussed earlier. All input data was formatted in ARFF format, (attribute relation file format). Below is a sample of an ARFF file for the famous Iris data (Weka, 2008):

```
@RELATION iris
@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
```

ARFF allows us to specify all the attributes we are using in the classification process, as well as a label for each new entry. All participants of the experiment were asked to record their voices while naturally reading the training and testing sentences. Participants recorded from their personal devices (laptops, tablets, or phones). All files collected were formatted as 16-bit wave files, for compatibility with the feature extraction library.

CHAPTER 4.3

CODE

Leveraging existing APIs and libraries allowed us to simply focus on generating the best models and evaluate our work to yield the best outcome. TarsosDSP library provided a simple way of extracting features. After building the archive files using Gradle, we only had to use the command *$java -jar FeatureExtractor-2.4.jar <type> <file>.wav* specifying the type of feature we wish to extract, and the name of the file in a 16-bit stereo wave format. Weka's API was also simple to handle with all the available documentation online. Figures 6.1-6.7 show some parts of the system's code. After building the classifier, we train the data, and use the trained data to classify new input, and finally evaluate the accuracy of the classification. Using similar methods, we were able to build and test other models to analyze accuracy and choose the model that yields the best results. Figure 6.1 shows the construction of the main classifier class that will then be used to create and test the various models we attempted before the MLP. While Figures 6.2-6.3 show how the various algorithms were trained, tested and evaluated.

When our attempts yielded unsatisfactory results from four classification algorithms, we opted for an ANN that we can control, in terms of parameters and construction. Figures 6.4-6.5 show some parts of building the MLP. The main model class, and the class that handles the heavy lifting. After the classifier is built and trained, we feed it the voice features discussed earlier. Figures 6.6-6.7 show how we use the MLP to train, test, classify and evaluate the model. Weka models use instances (multidimensional data points) as input. The MLP then proceeds to classify the test

data. In the final step, the MLP takes unlabeled instances from the test file and labels them based on the trained data. If the user exists in the list of known users, then it authenticates the user, otherwise it returns “failed authentication”.

```
public class Models {
	public static BufferedReader readDataFile(String filename) {
		BufferedReader inputReader = null;

		try {
			inputReader = new BufferedReader(new FileReader(filename));
		} catch (FileNotFoundException ex) {
			System.err.println("File not found: " + filename);
		}

		return inputReader;
	}

	public static Evaluation classify(Classifier model,
			Instances trainingSet, Instances testingSet) throws Exception {
		Evaluation evaluation = new Evaluation(trainingSet);

		model.buildClassifier(trainingSet);
		evaluation.evaluateModel(model, testingSet);

		return evaluation;
	}

	public static double calculateAccuracy(FastVector predictions) {
		double correct = 0;

		for (int i = 0; i < predictions.size(); i++) {
			NominalPrediction np = (NominalPrediction) predictions.elementAt(i);
			if (np.predicted() == np.actual()) {
				correct++;
			}
		}

		return 100 * correct / predictions.size();
	}
```

Figure 6.1 shows core model of building a classifier using Weka.

```
    public static void main(String[] args) throws Exception {
            long startTime = System.nanoTime();
            BufferedReader datafile = readDataFile("/path/to/training.arff");

            Instances data = new Instances(datafile);
            data.setClassIndex(data.numAttributes() - 1);

            // Do 10-split cross validation
            Instances[][] split = crossValidationSplit(data, 20);

            // Separate split into training and testing arrays
            Instances[] trainingSplits = split[0];
            Instances[] testingSplits = split[1];

            // Use a set of classifiers
            Classifier[] models = {
                            new J48(), // a decision tree
                            new RandomForest(),
                            new NaiveBayes(),
                            new Logistic(),|
            };

        // Run for each model
        for (int j = 0; j < models.length; j++) {

                // Collect every group of predictions for current model in a FastVector
                FastVector predictions = new FastVector();

                // For each training-testing split pair, train and test the classifier
                for (int i = 0; i < trainingSplits.length; i++) {
                        Evaluation validation = classify(models[j], trainingSplits[i], testingSplits[i]);

                        predictions.appendElements(validation.predictions());

                        // Uncomment to see the summary for each training-testing pair.
                        //System.out.println(models[j].toString());
                }

                // Calculate overall accuracy of current classifier on all splits
                double accuracy = calculateAccuracy(predictions);

                System.out.println("Accuracy of " + models[j].getClass().getSimpleName() + ": "
                                + String.format("%.2f%%", accuracy)
                                + "\n----------------------------------");
        }

    long endTime   = System.nanoTime();
    double totalTime = (endTime - startTime)/1_000_000_000.0;
    System.out.println(totalTime);

}
```

Figures 6.2-6.3 show the main and how to run and evaluate Weka models.

```
public class ModelGenerator {
        public Instances loadDataset(String path) {
        Instances dataset = null;
        try {
            dataset = DataSource.read(path);
            if (dataset.classIndex() == -1) {
                dataset.setClassIndex(dataset.numAttributes() - 1);
            }
        } catch (Exception ex) {
            Logger.getLogger(ModelGenerator.class.getName()).log(Level.SEVERE, null, ex);
        }

        return dataset;
    }

        public Classifier buildClassifier(Instances traindataset) {
                MultilayerPerceptron m = new MultilayerPerceptron();

                try {
                        m.setLearningRate(0.2); //0.2, 9, 0.6, 300 best results
                        m.setHiddenLayers("9");
                        m.setMomentum(0.6);
                        m.setTrainingTime(300); //epochs
                        m.buildClassifier(traindataset);

                } catch (Exception ex) {
                         Logger.getLogger(ModelGenerator.class.getName()).log(Level.SEVERE, null, ex);
                }
                return m;
        }

            public String evaluateModel(Classifier model, Instances traindataset, Instances testdataset) {
            Evaluation eval = null;
            try {
                // Evaluate classifier with test dataset
                eval = new Evaluation(traindataset);
                eval.evaluateModel(model, testdataset);
                double acc = eval.pctCorrect();
                double wrong = eval.pctIncorrect();
                double unclass = eval.pctUnclassified();
                System.out.print("acc: " + acc + "\n wrong: " + wrong + "\n unclassified: " + unclass + "\n");
            } catch (Exception ex) {
                Logger.getLogger(ModelGenerator.class.getName()).log(Level.SEVERE, null, ex);
            }
            return eval.toSummaryString("", true);
        }

        public void saveModel(Classifier model, String modelpath) {

            try {
                SerializationHelper.write(modelpath, model);
            } catch (Exception ex) {
                Logger.getLogger(ModelGenerator.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
```

Figures 6.4-6.5 show the MLP model generator.

```
public class MLP {

        public static final String DATASETPATH = "/path/to/training.arff";
        public static final String TESTPATH = "/path/to/test.arff";//
        public static final String MODElPATH = "/path/to/model.bin";
        public static final String UNLABELED = "/path/to/labeled.arff";

public static void main(String[] args) throws Exception {
                long startTime = System.nanoTime();

        ModelGenerator mg = new ModelGenerator();

        Instances dataset = mg.loadDataset(DATASETPATH);

        Filter filter = new Normalize();

        Instances unlabeled = new Instances(new BufferedReader(new FileReader(TESTPATH)));
        unlabeled.setClassIndex(unlabeled.numAttributes() - 1);

        int trainSize = (int) Math.round(dataset.numInstances() * 0.95);
        int testSize = dataset.numInstances() - trainSize;

        dataset.randomize(new Debug.Random(1));

        //Normalize dataset
        filter.setInputFormat(dataset);
        Instances datasetnor = Filter.useFilter(dataset, filter);

        Instances traindataset = new Instances(datasetnor, 0, trainSize);
        Instances testdataset = new Instances(datasetnor, trainSize, testSize);

        // build classifier with train data set
        MultilayerPerceptron ann = (MultilayerPerceptron) mg.buildClassifier(traindataset);

        // Evaluate classifier with test data set
        String evalsummary = mg.evaluateModel(ann, traindataset, testdataset);
        System.out.println("Evaluation: " + evalsummary);


          for (int i = 0; i < unlabeled.numInstances(); i++) {
                      double clsLabel =ann.classifyInstance(unlabeled.instance(i));
                      unlabeled.instance(i).setClassValue(clsLabel);
                    }
                    // save labeled data
          BufferedWriter writer = new BufferedWriter(new FileWriter(UNLABELED));
          writer.write(unlabeled.toString());
          writer.newLine();
          writer.flush();
          writer.close();

          mg.saveModel(ann, MODElPATH);
          long endTime   = System.nanoTime();
          double totalTime = (endTime - startTime)/1_000_000_000.0;
          System.out.println(totalTime);
 }
```

Figures 6.6-6.7 show the main of using the model generator and other files to build, train, test and evaluate the MLP.

CHAPTER 5.1

TRAINING DATA

Since speech rhythm pivots on vowels, consonants and syllables we constructed the training paragraph in such a way that it meets all the requirements for effective training, while not bothering the user with extensive reading or cumbersome repetitiveness of sentences. The paragraph used for training is: "I have been waiting at the airport since 2:30 in the afternoon. My bags from flight 57, which left Dallas this morning at 9am have not arrived yet. I cannot believe that the bags are delayed, I am going to be late for my niece's 1st birthday party. It starts in 30 minutes! I have a stuffed dog I want to give her. She loves animal toys, especially dogs and dolphins." The above paragraph has 88 syllables and takes on average 23.7 seconds to say in a normal tone of speech. This results in about 65 data points of training per person after extraction of features. We collected data from13 participants.

CHAPTER 5.2

TESTING DATA

For testing we used "Hey why are you not answering my calls? There is something important I want to talk to you about". The sentence has 25 syllables and takes on average 6.5 seconds to read in a normal tone of speech. This results in about 12 data points after feature extraction that are used as new input to do comparison and classification. After establishing an accuracy baseline of the system, we experimented with random, but continuous, speech from the users. We asked the participants to speak at random without taking extended breaks between sentences to test accuracy under different circumstances.

CHAPTER 6.1

VARIOUS ALGORITHMS

As stated earlier one of the reasons we developed this system was the convenience for the user. Other systems require cumbersome steps to make accurate predictions. The training and testing sentences we scripted were fairly short and take little time on part of the user (less than 30 seconds to read and record). Once the training is done, the authentication process takes about a second. Each training sample produces about 65 data multidimensional data points after feature extraction.

As part of the process of developing the best possible system, we attempted a number of combinations of features to determine which combination produces the most accurate, yet efficient algorithm. Therefore, we tinkered with the number of dimensions in the MLP to see if accuracy can be improved. We introduced pitch in addition to speech rhythm as a fifth dimension, the accuracy increased with some of the algorithms, however it decreased in the MLP. We also kept track of the time it took each algorithm to train the data. Tables 1 and 2 show the time each algorithm took to run and the corresponding accuracy for four and five attributes.

	Time (seconds)	**Accuracy - 4 Features**
Decision Tree	1.12	45%
Random Forest	12.8	50%
Naive Bayes	0.36	53%
Logistic Regression	5.93	52%
MLP	1.08	93.3%

Table 1 shows the accuracy of each of the five algorithms used with four features

	Time (seconds)	Accuracy – 5 Features
Decision Tree	1.35	62%
Random Forest	14.4	63%
Naive Bayes	0.45	59%
Logistic Regression	7.79	54%
MLP	1.25	84%

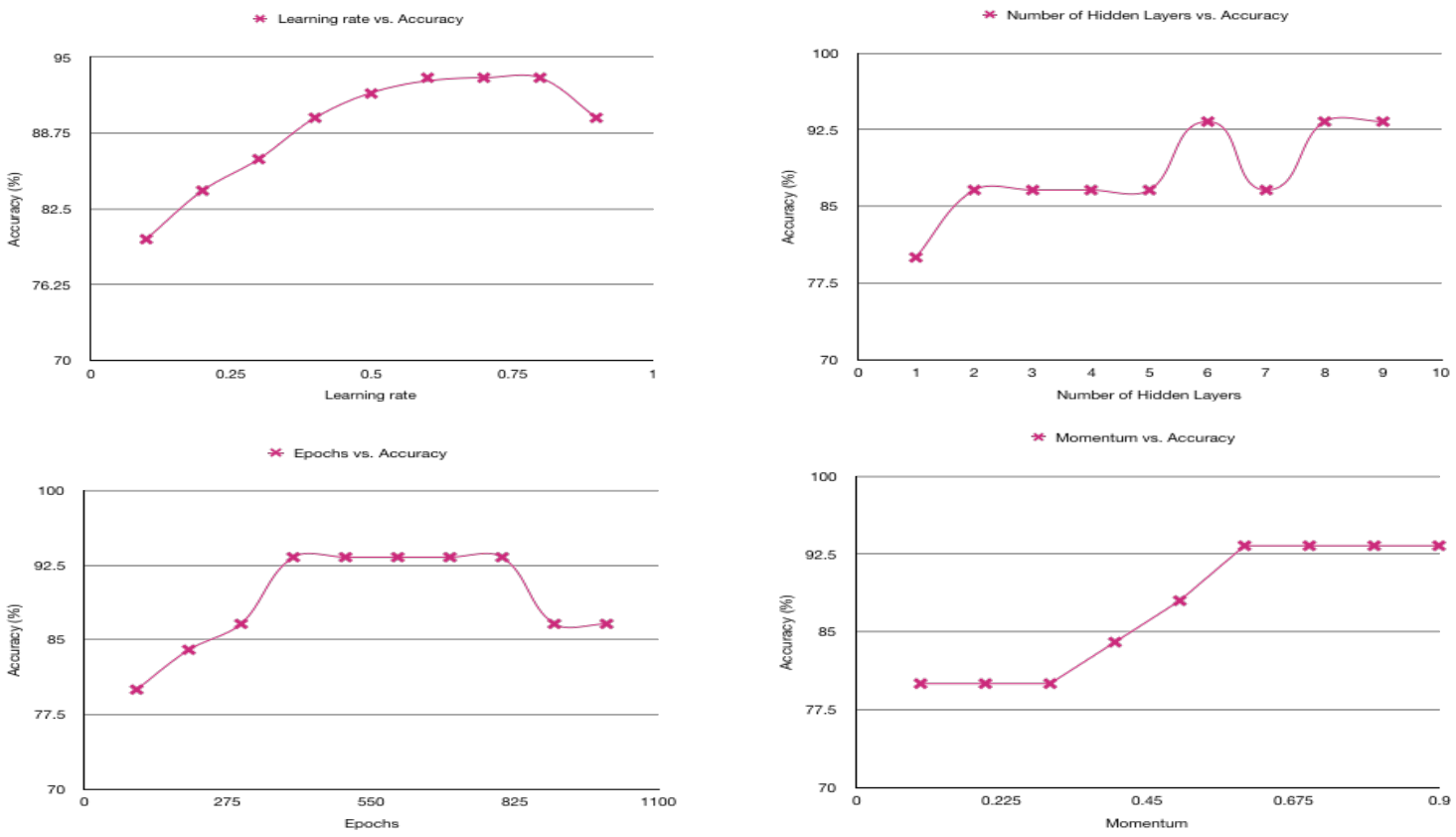
Table 2 shows the accuracy of each of the five algorithms used with five features.

Both tables show how MLP beats all other algorithms in performance, some by a significant margin. Even though it is not the fastest algorithm but completing the training process in a little over a second with the highest accuracy shows it clearly as the superior choice. Table 1 also shows better performance with a smaller number of dimensions, which translates to reduced overhead. MLP can be 93.3% accurate in as little as 1.08 seconds. We also attempted introducing other dimensions, such as intensity, and beat but the results did not improve for any of the algorithms.

CHAPTER 6.2

PARAMETERS OF MLP

One of the reasons we saw a jump in accuracy in results between MLP and the other four algorithms was the ability to control the constraints and parameters of the network. Weka allows us to control parameters such as number of epochs, learning rate, momentum, train/test split, and number of hidden layers. As we trained the network we tinkered with the numbers until we found the set of numbers that yielded the best result. Figure 7.1 shows the how changing learning rate changes the accuracy of the algorithm. The figure shows how setting the learning rate $\alpha \in$ [0.6-0.8] yields the best outcome. We also notice a dip in accuracy when the learning rate increases past the [0.6-0.8] range. While Figure 7.2 shows how the accuracy varies with the number of hidden layers (H) in the ANN. We noticed and immediate increase in accuracy with the increased number of layers, however the accuracy stayed consistent until it maxes out at six. Figure 7.3 is particularly interesting because it shows how changing the number of epochs (e) past 400 makes no difference, as the data consistently yields the same accuracy, until it drops at 900 epochs, which also yielded much longer runtime. Momentum (m) was another parameter we varied to find best outcome. Figure 7.4 below shows how momentum almost linearly increases until it maxes out at 0.6, after which accuracy stays consistent. Therefore, we conclude when $\alpha = 0.6$, e = 400, H = 6, and m = 0.6 the accuracy is 93.3% and the algorithm runs in a fraction over a second, on average 1.08 seconds.



Figures 7.1-7.4 show how the accuracy varies with α, e, H, m.

The last metric we used to solidify the accuracy of the system was varying the training/testing split to determine the best accuracy. Figure 7.5 shows how the accuracy changes as the training sample increases in size. We held all other parameters constant to evaluate the system for best possible results. The figure clearly shows how the increased training sample size, increases the accuracy of the algorithm. We also assured randomness of the samples using Weka's randomizer. One decision we had to make was how big of a split we wish to use, because as the training size set increases, the runtime of the algorithm also increases. Figure 7.6 below shows how the split percentage affected the run time of the algorithm. We settled for 0.97/0.03 split for best results.

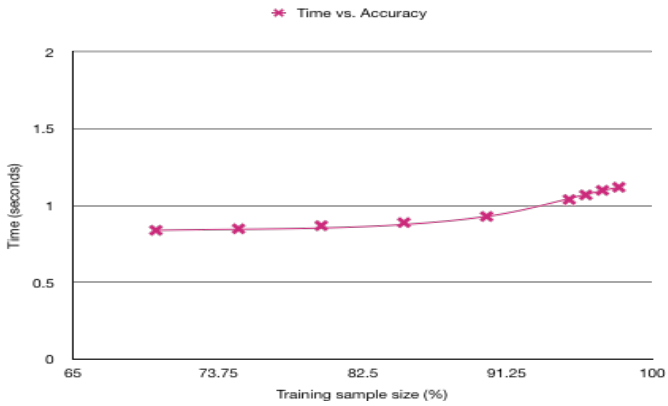


Figure 7.5-7.6 show how accuracy and runtime change with the train/test split percentage.

CHAPTER 6.3

RESULTS FROM DISTANCE

A major advantage of our system is the ability to handle speakers from afar. This can be pictured when a user is talking to a central home unit (Alexa for example), that is in a different room. We recreated this environment and tested the accuracy of our system. Figure 8 (BiomedGuy) shows the setup for the experiment. In this experiment the speaker (labeled Actor in Figure 8) talked to the system from a range of distances, starting with holding the device by hand, until the speaker was behind an open door. We recorded the results of accuracy vs. distance in the Figure 8 below.

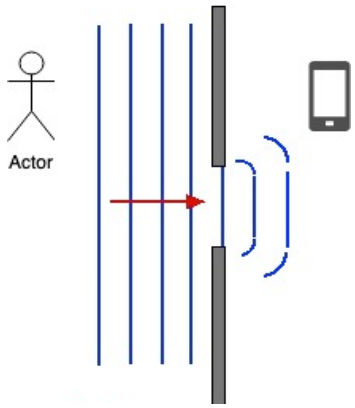


Figure 8 shows the experimental setup for a user speaking to the system from behind an open door.

The system managed to maintain high accuracy even at approximately 13 feet away. The authentication accuracy never dropped below 77% even at such distance. This goes to show how our system is not prone to disturbances easily and can still work accurately enough even without physical proximity to the main device. Note in this experiment the speakers were asked to speak naturally without shouting or raising

their voices to above normal levels of speech. This experimented was done with two users, each spoke randomly from various distance up to 13 feet.

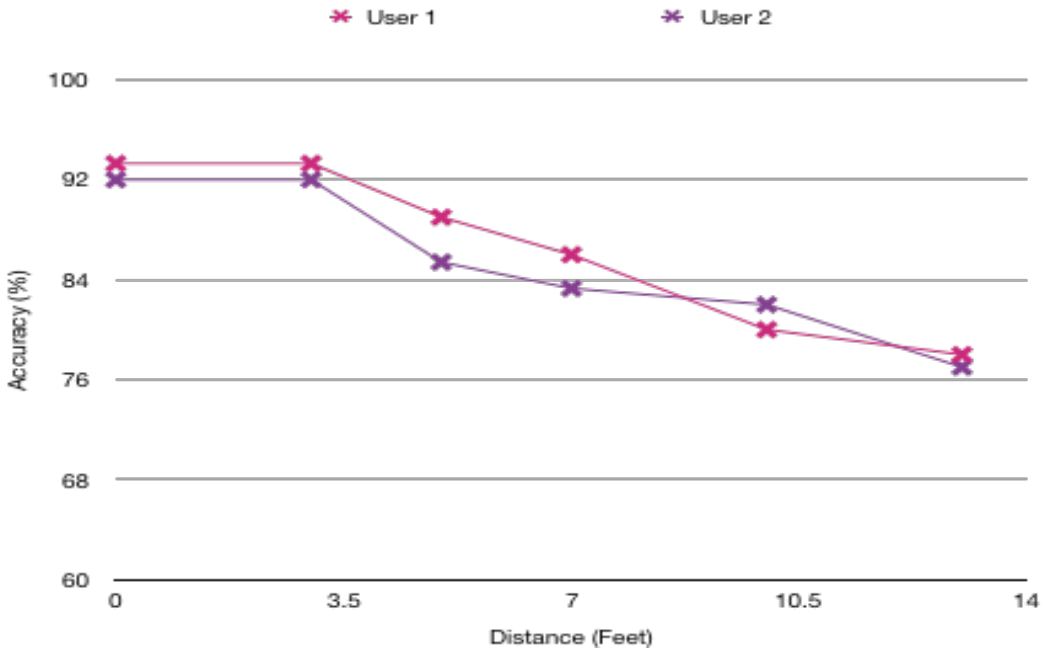


Figure 9 shows of accuracy of the system with increasing distance for two users, and how the accuracy did not drop below 77% even from 13 feet away with a barrier separating the speaker from the device.

CHAPTER 6.4

FURTHER EVALUATION METRICS

To evaluate the success of our authentication system we conducted a series of experiments based on the dataset available. We used one of the users as legitimate user, while the others were designated as impostors. Evaluation metrics:

- False rejection rate (FRR): the probability that an actual user is treated as an imposter
- False acceptance rate (FAR): the probability that an imposter is treated as a legitimate user

We chose samples randomly to be the training set, while the rest was for testing. For each user we repeated the process 12 times. We ran this experiment with three train/test ratio variations and noticed significant jumps in the results. The system achieved best results with 97% training/testing split, which yielded an FRR of 22.5% and FAR of 10.2%.

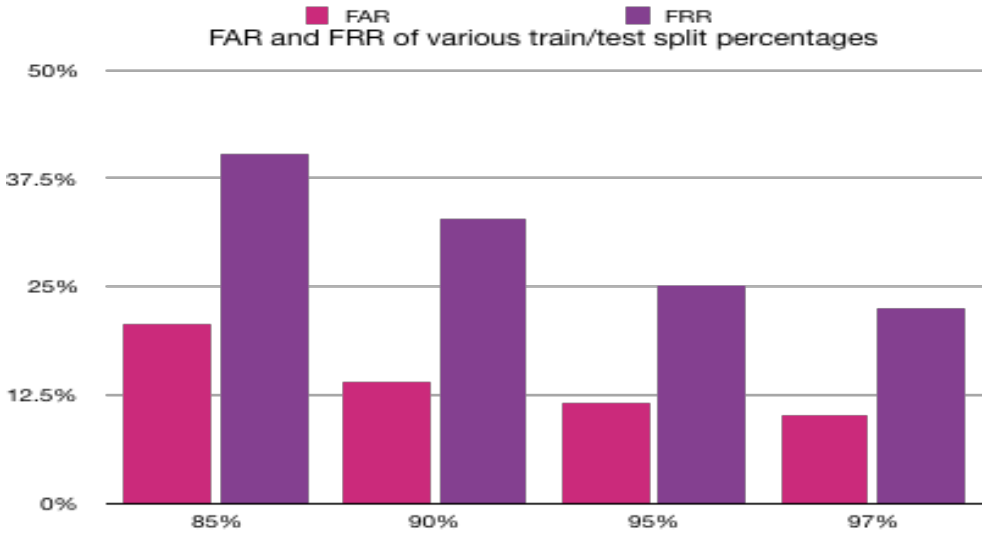
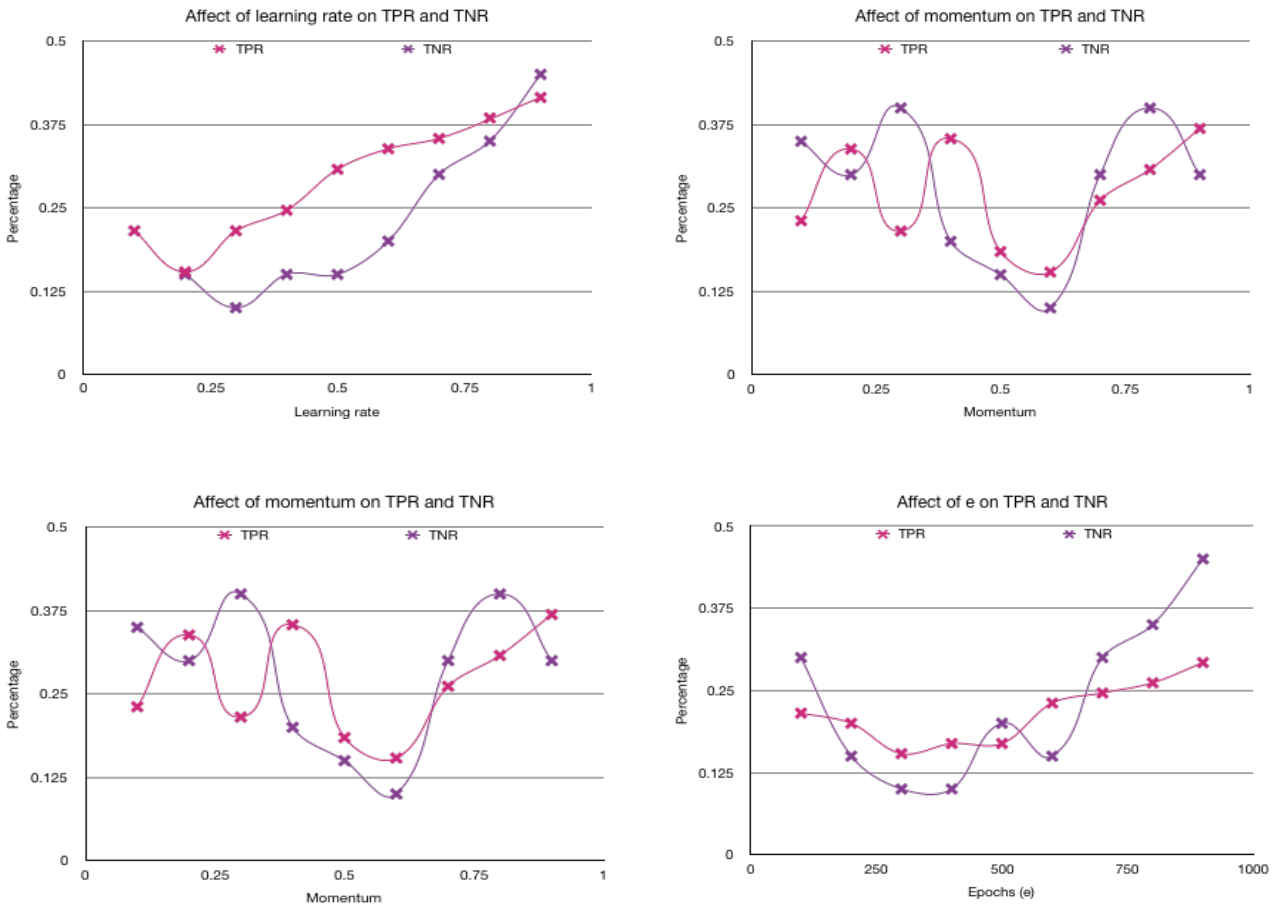


Figure 10 shows the varying FRR and FAR of different train/test splits.

One thing we observed as we were conducting the experiments that highest error occurred in speakers who spoke faster than the average, as well as speakers who spoke much slower than the average. Most of the error occurred in 4/13 users, two of which finished the training sentence in about 18 seconds, and the testing sentence in about 3 seconds. While the slower speakers took over 27 and 8 seconds to complete the training and testing sentences, respectively. The average speech time for the training sentence was 23.7 seconds, while that of testing was 6.5 seconds.

To further solidify the success of our system, we conducted another series of experiments to check how the four parameters we varied in chapter 6.2 affected the true positive rate (TPR) and true negative rate (TNR). In these experiments we held three parameters constant and varied the fourth and recorded the percentage of TPR and TNR. Figures 10.1-10.4 show that finely tuning the parameters led to the best accuracy, otherwise the system yielded unsatisfactory results. In some cases, we saw the false negative rate go to as high as 45%, while the false positive rate reached 43%. In chapter 6.1 we highlighted how other algorithms performed poorly in the problem we have at hand. We resorted to an MLP because we have full control of the parameters and can overcome such problems by fine tuning the system to suit our needs. Figures 10.1-10.4 further confirm that an ANN was the ideal selection for classification.



Figures 10.1-10.4 show how the TPR and TNR change with varying parameters of MLP

CHAPTER 6.5

COMPARISON TO VOICE-BASED SYSTEMS

In this paper we introduce an elegant system that offers flexibility for users, while maintaining high accuracy. In chapter 6.3 we showed how well our rhythm-based system works from distance. In this section we compare our system with existing voice-based authentication systems that are in the market, especially from distance. We examine the FRR of our rhythm-based system against some of the most accurate voice-based systems in the market. Below is a list of a few voice-based system, their stated accuracy, and some of their drawbacks:

- VoiceVault managed to achieve 0.1% FAR, TNR of 5%, and 3% TPR (VoiceVault, 2019); their system is limited by physical proximity to the device, the user is required to be talking on the phone for their system to work, as our experiments below prove.
- Barclays uses voice authentication to verify their enrolled users when the call the bank for customer support. Their service is 95% accurate, however, their system is also limited by physical constraints. The voice authentication is only accessible via phone calls and therefore cannot be done from a distance (Nuance, 2014; Shead, 2013).
- HSBC is another global bank that adopted voice-based authentication for mobile banking (HSBC, 2019). Unlike Barclays, HSBC uses authentication for users to access their mobile banking application. BBC, however, showed a case study in which they managed to fool their voice-based authentication system (Simmons, 2017).

- Microsoft Speaker Recognition API: this API allows the user to identify and verify users based on their voice. This API mostly works with a person's pitch, and length of their larynx (Boelman, 2018). The API works accurately, up to 95% (Shu, 2017) in a quiet environment without outside noise. However, the system requires 60 seconds of straight talk, no interruptions or breaks. In addition, the enrollment sentences required are available online, and do not use user-generated sentences and therefore increasing risk of an attack (Annie Shoup).
- VoiceIt is a company that provides biometric solutions for security and works with Twilio for authentication of their users. Their system has a FAR of 0.0001% and FRR of 1%-10% (VoiceIt). However, their system is extremely limited by the distance, they recommend users to be wearing headsets and speak directly into microphones for accurate authentication.

Some of the service providers above mention outstanding numbers for their accuracy, FAR and FRR. We also looked at their range of functionality, as well as their recommended usage method. Table 3.1 below show the stated accuracy, and method of usage of each system.

	Accuracy	Usage Range
VoiceVault	97%	Contact with device
Barclays	95%	Contact with device
Microsoft	94.9%	Contact with device
VoiceIt	90%-99%	Headset
Speech-rhythm	93.3%	Up to 13 feet

Table 3.1 shows the accuracy of various systems and their usage range.

We also managed to secure some demos and test two of these voice-based systems. After creating a few accounts for testing, we replicated our experiment in chapter 6.3. Both systems worked with predefined words/sentences. To create a profile the user had to repeat the given set of words or sentences a few times. Table 3.2 below show the how long it took to enroll a user, and the time for a successful authentication after registration. It is important to note that authentication time is for a successful attempt, when a user failed to authentication, they were prompted to repeat their attempt. In some cases, the user had to repeat three times for a successful attempt. VoiceIt only allows three attempts after which the user is no longer eligible to attempt for a fixed period of time.

	VoiceIt	**VoiceVault**	**Speech-rhythm**
Enrollment Time (seconds)	30	42	35
Successful authentication time (seconds)	7.6	6.5	7.2

Table 3.2 shows the enrollment and successful authentication time for VoiceIt, VoiceVault and compares them to speech-rhythm system.

In our experiment we repeated the authentication attempt from each distance five times. Figure 11 below reflects the percentage of successful authentication of each method vs. distance in feet. Figure 11 clearly shows how the accuracy drastically drops as distance increases. Both systems failed to authenticate a single time from any distance of seven feet or more. This is a limitation of voice-based systems. On the other hand, our rhythm-based system showed great success from distance. In the

seven-foot range, where voice-based systems failed, our system was more than 85% accurate.

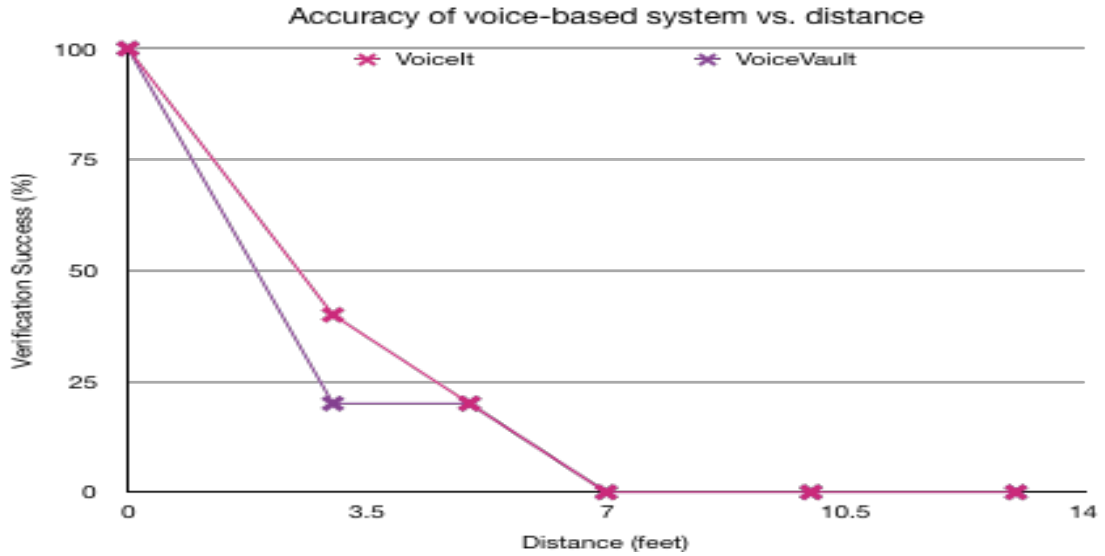


Figure 11 shows the accuracy of VoiceIt and VoiceVault vs. distance

We also compared the FRR of our system against some of the voice-based systems mentioned above. Figure 12 below compares the success rate of our system compared voice-based systems'. Our system managed to outperform some of the voice-based system in terms of FRR. It was also not far off from the best voice-based systems. Earlier we established that our system beats voice-based system in terms of range of success, coupling that with comparable FRR testifies to strength of rhythm-based systems for user authentication. Table 3.1 and Figures 11 and 12 demonstrate how our system is superior in terms of flexibility, and range of work. It maintains high levels of success, even when compared to systems with much larger datasets, and computational power. The service providers only shared limited statistics of their products, even after getting in touch with them in person to collect more detailed numbers about their systems to provide a more comprehensive comparison.

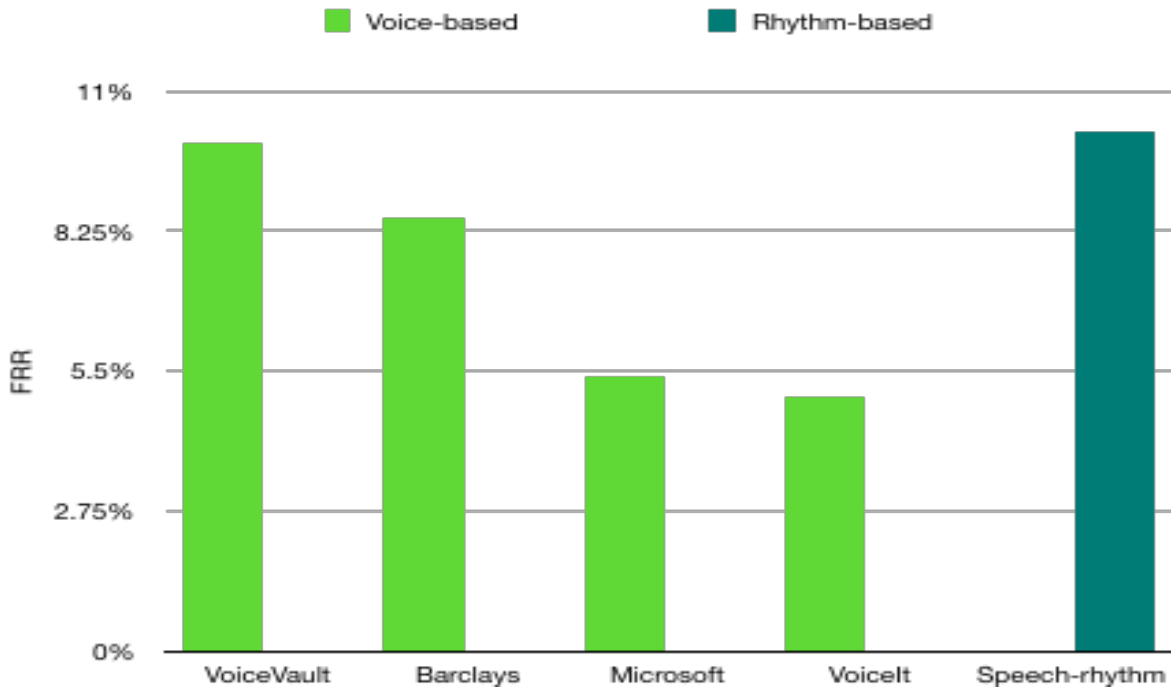


Figure 12 shows the FRR of voiced-based system against our rhythm-based system

CHAPTER 7

CONCLUSION

In this paper we introduced a user authentication system using speech rhythm. The system only requires a device equipped with a simple microphone. The efficient layer of software works in a little over a second. The system first extracts features from the input speech, then analyzes them and classifies the user as an existing user or unknown using a multilayer perceptron. Experimental evaluation validates the accuracy of the system under a variety of conditions. Overall, the User Syndication System Using Speech Rhythm can achieve 93.3% accuracy. In addition, the system showed success even when the speaker was separated by a barrier and large distance from the system. When a speaker said a phrase from 13 feet away, behind an open door, it yielded 77% user classification accuracy, which is a significant improvement from the existing authentication systems, as well as voice-based systems in the market. Keeping in mind that most of these authentication systems require the user to be in direct contact with the device. Our elegant solution works well in situations where existing systems fail to deliver, or simply cannot function. Experimental results yielded FRR of 10.2% and FAR of 22.5%. The paper presents a baseline for a novel and flexible system that can be extended and used in a variety of situations. We provide the user with an accurate, convenient, and hassle-free process that delivers high accuracy.

CHAPTER 8

APPLICATIONS

The practicality of this system extends to a lot of fields and applications. The cheap cost, usability, and quickness makes it ideal for authentication of handheld and wearable devices. When deployed on a phone or a smart watch this system can authenticate users to unlock the device, make banking transactions though applications, and e-commerce. It can also be used in a two-factor authentication instead of sending codes via emails or text messages to the user. That idea can be extended to a system that does both speech recognition and rhythm-based authentication, in which a user says a user chosen passphrase. The system then proceeds to analyze their rhythm, as well as the sentence for increased security. More application-specific versions can be produced based on the specific need. We believe the system can turn into a live speaker detector, in which it does the processing in real time and yields instantaneous results for home devices such as Alexa and Google Assistant. The system shows great promise for practical use at home even when the device is not near the user. It can also be used for long distance control of devices, such as home security systems, lights and other appliances.

CHAPTER 9

FUTURE WORK

This paper showed that there is a lot of promise using speech rhythm as a method to classify users. The uniqueness of each person's rhythm in speech makes it a reliable approach. To achieve better results, a more specific feature extractor (one that can handle more rhythm-based features) could improve the accuracy of the system even more. Also adding more dimensionality for higher accuracy while maintaining effectiveness would be the next step to take. Additionally, we can extend the system for application specific versions based on the goal in mind. In general, a larger training set would definitely help in increasing the accuracy, as the data can be used for a training, development and test sets instead of just the classic train/test split we used. The development set provides the ability of examining which specific aspects of the system need to be improved to achieve higher accuracy.

REFERENCES

1. Almog Aley-Raz, N. M. (2017, October 30). Device, system, and method of liveness detection utiziling voice biometrics. *CCS*, 57-71.
2. Amazon. (n.d.). *Alexa Voice Service*. Retrieved July 15, 2019, from amazon alexa: https://developer.amazon.com/alexa-voice-service
3. Andreas Kipp, M.-B. W. (n.d.). Automatic detection and segemntation of pronounciation variants in German speech. *IEEE*.
4. Annie Shoup, T. T. (n.d.). An Overview and Analysis of Voice Authentication Methods. *MIT*.
5. Arthur Janicki, F. A. (2016). An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks. *Security and Communication, 9*, 3030-3044.
6. Beare, K. (2019, June 26). *Voiced vs. Voieless Consonants*. Retrieved July 2019, from Thoughtco.: https://www.thoughtco.com/voiced-and-voiceless-consonants-1212092
7. Bell, K. (2015, September 11). *A smarter Siri learns to recognize the sound of your voice in iOS 9*. Retrieved from Mashable: https://mashable.com/2015/09/11/hey-siri-voice-recognition/
8. BiomedGuy. (n.d.). *Diffraction.* Retrieved from Fandom: https://bmet.fandom.com/wiki/Diffraction?file=Diffgaps.gif
9. Boelman, H. (2018, January 11). *Microsoft Speaker Recognition API*. Retrieved from HenkBoleman: https://www.henkboelman.com/speaker-recognition/
10. Boone, D. R. (2016). *Is Your Voice Telling On You? How to Find and User Your Natural Voice* (Vol. 3). Plural Publishing.
11. Boyd, C. (2018, January 10). *The Past, Present, and Future of Speech Recognition Technology*. Retrieved June 19, 2019, from The Startup: https://medium.com/swlh/the-past-present-and-future-of-speech-recognition-technology-cf13c179aaf
12. Castro, A. (2019, January 4). *The Verge*. Retrieved July 15, 2019, from https://www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp
13. Clark, D. (2016, October 26). *IBM: A Billion People to Use Watson by 2018*. Retrieved May 10, 2019, from https://www.wsj.com/articles/ibm-a-billion-people-to-use-watson-by-2018-1477496455
14. Dafyd Gibbon, U. G. (2001). Measuring speech rhythm. *EUROSPEECH*, 95-98.
15. Dictionary. (n.d.). *Why Are A,E,I,O,U, And Y Called "Vowels"?* Retrieved January 10, 2019, from https://www.dictionary.com/e/vowels/
16. dictionary.com. (n.d.). *Syllable*. Retrieved 2019, from https://www.dictionary.com/browse/syllable
17. Dyroff, C. (2018, July 25). *Insider: here's how much cellphones have actually changed over the years*. Retrieved July 1, 2019, from https://www.insider.com/the-history-of-the-cellphone-2018-7
18. Felix Xiaozhu Lin, D. A. (n.d.). RhythmicLink: securely pairing I/O-constrained devices by tapping. *Proceedings of ACM sumposium on user interface software and technology*.

19. Fisher, C. (2019, March 1). *'OK Google' will no longer fully unlock your phone*. Retrieved from Engadget: https://www.engadget.com/2019/03/01/ok-google-voice-match-unlock-update/
20. Flowerdew, J. L. (1992). Salience in the performance of one speech act: the case of definitions. *Discourse PRocesses*, 165-181.
21. FSU. (2017, October 30). Hearing Your Voice is Not Enough: An Articulatory Gesture Based Liveness Detection for Voice Authentication. *CCS*.
22. Google. (2019). *Smart Lock*. Retrieved July 15, 2019, from https://get.google.com/smartlock
23. Gustke, J. (2019, July 22). *Cell Phone Cost Comparison Timeline*. Retrieved from ooma.com: https://www.ooma.com/blog/cell-phone-cost-comparison/
24. Howard B. Demuth, M. H. (2014). *Neural Network Design* (Vol. 2nd). Martin Hagan.
25. HSBC. (2019). *Your voice is yourr password*. Retrieved from HSBC: https://www.us.hsbc.com/customer-service/voice/
26. IBM. (n.d.). *Watson*. Retrieved July 15, 2019, from https://www.ibm.com/watson/services/speech-to-text/
27. IPA. (2015). *Full IPA Chart*. Retrieved 2018, from https://www.internationalphoneticassociation.org/content/full-ipa-chart
28. Jinxi Guo, T. N. (2019). A Spelling Correction Model For End-To-End Speech Recognition.
29. Joren Six, O. C. (2014). TarsosDSP, a Real-Time Audio Processing Framework in Java. *AES*, 27-29.
30. Kaavya Sriskandaraja, V. S. (2016). Front-End for Anti-Spoofing Countermeasures in Speaker Verification: Scattering Spectral Decomposition. *IEEE Journal of Selected Topics in Signal Processing*.
31. Kaur, J. (2015, October). Factors Influencing Voice Onset Time (VOT): Voice Recognition. *IJRASET*, 174-179.
32. Ladeforged, P. (2014). a course in Phonetics. 99-101.
33. Lerhman, J. (2017, January 13). *Why Do Our Recorded Voices Sound Weird to Us?* Retrieved July 5, 2019, from New York Times: https://www.nytimes.com/2017/01/13/science/recorded-voices.html
34. Linghan Zhang, S. T. (2016, October 24). VoiceLive: Aphoneme Localization based Liveness Detection for Voice Authentication on Smartphones. *CCS*, 1080-1091.
35. MacRumors. (2018). *MacRumors*. Retrieved from https://www.macrumors.com/2018/08/14/strategy-analytics-homepod-2q18/
36. MarketsAndMarkets. (n.d.). *Speech and Voice Recognition Market Worth*. Retrieved July 13, 2019, from https://www.marketsandmarkets.com/PressReleases/speech-voice-recognition.asp
37. Merrian-Webster. (n.d.). *sounds spectrogram*. Retrieved November 2018, from https://www.merriam-webster.com/dictionary/sound%20spectrogram
38. Millward, S. (2012, November 29). *Open Sesame: Baidu Helps Lenovo User Voice Recognition to Unlock Android Phones*. Retrieved June 15, 2019, from techinasia: https://www.techinasia.com/baidu-lenovo-voice-recognition-android-unlock
39. Nuance. (2014). *Barclays improves their customer experience.* Retrieved from nuance.com: https://www.nuance.com/content/dam/nuance/en_au/collateral/enterprise/case-study/cs-

barclays-en-us.pdf?ppi=Enterprise%20-%20Voice%20Biometrics&campaign=DEO-2017-WL-VB-Fraudminer-Barclays-CS&campaignID=7010W000002S1VrQAK&formName=Enterprise-Web-Default
40. O'Boyle, B. (2019, June 14). *What is Alexa and what can Amazon Echo do?* Retrieved from Pocket-Lint: https://www.pocket-lint.com/smart-home/news/amazon/138846-what-is-alexa-how-does-it-work-and-what-can-amazons-alexa-do
41. Parthasarathi, H. (2019, April 4). *New Speech Recognition Experiments Demosntrate How Machine Learning Can Scale*. Retrieved from Amazon Alexa: https://developer.amazon.com/blogs/alexa/post/9e8392c6-5476-4a34-a2d8-c4e479677954/new-speech-recognition-experiments-demonstrate-how-machine-learning-can-scale
42. Philip L. De Leon, M. P. (2012). Evaluation of speaker verification security and detection of HMM-based synthetic speech . *IEEE*.
43. Prakash, S. P. (2014). Crowdsourcing attacks on biometric systems. *Symposium on Usable Privvacy and Security*, 257-269.
44. Quan Zhang, L. R. (2013). HONEY: A Multimodality Fall Detection and Telecare System. *TELEMEDICINE and e-HEALTH*.
45. Reisinger, D. (2017, March 6). *Here's How Many iPhones are Currently Being Used Worldwide*. Retrieved from Fortune: https://fortune.com/2017/03/06/apple-iphone-use-worldwide/
46. Rosa Gonzalez Huatamaki, T. K.-M. (2014). I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry. .
47. Ryan, K. J. (2019, July). *Who's Smartest: Alexa, Siri, and or Google Now?* Retrieved from Inc.com: https://www.inc.com/kevin-j-ryan/internet-trends-7-most-accurate-word-recognition-platforms.html
48. *Saypay Technologies*. (2017, January 1). Retrieved July 15, 2019, from https://www.say-tec.com
49. Shead, S. (2013, May 9). *Barclays puts Nuance voice recognition tech to work at identifying customers.* Retrieved from ZDNet: https://www.zdnet.com/article/barclays-puts-nuance-voice-recognition-tech-to-work-at-identifying-customers/
50. Shu, C. (2017). *Microsoft's speech recognition System hits a new accuracy milestone*. Retrieved from TechCrunch: https://techcrunch.com/2017/08/20/microsofts-speech-recognition-system-hits-a-new-accuracy-milestone/
51. Si Chen, K. R. (2017). You can Hear But You cannot Steal: Defending against Impersonation Attacks on Smartphones. *IEEE*, 183-195.
52. Simmons, D. (2017, May 19). *BBC Fools HSBC voice recognition security system*. Retrieved from BBC: https://www.bbc.com/news/technology-39965545
53. Sylvian Sardy, P. T. (2001). Efficient Algorithms for Speech Recognition. *IEEE Transaction on Signal Processing*, 1146-1152.
54. Technology. (2017, January 1). *Speech and Voice Recognition Market*. Retrieved July 15, 2019, from https://www.marketsandmarkets.com/PressReleases/speech-voice-recognition.asp
55. *The Vocal Tract*. (n.d.). Retrieved May 24, 2019, from Voice Science Works: https://www.voicescienceworks.org/vocal-tract.html
56. Tode, C. (2017). *Barclays expands use of voice security for phone banking convenience*. Retrieved from ReailDive:

https://www.retaildive.com/ex/mobilecommercedaily/barclays-expands-use-of-voice-security-for-phone-banking-convenience
57. Twilio. (2019). *Twilio & VoiceIt: Building an API Business for Voice Biometrics*. Retrieved from Twilio: https://signal.twilio.com/2017/sf/sessions/7lDenNPUekKKcIGyokskO6/twilio-and-voiceit-building-an-api-business-for-voice-biometrics
58. University of Nevada, Reno. (2018, June). Beat-PIN: A User Authentication Mechanism for Wearable Devices Through Secret Beat. *ASIACCS*, 101-115.
59. VocalPassword. (2016). *VocalPassword*. Retrieved July 15, 2019, from https://www.nuance.com/omni-channel-customer-engagement/security/identification-and-verification.html
60. VoiceIt. (n.d.). *VoiceIt Your Are Key.* Retrieved from Voiceit.io: https://voiceit.io/assets/VoiceItVideoBiometricsPreview-DiscoveryDocument.pdf
61. Voiceprint, W. (2015, May 21). *WeChat Blog*. Retrieved June 1, 2019, from https://blog.wechat.com/2015/05/21/voiceprint-the-new-wechat-password/
62. VoiceVault. (2019). *VoiceVault*. Retrieved July 15, 2019, from https://voicevault.com
63. Weka. (2008, November 1). *Attribute-Relation File Format (ARFF)*. Retrieved March 2019, from https://www.cs.waikato.ac.nz/~ml/weka/arff.html
64. Wobbrock, J. O. (n.d.). Tapsongs: tapping rhythm-based passwords on a single binary sensor. *Proceedings of ACM symposium on user interface software and technology*.
65. Wood, S. (n.d.). *What are formants?* Retrieved July 2019, from Person2: https://person2.sol.lu.se/SidneyWood/praate/whatform.html
66. Zahran, M. (2015, December). *Assessment of Artificial Neural Network*. Retrieved 2019, from researchgate: https://www.researchgate.net/figure/A-hypothetical-example-of-Multilayer-Perceptron-Network_fig4_303875065
67. Zhang, D. D. (2012). Biometric solutions: For authentication in an e-world. 877-886.
68. Zhi-Feng Wang, G. W.-H. (2011). Channel pattern noise based playback attack detection algorithm for speaker recognition . *IEEE*.
69. Zhizheng Wu, N. E. (2015). Spoofing and countermeasures for speaker verification: a survey. *Speech Communication 66*, 130-153.
70. Zhizheng Wu, S. G. (2014). A study on replay attack and anti-spoofing for text-dependent speaker verification. *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA).* , 1-5.