Detecting Lies with Machine Learning Classification

John McCormick
Department of Electrical Engineering
Rochester Institute of Technology
Rochester, NY
jxm1522@rit.edu

Nikhil Deshmukh
Department of Electrical Engineering
Rochester Institute of Technology
Rochester, NY
nxd4573@rit.edu

***Abstract*—Lie detection has been sought after for many reasons. From interpersonal to interrogative reasons, lie detection is useful for determining the value of what someone says. Focusing on more discrete methods for determining lies, machine learning can be used to classify truths and lies from video, audio, gaze, and even EEG data. However, the scope of this project only focuses on using audio data to determine if someone is lying or not.**

***Index Terms*—Lie detection, Machine Learning Classification, Audio Processing**

I. Problem Statement

Lie detection is normally done by connecting a polygraph to someone and having someone else interpret the data to see if the person is lying or not. The accuracy depends on who is interpreting the data, so this could lead to adverse results. In addition, special equipment is needed to perform the analysis. However, a less invasive method of lie detection would be to analyze the audio signals of someone speaking. With only a microphone, there may be an interrogation where you do not want someone to know if you are trying to detect if they are lying, either audio or visual analysis would be best. Also, recordings can be taken from political speakers to determine if they are lying. The end goal is to detect a lie on simple readily available data source. Everyone nowadays has phone capable of recording audio, so the data is widely available.

The common ways of analyzing the audio and video data are through other types of machine learning algorithms. [12] uses random forest and K-Nearest Neighbor for its audio deception detection. In addition to the audio data they collected, they also utilized other algorithms that process video, EEG, and gaze detection to determine when someone is lying. Then a final decision is made based on all of those factors. The approach used here involves a classification algorithm to relate specific features extracted from audio that would be considered a lie. The primary extracted features to be observed are pitch contours and Mel frequency cepstral coefficients (MFCC). If needed, other data modes could be utilized in order to get a better result.

First, the data set will be explored. A study based on determining a multi modal way of collecting truth and lie data for machine learning algorithms produced a data set that will be used for this method of lie detection. Then, the method will be explained with the data set that is going to be used. This includes how the audio is going to be processed in order to determine qualities that would expose a lie. Finally, the verification method will be discussed with the metrics of performance and prediction of how well the detection will perform.

II. Literature Review

One common theme in other works is they use multiple modes to determine lies. [9] and [10] both use multiple modalities to determine the result, whether it be audio, visual, gaze detection or other means. [9] uses all forms of data the study collected to determine lies. Video, audio, EEG, and gaze data were categorized and processed with their own algorithms based on best performing baselines. Then the results of each mode are weighted to calculate the final guess of a truth or a lie. These attributes were isolated and observed as well as combined and observed. The combination of all four attributes yielded a higher accuracy. Isolating only the audio section, there was a 53.24%-56.22% accuracy using K-Nearest Neighbor (KNN) and Random Forest machine learning algorithms [9].

The C.I.A also conducted their own study on the validity of audio-based lie detection. The human voice is built on three distinct sounds: the basic sound, format sound, and microtremor [2]. The basic sound is the signal between 100 - 300 Hz and these frequencies create the base of what is considered to be human speech. The format sounds are resonances created by cavities in the head, especially the mouth. These cavities add a second amplitude-modulated sound to the voice. Lastly, microtremors are signals typically ranging in 8 to 12 Hz. It is super imposed on the basic and format sounds and is present in all normal speech. However as a person becomes stressed, the microtremors become suppressed and disappear from the overarching signal. The presence of microtremors, or lack there of, can help determine if a person is under stress if properly recorded. [7] Utilizing these attributes in conjunction with the classification algorithm, a new data set can be processed to find discrepancies within these frequencies.

While most programs use multiple modes of data, the proposed method here is to use only one modality: audio. Although [12] does use machine learning algorithms for their audio, the accuracy of Random Forest and K-Nearest Neighbor were not impressive. They were about the average for a human guessing, which is 54% [7]. Also, the method they used took the better of the two results to be sent to the weighted output. A similar scheme can be seen in [10], but there were only audio

Modality	Method	Average Accuracy	
		Set A	Set B
Only EEG	Random Forest	58.71	-
	EEG Net	54.25	-
	MLP	53.79	-
Only Gaze	Random Forest	61.70	57.11
	MLP	57.71	53.51
Only Video	LBP + SVM	55.21	53.25
	LBP + Random Forest	56.20	55.26
	LBP + MLP	54.22	49.90
Only Audio	Random Forest	53.24	54.89
	KNN	53.22	56.22

TABLE I
ACCURACY RESULTS FROM MULTIPLE MODALITIES AND ALGORITHMS [12]

and visual data to go into a majority voting output. In both cases, there was audio feature extraction/processing before the classification. The proposed method here will include pre-processing before classification, but the exact operations are still to be determined. The exact operations that could be done are shown in the Method section under Proposed Methodology.

Based on public papers, many have not been utilized multiple classifiers in lie detection. The attempt is to test our validation accuracy against other public audio lie detection methods. There can be various ways to attempt to cheat the detection, because it is solely audio based. Also due to using a singular method of verification, the validation may not have a high accuracy in comparison to other studies that use multiple forms of verification. Table I illustrates how gaze detection has a higher accuracy than audio detection when compared individually. Gaze detection involves looking at pupil dilation and where a subject's eyes move towards. This allows for more variability in the data to compare against. However, high end equipment is needed to measure these attributes. For audio detection, while the chart displays a lower accuracy, this was again conducted using a different form of machine learning. To add, high end equipment is necessarily needed to achieve a workable result, however better recording equipment could yield to more precise data.

III. METHOD

A. Task Environment

The data set used is a result of [2], specifically for deception detection. With a sample size of 80 people, there are 40 female participants, and 40 male participants. Each participant recorded 4 videos, each containing a positive truth, negative truth, positive lie, and a negative lie. In this context, a positive and negative truth or lie is the connotation of what the person is saying. In total, 160 lies and 160 truths were said. There should be a balanced and sufficient data set that the classification alogorithm can learn from. This data was in video mp3 format, so the audio needed to be extracted to save file space. A small Python script was created to call the program, ffmpeg, from the command line for each video. The result was 320 .wav files with 48kHz sampling frequency with various bitrates.

To split the dataset, all of the male and female sets were split from the original set. From those two subsets, 80% of the samples were put into the training set, and the other 20% was put into the validation set. Then the training and validation sets were shuffled so that there would not be a bias with the male or female subset being appended to each other in the training and validation sets.

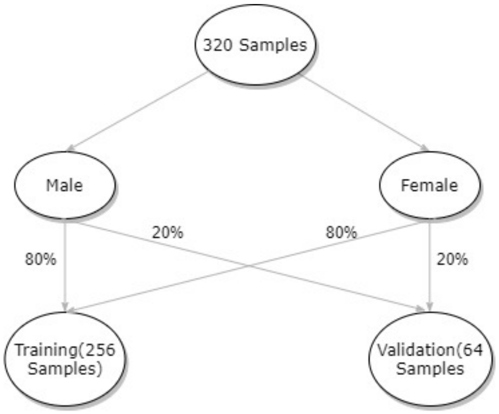


Fig. 1. Graph of splitting dataset

B. Proposed Methodology

The method of determining lies through audio is through classification. Classifiers utilize training data to learn how the data can be divided into classes based on their relations. There are various types of classifiers, and specifically, six will be observed. Logistic regression, decision tree classifier, support vector machine (SVM), stochastic gradient descent classifier (SGD), K-nearest neighbor (KNN), and gradient boosting classifier (GBC), will all be utilized. Logistic regression uses a logistic function (sigmoid function) at its core to determine the probability of a given outcome [11]. While this a regression model, given a decision threshold, it becomes a classification technique. Decision tree classifiers make observations about an item, known as branches, and then draw conclusions about the target's value, known as leaves [4]. In classification, the leaves represent the class labels, while the branches represent the connections to and from each label. A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin, the lower the generalization error of the classifier [5]. Stochastic gradient descent is essentially an optimized version of a loss function. The loss function is determined by the hyperparameter, and is used to provide a more efficient alternative to the standard loss function defined by the hyperparameter [1]. KNN operates under the assumption that similar traits exist in close proximity. An example of KNN is shown in Figure 2.

It relies on this assumption and based on the chosen number of neighbors will gather its result based on the closeness between data points [3]. Gradient Boosting functions by taking weak learning algorithms, typically decision trees, and combining them through boosting stages, defined by hyperparam-

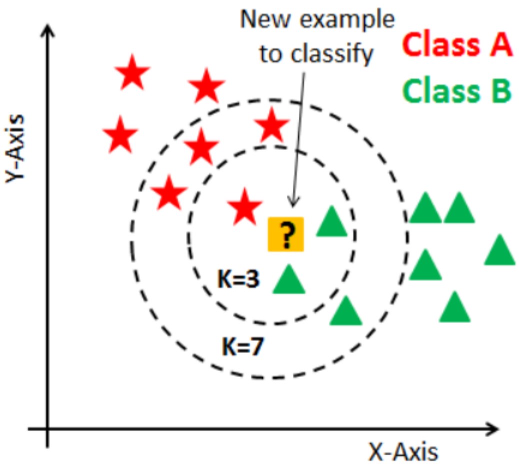


Fig. 2. K-Nearest Neighbor Example Diagram

eters [8]. In addition to each all of these classifiers, ensemble learning will also be conducted on the dataset to try and improve the overall accuracy. Ensemble learning essentially combines multiple models in an attempt to improve the overall accuracy, and is depicted in Figure 3.

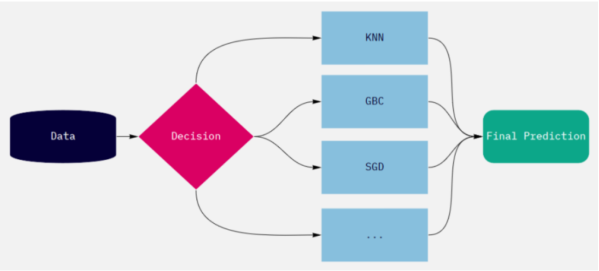


Fig. 3. Ensemble Learning Flow Chart

The predicted outcomes from the classifiers are combined and a majority vote is taken from each iteration. The accuracy from this new set is determined and defined as the new overall accuracy. In certain situations, only the top performing classifiers will be used for ensemble learning.

C. Audio Feature Extraction

All of the feature extraction was completed with the Python package Librosa. The two features that were extracted were MFCC, or Mel Frequency Cepstral Coefficients, and pitch data. Figures 4 and 5 show the extracted spectrogram and MFCC graphs through Librosa. Spectrograms show the intensity of each frequency at a given time. Since Figure 4 is a Mel Spectrogram, the spectrogram has been scaled to the intensity of what humans would perceive by using a Mel scale. This allows the classifier to have the frequencies that a human would perceive. The pitch data extracted is simply the instantaneous frequencies through time. It was not scaled with the Mel scale.

MFCCs are usually used in voice recognition because they take human's perception of frequencies into account through the Mel scale. Humans do not perceive frequency on a linear scale; we notice changes in lower pitches better than high pitches. The "Cepstrum" part of MFCC means that the rate of change in the spectral bands are what is plotted.

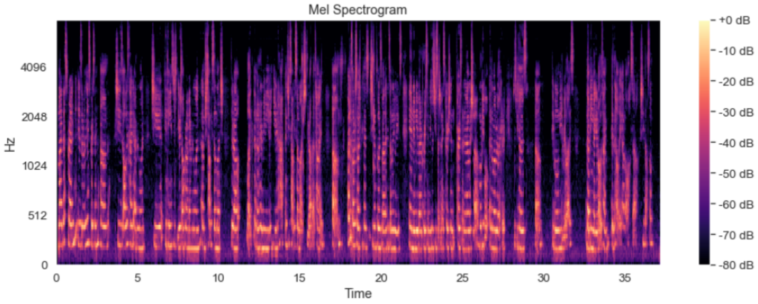


Fig. 4. Spectrogram graph of one sample

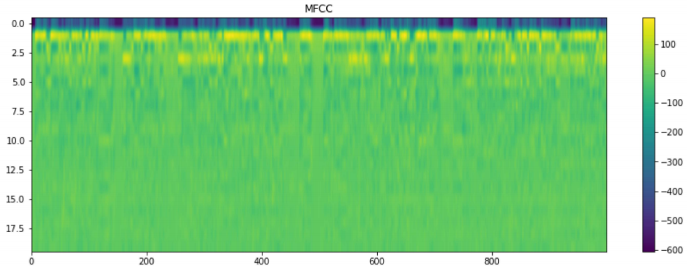


Fig. 5. MFCC Chart of one sample

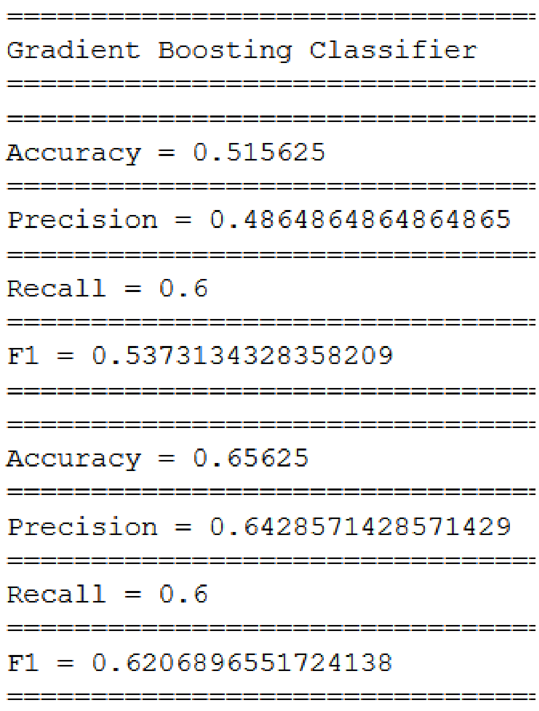
D. Validation

80% of the original set will be used to train the classification algorithm due to the large data set of both truths and lies. This leaves the rest of the truths and lies to be validated to see if they get categorized correctly or not. The goal is to get a classification accuracy of higher than 54% of the time as this is the accuracy a human has when detecting a true or a lie [7]. While the goal is to be above 54%, this may be difficult to achieve due to the quality of audio recording equipment. Audio signals can hold a large amount of data and attributes, and a high precision audio recorder is needed to document the complete data.

IV. Results

Preliminary results were not the best. At most, the accuracy was around 53%. A common occurrence during validation was the MFCC trained classifiers usually provided worse accuracy than the pitch trained classifiers. Sometimes it would be the opposite, but in most runs, the pitch accuracy was better.

The best result was from the Gradient Boosting Classifier, as can be seen in Figure 6. The first accuracy shows the result of the MFCC trained classifier, and the second accuracy shows the result of the pitch trained classifier. There is a drawback to this result though. The classifier took around 1.5 hours to run, so it was not viable to run multiple hyperparameter tuning passes. With that being said, this is the best result that any of the individual classifiers could produce. The next best was K-Nearest Neighbor with a best accuracy of 62.5%. With better hyperparameters, the two promising classifiers could give better results.



```
================================
Gradient Boosting Classifier
================================
================================
Accuracy = 0.515625
================================
Precision = 0.4864864864864865
================================
Recall = 0.6
================================
F1 = 0.5373134328358209
================================
================================
Accuracy = 0.65625
================================
Precision = 0.6428571428571429
================================
Recall = 0.6
================================
F1 = 0.6206896551724138
================================
```

Fig. 6. Best Result from All Trials

With ensemble learning, the best result was around 59% when the single classifier accuracy's were all lower than 59%. The pitch-trained stochastic gradient descent, K-Nearest Neighbor, Gradient Boosting and MFFC-trained Gradient Boosting classifiers had an individual accuracy of 54.68%, 56.25%, 56.25%, and 57.81% respectively. Figure 7 shows the true positive rate and false positive rate on one plot. Ideally, there would be a bend more in the top left corner, but with an ensemble accuracy of 59%, the bend is not too pronounced.

Unfortunately, the results here cannot be easily replicated because each classifier randomly shuffles the data, so the order of the training and validation set changes every time the program is run.

V. Conclusions

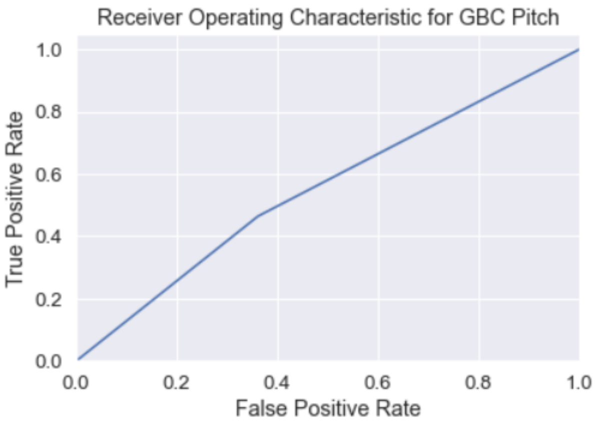


Fig. 7. AUC ROC plot for ensemble learning

Overall, the program was able to identify a truth and a lie with a greater accuracy than a human could [7]. Both the ensemble accuracy and best scenario case illustrate better results. While the accuracy is not as high as initially anticipated, there is merit to the overall program and methodology. Improvements can be made such as further tweaks to hyperparameters to yield greater accuracies. The simplicity of the source medium combined with the single modal technique of detection proved to be a challenge. However, the simplicity of using any audio recording was the ultimate goal to maintain throughout the project. It was found that the pitch data yielded better results, and the classifiers that performed the best were generally, GBC, KNN, and SGD. It is important to note that the dataset utilized contained isolated samples, so there was no background noise. Modifications to the pre-processing would need to be made to try and reduce background noise from new source samples. While extra noise may not affect the MFCC data, it will most likely have an adverse effect on the pitch contour data. Future implementations could include the ability to detect a specific person's voice and isolate it from a sample to be used for lie detection. All in all, the program was a success given the inherent constraints of the problem.

Acknowledgment

We would like to thank Miami University for providing the dataset.

References

[1] "1.5. Stochastic Gradient Descent¶." Scikit, scikit-learn.org/stable/modules/sgd.html.

[2] C. Link, Frederick, "Lie Detection Through Voice Analysis", Mar 7 2001, https://www.cia.gov/library/readingroom/docs/CIA-RDP96-00788R002000150003-2.pdf

[3] Harrison, Onel. "Machine Learning Basics with the K-Nearest Neighbors Algorithm." Medium, Towards Data Science, 14 July 2019, towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[4] https://en.wikipedia.org/wiki/Decision_tree_learning

[5] https://en.wikipedia.org/wiki/Support_vector_machine

[6] Lloyd, E.P., Deska, J.C., Hugenberg, K. et al. Miami University deception detection database. Behav Res 51, 429–439 (2019). https://doi.org/10.3758/s13428-018-1061-4

[7] Melendez, Steven. "Goodbye Polygraph? New Tech Uses AI to Tell If You're Lying." Fast Company, Fast Company, 24 May 2018, www.fastcompany.com/40575672/goodbye-polygraphs-new-tech-uses-ai-to-tell-if-youre-lying.

[8] Nelson, Dan. "Gradient Boosting Classifiers in Python with Scikit-Learn." Stack Abuse, Stack Abuse, stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/.

[9] Prieto, Miguel Delgado, et al. “Evaluation of Novelty Detection Methods for Condition Monitoring Applied to an Electromechanical System.” IntechOpen, IntechOpen, 31 May 2017, www.intechopen.com/books/fault-diagnosis-and-detection/evaluation-of-novelty-detection-methods-for-condition-monitoring-applied-to-an-electromechanical-sys.

[10] S. Venkatesh, R. Ramachandra and P. Bours, ”Robust Algorithm for Multimodal Deception Detection,” 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 2019, pp. 534-537, doi: 10.1109/MIPR.2019.00108.

[11] “Understanding Logistic Regression.” GeeksforGeeks, 30 May 2019, www.geeksforgeeks.org/understanding-logistic-regression/.

[12] V. Gupta, M. Agarwal, M. Arora, T. Chakraborty, R. Singh and M. Vatsa, “Bag-of-Lies: A Multimodal Dataset for Deception Detection,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 2019, pp. 83-90, doi: 10.1109/CVPRW.2019.00016.