An Acoustic Automated Lie Detector

Alice Xue

Dr. Hannah Rohde[1], Professor Adam Finkelstein[2]

[1] School of Philosophy, Psychology and Language Sciences, University of Edinburgh
[2] Department of Computer Science, Princeton University

I pledge my honor that this paper represents my own work in accordance with University regulations. – Alice Xue

Abstract

Current methods of lie detection are highly inaccurate and dependent on physiological and behavioral patterns. Less research has focused on creating a computational model to automate lie detection. This paper trains several machine learning models and a sequential neural network using solely acoustic features in speech for lie detection. Mel-frequency cepstral coefficients (MFCC), energy envelopes, and pitch contours are generated from a balanced dataset of deceptive and non-deceptive speech recordings collected from a 2-person lying game. The best model presented is a majority-voting ensemble learning classifier constructed from a Gradient Boosting Classifier (GBC), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) trained on MFCC and energy features. The maximum accuracy for lie detection achieved using this model is 55.8%, which outperforms the baseline chance accuracy of 50% and human accuracy of 48%. These results achieve an incremental improvement on a task which has monumental applications ranging from criminal investigations to national security.

1. Introduction

Detecting lies is a challenging and necessary pursuit, with widespread implications in many high-stakes scenarios, including police investigations, court decisions, and military circumstances. However, despite the significance of lie detection, modern-day methods of detecting deception are inaccurate. The polygraph, a common tool for lie detection, is dependent on physiological measures, but is easily fooled if its subject can suppress signs of physical discomfort.

The vast majority of previous research done in this regard has focused on studying visual and behavioral cues as evidence for deception (extensive review cited in **[1]**). Physiological responses, such as skin temperature and blood volume, has also been analyzed in laboratories as deceptive cues **[2].** Less work has been done to computationally and statistically predict lies. Therefore, it is necessary for the development of an automated system to approach this task, in hopes that a machine learning model may learn deceptive cues beyond the realm of human detection.

Various machine learning models and techniques have been researched in this regard. Unimodal deception detection using visual features has been implemented previously using Support Vector Machines (SVM) and Logistic Regression (LR) models, achieving an accuracy of 76.2% when using facial landmarks as input vectors **[3]**. Past research has also been done using multilayer perceptron (MLP) trained with 3D video features, textual information, audio features, and micro-expressions, achieving an accuracy of 96.14%. However, due to the small size of the dataset, the MLP was prone to overfitting and only applicable for a small domain **[4]**.

Aside from the visual cues to lying, multiple studies have been done to detect lies based purely on speech information. The Columbia-SRI-Colorado (CSC) Corpus has been widely used

as a deceptive speech dataset for training and testing machine learning models. The CSC Corpus was collected form 32 different native English speakers undergoing a laboratory interview process, where they were financially incentivized to lie to portray themselves to the interviewer as a target entrepreneur **[5]**. As the interviews were being conducted, subjects pressed a pedal below the table to indicate whether, at a specific time, they were telling a truth or lie. In total, the CSC corpus contains approximately 7 hours of subject speech, all of which were transcribed and aligned with audio **[5]**.

A study in 2005 used prosodic features of the CSC corpus (pitch, duration patterns, and energy) and lexical features (positive/negative flag words and filled pauses such as 'um's or 'ah's) as inputs to an SVM **[5]**. Spectral-based Mel cepstral features with energy were extracted to be fed into a Gaussian Mixture Model. A combination of the prosodic-lexical SVM system and audio-based Gaussian Mixture Model were found to increase accuracy metrics to 64.4%, with baseline chance accuracy being 60.4% **[5]**. Similarly, using a lexical-prosodic-acoustic combined feature set to train a Ripper rule-induction classifier, researchers were able to reduce the error rate from baseline (39.8%) to 33.6% on the CSC dataset **[6].** Other approaches to training a model on the CSC corpus have been to identify critical segments which are "hot spots" to tell if a speaker is telling a truth or lie **[7]**. After hand-selecting portions of the interviews that involved a direct question about the subjects' test scores, researchers were able to increase accuracy over baseline by 23.8% **[7]**.

Acoustic information is critical for the development of accurate machine learning models, and requires the preprocessing of data into different audio representations such as Mel-frequency cepstral coefficients (MFCC), pitch contours, and energy. Fundamental frequency contouring, or pitch information, can approximate the tonal features even in the presence of noise. However, the computation of pitch in audio is error-prone due to the ambiguity of pitch in human voice **[8]**, so MFCCs have also been widely-adopted. Based on a frequency domain scale that closely approximates the auditory processing of the human ear, MFCCs are widely used for speech recognition **[9]**. Along with energy information, both MFCCs and pitch information are widely used as machine learning input features for audio recognition and classification tasks. For a sample audio recording of human speech **[10],** these different audio representations are shown for side-by-side comparison **(Figure 1**).

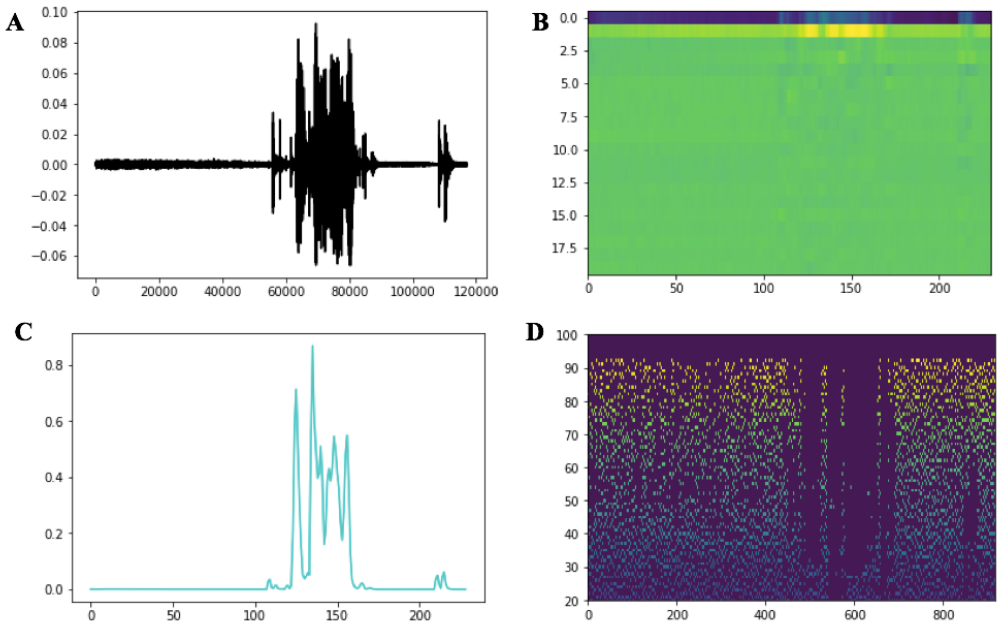

Figure 1: *Audio representations for a sample audio file in the dataset generated by Loy et al 2018* ***[10]*** ***(A)*** *Original audio waveform* ***(B)*** *Mel-frequency cepstral coefficients* ***(C)*** *Energy envelope* ***(D)*** *Pitch contours*

There has been substantially less research published on using purely acoustic data without supplementation of lexical and prosodic features. In practice, preprocessing lexical and prosodic features require hand-coding, thus adding extra overhead for anyone looking to use speech recordings for lie detection. Additionally, machine learning-oriented deceptive speech research has primarily focused on the CSC corpus, but for the purposes of broadening the applicability of lie detection models, more research is necessary on other speech datasets. Therefore, the research presented in this study is importantly unique in two regards: first, I focus on pure acoustical input features as inputs to a classifier, and second, the dataset used here has not been extensively used before to construct machine learning models for lie classification.

This study seeks to fully automate speech-based lie detection using machine learning models trained on acoustic data. The dataset consists of speech data from an interactive lying game, and is described in more detail in **Section 3.2**. MFCC, energy, and pitch information generated from speech recordings are fed into several different classifiers for comparison. The best model presented is a mode-based ensemble learning classifier which aggregates a Gradient Boosting Classifier (GBC), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) trained on MFCC and energy information.

2. Statement of Purpose

The purpose of this research is to automate lie detection in speech recordings using solely acoustic data.

3. Approach & Implementation

3.1 Tools

Coding was done in Python in Colab Notebooks. SciKit learn and TensorFlow with Keras API, a deep learning library wrapped around deep learning software in Theano, was used to construct the neural network. Librosa, a python package for audio analysis, was used for generating mel-frequency cepstral coefficients and pitch contour information. Numpy, a Python package for scientific computing, was used to store and retrieve information due to its support for large, multi-dimensional matrices.

Evaluation metrics, such as confusion matrices and epoch behavior, were visualized and plotted with SciKit learn, a machine learning library for python. MFCCs, energy, and pitch contours were graphed using Matplotlib, a Python 2D plotting library.

3.2 Dataset

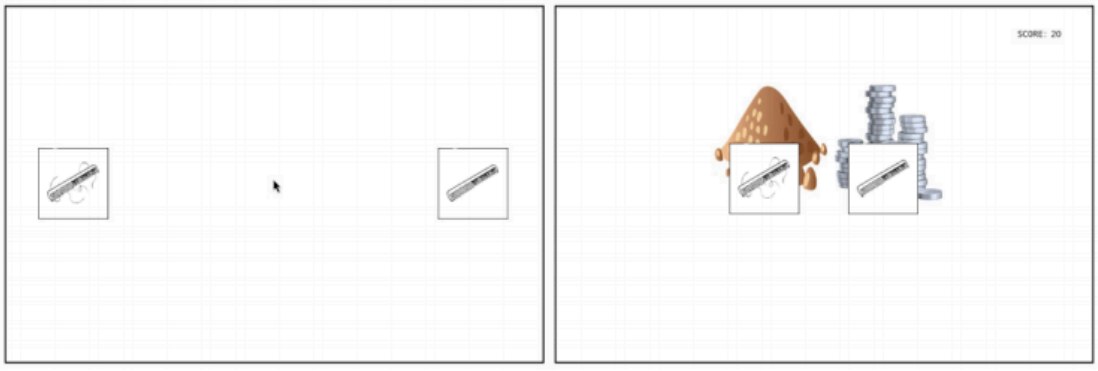

Figure 2: *Displays shown in 2-person interactive lying game, taken from paper published by Loy et al, 2018* ***[10]****.* ***(Left)*** *Guesser's display. Guesser must choose if the coins are behind comb with hair or comb without hair.* ***(Right)*** *Speaker's display, showing the position of the coins. Speaker then decides whether to lie and say the treasure's behind the comb with hair or comb without hair.*

The dataset used was collected in 2018 as part of an experimental study to test the validity of established cues to lying **[10]**. In the study, researchers created a 2-person, interactive lying game

where the two participants were seated in front of each other. They were shown two different displays (**Figure 2)**, where the speaker was shown a pile of coins behind one of two objects. The participant with role of speaker was financially incentivized to make their opponent, the guesser, incorrectly select the position of a "treasure." **[10]** All interactions were recorded and marked a truth or lie. For instance, in a sample recording, the speaker says "the treasure's behind the big crocodile with its mouth shut" and this recording is labeled as truth **[10]**. The dataset which was provided for this research initially had 932 total recordings for 24 subjects, which each recording averaging around 8 seconds each.

Initially, the dataset was skewed, with 498 truth recordings and 434 lie recordings. Under-sampling of the majority class was used prevent class imbalance from interfering with model training, and to standardize evaluation metrics to a baseline of 50%. Duplicate recordings were removed, and 61 truth samples were chosen at random to be omitted from the classification set. After cleaning and balancing the dataset, 830 total recordings were left, with 415 truths and 415 lies.

3.3 Classification

3.3.1 Feature Extraction

Three different features were extracted from each audio file in the dataset for use in classification: MFCC, energy, and pitch contours (examples of these audio representations for a sample recording is shown in **Figure 1**).

a. MFCCs were generated using Librosa (librosa.feature.mfcc), with 20 MFCCs generated per audio file, each MFCC padded with zeros to a fixed length of 1000.

b. Energy envelopes were calculated as the sum of the square of the magnitude of the speech signals **(Equation 1)**. The frame length was 512, and 512 samples were skipped between successive frames. All energy signals were padded with zeros to a fixed length of 500.

$$E_x = \sum_n |x[n]|^2 \quad \textbf{(1)}$$

c. Pitch contours were generated with Librosa (librosa.core.piptrak), which provides pitch tracking using the sinusoidal peak interpretation. The length of FFT window was set at 512. Each pitch contour was padded with zeros in the second dimension to length 2000 to standardize the size of inputs fed into the classifier.

3.3.2 Models

After extracting features from the dataset, various models were trained and tested on either MFCCs, energy, or pitch. The models were implemented using scikit-learn's library for logistic regression (LR), decision tree classifier (DTC), random forest (RF), gradient boosting classifier (GBC), linear kernel support vector machine (SVM), and stochastic gradient descent classifier (SGD). After training all models on either MFCC, energy, or pitch, evaluation metrics were taken (described below) and used for comparison.

3.3.2.1 Sequential Model

Aside from simpler models provided in scikit-learn, sequential models such as the recurrent neural network (RNN) have been widely adopted for sequential speech recognition and natural language processing tasks. In particular, the long short-term memory (LSTM) is an RNN architecture has been shown to allow state-of-the-art performance in speech recognition due to its ability to remember long-term patterns better than a simple RNN **[11]**.

In this research, a LSTM-based sequential model was implemented using Keras API for sequential layering. Two LSTM layers are used, the first with 16 LSTM units, and the second with 8 LSTM units. Due to the small size of the dataset, the models were prone to overfitting, which is a phenomenon where the model starts to learn the nuances in the training set rather than general features. To prevent this, dropout of 5% was added at each layer and batch normalization was added in between the layers. Finally, the LSTM outputs were flattened and fed into a fully-connected layer. The model was compiled with a binary cross-entropy loss function to calculate final probabilities of the original input being in each class. The model here uses the softmax loss function (**Equation 2),** which outputs probabilities for each class that sum to 1 and selects the highest probability as the selection of the predicted class. The optimizer used was the Adam optimization algorithm. The learning rate was set as 0.001 with a decay of 0. The final output of all classifiers was a binary determination of whether a speech recording is a truth or lie.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}} \quad \textbf{(2)}$$

3.3.2.2 Ensemble Learning

Ensemble learning techniques have been used to improve the results of weaker machine learning classifiers by aggregating their individual decisions using an ensemble algorithm. Ensembling techniques include majority voting, in which the mode of the classifiers' outputs is the ultimate output, bagging, boosting, and weighted average (see **[12]** for in-depth review). In order to increase the accuracy and F1 scores of the models in this research, majority voting ensemble learning was utilized for a combination of the best-performing classifiers trained on either MFCCs, energy, or pitch contours.

Several different combinations of ensembling were tried and tested in this paper. Ensembling was done both inter- and intra- feature. For each of the three types of audio features (MFCC, pitch, energy), ensembling was done between the other seven models tested on that specific feature. For instance, one intra-feature ensembling classifier was constructed from the outputs of LR, DTC, RF, GBC, SVM, and SGD models trained and tested on MFCC data only. These intra-feature ensembling classifiers were compared with inter-feature ensembling done between a combination of the strongest models trained on MFCC, pitch, or energy. The two inter-feature ensembles were 1) MFCC-GBC, MFCC-SVM, energy-GBC, energy-SVM, energy-SGD, pitch-RF, and 2) MFCC-GBC, MFCC-SVM, energy-GBC, energy-SVM, and energy-SGD.

3.3.3 Training

Input data was split into 70% training and 30% testing. Among the training set, 20% was set as validation set for each of the 15 epochs. Batch size was set at 20. During each epoch, each batch is sent to the network and then the validation set is tested on the network to find validation accuracy for that epoch.

3.3.4 Evaluation Metrics

Because the dataset was initially unbalanced, F1 scores were generated in initial model testing. F1 scores were later also generated on subsequent models trained on a balanced dataset, though since the classes were balanced, accuracy became a better measure for model performance. F1 score considers both accuracy and precision **(Equation 3)**. Accuracy is the percentage of cases the model correctly classified over the total population **(Equation 4).** Aside from F1 scores and accuracy,

confusion matrices, ROC-AUC plots and scores, and model loss and accuracy plots were generated for LSTM classifier outputs.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad \textbf{(3)}$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad \textbf{(4)}$$

There were several different metrics that could be successfully used as baselines to evaluate model performance in this research. One such baseline was the percent of cases that would be correctly identified if only truth was guessed for the entire dataset. This baseline, based solely on chance, was 50% for both accuracy and F1 for the balanced dataset. Another metric of comparison was the success of humans in the initial laboratory experiment from which this dataset was collected. The truth-lie discrimination accuracy for the guessers in the interactive lying game was 48%, so that was set as a "real-life" standard by which to judge the machine learning model because both received the same auditory information **[10]**.

4. Results and Evaluation

4.1 Models trained on only MFCC, Pitch, or Energy

Models were first trained on either MFCC, pitch, or energy individually, without any combination of the features. In total, there were 8 different models for each MFCC, pitch, and energy. In order to better visualize the values, bar graphs of are created of the evaluation metrics for each of the three features and will be further described below **(Figure 3)**. All raw values can be seen directly in **Table 1.** Metrics for pitch-trained gradient boosting classifiers are not shown because there was not enough RAM to run the GBC training using pitch as input features.

Feature	Model	Accuracy	F1 Score
MFCC	LR	0.514	0.502
	DTC	0.502	0.496
	RF	0.534	0.491
	GBC	0.518	0.524
	SVM	0.518	0.516
	SGD	0.482	0.303
	LSTM	0.546	0.580
	Ensemble	0.546	0.539
Pitch	LR	0.511	0.397
	DTC	0.504	0.467
	RF	0.539	0.370
	GBC	-	-
	SVM	0.507	0.395
	SGD	0.496	0.468
	LSTM	0.518	0.388
	Ensemble	0.511	0.422
Energy	LR	0.434	0.436
	DTC	0.498	0.473
	RF	0.494	0.452
	GBC	0.514	0.498
	SVM	0.502	0.541
	SGD	0.502	0.520
	LSTM	0.470	0.496
	Ensemble	0.53	**0.624**
MFCC, Energy	Ensemble	**0.558**	0.574
MFCC, Pitch, Energy	Ensemble	0.534	0.525

Table 1: *Evaluation metrics for all models. Highest accuracy and F1 score are highlighted in yellow.*

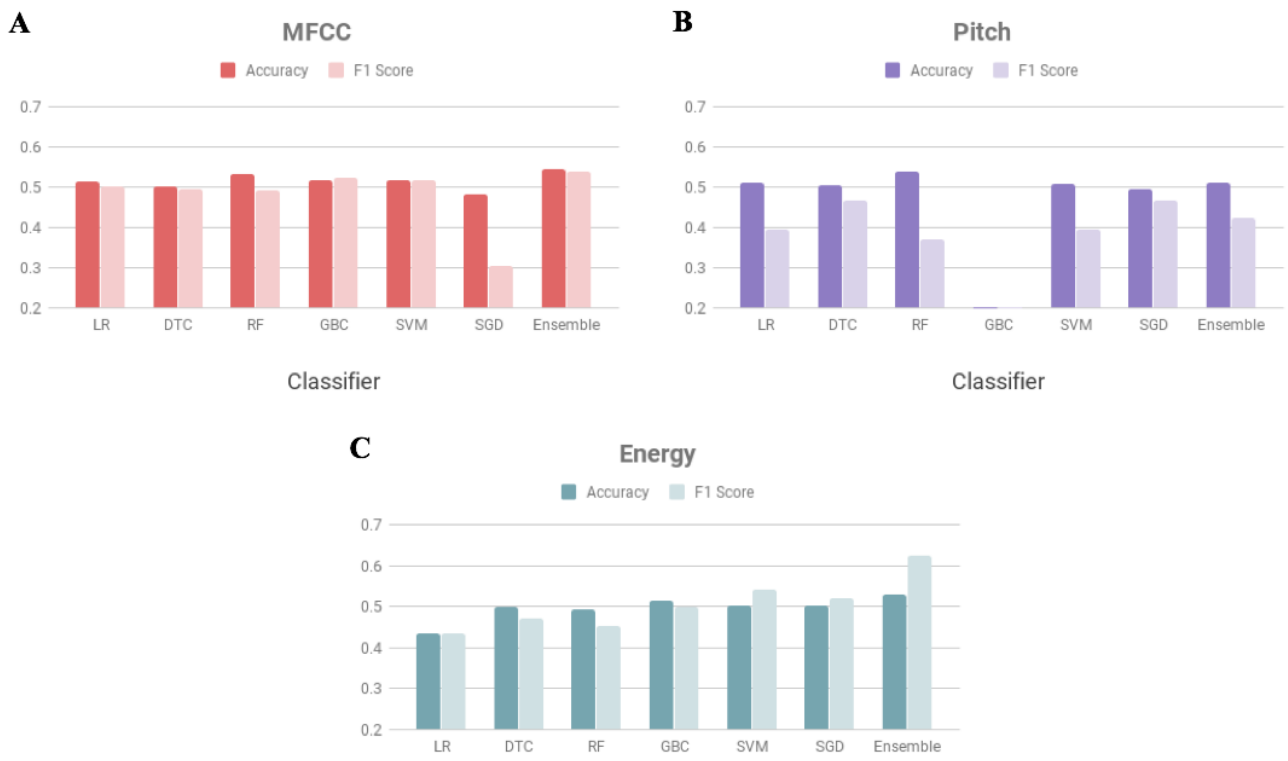

Figure 3: *Accuracy and F1 scores for MFCCs **(A)**, pitch* **(B)**, and energy **(C)** trained models. Abbreviations for classifiers are as follows: LR = logistic regression; DTC = decision tree classifier; RF = random forest; GBC = gradient boosting classifier; SVM = linear kernel support vector machine; SGD = stochastic gradient descent classifier; Ensemble = majority-voting based ensemble classifier of all the other 7 models trained on either MFCC, pitch, or energy.*
**GBC metrics are missing from pitch due to computational limitations (too little RAM).*

The highest-performing model for MFCC was the LSTM-based sequential model, with an accuracy of 54.6% and F1 score of 0.580, which was greater than even the ensemble classifier whose F1 score was 0.539. The worst model trained on MFCCs was the stochastic gradient descent, which had the lowest accuracy and F1 score. Ensembling for MFCC did not increase model performance over the LSTM and instead had a lower F1 score than the LSTM.

Within the models for pitch, the highest-performing model in terms of accuracy was the random forest classifier, at 53.9%. The worst model for pitch was again the stochastic gradient descent classifier, with an accuracy of 49.6% and F1 score of 0.468. None of the F1 scores in the

pitch-trained models are an improvement over the baseline of 0.5. Pitch-trained models were thus the worst-performing models overall, compared to MFCC and energy.

The ensemble classifier for energy had the highest F1 score across the board, at 0.624, even higher than the inter-feature ensemble classifiers. For both MFCC and energy, ensembling the seven classifiers was an overall improvement over the other models. However, this was not the case for pitch, in which the ensemble was the runner-up classifier for accuracy.

4.1.1 LSTM Results

Because of the dearth of research on sequential models for use in speech-based lie detection, this paper will expand on results obtained from training the LSTM-based sequential model. It was hypothesized that, due to the LSTM's ability to learn and remember long-term patterns, it may be able to pick up on nuanced features in long inputs and correlate those to speech deception.

Surprisingly, the LSTM did not outperform the other models and was, in general, worse than any of the ensembling classifiers, with the exception of the MFCC-trained LSTM. This is most likely due to overfitting of the model, seen from the gap between the training and testing accuracies. For MFCC, the sequential model had a testing accuracy of 54.6% and training accuracy of 75.4%. For energy, the sequential model had a testing accuracy of 47.0% and training accuracy of 76.9%. For pitch, the gap was smaller with a testing accuracy of 51.8% and training accuracy of 60.1% The substantial decrease in accuracy for the testing set is a strong indicator that the model is being fitted to the specific nuances of the training set, rather than learning the overall features as they pertain to deceptive speech. However, due to the smaller size of the dataset, with only 830

files, training a multi-layered sequential model is bound to be prone to overfitting. More data is needed for a successful sequential model.

The MFCC-trained LSTM had an AUC-ROC score of 0.59, which is higher than the baseline AUC-ROC score of 0.5. With an accuracy of 54.6%, it was also the most accurate model, both across the other LSTMs and across the other MFCC-trained models. Pitch-trained LSTMs also had an AUC-ROC score above baseline (0.538). The energy-trained LSTM was worst-performing and had AUC-ROC score below the baseline, at 0.493. These results indicate that overall, the MFCC-trained LSTM delivers the most promising results for further experimentation with larger datasets **(Table 2)**.

Feature	Accuracy	AUC
MFCC	0.546	0.59
Pitch	0.518	0.538
Energy	0.470	0.493
Baseline	0.5	0.5

Table 2: *Evaluation metrics for LSTM.*
AUC = area under ROC curve.

4.2 Inter-feature Ensemble Learning: Combining MFCC, Pitch, Energy

Further improvements to machine learning models are made through ensemble learning, which allows mistakes from an individual model to be buffered through a majority-voting algorithm. The output with the most "votes" from the outputs of each classifier in the ensemble is deemed the ultimate output. The ensemble results for intra-feature ensembling were described in **Section 4.1**.

Inter-feature ensembling was constructed with various combinations of classifiers. One ensemble was a combination of the best classifiers among each of the features. This comprised of the gradient boosting classifier for MFCC and energy, support vector machines for MFCC and

energy, the stochastic gradient descent classifier for energy, and the random forest classifier for pitch. This ensemble had an accuracy of 53.4% and F1 score of 52.5%. Pitch-trained models were overall the worst performing models, so only one model for pitch was selected.

The best-performing ensemble was created from the same combination of classifiers just described, but without the pitch-trained random forest classifier. This was not only the best-performing ensemble, but the best model out of all the models presented here in this study **(Figure 4)**. The accuracy for this ensemble was 55.8% and F1 score was 0.57. The ensemble outperformed all models that had been individually trained on either MFCC, pitch, and energy **(Figure 4)**.

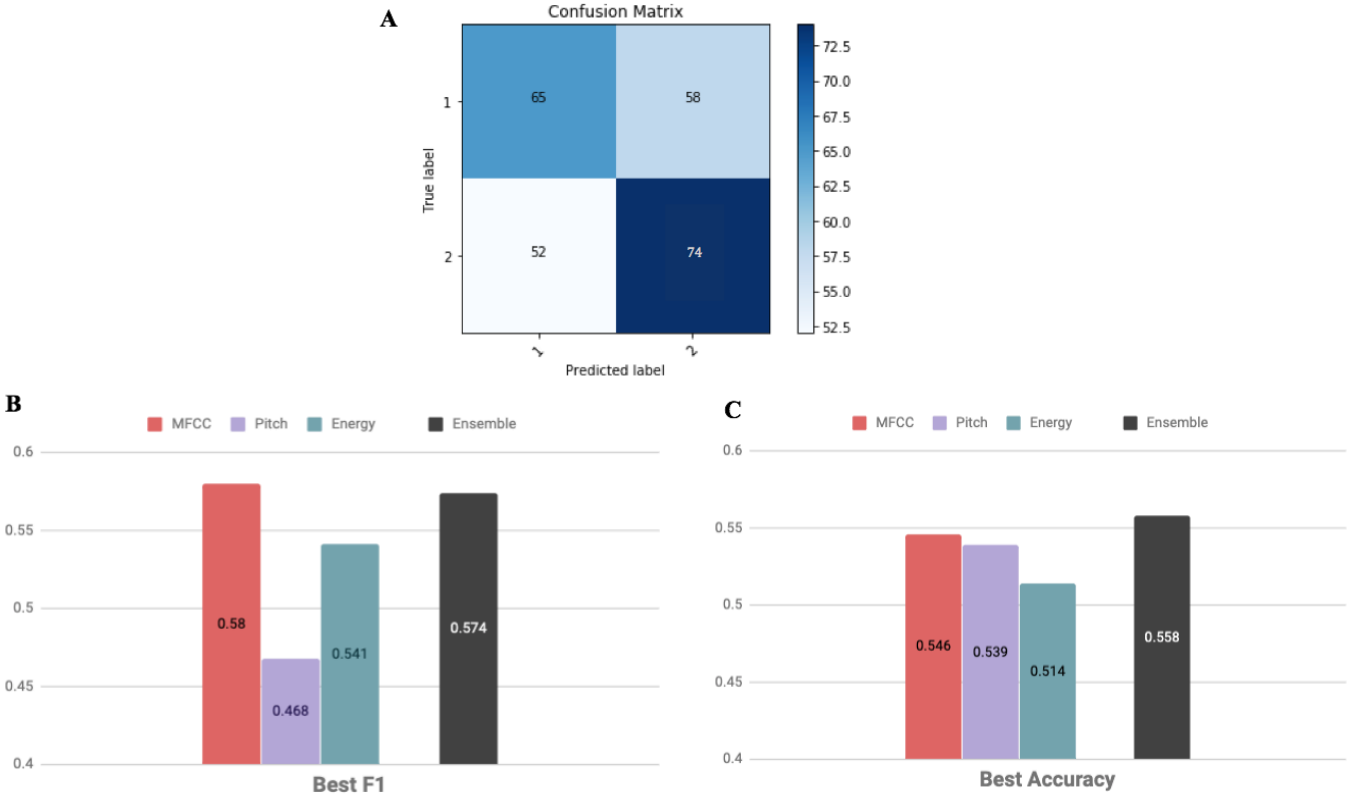

Figure 4: *Evaluation metrics for the best-performing classifier, an ensemble-learning classifier based on MFCC and pitch.*
(A) *Confusion matrix for the ensemble classifier*
(B) *Bar graph showing the F1 score of the ensemble classifier compared with the best F1 scores among models trained individually on either MFCC, pitch, or energy*
(C) *Bar graph showing the accuracy of the ensemble classifier compared with the best accuracy among models trained individually on either MFCC, pitch, or energy.*

5. Discussion

Previous research done with the same dataset used in this research reported a 48% truth-lie discrimination accuracy for human subjects undergoing the interactive lying game. The other baseline used in this research was the chance accuracy, which was 50% due to the fact that the dataset was balanced evenly between truth and lie recordings for purposes of model training. With a 55.8% testing accuracy, the highest-performing model presented in this study improves upon both the chance and human subject baseline accuracies (**Table 3**). An LSTM-based sequential model also delivered promising results, at a 54.6% accuracy when using MFCC as input features.

	Accuracy (%)
Chance	50
Human subjects (Loy, 2018)	48
Current study	**55.8**

Table 3: *Current study vs baseline accuracies*

It is important to note that the human guessers not only had auditory information but also full visual, prosodic, and lexical information of the lying subject. Even with a full range of audio-prosodic-visual cues that could help them interpret if the speaker's deception, the human guessers could not outperform a machine learning model trained with only acoustic information.

However, an accuracy of 55.8%, though higher than baseline, is still only a modest improvement. This may be explained by several limitations of this research. First is the general lack of data which limits the accuracy of neural network models due to the tendency for models trained on small datasets to overfit. In order to balance the dataset 61 recordings labeled "truth"

were discarded randomly, so the dataset was even smaller than what was already a small dataset of 932 files. Second, in listening to the truths and lies in the dataset itself, it is hard to discern noticeable fluctuations in speech characteristics from lie to truth. The game itself has more complex workings that just telling a lie or truth in isolation. Using psychological tactics such as reverse psychology, the speakers may choose to purposefully insert filled pauses or manipulate the tone of their voice to deceive the guesser into thinking that a lie is being told.

Training the model with male and female data together also introduced unwanted variability in pitch within a single class, due to the fact that male and female voices have intrinsic differences in pitch. This was perhaps also the reason why pitch inputs to the models were generally worse-performing, especially in F1 scoring, than MFCC or energy-trained models, which were more robust to gender differences. For future machine learning research, this observation should be taken into account if using a dataset consisting of male and female speakers.

The fact that the ensemble learning classifier arrived at an accuracy higher than chance signifies that it indeed found a pattern in acoustic information that can be correlated to deception. Thus, I conducted an experiment to try to interpret the model and understand what exactly it was learning. To this end, a list of files that had been scored unanimously by all the best-performing classifiers was compiled. Recordings and MFCC data were displayed side by side in Colab, and I subjectively examined the list of files to see if I could detect a pattern between the files that the models confidently labeled true or false. However, I could detect no such pattern. Recordings that sounded similar were labeled truth in one recording and false at another. It would be an interesting experiment to recruit more human ears to see if anyone can find a detectable pattern within recordings labeled truth and recordings labeled lie.

6. Research Significance

Past research in lie detection has primarily focused on physiological and behavioral signs as cues for deception, but an automated system this task can be better achieved using computational models. This research evaluates lies based solely on acoustic information, which, unlike lexical and prosodic speech features, does not require extra overhead of hand-coding and aligning transcripts to audio. Additionally, the dataset used in this research has not previously been used to build both computational models and neural networks. Presented here is a model that is able to detect acoustic patterns in speech recordings to detect lies. This research provides an important incremental step toward automating lie detection, which is a challenging task that cannot be perfected using one specific measure. Lie detection has far-reaching implications in the political and security sphere, where accurate lie detection is integral for criminal investigations, evaluating government reports, and high-stakes military scenarios.

7. Future Work

Because this dataset has not been extensively studied through the lens of machine learning, it would be interesting to further develop models, expanding from solely acoustic information to encompass visual, prosodic, and lexical information. This would allow us to see if research methods done on the CSC corpus may be similarly extended to other types of data. With a combination of video, audio, and transcriptional information, it may possible to build a powerful machine learning model on this dataset that can more accurately classify truth or lie.

Furthermore, interpreting the model may have important implications in psychological and behavioral understandings of cues to deception. If we are able to understand the patterns of the model and upon what it bases its classifications, we can apply that to further understand human behavior.

Lastly, there is little literature surrounding the CSC corpus that makes use of ensemble learning to aggregate and strengthen machine learning classifications. Using the ensemble methods tested in this research, further research done on the CSC corpus may be conducted to improve upon reported prosodic, lexical, and acoustic based models in previous literature.

8. Conclusions

This study focused on acoustical features such as MFCC, energy, and pitch contour information to build machine learning models to automate lie detection. An LSTM-based sequential model is tested and found to produce higher-than-chance accuracy (54.6%) using MFCCs as input features. The best-performing model in this research is an ensemble-learning classifier which is constructed from a majority voting system, based on outputs from the Gradient Boosting Classifier (GBC), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) trained on MFCC and energy information. Using this classifier, the maximum accuracy achieved is 55.8%, which presents an improvement over the baseline chance accuracy of 50% and human accuracy of 48%. This has important implications in lie detection automation, which, if perfected, will revolutionize criminal investigations, military proceedings, and national security protocols.

9. Acknowledgements

I would like to acknowledge my advisors, Dr. Hannah Rohde from the University of Edinburgh and Professor Adam Finkelstein from Princeton University, for all their support and guidance. I would also like to thank Dr. Zeyu Jin for his advice. The dataset used in this research was generously provided by Dr. Jia Loy.

10. Bibliography

[1] Bella. M. DePaulo, James J. Lindsay, Brian E. Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. 2003. Cues to Deception. *Psychological Bulletin*, vol 129, no 1. Pages 74–118. http://www.angelfire.com/lieclass/C2D.pdf

[2] Veronica Perez-Rosas, Rada Mihalcea, Alexis Narvaez, and Mihai Burzo. 2014. A multimodal dataset for deception detection. *Language Resource Evaluation Conference*. pages 3118–3122. http://www.lrecconf.org/proceedings/lrec2014/pdf/869 Paper.pdf.

[3] Samin Azhan, Anik Zaman, Monjur R. Bhuiyan, 2018. Using Machine Learning for Lie Detection: Classification of Human Visual Morphology. *Brac University*. http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/10144/14101005%2C17241023%2C17241022_CSE.pdf?sequence=1&isAllowed=y

[4] Gangeshwar Krishnamurthy, Navonil Majumder, Soujanya Poria, and Erik Cambria, 2018. A Deep Learning Approach for Multimodal Deception Detection. *CoRR.* https://arxiv.org/pdf/1803.00344.pdf

[5] Martin Graciarena, Elizabeth Shriberg, Andreas Stolcke, Frank Enos, Julia Hirschberg, and Sachin Kajarekar. 2006. Combining prosodic lexical and cepstral systems for deceptive speech detection. *IEEE International Conference on Acoustics, Speech and Signal Processing* http://www.cs.columbia.edu/nlp/papers/2006/graciarena al 06.pdf.

[6] Julia Hirschberg, Stefan Benus, Jason M. Brenier, Frank Enos, Sarah Friedman, Sarah Gilman, Cynthia Girand, Martin Graciarena, Andreas Kathol, Laura Michaelis, Bryan Pellom, Elizabeth Shriberg, and Andreas Stolcke. 2005. Distinguishing deceptive from non-deceptive speech. *INTERSPEECH* pages 1833–1836. http://www.cs.columbia.edu/julia/files/hirschbergal_05.pdf.

[7] Frank Enos, Elizabeth Shriberg, Martin Graciarena, Julia Hirschberg, and Andreas Stolcke. 2007. Detecting deception using critical segments http://www.cs.columbia.edu/frank/papers/enos_et_al_is2007.pdf.

[8] R. N. Shepard, "Circularity in judgments of relative pitch," *The Journal of the Acoustical Society of America*, vol. 36, no. 12, pp. 2346–2353, 1964.

[9] N. Dave. "Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition," in *International Journal for Advance Research in Engineering and Technology,* 2013.

[10] Loy, J. E., Rohde, H., & Corley, M. (2018). Cues to Lying May be Deceptive: Speaker and Listener Behaviour in an Interactive Game of Deception. *Journal of Cognition*, *1*(1), 42. DOI: http://doi.org/10.5334/joc.46

[11] Hasim Sak, Andrew Senior, Francoise Beaufays. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. 2018. *CoRR*. http://arxiv.org/abs/1402.1128.

[12] Dietterich T.G. (2000) Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. *Lecture Notes in Computer Science*, vol 1857. Springer, Berlin, Heidelberg