Determining the Likelihood of a COVID-19 Diagnosis Using Audio Data and Ensemble Machine Learning

Drake Spence & Jared Santos

Wichita State University

BSAN 735

**Abstract**

Previous literature has traditionally focused on converting cough audio data into spectrograms for training convolutional neural networks. In this study, we compare the results of the “Screening COVID-19 by Cough Audio Data Using Attention-Based Convolutional Recurrent Neural Networks” paper with our own methods to accurately classify coughs as COVID-19. While the original tested between four models and then chose the best one, our team believes that deploying an ensemble of three different classification algorithms (Logistic Regression, Long Short-Term Memory Recurrent Neural Network, and a Convolutional Neural Network) would lead to superior results that are more generalized for a real-world application. The initial results of the experiment were very promising; our individual neural networks outperformed the models in the original paper with AUCs of .89 and .79 vs the original paper’s AUCs of .53 (for both individual CNNs) and .80 for a CNN with Attention. Additionally, the overall ensemble matched the performance of the original paper’s best model with an AUC of .80. However, we believe that this ensemble approach results in a more generalized algorithm that can leverage the specific context surrounding the audio data to more accurately make a diagnosis. There is still room to improve the ensemble, namely by using a stronger classification algorithm in place of the Logistic Regression and leveraging self-attention in our ensemble.

**Introduction**

Although the COVID-19 pandemic is mostly in the past, the effects of it will be felt for generations to come. As such, it will be ever more important to improve our response to a pandemic to be as quick, efficient, and healthy as it can be. Currently the standard ways that medical professionals test for covid involve some form of physical exposure to the patient that may lead to infection. Additionally, the tests are also invasive and require an uncomfortable swab of the patient’s nostrils. However, what if one was able to accurately determine whether someone had covid based on the location an exam took place and an audio clip of the person’s cough?

To improve upon the study conducted by Gu, C. et al (15), the team decided to leverage ensemble machine learning and change the data preparation and feature selection process to find an answer for the above question. The ensemble consists of a recurrent neural network using “Long Short-Term Memory” (LSTM) and a Convolutional Neural Network (CNN) for the audio clips due to the sequential nature of the data and a logistic regression for the non-sequential metadata that provides context to the audio clips (such as location, time, date, etc.).

**Related Works**

To find less invasive methods to make COVID-19 diagnosis, there has been increasing interest in the use of audio data to diagnose COVID-19 by analyzing coughs, breathing patterns, and speech (1). This study will equally focus on cough sounds and the metadata surrounding them. As such, the team has identified four primary datasets leveraged in other literature: COUGHVID (2), Virufy (3), Cambridge (4), and Coswara (5). Prior research deployed feature selection techniques such as Spectral Contrast, Mel-scale spectrogram, Chromagram, and Tonal Centroid (6) before applying machine learning models such as Convolutional Neural Networks (CNN), fully connected models, XGBoost, and random forest to classify cough sounds (3, 6).

Across the literature that the team reviewed, there are various methods and techniques used to complete this type of work. For example, Chaudhari et al. (3) used an ensemble model of CNN and fully connected models with various features, while Esposito et al. (7) employed an ensemble of Convolutional Recurrent Neural Networks (CRNNs) adapted from Visual Geometry Group 13 (VGG13) architecture. Since there is no clear benchmark for COVID-19 diagnosis using audio data, it is difficult to compare results across studies. Thus, it is still too early to determine what "state-of-the-art" looks like with regards to specific machine learning architectures and research methods for this use case. However, neural networks are commonly used in deep learning architectures, and the recurrent and convolutional variations of them will be used as the baseline for this study.

**Dataset and Features**

Before creating the initial data sets that the ensemble would consume, the team completed Exploratory Data Analysis (EDA) to get a better understanding of what the raw data set contained and develop a strategy to create the machine learning data sets based upon the EDA which can be seen below:

Application

Description automatically generated

In the exploratory analysis above, the top left image is a visual representation of the raw audio files the team will be feeding into the LSTM and CNN. Further, the top right image is a histogram showing the balance of gender and whether a cough was detected in the individual. That visualization on its own is not enough to show that gender may be a significant factor in predicting COVID-19 (especially with the large amount of “No Answer” responses), however when compared against the image on the bottom right, the complete story shows that there is a difference in positive COVID-19 diagnosis across reported genders (male and female) of about 10%. This shows that males are more likely to be diagnosed with covid, so the team was sure to include the feature in the dataset for the logistic regression. Lastly, the bottom left image is a geographic distribution of the positive COVID diagnosis in our data set. As one can see, the data has a large concentration of positive diagnosis in Europe and several pockets across Southwest Asia and Latin America. As such the team rolled up the geographic data into “regions” which could then be used as labels for the individual coughs indicating where they are.

Although the overall data set will be feeding the logistic regression, the recurrent neural network, and the convolutional neural network, the specific features of interest for these models will be different. The features of interest for the logistic regression can be seen in the table below:

|  |  |
| --- | --- |
| **Variable Name (Independent or Dependent)** | **Data Type** |
| **Cough\_Detected (Independent)** | **Float64** |
| **Age (Independent)** | **Int** |
| **Is\_Female (Independent)** | **Boolean** |
| **Is\_Male (Independent)** | **Boolean** |
| **Symptomatic (Independent)** | **Boolean** |
| **Season\_Autumn (Independent)** | **Boolean** |
| **Season\_Summer (Independent)** | **Boolean** |
| **Season\_Spring (Independent)** | **Boolean** |
| **Africa (Independent)** | **Boolean** |
| **Asia (Independent)** | **Boolean** |
| **Europe (Independent)** | **Boolean** |
| **North America (Independent)** | **Boolean** |
| **Oceania (Independent)** | **Boolean** |
| **South America (Independent)** | **Boolean** |
| **COVID-19 (Dependent)** | **Boolean** |

The table above is comprised of the “metadata” that gives context to the audio files that are at the heart of the data set. In the original paper, this information was only really used as a part of the exploratory data analysis, however the team believes this information will be critical to creating a generalized ensemble model for predicting the presence of covid in someone’s cough.

Likewise, a numeric representation of the “.wav” files that the team will be feeding into the LSTM and CNN can be seen below:

Table

Description automatically generated with medium confidence

The reason the data needs to be converted into a numeric representation (a collection of arrays), is because a machine learning model does not “listen” to the audio the same way a human’s ears do. Instead, one must represent the audio one would normally hear into a format the model can understand which is numeric.

Lastly, the team has also opted to classify both symptomatic responses and non-symptomatic responses as “healthy.” This is because the prediction of whether someone has COVID-19 is really what is of interest, and in this dataset being symptomatic is an entirely different classification. After the reclassification of said variable, the team then moved to balance the data set by using over sampling to ensure our target observation numbers aligned with the other observations.

**Comparison Strategy**

To establish a baseline for comparison, the team evaluated the results of the ensemble against the performance scores reported in (15). An overview of the architectures of all three models within the ensemble can be found below:

Text

Description automatically generated

A key difference between this study and (15) is the process by which feature selection and data preparation is being conducted (explained in the previous section). As such, there is no proper baseline that can be established between the data sets as they are fundamentally different. However, these changes were made to improve upon the results reported in the original study. Therefore, outside of the results from the original study, there is no further interest in establishing a baseline in any other way.

**Methods**

As discussed in the introduction, the goal of this project is to create an ensemble machine learning model comprised of a logistic regression, Long Short-Term Memory (LSTM) recurrent neural network, and a convolutional neural network (CNN). The logistic regression is being trained using K-Fold cross validation (5 folds) on 80% of the data and then evaluated on the remaining 20% of the data. Further, the team is also calculating the probability of the prediction being made by the logistic regression for later use. The LSTM is being trained and validated using K-Fold cross validation (5 folds) on 80% of the data and then evaluated on the remaining 20%. The CNN is being trained with a 60% training split, being validated on a 20% validation split, and being evaluated on a 20% testing split.

For both LSTM and CNN, we are using the .wav audio files as the feature (X) and the covid variable as the target variable (y). Initially the dataset was comprised of one thousand five hundred and twenty-four cases where the person did not have covid and six hundred and twenty cases where the person did have covid. To address this imbalance, we decided to oversample to increase the number of samples in our minority class. The length of our audio files varied between two seconds and fifteen seconds. Using librosa, we padded or truncated every audio clip to be 10 seconds long. To further process these audio clips, the team removed two seconds of null audio data bringing the length every audio file to 8 seconds and set the sampling rate for each audio clip eight thousand. Next, we used Mel Frequency Cepstral Coefficients (MFCCs) to represent the spectral characteristics of an audio signal in a compact form that can be read by the computer and used in machine learning algorithms.

The architecture of our LSTM model is a 3-layer sequential model. The first layer is comprised of sixty-four units with an input shape being time steps and number of coefficients. Time steps refers to the number of time steps in the input sequence and number of coefficients refers to the MFCCs used to represent each frame of the audio signal. The output of this layer will be a sequence of vectors. The second layer is comprised of thirty-two units. This layer takes the output sequence from the first layer as the input and produces a single output vector. The final layer is a dense layer with one unit and a sigmoid activation function. This layer produces a binary output indicating whether the input sequence belongs to a certain class. This class would be binary and show that either the audio suggests the individual does not have COVID (0) or does have COVID (1). The model also uses Adam optimizer and a binary crossentropy loss function as hyper parameters. Adam optimizer is a stochastic gradient descent optimization algorithm, and it calculates adaptive learning rates for each parameter based on the first and second moments of the gradients. This then allows the algorithm to converge faster and more efficiently than other optimization methods. The binary crossentropy loss function measures the difference between the predicted probability distribution and the actual probability distribution of the binary class variables. The goal is to minimize the difference. Since this LSTM is in a KFOLD, we are using ten epochs and our batch size is thirty-two. We found this combination of epochs and batch size to produce the best results.

Our CNN model is a standard 13-layer sequential model. We used two Conv2d layers, three batch normalization layers, two max pooling2d layers, three dropout layers, one “flatten” layer, and two dense layers. The conv2d layers apply convolution operation and the activation for both are relu. The batch normalization layers normalize the output of the previous layers which reduces internal covariate shift and improves the stability and speed of the training. The max pooling layers help reduce the spatial dimensions of the feature maps which makes the model more computationally efficient. Our first two dropout layers drop out 0.25 of the input units and the last one drops 0.50. The reason to have these dropout layers is to prevent the model from overfitting. The flatten layer takes the previous layer and flattens it to a one-dimensional array, which is then passed to our dense layer. Lastly, our first dense layer has a relu activation and our last layer, which is also a dense layer, has a SoftMax activation function. The SoftMax function produces a probability distribution over the output classes, and the class with the highest probability is chosen as the predicted output. The optimizer being used on our CNN model is the Adam optimizer and our loss is the sparse categorical crossentropy. Our batch size is thirty-two and we have fifty epochs. Having lots of epochs could lead to overfitting, but since we have three dropout layers, we did not experience any overfitting issues. The input data was reshaped to 4D since our Conv2d layer expects a 4D tensor.

Finally, to bring together the results of each of the models described above, the team has decided to deploy a simple voting method. This works be compiling the predictions of each of the models, and then calculating the mode of the predictions being made. Whichever prediction shows up the most often, is what the ensemble will assign as it is true “prediction.” In the case of a tie, we will instead take the probability of each of the predictions made by the models and average them across all the models. If the average probability of every prediction is over .5, the ensemble will determine that the ultimate prediction is true. Otherwise, it will assign a “false” value to the prediction.

**Experiment & Results**

After we trained and validated our models, the main scores we tested for include test loss, test accuracy, classification report, AUC-ROC, and confusion matrix. The test loss is important because it measures the performance of a model on data that it has not seen during training. A good test loss means that the model has generalized well to new data and should make accurate predictions on unseen examples.

The accuracy on the test set indicates how well the model can perform on new, unseen data, and is a crucial metric for assessing its overall performance. A high-test accuracy means that the model is generalizing well and making accurate predictions on new data. On the other hand, a low-test accuracy may indicate that the model is overfitting to the training data and is not able to generalize well to new data.

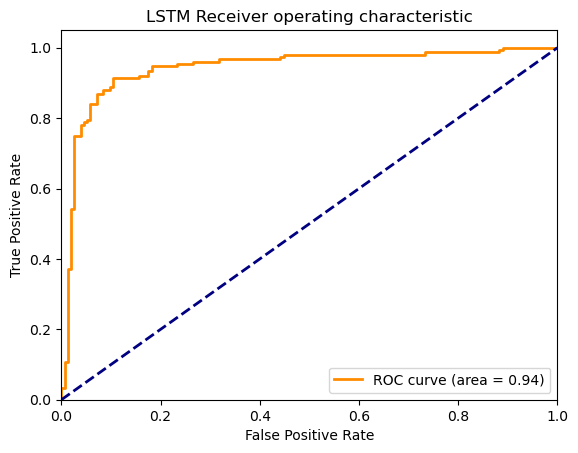
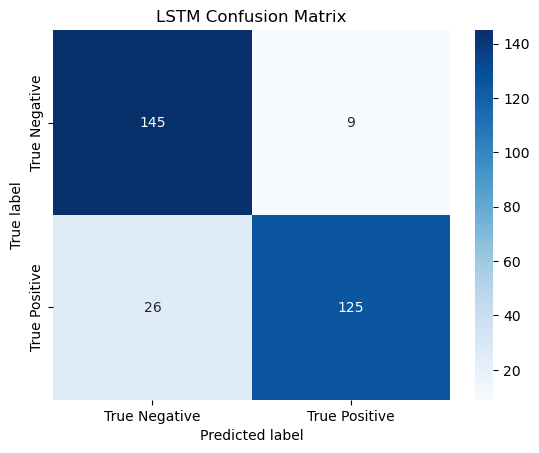
The classification report is important because it provides a comprehensive evaluation of the performance of a classification model. It gives an overview of how well the model is performing in terms of precision, recall, f1-score, and support for each class.

Precision is the ratio of true positives to the total number of positive predictions made by the model, and it measures the model’s ability to identify only the relevant instances. Recall, on the other hand, is the ratio of true positives to the total number of actual positive instances, and it measures the model's ability to identify all the relevant instances. F1-score is the harmonic mean of precision and recall, and it provides a balanced measure of both precision and recall.

The AUC-ROC score summarizes the performance of the model across all thresholds and provides a single number that represents the model’s ability to distinguish between positive and negative classes. A higher AUC-ROC score indicates better classification performance. Lastly, the confusion matrix helps identify the number of true positives, false positives, true negatives, and false negatives.

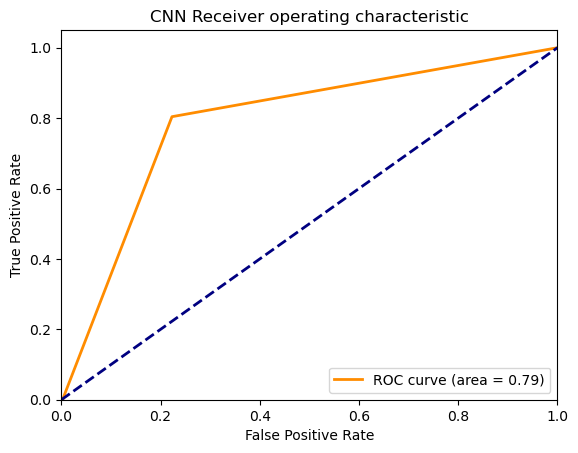
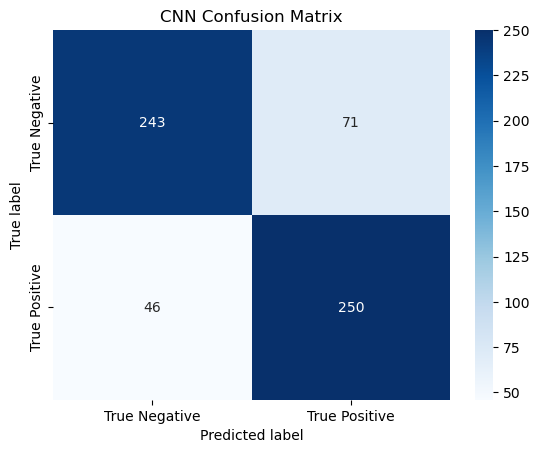
The LSTM test results for test loss is 0.3958, test accuracy is 0.8144, and the AUC-ROC score is 0.8907. Our average classification results and confusion matrix are

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Accuracy** | **AUC-ROC** | **Loss** | **F1 Score** | **Precision** | **Recall** | **Support** |
| **0.8144** | **0.8907** | **0.4030** | **0.8133** | **0.8218** | **0.8172** | **153** |



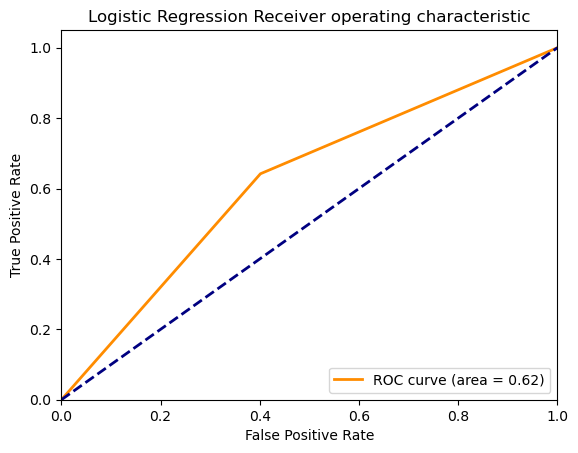
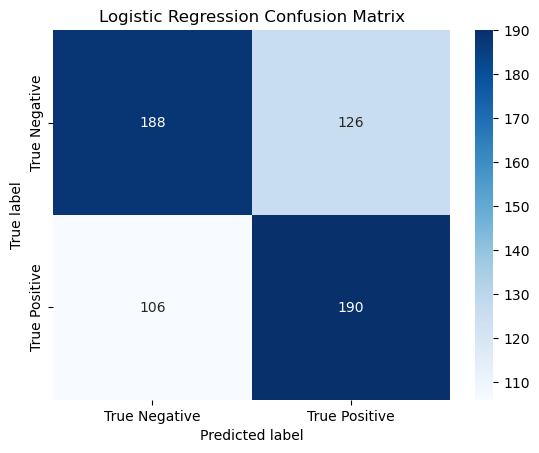
For our CNN, the test loss is 0.7945, test accuracy is 0.8082, and our AUC-ROC score is 0.7897. Classification results and Confusion matrix are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Precision** | **Recall** | **F1-Score** | **Support** |
| **0** | **0.84** | **0.77** | **0.81** | **314** |
| **1** | **0.78** | **0.84** | **0.81** | **296** |
| **Accuracy** | **-** | **-** | **0.81** | **610** |
| **Macro Avg** | **0.81** | **0.81** | **0.81** | **610** |
| **Weighted Avg** | **0.81** | **0.81** | **0.81** | **610** |



With our logistic regression, the things we checked for are the classification report, AUC-ROC score, confusion matrix, and p-values. The test accuracy is 0.6197 and the AUC-ROC score is 0.6203. Here are our classification results and confusion matrix.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Precision** | **Recall** | **F1-Score** | **Support** |
| **0** | **0.64** | **0.60** | **0.62** | **314** |
| **1** | **0.60** | **0.64** | **0.62** | **296** |
| **Accuracy** | **-** | **-** | **0.62** | **610** |
| **Macro Avg** | **0.62** | **0.62** | **0.62** | **610** |
| **Weighted Avg** | **0.62** | **0.62** | **0.62** | **610** |

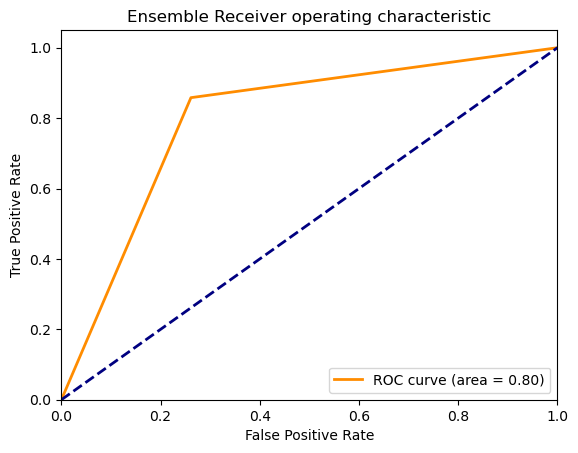
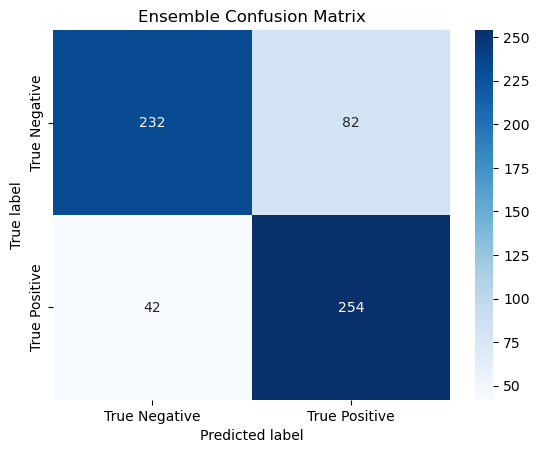
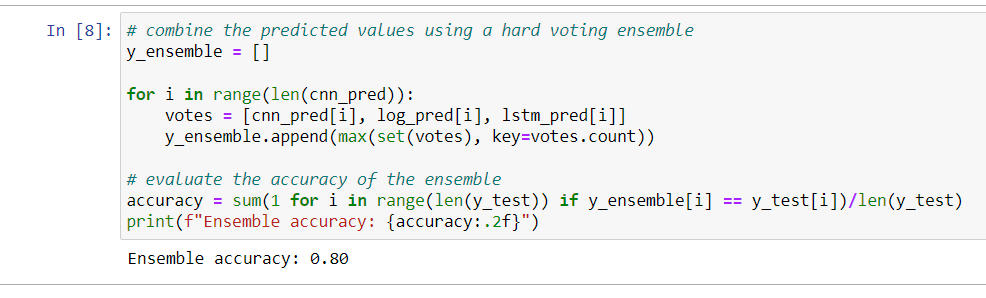


When it comes to our logistic regression, we wanted to check the p-values so we could see which features are statistically significant. These results indicate the statistical significance of each feature in predicating the target variable. The null hypothesis for each feature is that it has no effect on the target variable, and the alternative hypothesis is that it does have an effect. A p-value less than 0.05 is considered statistically significant. Here are our p-value results for our 13 features.

|  |  |
| --- | --- |
| **Features:** | **P-Value:** |
| **Cough Detected** | **0.0002** |
| **Age** | **0.1304** |
| **Female** | **0.2083** |
| **Male** | **0.2083** |
| **Autumn** | **0.0000** |
| **Spring** | **0.0000** |
| **Summer** | **0.0264** |
| **Africa** | **0.6680** |
| **Asia** | **0.0000** |
| **Europe** | **0.0000** |
| **North America** | **0.0038** |
| **Oceania** | **1.0000** |
| **South America** | **0.0472** |

From our results, feature numbers 2,3,4,8, and 12 have a p-value greater than 0.05 so we can conclude that they have no effect on the target variable. Feature numbers 1,5,6,9,10, and 11 all have p-values less than 0.05, which means they are statistically significant in predicting the target variable. These statistically significant features are cough detected, season autumn, season spring, Asia, Europe, and North America.

Finally, we took the results of all our models and created a simple hard voting function within our project to make sure that the reported prediction against the dependent variable agreed with most of our models. The logic for this voting function can be seen below:



**Conclusion & Future Work**

When the above models are compared with the results found in the paper “Screening COVID-19 by Cough Audio Data Using Attention-Based Convolutional Recurrent Neural Networks”, it is difficult to complete a thorough comparison across the performance of the models. This is mostly due to the original paper lacking in its reporting of performance metrics other than AUC. However, since the original tested between four models, our team believes that deploying an ensemble of three different classification algorithms will lead to results that are more generalized for a real-world application.

Additionally, the original paper focused solely on the use of the audio data as an input to the machine learning algorithms that were trained, while ignoring all the meta data available that added context to that audio data. The team believes that the data available here (region, sex, symptoms, etc.) is critical to fully rounding out the ensemble’s ability to accurately predict whether a person who is coughing is COVID-19 positive.

A brief view of the ensemble results with our models and Stanford's models results:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Learning Rate | Weight Decay | Dropout | Epochs | AUC | Recall | F1 Score | Accuracy |
| Ours  CNN |  |  |  |  |  |  |  |  |
| 1e-4 | - | 0.5 | 50 | 0.79 | 0.81 | 0.81 | 0.81 |
| LSTM | 1e-3 | - | - | 10  (per fold) | 0.89 | 0.82 | 0.81 | 0.81 |
| Logistic Regression | - | - | - | - | 0.62 | 0.68 | 0.62 | 0.62 |
| Ensemble | - | - | - | - | 0.80 | 0.86 | 0.80 | 0.80 |
| Stanford Results |  |  |  |  |  |  |  |  |
| Small CNN | 1e-4 | 1e-5 | 0.2 | 38 | 0.53 | - | - | - |
| Large CNN | 1e-4 | 1e-5 | 0.2 | 49 | 0.53 | - | - | - |
| CRNN | 1e-4 | 1e-5 | - | 32 | 0.80 | - | - | - |
| CRNN + Self-Attention | 1e-4 | 1e-5 | - | 39 | 0.80 | - | - | - |

For future work, to enhance the performance of the model, it would be beneficial to consider reducing the sampling frequency and removing inaudible sections from each audio clip. The team would also like to expand the ensemble and its child models to see how it performs with other respiratory infections using transfer learning. The team would also like to improve upon the ensemble architecture as it stands today by deploying more advanced algorithms as part of the ensemble. Specifically, replacing the simple Logistic Regression model being used against the metadata for the audio clips. It may be better to leverage a Random Forest, Gradient Boosted Machines, or another neural network (perhaps a Deep Neural Network) instead. This would most likely bring the performance levels of the currently weakest link in the team’s ensemble to much greater heights which would push the results of the ensemble to be better than the performance of the original model described above.

Lastly, all the work we completed was without the use of self-attention as a part of our algorithms. If we apply that to the neural networks within the ensemble, we believe we will be able to see results that are not only greater than the performance from the original paper we are comparing to, but also greater than the results found in (9). This is especially interesting because in the paper the team is comparing to, they did deploy the use of self-attention, but it was added onto models that were very poor performing in the first place (increasing their performance from .53 AUC to .80). Self-attention is an advanced machine learning technique that allows for a more adaptable approach to traditional fixed length encoding methods (such as LSTM). As such, if we were to leverage this as a part of our ensemble, we believe our results would improve as well.

**Contributions**

Drake Spence:

* Code:
  + Initial and final model development for the RNN (LSTM), Logistic Regression, and CNN.
  + Execution of the initial and final data preparation, cleansing, and aggregation with input.
* Report:
  + Review of the initial report and instrumental contributions toward the Methods and Experiment & Results sections.

Jared Santos:

* Code:
  + Review of code and model improvement as required.
  + Advised with methods by which to complete initial and final data preparation, cleansing, and aggregation.
* Report:
  + Initial report development and structure. Specifically focused on the Abstract, Introduction, Literature Review, Comparison Strategy, Data Set & Features, and Conclusion and Future work.

**References**

[1] G. Deshpande and B. W. Schuller, “Audio, speech, language, & signal processing for covid-19: A comprehensive overview,” arXiv preprint arXiv:2011.14445, 2020.

[2] L. Orlandic, T. Teijeiro, and D. Atienza, “The coughvid crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms,” Scientific Data, vol. 8, no. 1, pp. 1–10, 2021.

[3] G. Chaudhari, X. Jiang, A. Fakhry, A. Han, J. Xiao, S. Shen, and A. Khanzada, “Virufy: Global applicability of crowdsourced and clinical datasets for AI detection of covid-19 from cough,” arXiv preprint arXiv:2011.13320, 2020.

[4] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo, “Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data,” arXiv preprint arXiv:2006.05919, 2020.

[5] N. Sharma, P. Krishnan, R. Kumar, S. Ramoji, S. R. Chetupalli, P. K. Ghosh, S. Ganapathy, et al., “Coswara–a database of breathing, cough, and voice sounds for covid-19 diagnosis,” arXiv preprint arXiv:2005.10548, 2020.

[6] N. K. Chowdhury, M. A. Kabir, M. M. Rahman, and S. M. S. Islam, “Machine learning for detecting covid-19 from cough sounds: An ensemble-based mcdm method,” Computers in Biology and Medicine, vol. 145, p. 105405, 2022.

[7] M. Esposito, S. Rao, V. Narayanaswamy, and A. Spanias, “Covid-19 detection using audio spectral features and machine learning,” in 2021 55th Asilomar Conference on Signals, Systems, and Computers, pp. 1146–1150, IEEE, 2021.

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[9] S. Hamdi, M. Oussalah, A. Moussaoui, and M. Saidi, “Attention-based hybrid cnn-lstm and spectral data augmentation for covid-19 diagnosis from cough sound,” Journal of Intelligent Information Systems, pp. 1–23, 2022.

[10] S. R. Chetupalli, P. Krishnan, N. Sharma, A. Muguli, R. Kumar, V. Nanda, L. M. Pinto, P. K. Ghosh, and S. Ganapathy, “Multi-modal point-of-care diagnostics for covid-19 based on acoustics and symptoms,” arXiv preprint arXiv:2106.00639, 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

[12] Q. Zhou, J. Shan, W. Ding, C. Wang, S. Yuan, F. Sun, H. Li, and B. Fang, “Cough recognition based on mel-spectrogram and convolutional neural network,” Frontiers in Robotics and AI, p. 112, 2021.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in Advances in Neural Information Processing Systems 32, pp. 8024–8035, Curran Associates, Inc., 2019.

[14] D. Jiang, W. Li, M. Cao, W. Zou, and X. Li, “Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning,” arXiv preprint arXiv:2010.13991, 2020.

[15] Gu, C., Tantivasadakarn, N., & Zhang, S. Screening COVID-19 by Cough Audio Data Using Attention-Based Convolutional Recurrent Neural Networks.

Project Code Link:

<https://outlook.office.com/mail/group/shockers.wichita.edu/bsan735fileshare/files>

Raw Data Link:

<https://zenodo.org/record/7024894#.ZEiHQHbMKbg>