Scientific Research Publishing

Journal of Biosciences and Medicines, 2023, 11, 158-170
https://www.scirp.org/journal/jbm
ISSN Online: 2327-509X
ISSN Print: 2327-5081

HealthNet: Machine Learning for Cystic Fibrosis Characterization

Manasvi Pinnaka[1], Eric Cheek[2]

[1]Basis Independent Silicon Valley, San Jose, USA
[2]Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, USA
Email: maavistar@gmail.com, echeek@umich.edu

How to cite this paper: Pinnaka, M. and Cheek, E. (2023) HealthNet: Machine Learning for Cystic Fibrosis Characterization. *Journal of Biosciences and Medicines*, **11**, 158-170.
https://doi.org/10.4236/jbm.2023.119014

Received: August 14, 2023
Accepted: September 17, 2023
Published: September 20, 2023

Abstract

Cystic fibrosis patients often develop lung infections because of the presence of thick and sticky mucus that fills their airways. The presence of this thick mucus prevents the lungs from filtering out certain dominant bacterial types, making patients highly susceptible to infections that can range anywhere in severity from mild to life-threatening. These infections can cause great distress for patients as it becomes harder for patients to breathe and increases the chance of mortality by respiratory failure. It is important to be able to track the progression or regression of cystic fibrosis to determine the best course of treatment. Thus, this project focuses on the use of an AI model to examine the microbiology of cystic fibrosis patients and predict the condition or stage of lung function in the future, as a way to guide doctors with their treatment plan. Due to the limited amounts of publicly available patient data, we used all of the data in the training and testing of our machine learning algorithms initially and then tried a 50% training, 10% validation, and 40% testing split. Our results show that with relatively simple models (cubic polynomials), we can predict FEV1 from statistically significant bacteria sequences within 98% accuracy when training on sufficiently large samples.

Keywords

Cystic Fibrosis, Lung Microbiome, Machine Learning

1. Introduction

Cystic Fibrosis is a fatal genetic disorder that affects a number of organs, especially the lungs [1]. Mutations in the CFTR gene, known as the Cystic Fibrosis Transmembrane Conductance Regulator, disrupt the flow of chloride ions and water across cell membranes. Because the mucus becomes dehydrated with this

disruption, the cilia that normally move the mucus along the passageways can no longer perform this function on the thick and sticky mucus that builds up. Compared to healthy individuals, this abnormality of the mucus in CF patients provides germ with a more favorable environment to inhabit, thereby increasing the frequency of infections. As the disease progresses, a few bacterial types like Pseudomonas Aeruginosa tend to dominate in the lung as CF pathogens as a result of their ability to quickly adapt to the lung conditions of patients. CF patients often experience constant declines in FEV1 scores as they are faced with frequent lung infections and inflammation. In comparison to personal baselines, FEV1 scores for patients can help determine lung condition as the disease progresses.

For this project, we used several machine learning approaches to predict the FEV1 score from current bacteria levels. The first two machine learning models used were the linear regression and polynomial regression models. After finding the baseline performance using these linear and non-linear models, we further experimented with training a neural network for this task. The type of neural network architecture selected to predict cystic fibrosis progression is the classic Multi-Layer Perceptron (MLP), which consists of an input layer that takes in the information that needs to be processed, hidden layers that drive the computations, and an output layer that performs the final classification or prediction. After feeding forward the values, MLPs are also trained with a backpropagation algorithm to correct and reduce errors [2].

Machine learning has a rich history of successfully processing sequential data, especially time-series information. Connor and coauthors [3] showed reliable predictions of weather data using RNNs where other machine learning (ML) methods like linear models and neural networks failed. This is also not the first time ML has been applied to the medical field [4]. Chang and coauthors [5] used ML to identify biomarkers from the microbiomes of study participants to aid in disease prediction and treatment. In the past decade, various types of machine learning algorithms have been developed with remarkable progress to analyze data a large scale in a time effective and cost-effective manner for applications in numerous fields, from medicine to agriculture to cybersecurity [6].

In addition to using recurrent neural networks, we will also experiment with simpler models, such as linear regression, polynomial regression, kernelized regression methods, and multi-layer perceptron networks. These types of machine learning models can learn linear and non-linear relationships between the input features like the presence of certain bacteria and FEV1 score.

Multi-Layer Perceptrons are machine learning models that aim to learn important information from data through a process called training. The weights of the network are often called neurons. These neurons mimic neurons of the human brain and learn to become activated and transmit information to one another when activated. MLPs have shown promise in several machine learning applications ([7] [8]), and should be expressive enough to learn the relationship

between FEV1 score and the presence of certain bacteria. The inputs to our neural network will be the number of bacteria present in the lungs on a certain day, and the output will be the predicted FEV1 score at a later point in time. We will train our MLP using the ADAM optimizer to minimize the mean-squared error over all of our training samples to learn the optimal weight setting.

2. Data

The data used in this experiment is publicly available and acquired from Qitta.com [9]. The majority of preprocessing was done using usegalaxy.org. There were 51 patient sputum samples in the dataset that were collected and analyzed for certain bacteria using next-generation sequencing technology. Of these 51 samples, 6 had invalid readings of FEV1 score due to faulty measurements.

3. Experiments

3.1. Data—Preprocessing

For our initial experiment, we first wanted to show that it is possible for a machine learning model to predict FEV1 score based on the presence of certain bacteria in the patient's sputum sample. To ensure that we were only providing relevant information to our machine learning models, we used only bacteria which had a statistically significant correlation to FEV1 score as measured by the spearman correlation metric. Still, some patients had extremely low numbers of bacteria counts. We further pre-processed our data by removing patients who had fewer than 10 statistically significant bacteria present in their sputum sample which left a total of 19 patients' data. For all machine learning models, we did two separate tests by partitioning the data in separate ways.

1) Use all of the data for training and testing. This was done to see if the model was expressive enough to describe the relationship between bacteria counts and FEV1 score.

2) Separate the data into training, testing and validation sets. Here, 50% of the data was used for training, 10% was held out for validation to optimize our model hyper-parameters, and 40% was held out for testing.

In each case, we trained our machine learning models in the simple supervised learning approach, where the input to the models was the bacterial counts, and the predicted output was FEV1 score.

3.2. Same-Day Prediction

We trained a linear regression model, a polynomial regression model, and an MLP model to predict FEV1 score based on bacteria counts.

Let $w \in R^N$ denote a vector of the learned weights for the linear regression model, $y \in R^M$ denote a vector of FEV1 scores for each patient, and $X_{tr} \in R^{M \times N}$ denote our matrix of training data, and $X_{te} \in R^{M \times N}$ denote a matrix of our test data. To learn the weights for our linear regression model, we solved the linear-least squares problem. The closed-form pseudo-inverse solution to this

problem is:

$$w = X_{tr}^{+} y_{tr} \tag{1}$$

To make a prediction with these weights, we multiply them by the matrix of our training data.

$$y_{pred} = X_{tr} w \tag{2}$$

To make a prediction on test data, we can do the same thing with the test data.

$$y_{pred} = X^{te} w \tag{3}$$

The polynomial regression model fits a polynomial of degree n to the data, and can capture more complex relationships than our linear regression model. The procedure for learning the weights of our polynomial regression model is exactly similar to the process of learning the weights of our linear model, except we augment our data matrix X by repeating the existing columns raised to a certain degree according to the degree of the polynomial we decided to use for our model. For our experiments, we used a polynomial model of degree two. For implementation, we used the function from the sklearn.kernel ridge library with $\alpha = 0$ to create, fit, and test our polynomial model [10].

Lastly, we decided to train a shallow neural network model to learn the relationship between statistically significant bacteria and FEV1 score. Our neural network had 3 layers, with 10, 5, 1 neurons in each layer, respectively. The output layer was programmed to have only one output and return a single number, the predicted FEV1 score for that patient. We used a ReLu activation function for layers 1 - 8 and a linear activation function for the output layer. We used the Adam optimizer to update the weights of the network during training, with an initial learning rate α of 0.001, a value of β_1 of 0.9, and a value of β_2 of 0.999.

For implementation of the neural network, we used the Keras and tensorflow libraries ([11] [12]).

3.3. Predicting Future FEV1 Scores

To create our dataset of forecasted FEV1 scores, we modeled three scenarios for patients. The first is where the patient's condition gets worse, and the bacteria counts increase along with FEV1 score. By trial and error, we determined the probability of cells splitting (0.2) and the probability of cells dying (0.5) that would allow for a general moderate increase in the number of cells over the 10 days, starting with 100 cells. The second scenario is where the patient's condition gets better, and the bacteria counts decrease along with FEV1 score. We kept the probability of cells splitting the same and decided once again by trial and error what probability of cells dying (0.75) would allow for a general moderate decrease in the number of cells over 10 days. Finally, the third scenario is where the patient's condition stays the same, so the bacteria counts fluctuate around the same value. A graphical representation of this bacteria growth for a hypothetical scenario can be seen in **Figure 1**. To create this fluctuation, we generated a normal distribution, with a mean of the bacteria count for that day and a standard

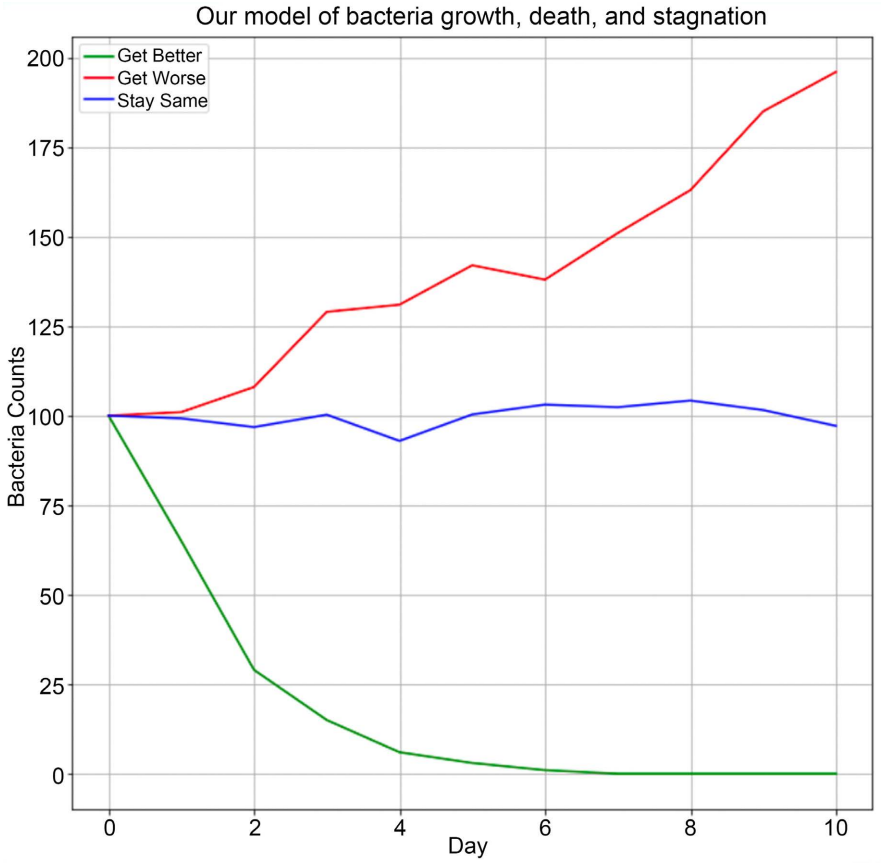

Figure 1. Model for bacteria growth, bacteria decay, and bacteria stagnation.

deviation of 0.05, and allowed for random selection of the bacteria count values based on the normal curve. The value of 0.05 was chosen through trial and error based on what would allow for the selection of bacteria count values with moderate differences between each day. The figure below shows the projected bacteria counts for each scenario over the 10 days.

We then created the projected FEV1 score values that would model each of the three scenarios. For the first scenario where the patient's condition worsens over the 10 days, FEV1 score was increased by 0.05% each day. For the second scenario where the patient's condition improves, FEV1 score was decreased by 0.05% each day. Finally, for the third scenario where the patient's condition stays the same, FEV1 score was kept the same as the day 0 value through day 10. Figure 2 shows the projected FEV1 scores for each scenario over the 10 days.

All this data was aggregated into a 3D array: 19 patients with 5 bacteria count values for each of the 10 days.

4. Results

4.1. Same-Day Prediction

4.1.1. All Data Used for Training

We first tested how well a linear model (LSTQ) can fit the data if we use all of it for training. As shown in Figure 3, this model performed poorly on the data,

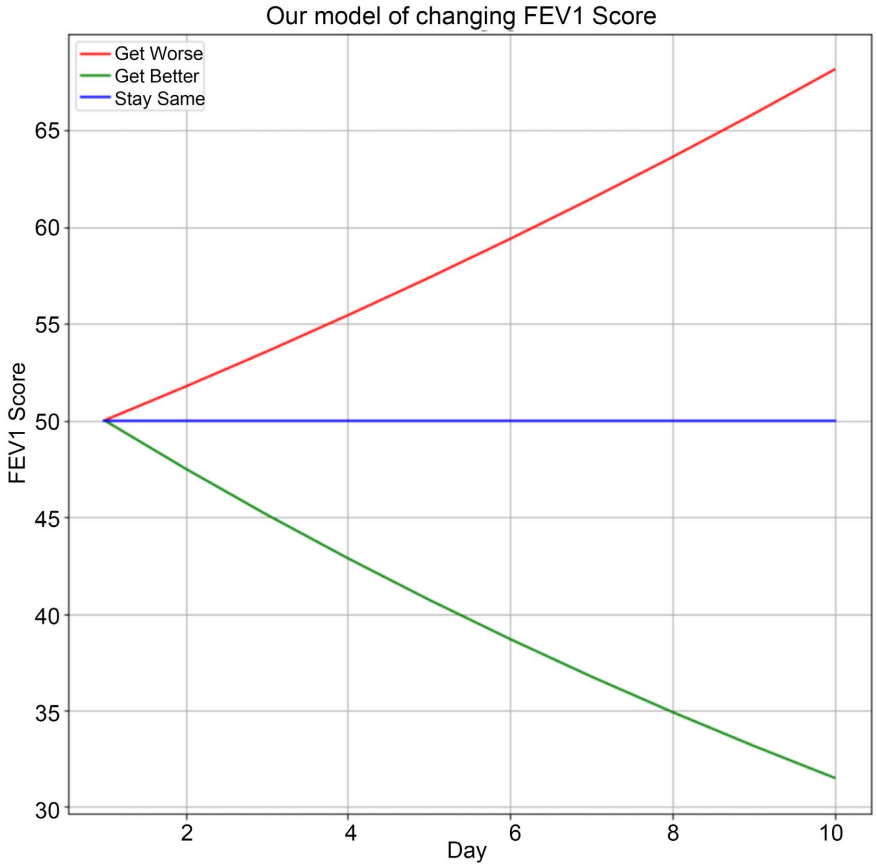

Figure 2. Model for forecasting FEV1 scores.

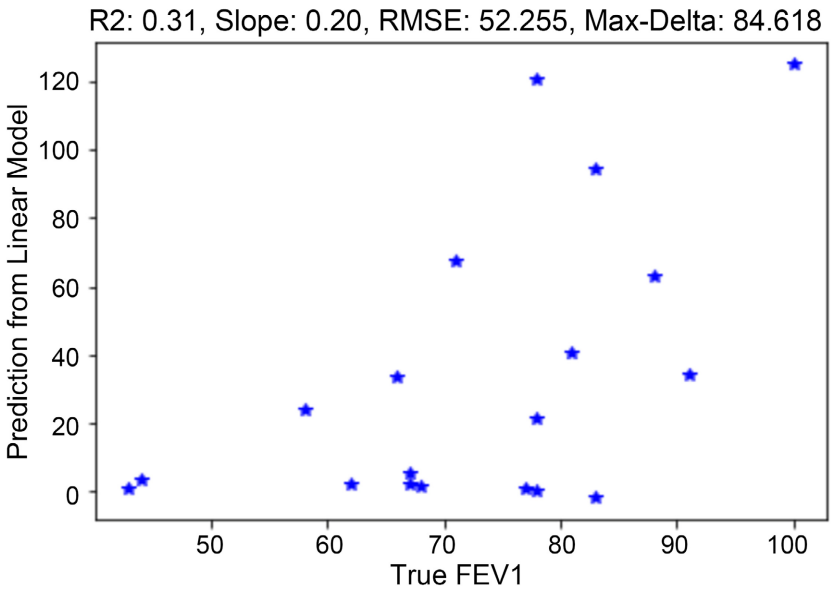

Figure 3. LSTQ Model for FEV1 score prediction.

suggesting that the relationship between FEV1 and the bacterial counts is non-linear.

Next, we tried a slightly more sophisticated model, our polynomial regression

model. Here, we set the degree of our model to 3. Figure 4 shows that it performed much better than a simple linear model. Lastly, we tried a neural network with nine layers and a learning rate of 0.01. Figure 5 shows how well the model fits the data.

As expected, when all of the data is used for training, more complex models are able to effectively express the relationship between the presence of certain bacteria and FEV1 score. The next question is how will these results change when separating data into training and testing sets?

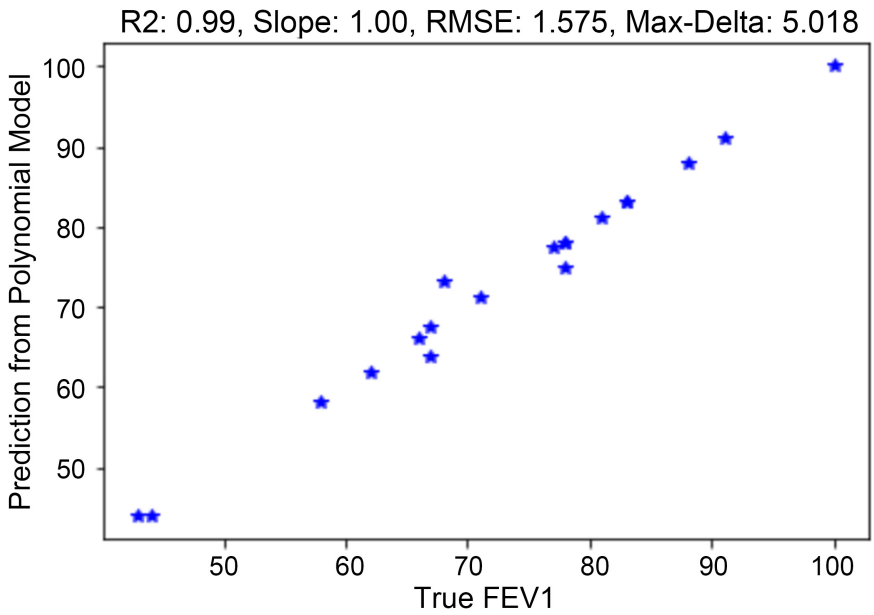

Figure 4. Polynomial regression model for FEV1 score prediction.

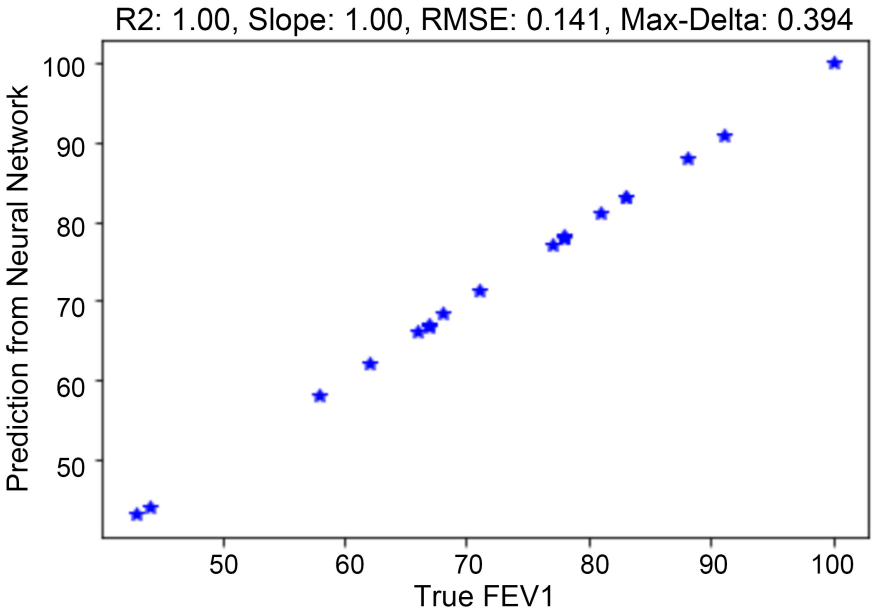

Figure 5. Shallow neural network for FEV1 score prediction.

4.1.2. Separate Training and Testing Sets

Figures 6-8 below show the results for training the same machine learning models with 50% of the data used for training, 10% used for validation (for neural network training) and 40% used for testing.

Unsurprisingly, the linear model performed equally as good on training data when using all of the points for training and testing (Figure 6). It also performed the best in terms of having the smallest maximum error over all test samples. This is most likely because the simplicity of the linear model and the concept of bias-variance trade-off. The linear model has low variance, but a higher bias.

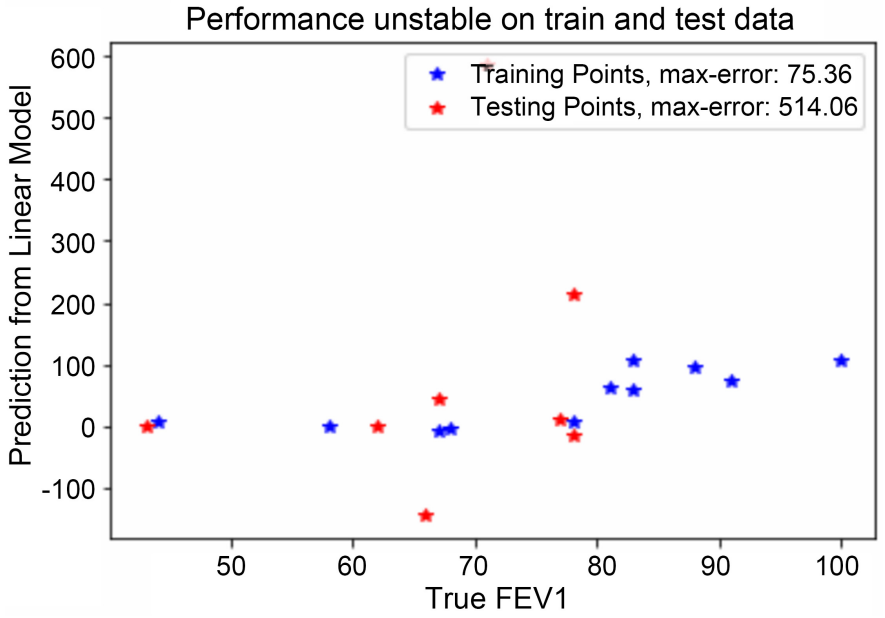

Figure 6. Linear model results on train and test data.

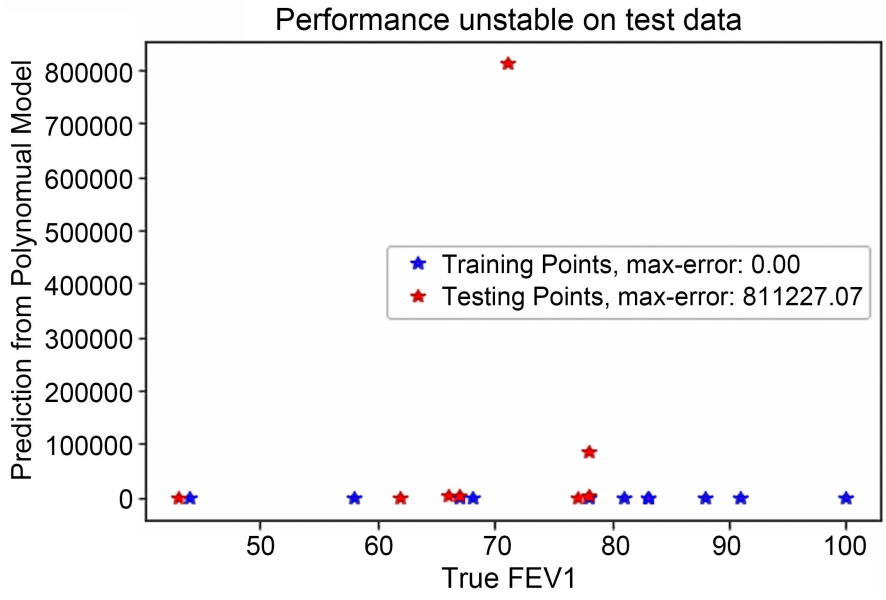

Figure 7. Polynomial model results on train and test data.

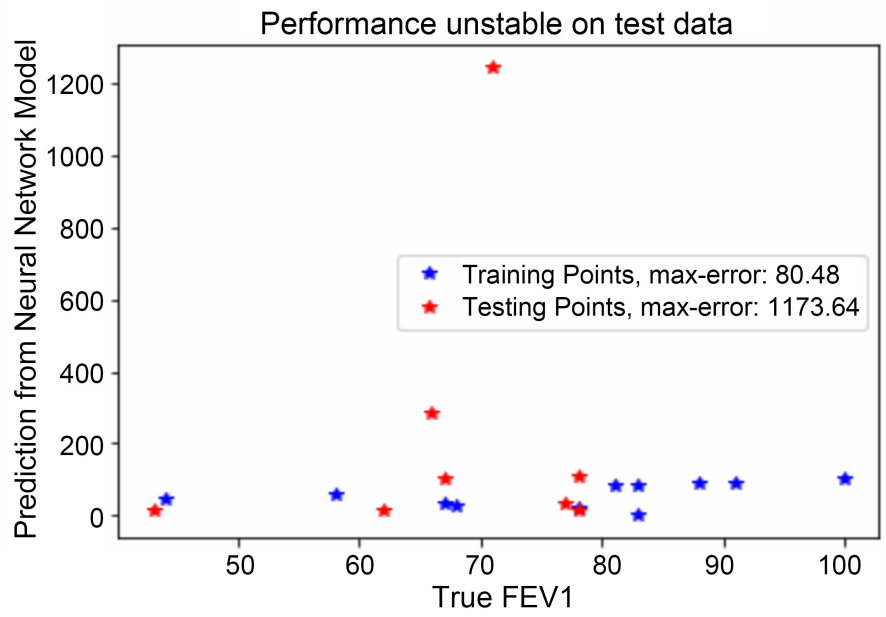

Figure 8. Shallow neural network results on train and test data.

The polynomial model performed the best on training data and the worst on test data (Figure 7). This is somewhat expected, as perfect performance on training data is usually indicative of overfitting. This was extremely likely to occur anyways based on the size of our dataset after preprocessing.

To our surprise, the neural network actually performed the 2nd best on test data and worst on the training data (Figure 8). Our explanation for this is that due to terminating training of the neural network model based on our one sample used for validation, we didn't overfit the test data. The one point with a large error in the training set ended up being the one point left out for validation.

4.2. Predicting Future FEV1 Scores

Separate Training and Testing Sets

Using our self-generated dataset where we modeled bacteria growth, death, and stagnation, we moved on to predicting future FEV1 scores. To do this, we used the same three machine learning models as in the Same-Day prediction experiments.

The input to the machine learning models was the number of bacteria counts for each of the statistically significant bacteria over k sequential days, and the output predicted by the model was the FEV1 score of the patient on the 10^{th} day. For these experiments, we selected $k = 9$ to give our machine learning models their best chance at predicting the value on the 10^{th} day.

Figures 9-11 below show the results for the linear, polynomial, and neural network models on both training and testing sets, consolidated for simplicity in this section.

The results were very similar to the previous case, except now the neural network had the best performance on test data (Figure 11), striking the best balance

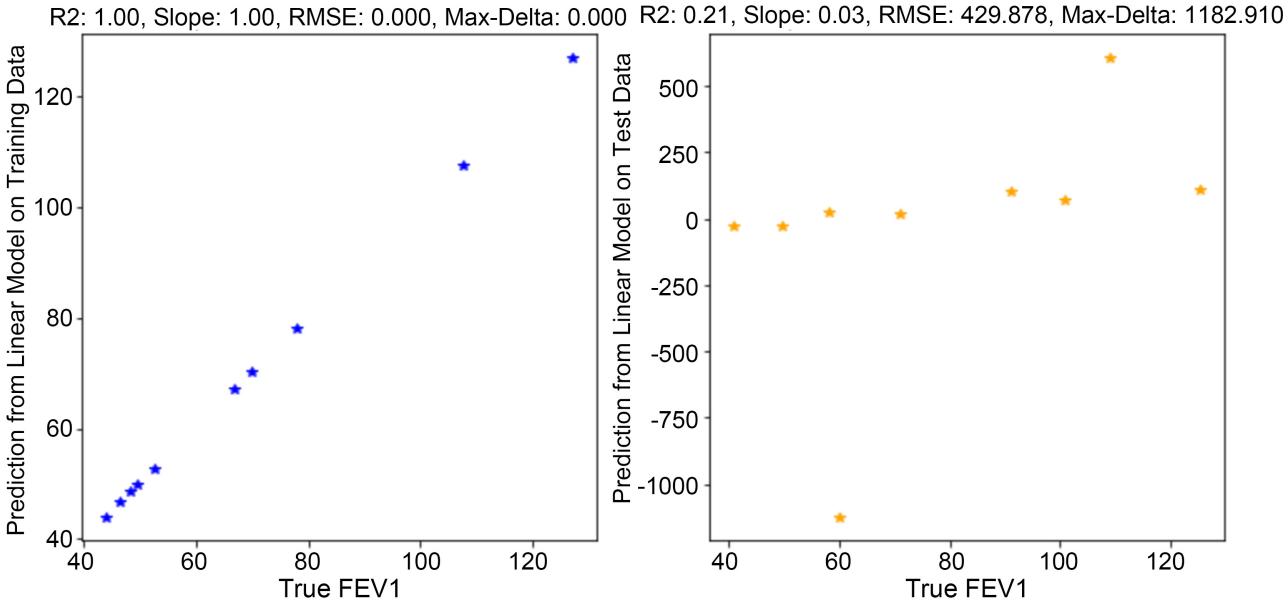

Figure 9. Simple linear model performance for future FEV1 score prediction.

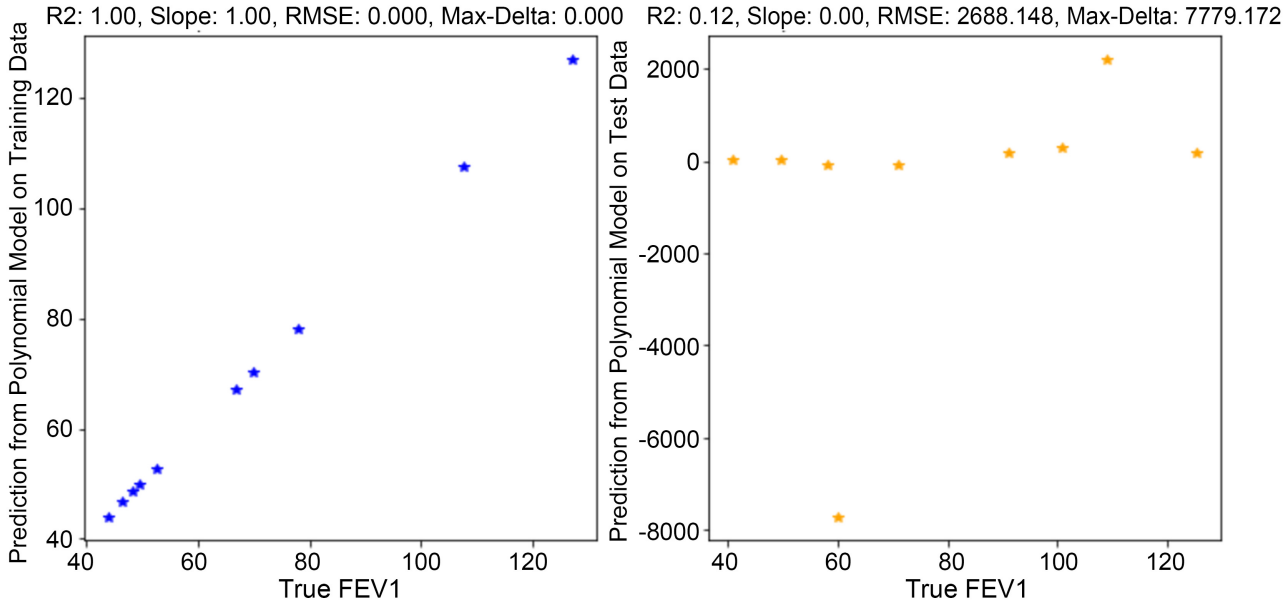

Figure 10. Polynomial model performance for future FEV1 score prediction.

between overfitting and underfitting given our very limited dataset. The performance on training data for each model also improved, even for the linear model (Figure 9). This suggests that the increase in features given to the network helps aid in remedying the lack of data problem.

Table 1 summarizes the results on both the training and testing datasets for future prediction. This gives a concise summary which shows that the neural network (NN) actually performed best on the test data, despite having the most parameters and performing the worst on the training data. In this table, let "tr" be an abbreviation for "training data" and "te" be an abbreviation for "test data".

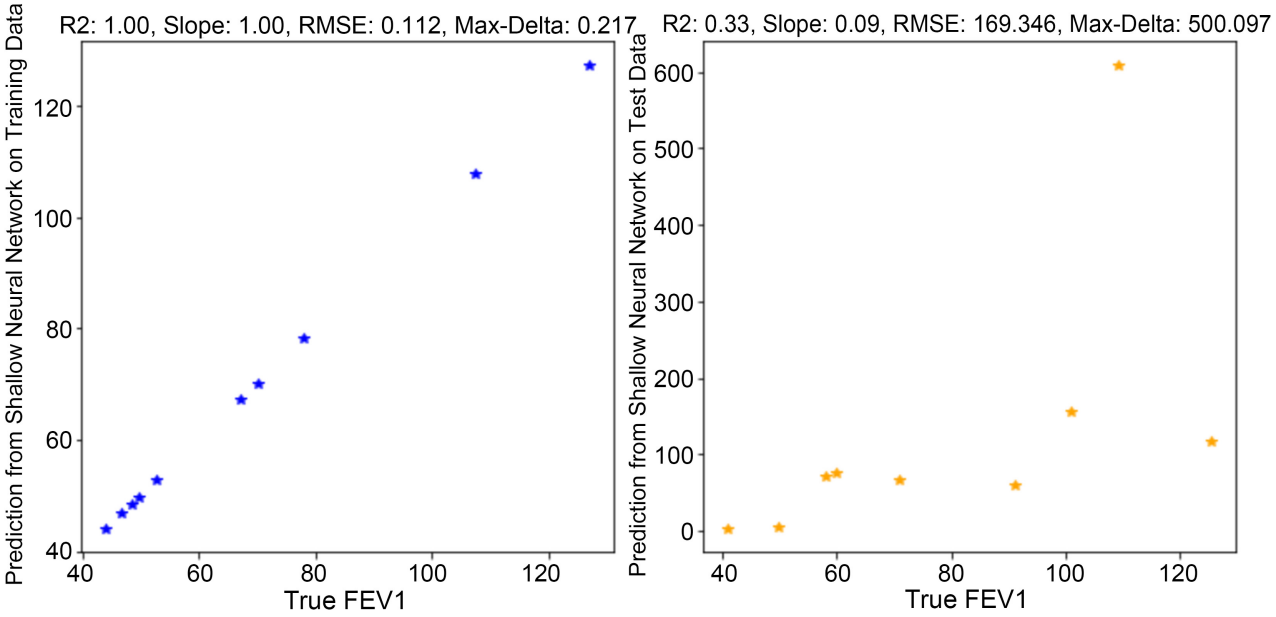

Figure 11. Shallow neural network performance for Future FEV1 score prediction.

Table 1. Models' performance for future FEV1 score prediction.

Metric	*Model Type*		
	Linear Model	**Polynomial Model**	**Shallow NN**
R^2 (tr)	1.0	1.0	1.0
R^2 (te)	0.21	0.12	0.33
Slope (tr)	1.0	1.0	1.0
Slope (te)	0.03	0.0	0.09
RMSE (tr)	0.0	0.0	0.112
RMSE (te)	429.89	2688.15	169.37
Max Error (tr)	0.0	0.0	0.22
Max Error (te)	1182.91	7799.17	500.01

4.3. Conclusion and Future Work

Using a publicly available dataset of DNA sequences from bacteria in the lung microbiomes of patients with cystic fibrosis, we investigated the existence of positive or negative correlations between the different microbial species in the lung and the extent of deterioration of lung function. After determining which bacteria were highly correlated with FEV1 score, we trained a linear model, polynomial model, and shallow neural network model for predicting the progression or regression of patients suffering from CF. Our results showed that, on training data, our deep learning model was highly accurate at predicting a patient's future FEV1 score based on their previous microbiome contents. However, with such a limited number of samples, all of the models performed poorly on test data. Factors such as data noise, training time, and data size all influence machine learning model accuracy, but it is evident that data size had the biggest

influence in this case as shown by the extent of improvement in the predictive abilities of all three models when the forecasted data points were added into the dataset [13]. Further work would have to be done with larger datasets to examine the relationship between bacteria counts and FEV1 score. A larger dataset would have also enabled more complex models made to process sequential data for future prediction more effectively (such as a recurrent neural network, for example). Additionally, one could use more features than the presence of certain bacteria, such as the specific families or genus the bacteria each belong to.

Acknowledgements

Special thanks to the individuals with cystic fibrosis and their families who gave their consent for their data to be included in the QIITA database.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Naehrig, S., Chao, C.M. and Naehrlich, L. (2017) Cystic Fibrosis. *Deutsches Ärzteblatt International*, **114**, 564-574. https://dx.doi.org/10.3238/arztebl.2017.0564

[2] Singh, J. and Banerjee, R. (2019) A Study on Single and Multi-Layer Perceptron Neural Network. 3*rd International Conference on Computing Methodologies and Communication*, Erode, 27-29 March 2019, 35-40. https://dx.doi.org/10.1109/ICCMC.2019.8819775

[3] Connor, J.T., Martin, R.D. and Atlas, L.E. (1994) Recurrent Neural Networks and Robust Time Series Prediction. *IEEE Transactions on Neural Networks*, **5**, 240-254. https://dx.doi.org/10.1109/72.279188

[4] Ahmad, M.A., Eckert, C. and Teredesai, A. (2018) Interpretable Machine Learning in Healthcare. *Proceedings of the* 9*th ACM international Conference on Bioinformatics*, *Computational Biology*, *and Health Informatics*, Washington DC, 29 August-1 September 2018, 559-560. https://doi.org/10.1145/3233547.3233667

[5] Chang, C.-H., Lin, C.-H. and Lane, H.-Y. (2021) Machine Learning and Novel Biomarkers for the Diagnosis of Alzheimer's Disease. *International Journal of Molecular Sciences*, **22**, Article 2761. https://dx.doi.org/10.3390/ijms22052761

[6] Sarker, I.H. (2021) Machine Learning Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, **2,** Article No. 160. https://dx.doi.org/10.1007/s42979-021-00592-x

[7] Popescu, M.-C., Balas, V.E., Perescu-Popescu, L. and Mastorakis, N. (2009) Multilayer Perceptron and Neural Networks. *WSEAS Transactions on Circuits and Systems*, **8**, 579-588.

[8] Bikku, T. (2020) Multi-Layered Deep Learning Perceptron Approach for Health Risk Prediction. *Journal of Big Data*, **7**, Article No. 50. https://dx.doi.org/10.1186/s40537-020-00316-7

[9] Gonzalez, A., *et al.* (2018) Qiita: Rapid, Web-Enabled Microbiome Meta-Analysis. *Nature Methods*, **15**, 796-798. https://doi.org/10.1038/s41592-018-0141-9

[10] Sklearn.kernel_ridge.KernelRidge.

https://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html

[11] Keras Documentation: The Model Class. https://keras.io/api/models/model/

[12] Tf.keras.model Tensorflow v2.11.0. https://tensorflow.org/api_docs/python/tf/keras/Model

[13] Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Naumann, F. and Harmouch, H. (2022) The Effects of Data Quality on Machine Learning Performance. arXiv: 2207.14529v4. https://arxiv.org/abs/2207.14529v4