Inferring Cosmological Parameters from Void Properties Using Fully Connected Neural Networks

Imanol Benitez, Vedang Bhelande, Sparsh Vashist, and Joseph Vera
Department of Physics and Astronomy, University of California, Los Angeles, CA 90095, USA
(Dated: March 17, 2024)

Abstract: Most of the volume of our universe is covered by cosmic voids, which are vast regions of relatively low densities. The properties of these voids are believed to offer important insights about the composition and evolution of our Universe. In this project, we show that machine-learning methods like Neural Networks can infer cosmological parameters from the properties of these voids. We use 2000 void catalogs from the GIGANTES dataset. We specifically focus on the decoding the relationship between properties of voids like ellipticity, density contrast, radius and cosmological parameters which include Ω_m, σ_8, and n_s. We train our fully-connected neural networks on the histogram of the void properties and are able to predict the values of Ω_m, n_s, and σ_8 with an R^2 value of 0.7582, 0.2889 and 0.7612 respectively as compared with the actual simulation parameters.

Keywords: Cosmology, Voids, Machine Learning, Neural Networks

MEMBER CONTRIBUTIONS

Imanol Benitez: Ran initial neural network models to obtain cosmological parameters. Implemented Optuna software to optimize hyperparameters and fine tune all neural network models. Ran best models to obtain final results and compare them to actual cosmological parameters. Wrote parts 5 and 6, and contributed to part 7 of report.
Vedang Bhelande: Responsible for void data acquisition and cosmological parameters acquisition from Globus, data exploration, and data cleaning. Prepared void data to be inputted into neural networks and train them. Researched different types of neural network architectures and contributed to code for their models. Wrote parts 2, 3, 4 and contributed to part 7 of report.
Sparsh Vashist: Established scope of research inquiry for the project, retrieved simulation data from 'Globus', helped with implementing all feature to all labels neural network architecture, evaluating our results and comparing them with our science objectives, wrote abstract, part 1, part 7, and conclusion of report.
Joseph Vera: Researched, coded and designed the architecture for all 3 neural network models used in this paper. Implemented custom loss functions converted input data into histograms to feed into neural networks. Obtained results for predicted cosmological parameters and compared them to actual values from test set made from data. Contributed to parts 5, 6, and 7 of report.

PART 1: SCIENTIFIC BACKGROUND AND THE BIG PICTURE

Acknowledgement: This project is based on the paper 'Machine-learning Cosmology from Void Properties' by Wang et. al (2023)[1].

The realm of cosmology aims to uncover the basic parameters describing the structure and the evolution of our universe. Cosmic voids, which are vast regions of relatively low density in the universe, offer a pristine laboratory for such investigations. These cosmic voids, accounting for most of the volume of the universe, are believed to provide hiding grounds for dark energy and are relatively unevolved parts of our universe. Hence, they are believed to be an excellent proxy for the primordial universe.

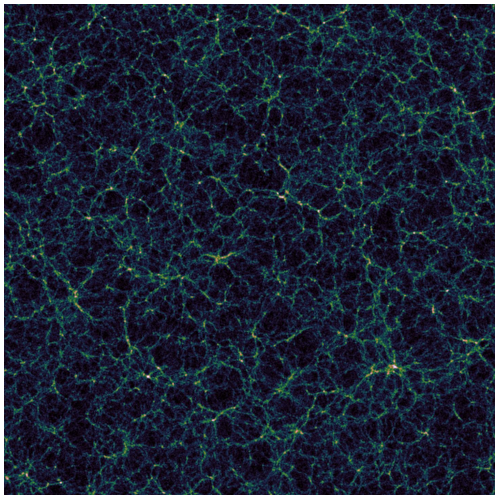

FIG. 1. **Visualising Cosmic Voids between large-scale structures of the Universe**

Earlier studies to investigate this includes the Hamaus et al. 2020[2] study which uses the void-galaxy cross-correlation function to constrain (Ω_m) to a value of 0.310 with a relative error of 5%. However, more exploration is required to explore the relationship between void properties like ellipticity, density contrast, radius and the cosmological parameters. Until recently, the number of discovered voids was relatively low. However, with increasing surveys and a vast amount of simulated void data

becoming available, specifically from the 'GIGANTES' data[3], studying these relationships has become more viable[1].

The large amount of simulated data, offers an opportunity to use Machine-Learning methods to study this relationship. This approach is advantageous for two reasons. Firstly, theoretical models to study this relationship involve robust modelling of void size function, density profile etc. which may not always be possible to compute or may be resource inefficient. Secondly, Machine learning methods have the potential to uncover complex, nonlinear relationships in the data that may be missed by theoretical models.

Hence, in this project, we use a fully-connected Neural Network architecture to study the relationship between void properties like ellipticity, density contrast, radius and the cosmological parameters, specifically the matter density parameter (Ω_m), the density perturbation amplitude (σ_8), and the density perturbation spectral index (n_s). Our data comprises of 2000 void catalogs from the 'GIGANTES' dataset[3] which includes void properties (features) and the cosmological parameters of the simulation runs (labels).

Our specific objective is to be able accurately predict the Cosmological Parameters of the the simulation runs given the void properties. A metric of success for this project is to be able to recreate the results from the Wang et al (2023) paper [1] within reasonable errors.Quantitatively, the efficacy of our ML model would be quantified through metrics including Root mean squared error (RMSE), mean absolute error (MAE), and the coefficient of determination (R^2) which would be able capture the accuracy and precision of the cosmological parameter predictions.

This project approach assumes that the properties of cosmic voids are robust indicators of underlying cosmological parameters and that the datasets used are sufficiently comprehensive and accurate to train machine learning models effectively[1]

PART 2: ACQUIRING THE GIGANTESVOIDS DATASET

To study how much information about cosmological parameters can be inferred from void properties, we needed to acquire a large amount of void data, enough to train a machine learning model. Going off of the paper we referenced (Wang et al[1]), we use the GigantesVoid dataset. This contains a catalog of over 1 billion voids made by running the VIDE void finder on QUIJOTE N-body simulations. However, to be consistent with the author's methods, we used a subset of the data in the folder 'latin_hypercube_params' at redshift z=0.0 which consisted of 2000 void catalogs providing us a total of around 4 million voids. The GigantesVoid dataset is open source and publicly available on Globus, a cyber infrastructure hosted by the University of Chicago developed to hold large amounts of data[3]. Hence, we did not have any authorization issues or legal obligations although we met with the authors to confirm this.

To have a workspace where all members of our group could access and use the data to train our machine learning models, we set up a shared folder in Google Drive. Each catalog of the data contains numerous files, primarily with different cuts to the data in terms of density and other features. For our study, we required unfiltered data and we accordingly sent batch requests to Globus to extract the centers and shapes files which contained all the necessary information needed to describe the voids. The final result was 1 file containing 2000 catalogs (folders) with 2 files each, one with shapes information and one with centers/position information.

To make the data easy to input for further inspection and training models, the void properties were compiled into a csv (comma separated value) file. However, this data alone is around 1 GB which would not be feasible to upload into the final project folder. Instead, we upload the features used to train our model instead (around 155 MB in size) which are explained in more detail in the following section. There was no sensitive information present. The data is simulation data which tells us about the shape of voids, as well as their positions, and other astrophysical properties. Each catalog containing a differing number of voids is simulated using a combination of 5 cosmological parameters shown as follows[1]:

1. $\Omega_m \in [0.1, 0.5]$
2. $\Omega_b \in [0.03, 0.07]$
3. $h \in [0.5, 0.9]$
4. $n_s \in [0.8, 1.2]$
5. $\sigma_8 \in [0.6, 1.0]$

Both the shapes and centers files contain 14 features each with a differing number of voids. We received access to a corresponding file called 'latin_hypercube_params.txt' which contained the cosmological parameters from which each catalog of voids were simulated. This file has dimensions 2000 x 5 since there are 2000 catalogs along with the 5 cosmological parameters used to simulate the voids. This is used to check the accuracy of our machine learning models and we split the catalogs along with their corresponding set of cosmological parameters into a training set and test set.

PART 3: EXAMINING THE DATA

According to our scientific objective, we want to find how much information we can gain about the aforementioned cosmological parameters from 3 void properties:

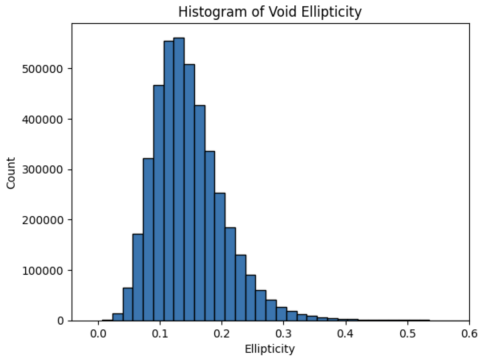

FIG. 2. Histogram showing void ellipticity for voids in all catalogs.

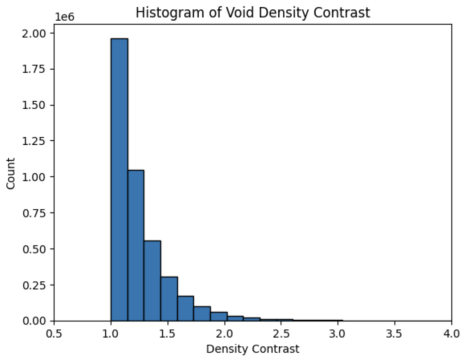

FIG. 3. Histogram showing density contrast for voids in all catalogs.

void ellipticity, void radius, and density contrast. The ellipticity is obtained from the shapes file while the radius and density contrast are obtained from the centers file. In both files, all 14 features each are of the data type float. However, we limit our exploration of data to the relevant void properties mentioned earlier. Searching for missing values in the dataset, we only found missing values only in the 1898th catalog. There was no information about any void properties, including the ones we are interested in. Thus, about 0.05% of our data is missing. The next step was to identify outliers in the relevant void features. For the void ellipticity, we plotted a histogram shown in Figure 2 and saw that all the values were between 0 and 1 meaning they were all physically meaningful and no outliers were present. We also plotted a histogram for the density contrast shown in Figure 3 and saw all the values were between 1.0 and 3.0 which is consistent with the range of density contrast values set by the authors. Hence, there were no outliers here either. For the void radius, we saw the histogram obeyed a right skewed Gaussian distribution as shown in Figure 4. We checked if any values were greater than $\mu + 3\sigma$ or less than $\mu - 3\sigma$ where μ = mean and σ = standard deviation. No outliers were identified, and the histograms are shown below. All data analysis was done in Python.

This was a supervised learning task and the target attributes were 5 cosmological parameters: $\Omega_m, \Omega_b, h, n_s, \sigma_8$. To reduce redundancy in the features used to train our machine learning models, we plotted a correlation matrix and found no significant correlations between any features as shown in Figure 5. This allows use to each feature independently or use a combination of them to train our machine learning models.

Given that there is no concrete physical model that relates our void properties of interest to the cosmological parameters we want to find, this problem cannot be solved manually and hence this motivates our choice to use neural networks (discussed further in Part 5). Since we are only interested in how these 3 void properties can give us information about cosmological properties,

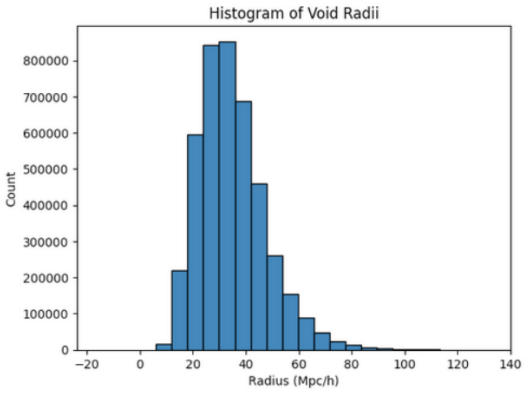

FIG. 4. Histogram showing void radius for voids in all catalogs.

we did not add extra data or features and did not apply any transformations to the data.

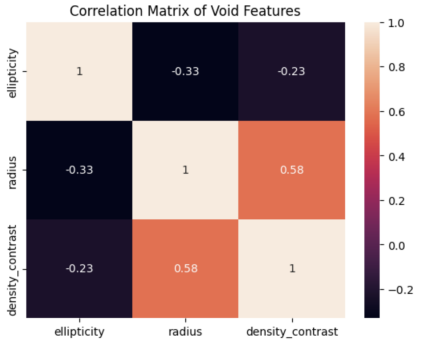

FIG. 5. Heat map showing the correlation matrix between void ellipticity, void radius, and density contrast. There are no significant correlations between any of these features.

	index	ellipticity	radius	density_contrast
0	0.0	0.198568	129.17	3.487631
1	0.0	0.162464	172.36	3.154508
2	0.0	0.118820	83.54	2.724835
3	0.0	0.120590	75.79	2.692753
4	0.0	0.101951	64.01	2.678991

FIG. 6. Sample of void features inputted to neural network. The index tells us which catalog each void belongs to.

	omega_m	omega_b	h	n_s	sigma_8
0	0.1755	0.06681	0.7737	0.8849	0.6641
1	0.2139	0.05557	0.8599	0.9785	0.8619
2	0.1867	0.04503	0.6189	0.8307	0.7187
3	0.3271	0.06875	0.6313	0.8135	0.8939
4	0.1433	0.06347	0.6127	1.1501	0.7699

FIG. 7. Sample of cosmological parameters used to simulate each catalog of voids.

PART 4: PREPARING THE DATA

As mentioned in the previous section, no data transformations were applied. From our exploration, no outliers were identified and hence the data did not have to be filtered in that regard. However, one cut was made and that was to the catalog which possessed no information about the void properties used to train our model. Since this was only 0.05% of our total data, dropping it will have negligible effects on the performance of our machine learning models.

The final dataset was trimmed to include only the void ellipticity, void radius, and density contrast. An additional column was added labelled 'index'. This corresponded to the catalog each void belonged to and proved useful in training our model in terms of classifying the data. No other transformations were applied to the raw data and no feature scaling was performed. Snapshots of the features and target attributes are shown in Figure 6 and Figure 7 as a pandas dataframe.

In order to prepare the data to be an input for the model used below we created histograms of each void in the catalog, which were separated into 18 bins of equal range. The values of the input data are:

1. ellipticity: $\epsilon \in [0.01.0]$
2. radius: $r \in [0.0, 650.0]$
3. density contrast: $\Delta \in [1.0, 3.0]$

PART 5: OVERVIEW OF NEURAL NETWORKS AND TRAINING THEM

In order to fully capture the complex relation of the void parameters to the cosmological parameters, we require a machine learning model capable of analyzing nonlinear data. Therefore, in this project, we have decided to proceed with a neural network model.

Neural Network Model	Model 1	Model 2	Model 3
Input Layer Shape	(18,)	(54,)	(54,)
Output Shape	1	1	5
Loss Function	MSLE	MSLE	Custom Loss
Learning Rate	0.001	4.3×10^{-5}	0.0002
Weight Decay	0.003	0.0001	0.0005

TABLE I. Differences between the three models used in this project MSLE refers to mean squared logarithmic error, the custom loss function is a combination of mean squared error and the logarithmic error.

In this project we have decided to create and train a fully connected neural network. The input to this neural network is a density histogram of the prepared data distributed over 18 bins. The neural network architecture comprises an input layer that accepts the histogram, followed by four dense layers, each with 100 nodes and the ReLU activation function. There's also a dropout layer with a 0.2 dropout rate, and an output layer. For the optimization process, AdamW is employed instead of the standard Adam. AdamW adds weight decay, which aids in preventing overfitting, complementing the regular dropout layer's functionality. This adjustment in the optimization strategy helps to ensure a more robust model against overfitting.

There are three different neural network models that we have trained. Each model diverges in its input and output configurations, as well as in certain aspects of its loss function and optimizer settings. Table 1 highlights the key differences between the models.

These models were trained using an 80/20 split for the training, test and validation data, additionally the models were trained using batches of 128 instances of data, for optimized selections of 1000, 500, and 1000 epochs for the three models respectively.

The first model takes as input a single histogram per simulation, specifically, void ellipticity, in order to predict a single cosmological parameter as output, specifically, the universe matter density Ω_m. The second model intakes a concatenated array of the ellipticity ϵ, the radius r and the density contrast Δ. And gives an output of a single target, in this case Ω_m. The third model has a similar intake as the second model, however it outputs predictions to all cosmological parameters Ω_m, Ω_b, h, n_s and σ_8.

Neural Network	Model 1	Model 2	Model 3
Trial	77	41	22
Learning Rate	0.005	4.3×10^{-5}	0.0002
Weight Decay	0.002	0.0001	0.0005

TABLE II. The optimized hyperparameters for the three models and the trial at which they were obtained from 100 trials.

PART 6: OPTIMIZING HYPERPARAMETERS AND FINE TUNING MODELS

The neural network can be improved by changing various hyperparameters, such as the learning rate, batch size, epochs, optimizer, loss functions and more. The authors of the original paper have established values for the batch size and the epochs, therefore we have decided to set them equal to their parameters in order to achieve similar results. The optimizer was selected based on its efficiency, easy tuning and adaptability. The Loss function was set based on testing of the models and directly analyzing the results, additionally the custom loss function used for the models is based on a similar loss function used by the authors of the original paper.

The two methods for optimizing the hyper parameters were grid search method and Optuna. The grid search method searches for the best performing hyperparameters in a multidimensional array containing all the possible combinations of a set of hyperparameters. The optuna method employs efficient search algorithms, such as Bayesian optimization, Tree-structured Parzen Estimator (TPE), and Pruned Random Search, to identify the most promising hyperparameter configurations for a given machine learning model.

We have ultimately opted for using Optuna instead of grid search as it proved most efficient in searching for the best hyperparameters for our models. Table II presents the optimized hyperparameters for each of the models using Optuna.

PART 7: RESULTS

To assess the performance of our neural networks, we employed various metrics including mean absolute error, mean squared error, root mean squared error, and the R^2 score. These metrics provide insights into the accuracy of our models in comparison with the results reported by the authors of the reference study. The following tables display key metrics for our models.

Table III shows the values for Ω_m by training Model 1 with ϵ histograms, the accuracy of these scores is compared to the results that the authors of the paper measured, comparing to their results we have very similar values for our metrics in Model 1. Table IV displays the results for Model 2, which trains on the (ϵ, r, Δ) histograms to the Ω_m target, similar to the previous table our results these values match the values that the authors predicted in their models. For table V we decided to measure the metrics for training Model 2 with respect to a different target value, σ_8 in this case, the results for these metrics display very similar values to table IV, except for the case of R^2 which is slightly higher than the previous table, this indicates a better fit for the model, which could be due to how the features relate to target parameters. Lastly table V displays the metric results for Model 3, which trains the neural network over all features and all targets. The results similar to the previous results are very similar to those given in the paper.

Ellipticity to Ω_m	
Metric	Value
MAE	0.0497
MSE	0.0039
RMSE	0.0626
R^2	0.6935

TABLE III. Table depicting metrics when only void ellipticity is used to find Ω_m.

All Features to Ω_m	
Metric	Value
MAE	0.0439
MSE	0.0031
RMSE	0.0556
R^2	0.7583

TABLE IV. Table depicting metrics when all 3 void properties are used to find Ω_m.

Table VI displays the values for model 3, which maps all the features (ϵ, r, Δ) to all of the targets (Ω_m, Ω_b, h, n_s, σ_8), for this Model we can observe that the metric values for some of the targets have low scores, specially for the values of Ω_m and n_s, which are the same values that the authors of the original paper have struggled to predict.

All Features to σ_8	
Metric	Value
MAE	0.0431
MSE	0.0031
RMSE	0.0556
R^2	0.7714

TABLE V. Table depicting metrics when all 3 void properties are used to find σ_8.

The prediction capabilities of our models were tested and plotted against the actual values, this can be seen in Figure 8 to figure 12. Figure 8 plots the predicted values of Model 1 against the actual values, for this Model we measured a value of $R^2 = 0.6852$. Figure 9 plots the predictions for Model 2 mapping to Ω_m, the $R^2 =$

Looking ahead, there are several areas for improve-

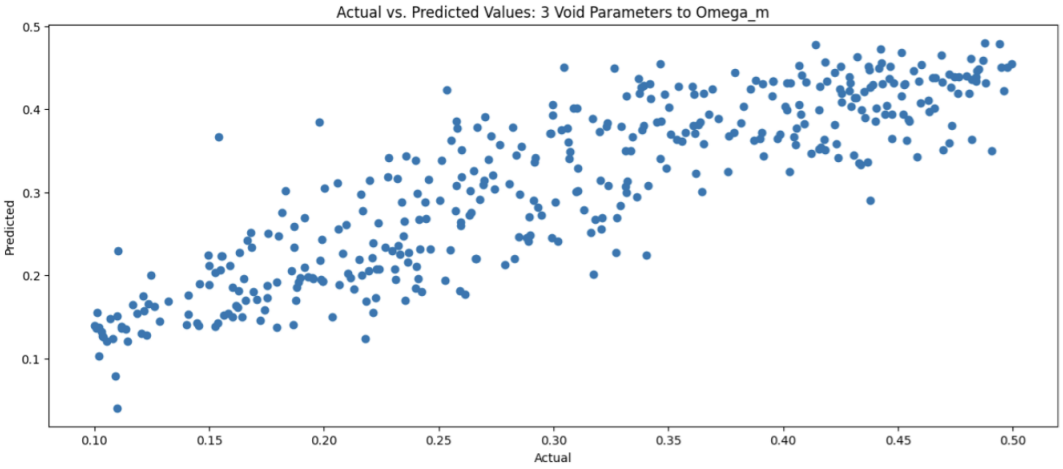

FIG. 9. Predicted v/s actual values for Ω_m when all 3 void features are used to train neural network.

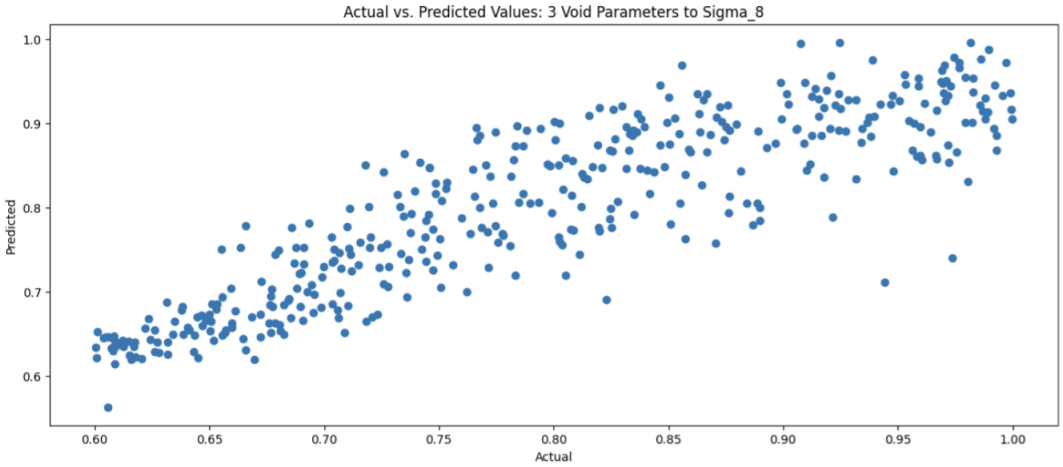

FIG. 10. Predicted v/s actual values for σ_8 when all 3 void features are used to train neural network.

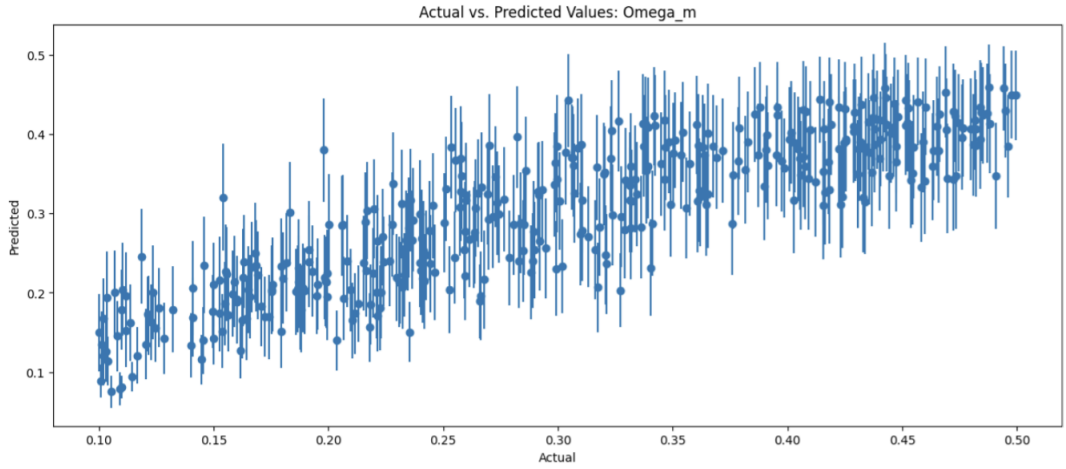

FIG. 11. Predicted v/s actual values for Ω_m when all 3 void features are used to train neural network to find all cosmological parameters.

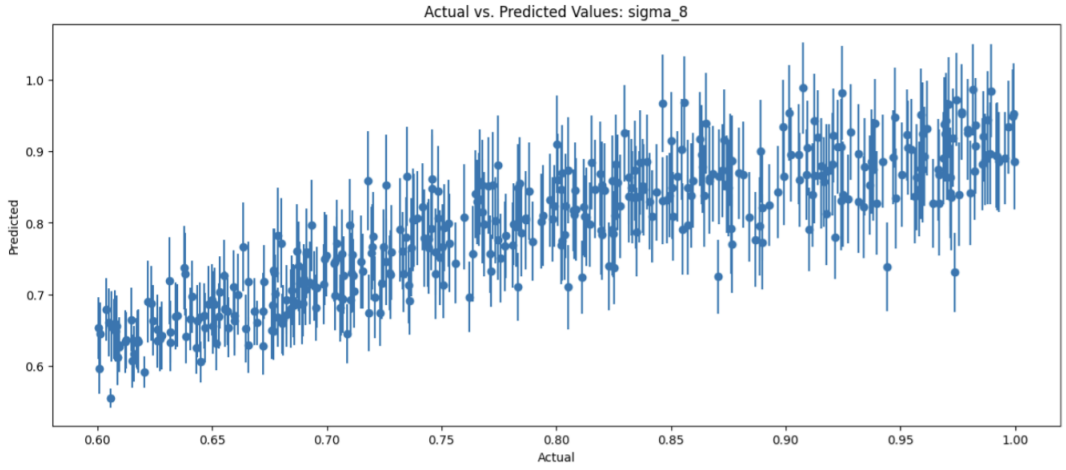

FIG. 12. Predicted v/s actual values for σ_8 when all 3 void features are used to train neural network to find all cosmological parameters.

All Features to	Ω_m	Ω_b	h	n_s	σ_8
Metric	Value	Value	Value	Value	Value
MAE	0.0457	0.0097	0.0958	0.0829	0.045
MSE	0.0031	0.0001	0.0127	0.0099	0.0034
RMSE	0.056	0.0113	0.1131	0.0999	0.0584
R^2	0.7542	0.0137	0.0318	0.2685	0.7489

TABLE VI. Table depicting metrics when all 3 void properties are used to find all cosmological parameters: Ω_m, Ω_b, h, n_s, σ_8.

ment and expansion. Further hypertuning of the models and exploration of additional architectural complexities could yield even more precise predictions. Incorporating a broader set of void properties and exploring their correlations with cosmological parameters could deepen our understanding. However, it's crucial to acknowledge the assumptions and limitations inherent in our models. The accuracy of our predictions relies on the assumption that the GIGANTES dataset is representative of true voids and captures the wide diversity of cosmic voids in the universe. Applying these models to diverse datasets from different sky surveys covering wider parts of the sky and varying observational conditions could further improve our results and understanding.

CONCLUSION

We were able to successfully implement an end-to-end Neural Network model using TensorFlow. Our findings are encouraging, aligning within reasonable error bounds as compared to Wang. et. al (2023). Through hypertuning, we significantly enhanced our model's performance, as evidenced by improvements in the R^2 and RMSE metrics. Crucially, this process has allowed us to establish a reliable mapping from the parameters of cosmic voids to key cosmological parameters, specifically Ω_m and σ_8. This achievement marks a promising step forward in our ability to decode the Universe's fundamental characteristics through machine learning techniques.

Our group would like to thank Professor Tuan Do along with TAs William Lake and Luke Finnerty for their guidance throughout this course and helping us set a project timeline with feasible goals. We would also like to thank the authors of the paper we referenced, Dr. Bonny Wang, Dr. Francisco Villaescusa-Navarro, and Dr. Alice Pisani for taking time out of their schedules and assisting us with understanding the scientific context of their paper and helping us acquire the void data from Globus. Their insights on how they trained their neural networks were invaluable and allowed us to think more deeply about how we could optimize our models and train them to infer cosmological parameters from void properties.

APPENDIX

Figures 13-15 shown below are for the predicted v/s actual values Ω_b, h, n_s. For these cosmological parameters, unsuccessful inferences were made meaning that our neural networks were unable to accuratly predict these parameters.

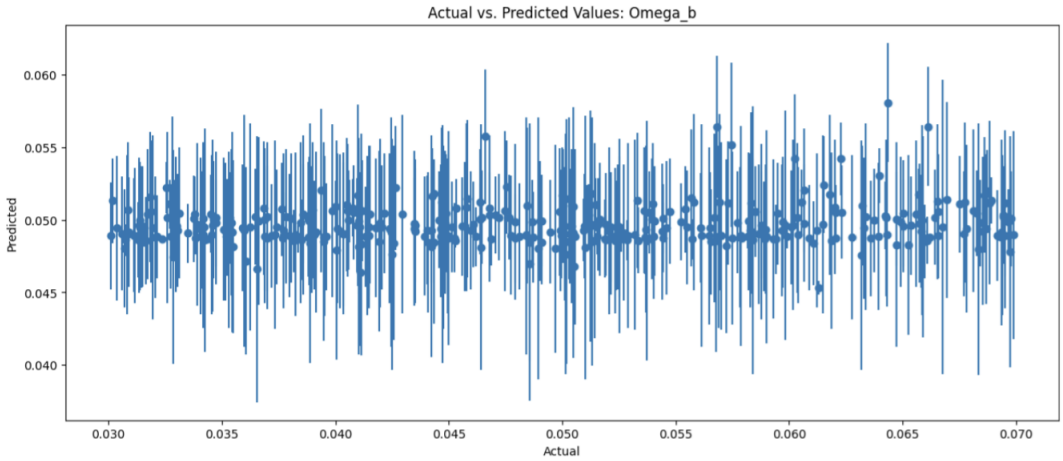

FIG. 13. Predicted v/s actual values for Ω_b when all 3 void features are used to train neural network to find all cosmological parameters.

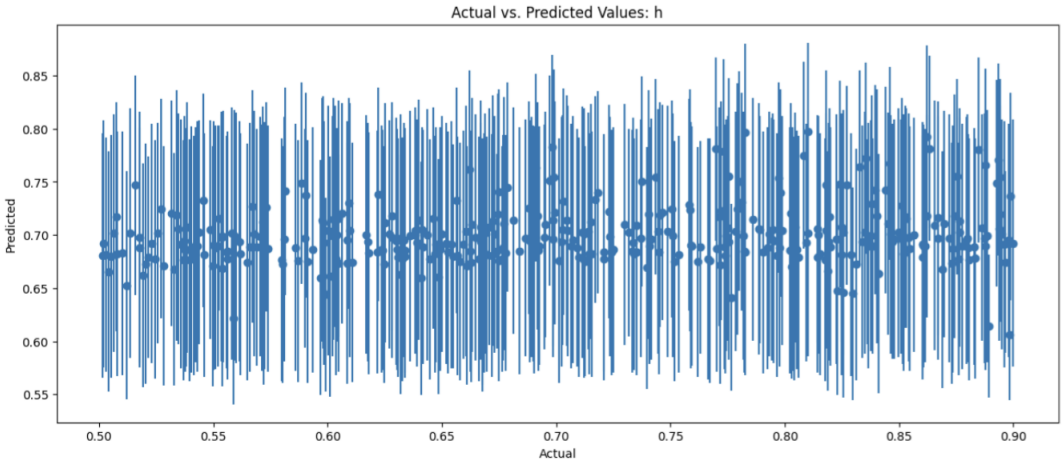

FIG. 14. Predicted v/s actual values for h when all 3 void features are used to train neural network to find all cosmological parameters.

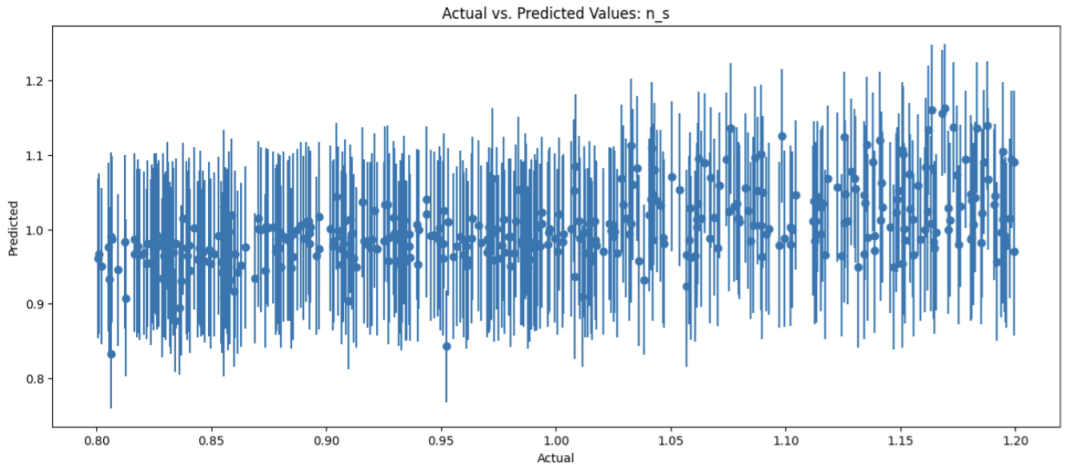

FIG. 15. Predicted v/s actual values for n_s when all 3 void features are used to train neural network to find all cosmological parameters.

REFERENCES

1. B. Y. Wang , A. Pisani, F. Villaescusa-Navarro, and B. D. Wandelt, Machine-learning cosmology from void properties, The Astrophysical Journal 955, 131 (2023).

2. N. Hamaus, A. Pisani, J.-A. Choi, G. Lavaux, B. D. Wandelt, and J. Weller, Precision cosmology with voids in the final boss data, Journal of Cosmology and Astroparticle Physics 2020 (12), 023–023.

3. C. D. Kreisch, A. Pisani, F. Villaescusa-Navarro, D. N. Spergel, B. D. Wandelt, N. Hamaus, and A. E. Bayer, The gigantes data set: Precision cosmology from voids in the machine-learning era, The Astrophysical Journal 935, 9100 (2022).