

Estimation of redshift and associated uncertainty of Fermi/LAT extragalactic sources with Deep Learning

Sarvesh Gharat¹,^{*} Abhimanyu Borthakur² and Gopal Bhatta³

¹Centre for Machine Intelligence and Data Science, Indian Institute of Technology Bombay, 400076 Mumbai, India

²Department of Electronics and Communication Engineering, Manipal Institute of Technology, 576104 Karnataka, India

³Janusz Gil Institute of Astronomy, University of Zielona Góra, ul. Szafrana 2, PL-65-516 Zielona Góra, Poland

Accepted 2023 November 17. Received 2023 November 6; in original form 2023 August 18

ABSTRACT

With the advancement of technology, machine learning-based analytical methods have pervaded nearly every discipline in modern studies. Particularly, a number of methods have been employed to estimate the redshift of gamma-ray loud active galactic nuclei (AGN), which are a class of supermassive black hole systems known for their intense multi-wavelength emissions and violent variability. Determining the redshifts of AGNs is essential for understanding their distances, which, in turn, sheds light on our current understanding of the structure of the nearby universe. However, the task involves a number of challenges, such as the need for meticulous follow-up observations across multiple wavelengths and astronomical facilities. In this study, we employ a simple yet effective deep learning model with a single hidden layer having 64 neurons and a dropout of 0.25 in the hidden layer on a sample of AGNs with known redshifts from the latest AGN catalogue, 4LAC-DR3, obtained from Fermi-LAT. We utilized their spectral, spatial, and temporal properties to robustly predict the redshifts of AGNs as well quantify their associated uncertainties by modifying the model using two different variational inference methods. We achieve a correlation coefficient of 0.784 on the test set from the frequentist model and 0.777 and 0.778 from both the variants of variational inference, and, when used to make predictions on the samples with unknown redshifts, we achieve mean predictions of 0.421, 0.415, and 0.393, with standard deviations of 0.258, 0.246, and 0.207 from the models, respectively.

Key words: methods: statistical – galaxies: active – galaxies: distances and redshifts – gamma-rays: galaxies – gamma-rays: general.

1 INTRODUCTION

Redshift, denoted as ‘ z ’, is a measure of the displacement of spectral lines towards longer wavelengths in the electromagnetic spectrum. This phenomenon arises due to the expansion of the Universe, stretching the wavelength of light emitted by distant celestial objects. Redshift estimation plays a fundamental role in understanding the properties of these objects, including their distance, cosmological evolution, and the nature of the Universe itself. In the realm of astrophysics, redshift estimation traditionally relies on spectroscopic measurements, where the light emitted by celestial objects is dispersed into its constituent wavelengths, revealing characteristic absorption or emission features. However, spectroscopic observations are often constrained by limited observational time, expensive resources, and the technical limitations of spectrographs. Consequently, obtaining spectroscopic redshift measurements for a large number of objects, as required by comprehensive surveys, becomes challenging and impractical.

The Fermi Gamma-ray Space Telescope (Fermi-LAT) has revolutionized the study of high-energy gamma-ray sources and contributed significantly to our understanding of the Universe. The

Fermi-LAT observatory observes celestial objects in gamma-ray wavelengths. However, efficiently extracting redshift information solely from gamma-ray observations poses a challenge as these observations are devoid of any spectral line, besides that of the 511 keV feature Skinner (2010). Therefore, the sole viable approach to gauging the distance involves linking the gamma-ray emitter with a recognized source that exhibits absorption or emission lines in other wavelengths, thereby enabling the calculation of redshift. The majority of discrete sources detected by Fermi/LAT are blazars, which consist of flat-spectrum radio quasars (FSRQs) exhibiting distinct optical emission lines over a broad-band continuum, and BL Lacs (BLLs), characterized by weak or absent emission line signatures (see Bhatta & Dhital 2020, and references therein). This indicates that while it may be relatively easier to estimate the redshifts of FSRQs, the redshift evaluation for BL Lacs is a complex and often computationally expensive task as it necessitates extensive optical spectroscopic observations along with comprehensive multi-wavelength observations involving diverse astronomical facilities.

To address these challenges, astronomers have turned to ML and DL techniques Dainotti et al. (2021); Narendra et al. (2022); Coronado-Blázquez (2023), which have demonstrated remarkable success. The study done by Dainotti et al. (2021) is one of the initial works in estimating the redshift of γ -Ray loud AGNs. The authors make use of an ensemble-based approach that combines

* E-mail: sarveshgharat19@gmail.com

Table 1. Classwise distribution of the data considered for this study.

	Known redshift	Unknown redshift	Total
BLL	738	433	1171
BCU	59	459	518
FSRQ	390	0	390
RDG	26	4	30
NLSY1	5	0	5
AGN	3	0	3
CSS	3	0	3
Total	1224	896	2120

Table 2. Neural Network Architectures: dropout of 0.25 between the hidden and output layers is common for each model.

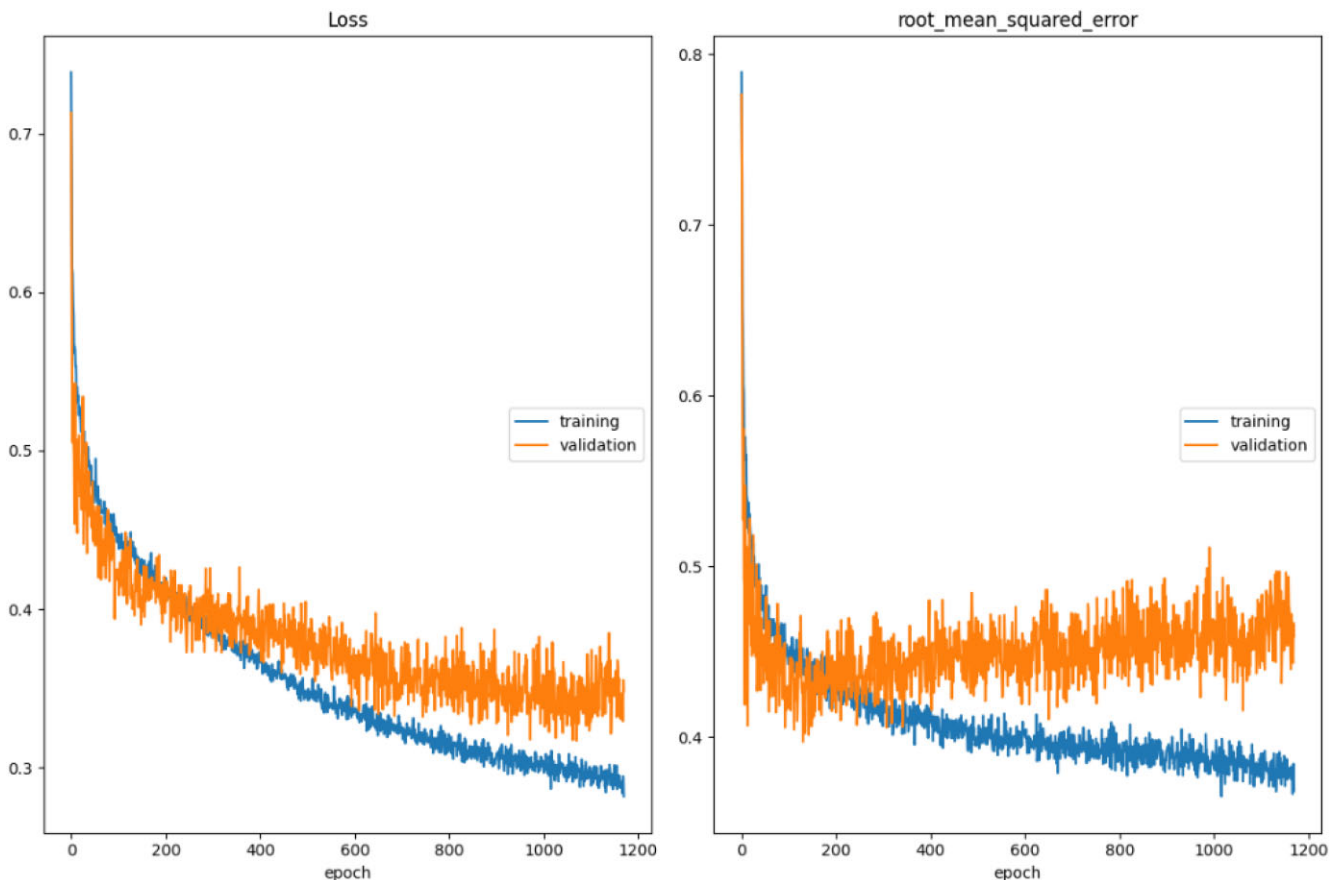
Model	Hidden layer	Output layer	Estimator
Frequentist	Dense (64 neurons)	Dense (1 neuron)	–
Variational inference	Dense (64 neurons)	DenseFlipout (1 neuron)	Flipout
Variational inference	Dense (64 neurons)	Dense Reparametrization (1 neuron)	Reparametrization

standard regression algorithms such as Random Forest, XG Boost, Big LASSO, and Bayes GLM to estimate the redshift of the corresponding input target. The authors make use of a 10 cross-fold validation technique iterated over 10 times to report a correlation

coefficient (r) ranging from 0.704 to 0.718. Moreover, they also reported a root-mean-squared error (RMSE) ranging from 0.432 to 0.438.

Narendra et al. (2022) is an advancement of Dainotti et al. (2021). The authors employed a similar ensemble-based technique as observed in Dainotti et al. (2021), however, the only difference besides an increase in the data points and the feature vector is the choice of machine learning models. The authors report an RMSE value of 0.212 when the sample size is 111 and 0.458 when the sample size is 1112. As RMSE is inversely proportional to the number of samples used during evaluation, it cannot be considered the best evaluation metric to compare different algorithms unless the sample size is the same across the algorithms. Also, the authors report a correlation coefficient of $r \approx 0.74$ in both of the aforementioned cases.

In Coronado-Blázquez (2023), the author makes use of the 4LAC DR3 catalogue, which is an updated version of the data used in Dainotti et al. (2021) and Narendra et al. (2022) with multiple additional features and a significant increase in the number of data points. To optimally use both the numerical as well as categorical features, the author relies on the CatBoost algorithm, which is a boosted decision tree-based algorithm capable of dealing with the categorical data. The author employs a five-cross validation technique to make effective use of the limited data. The reported ‘RMSE’ and ‘ r ’ values in this study are 0.46 and 0.71, respectively. Similar to Dainotti et al. (2021) and Narendra et al. (2022), the author also experiments with an ensembled approach having combined eight different algorithms; however, the performance of the CatBoost


Figure 1. Plots for Epochs versus Loss (MAE) and RMSE for variational inference (flipout estimator).

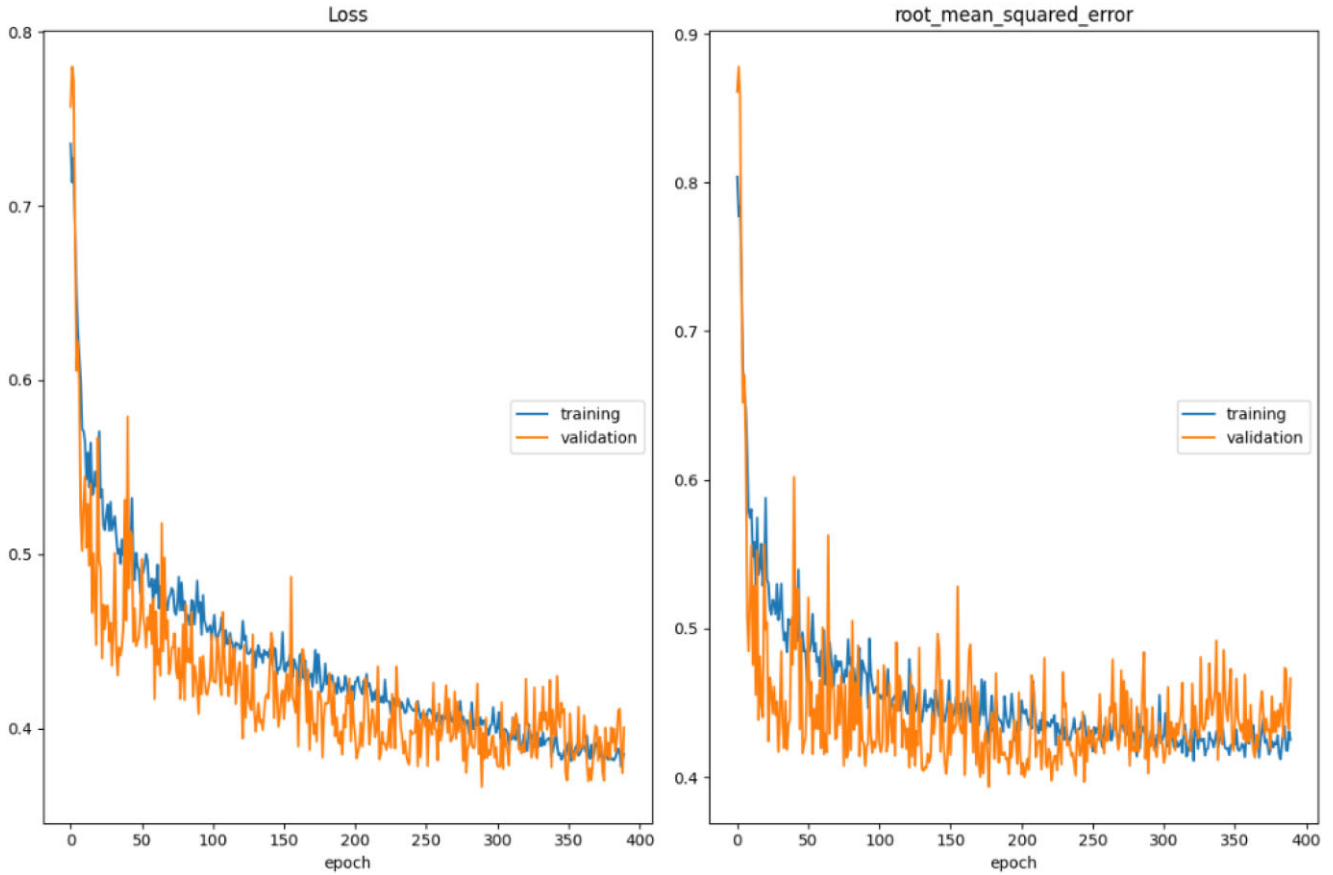


Figure 2. Plots for Epochs versus Loss (MAE) and RMSE for variational inference (reparametrization estimator).

Table 3. Performance Analysis: RMSE and correlation coefficient comparison between previous approaches and our proposed model. We make use of the mean predictions for the bayesian models to calculate the required metric.

	Dainotti et al. (2021)	Narendra et al. (2022)	Coronado-Blázquez (2023)	Frequentist algorithm	Variational inference (Flipout)	Variational inference (Reparametrization)
RMSE	0.432–0.438	0.458	0.46	0.415	0.406	0.438
Correlation Coefficient	0.704–0.718	0.74	0.71	0.784	0.777	0.778

Table 4. Assessing redshift predictions using variational inference: summary of true values, estimators, confidence intervals, and variance for a random set of samples.

True value	Estimator	68.2 per cent CI	95.4 per cent CI	99.7 per cent CI	Variance
0.1860	Reparametrized Flipout	0.1498–0.245	0.1022–0.2926	0.0546–0.3402	0.002
		0.1539–0.2533	0.1041–0.3031	0.0543–0.3529	0.002
0.2974	Reparametrized Flipout	0.3138–0.4124	0.2645–0.4617	0.2152–0.511	0.002
		0.3173–0.4177	0.2671–0.4679	0.2169–0.5181	0.002
0.4470	Reparametrized Flipout	0.2148–0.3588	0.1428–0.4308	0.0708–0.5028	0.005
		0.3563–0.4699	0.2955–0.5267	0.2427–0.5835	0.003
1.014	Reparametrized Flipout	0.7964–1.0154	0.6869–1.1249	0.5774–1.2344	0.011
		0.7924–0.979	0.6991–1.0723	0.6058–1.1656	0.008

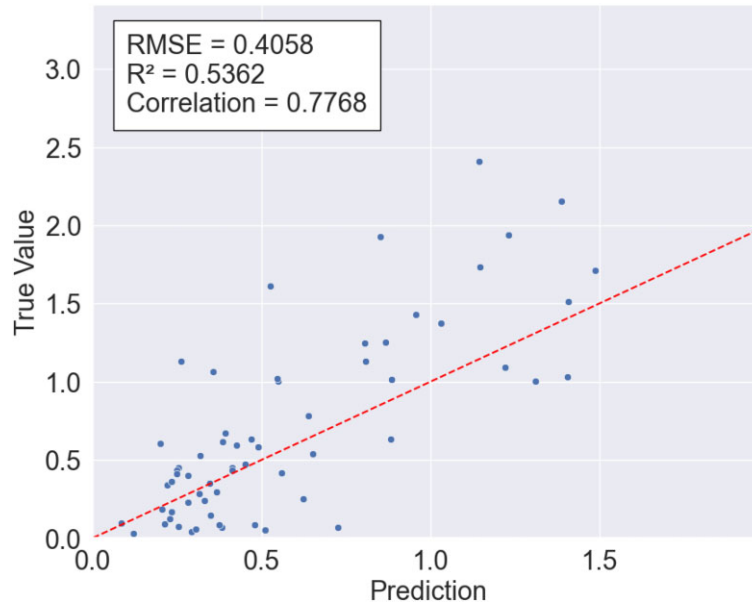


Figure 3. Scatter relation between the true value and the predicted mean value using variational inference (Flipout Estimator). The red diagonal represents a perfect prediction.

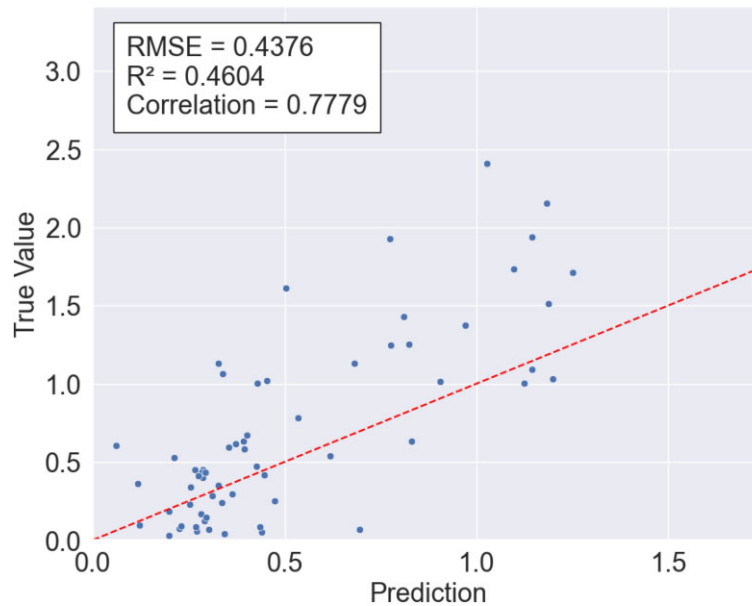


Figure 4. Scatter relation between the true value and the predicted mean value using variational inference (reparameterization estimator). The red diagonal represents a perfect prediction.

model is reported to be significantly better than what was observed in the ensembled algorithm.

Considering the limited number of studies conducted on this topic, none of which account for the uncertainty of the predicted redshifts, in this manuscript, we introduce an algorithm that employs a multilayer perceptron with a single hidden layer as the foundational model, which, when modified using variational inference allows us to not only quantify uncertainty but also augment our results.

2 METHODOLOGY

2.1 Data collection and processing

Since its launch in 2008, the Fermi Gamma-Ray Space Telescope's onboard instrument called the LAT has been continuously monitoring the high-energy sky (Atwood et al. 2009). In this study, we utilize the Fermi fourth catalogue of active galactic nuclei (AGNs) data release 3 (4LAC-DR3; Ajello et al. (2022a; 2022b)). The catalogue comprises 3407 individual sources, of which 1806 sources have known

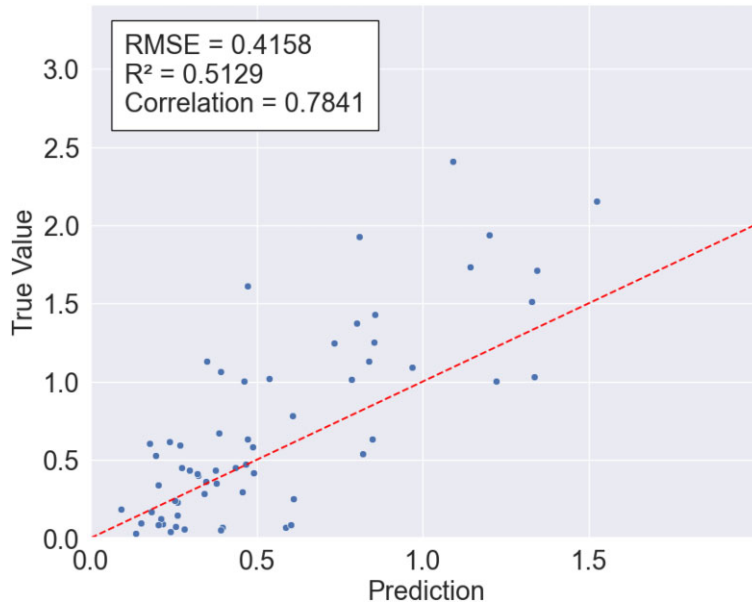


Figure 5. Scatter relation between the true value and the predicted value using Frequentist model. The red diagonal represents a perfect prediction.

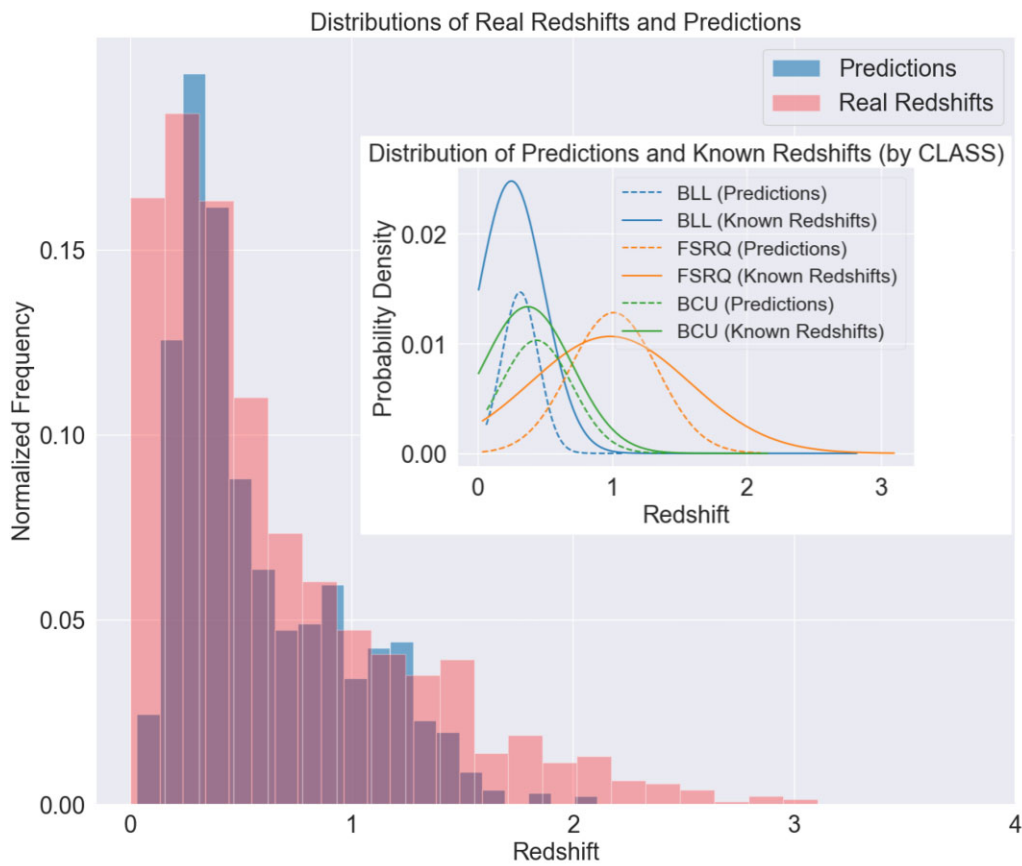


Figure 6. Variational inference (flipout estimator) – comparison between predicted mean redshift and true redshift using Histograms. The distribution of the redshift values for both the known and predicted redshifts, disaggregated by the ‘CLASS’ feature, is shown. Here, only those classes with more than 50 samples are represented.

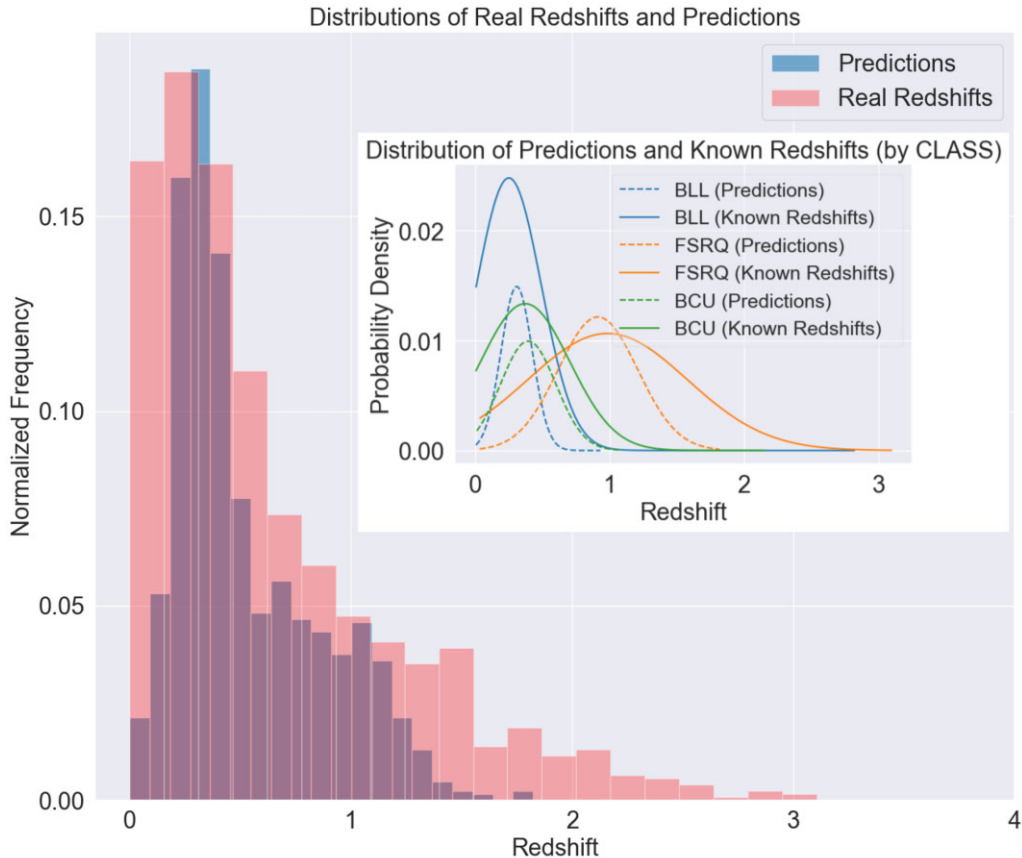


Figure 7. Variational inference (reparameterized estimator) – comparison between predicted mean redshift and true redshift using histograms. The distribution of the redshift values for both the known and predicted redshifts, disaggregated by the ‘CLASS’ feature, is shown. Here, only those classes with more than 50 samples are represented.

redshifts. Each source is characterized by a set of 41 different features with randomly missing values reported in this catalogue. Following Coronado-Blázquez (2023), we shortlist a set of 24 features for our study. Some of the features, such as ‘SED_class’, ‘Highest_energy’, and ‘Unc_LP_beta’ have a number of missing values. After sufficient experimentation with different imputing techniques, feature removal, and data removal, we proceed with the removal of the data points with missing values for the ‘Highest_energy’ and ‘Unc_LP_beta’ features. On the other hand, the missing values for the ‘SED_class’ are imputed using the mode estimation technique or, most frequent categorical value imputation Lin & Tsai (2020). To carry out the imputation process, we make use of sklearn’s Pedregosa et al. (2011a) SimpleImputer with appropriate arguments like setting strategy to most_frequent. This leaves us with 1224 data points (for detailed data distribution refer to Table 1) for our study with 90 per cent of the data used for training and 10 per cent of data used for validation and testing purposes equally divided among each other. It is crucial to note that the concept of the *validation data split* refers to the division of data used for evaluating and refining a deep learning model during its training process. This division serves as a means to optimize the model’s performance and make necessary adjustments. By subjecting the trained model to the validation set, we gain valuable insights into its ability to generalize on unseen data. The model’s performance on the validation set can be regarded as a reliable indicator of its performance on entirely new data at each training epoch. This evaluation helps in identifying potential issues, such as overfitting,

which can significantly affect the model’s effectiveness in real-world applications. It allows us to make unbiased estimations of critical hyperparameters, such as the number of neurons in the hidden layer or the dropout rate, essential for optimizing the model’s performance. The collected data consists of a number of numerical and categorical features. To deal with categorical data, we convert them to an integer-valued array using Sklearn’s ordinal encoder. Next, all the numeric data is normalized using the StandardScaler provided by Sklearn Pedregosa et al. (2011b); Buitinck et al. (2013). Prior to standard scaling, all numeric features except ‘Frac_Variability’, ‘GLAT’, ‘GLON’, ‘LP_Index’, ‘LP_beta’, ‘PL_Index’, ‘Unc_Flux1000’, and ‘Unc_PL_Index’ undergo log transformations. After pre-processing, we are left with a total of 1224 samples with known redshift values and 896 samples with unknown redshifts. Please refer Table 1 for class-wise distribution of the samples. The feature engineering and data engineering and proposed algorithms are implemented using PYTHON 3.7. The pandas library Wes McKinney (2010) is used to read the dataframes from the files and store them and once the input features are identified, we store them using numpy Harris et al. (2020) arrays, in order to feed them into our TensorFlow Abadi et al. (2015) models.

2.2 Model architecture and uncertainty quantification

In this study, we propose a multilayer perception Baum (1988); Murtagh (1991); Noriega (2005), with a single hidden layer having

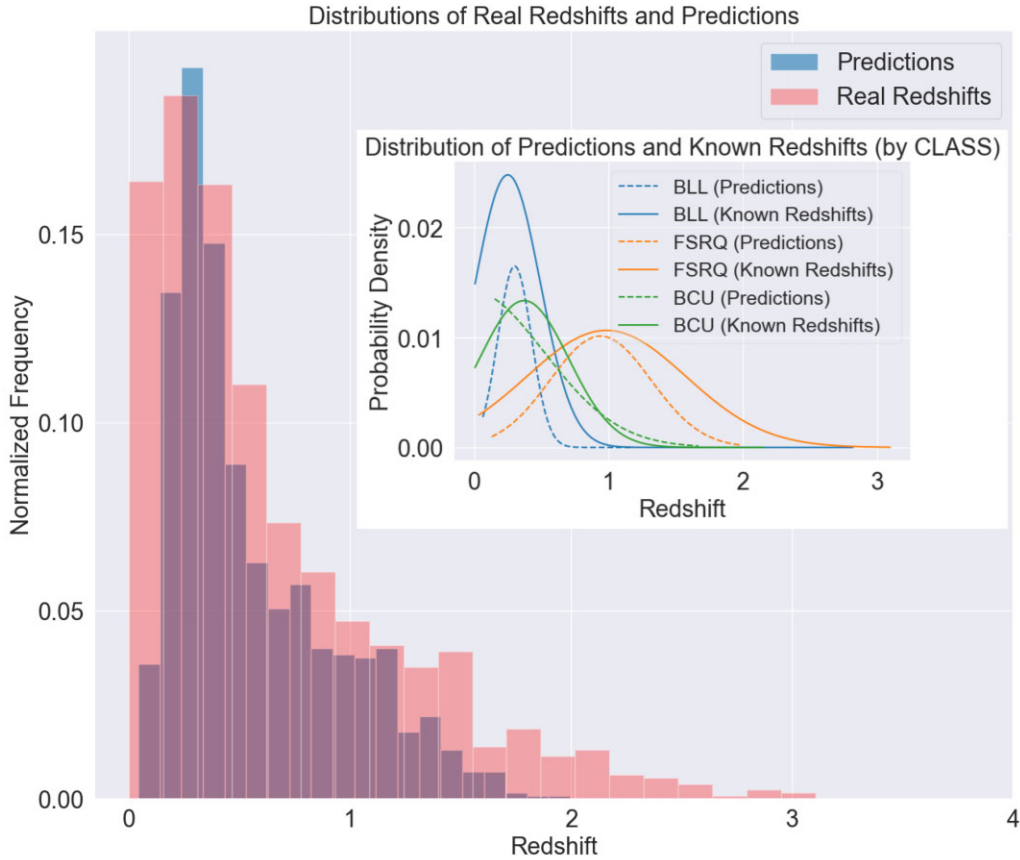


Figure 8. Frequentist model – comparison between predicted mean redshift and true redshift using histograms. The distribution of the redshift values for both the known and predicted redshifts, disaggregated by the ‘CLASS’ feature, is shown. Here, only those classes with more than 50 samples are represented.

64 neurons. A multilayer perceptron, often abbreviated as MLP, is a feed-forward Neural Network with at least three layers including the standard input, hidden, and output layer. Every layer has multiple nodes/neurons in it, which along with the number of hidden layers define the complexity of the model. Though there are many standard techniques to define the number of neurons in every hidden layer, in this study due to the simplicity of our model, we come up with the value of 64 after sufficient experimentation. These MLPs are fully connected, implying that every node in layer ‘ i ’ connects to each node in the subsequent layer ‘ j ’ through a weight value denoted as w_{ij} . The learning process is facilitated by adjusting the values of these weights as the data is processed, guided by the error between the MLP output and the target value. Further, to avoid overfitting, we introduce a dropout Srivastava et al. (2014); Srinivas & Babu (2016); Cai et al. (2019) of 0.25 in the hidden layer. This ensures that, during training, at any point in time, a neuron will be inactive with a probability of 0.25. This prevents the network from relying too heavily on specific neurons and encourages more robust, generalized learning. Next, to ensure non-linearity within the model, we apply ReLU – a widely used activation function Fukushima (1975). In the output layer, we utilize the softplus activation function Dubey, Singh & Chaudhuri (2022), which is just a smooth continuous version of ReLU. Placing an activation function at the end of each layer ensures that the layer’s output undergoes a non-linear transformation before being passed to the next layer. This is crucial in enabling the algorithm to learn and capture non-linear dependencies between the

input and the output. For the loss function, we employ the ‘Mean Absolute Error’ (MAE) Hodson (2022). This baseline model treats its parameters as point estimates and hence we refer to it as the ‘frequentist’ model. Moreover, to account for uncertainty, we employ the method of variational inference to modify our frequentist model using two different estimators, as discussed below. A summary of the architectures for the three models is listed in Table 2.

Variational inference is a technique that aims to approximate the true but often intractable posterior distribution of the model’s parameters (weights and biases) given the observed data Shridhar, Laumann & Liwicki (2019); Jospin et al. (2022). Instead of directly calculating the posterior, which is either challenging or impossible in complex models, variational inference introduces an approximating distribution (usually a known and tractable distribution).

To achieve this, a prior distribution is assigned to the model’s parameters, representing our initial beliefs about their values. As data is observed, the prior is updated using Bayes’ rule to obtain the posterior distribution. However, directly calculating the posterior is intractable for many models, especially neural networks. Thus, an optimization problem is formulated: we seek the closest approximating posterior distribution (in terms of the Kullback–Leibler (KL) divergence) that can be efficiently computed Bishop (2006). Both the prior and the approximating distributions are chosen as the Normal distribution, due to its desirable properties, like being a conjugate prior to itself.

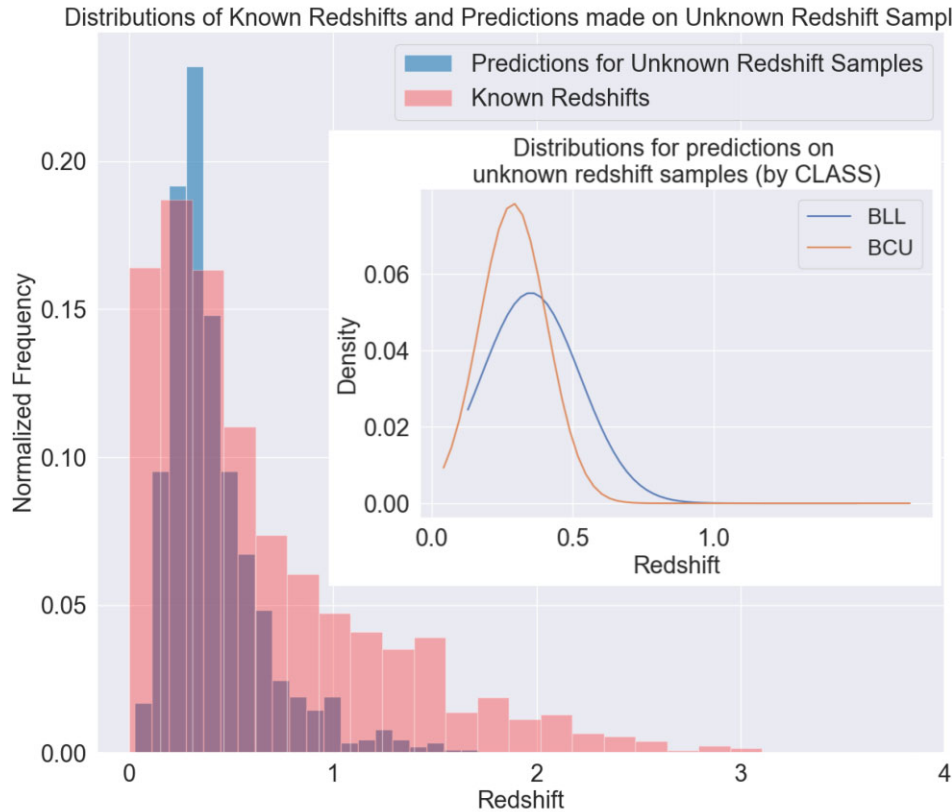


Figure 9. Variational inference (flipout estimator) – comparison between known redshift samples and predictions on unknown redshift samples using histograms. The distribution of the redshift values for the predictions made on the unknown redshift samples, disaggregated by the ‘CLASS’ feature. Only classes with more than 50 samples are represented.

Unlike traditional neural networks that rely on point estimates, variational inference provides a more meaningful measure of uncertainty and captures the complexity of the posterior distribution through this probabilistic approach. We use TensorFlow Probability Dillon et al. (2017) and Keras Chollet et al. (2015) to implement the proposed models. There are multiple methods to implement variational inference using tensorflow probability; however, we proceed with the DenseFlipout and DenseReparamtrization layers. In both of these methods, the layers implement the Bayesian variational inference counterpart to a Dense layer by drawing the parameter values from distributions. An important difference between both of these layers is that the flipout estimator uses roughly twice as many floating point operations as the reparametrization estimator. (Refer Wen et al. (2018) and Kingma & Welling (2013) for more information on both of these layers).

To quantify uncertainty, each sample is evaluated 1000 times, and the uncertainty is captured using the variance of the predictions. The resulting mean from the 1000 iterations is considered as the prediction of the Bayesian model. Due to the Bayesian nature of the variational inference algorithms, the output prediction at every iteration is an independent and identically distributed Gaussian sample. Having set the output predictions to be normally distributed for a fixed data point, we then calculate the mean and standard deviation of the predictions for each sample. As evident from the theory of Gaussian distributions, we then make use of the standard 3σ rule to come up with a possible range of redshifts containing the true value of the redshift with an associated confidence level.

Although this rule comments on the confidence levels being, 68.2, 95.4, and 99.7 per cent for 1, 2, and 3 standard deviations from the mean, respectively, it is easy to generalize it for any range of values depending upon the allowed tolerance.

2.3 Training and validation

Considering the computational requirement to train the algorithm, we make use of Google Collaboratory, a cloud-based jupyter environment for model training. An important aspect of any Machine or Deep Learning algorithm is its reproducibility. To ensure this, we train our algorithms on a fixed random seed over a maximum of 2500 epochs, and include the data splits pertaining to the training, validation, and testing sets in our GitHub repository. To reduce the computational overhead and avoid overfitting, we introduce early stopping Caruana, Lawrence & Giles (2001) with a validation patience of 100, and as a result, the proposed variational inference models stop after 1170th and 390th epoch, respectively, as shown in Figs 1 and 2, respectively.

As evident from Fig. 2, during the initial 100 epochs, the rate of decrease in ‘loss’ and ‘RMSE’ for both the training and validation data points is high. However, at later stages, it tends to saturate. This indicates that there’s a very high probability of having no further decrease in the loss. Having said this, the use of early stopping ensures that the algorithm stops its training once the rate of decrease in the validation loss tends to zero. This helps in avoiding unnecessary computations. Also, in Fig. 1, we observe that at later stages there’s a decrease in training loss; on the other hand, the validation loss tends

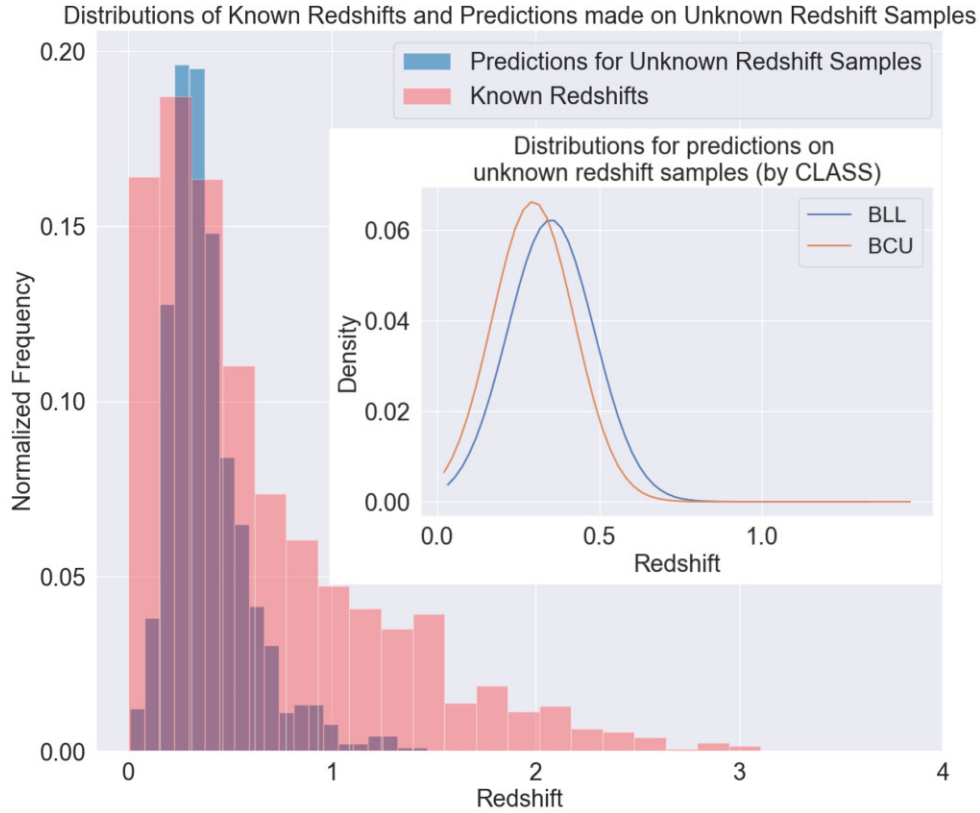


Figure 10. Variational inference (reparameterized estimator) – comparison between known redshift samples and predictions on unknown redshift samples using histograms. The distribution of the redshift values for the predictions made on the unknown redshift samples, disaggregated by the ‘CLASS’ feature. Only classes with more than 50 samples are represented.

to saturate and even increases in further epochs. This behaviour results in overfitting of the algorithm, if not stopped at the correct time, and the introduction of early stopping ensures the same.

To optimize the algorithm, we make use of ‘Adam’ Kingma & Ba (2014), which is one of the widely used optimizers in the Deep Learning community with a learning rate of 10^{-3} . One of the primary reasons for its popularity is that it incorporates momentum (for which we use the default values defined in TensorFlow) and is a variant of the AdaGrad optimizer, which facilitates quicker convergence.

3 RESULTS AND DISCUSSION

Blazars emitting γ -rays with known redshifts significantly contribute to our understanding of several fundamental aspects of cosmic phenomena. Determining their redshifts aids in constraining the nature of the Extragalactic Background Light (e.g. Ackermann et al. 2012a; Dwek & Krennrich 2013; Acciari et al. 2019). Additionally, these blazars shed light on the structures of intergalactic magnetic fields (e.g. Tavecchio et al. 2010; Finke et al. 2015; Aharonian et al. 2023) and the Universe’s star formation history (Ackermann et al. 2012b; Rojas-Bravo & Araya 2016; Fermi-LAT Collaboration et al. 2018). Also, by computing the luminosity function, we can estimate the evolution of blazars over cosmic time (Chiang et al. 1995; Ajello et al. 2012). This, in turn, can lead to the constraining of fundamental cosmological parameters (Domínguez et al. 2019; Zeng & Yan 2019).

The study contributes by providing an algorithm that rigorously estimates the possible range of redshifts with an associated confidence. To assess the effectiveness of our model, we conducted

evaluations using entirely new and unseen data, referred to as the test data. Since our study focuses on a regression problem, we utilized the ‘Root Mean Squared Error’ (RMSE) as one of our evaluation metrics. The Root Mean Squared Error (RMSE) calculates the square root of the average of the squared differences between predicted values and actual values in the test data. A lower RMSE generally indicates better model performance, as it signifies smaller prediction errors. However, the RMSE value is influenced by the number of samples. Therefore, we also utilized the ‘correlation coefficient’ to evaluate our model. The correlation coefficient measures the strength and direction of the linear relationship between two variables. A higher correlation coefficient indicates a better alignment between the predicted and actual values, demonstrating the model’s ability to capture the underlying patterns in the data. Table 3 clearly shows that our proposed algorithm yields improved results when compared to existing studies, with a maximal increase in the correlation coefficient of around 0.07.

Additionally, Table 4 presents a comparison between the actual redshift values and the predicted range of redshifts at fixed confidence levels for randomly selected data points from the test data set. The table clearly demonstrates that in the majority of cases, the true redshift value falls within the interval associated with a confidence level of 95.4 percent. Although in Table 4, we focus on the specific confidence levels, the range can be easily calculated for different confidence levels based on a real multiple of the standard deviation.

Figs 3, 4, and 5 present scatter plots that showcase the relationship between predicted and true redshifts obtained from various models.

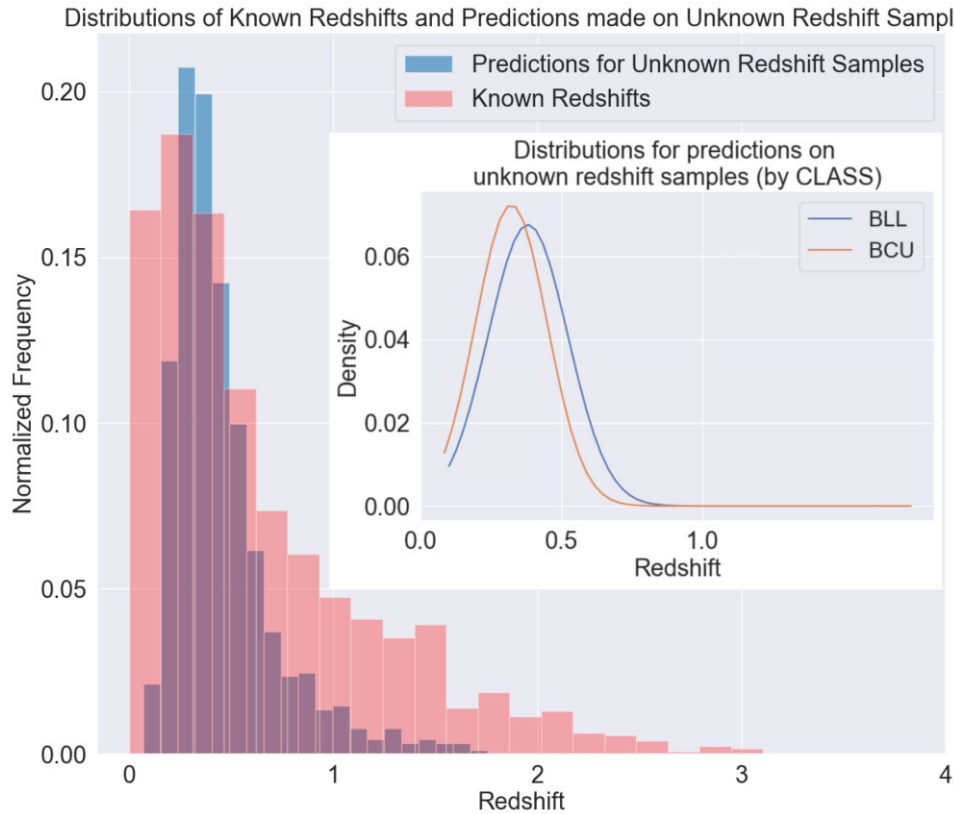


Figure 11. Frequentist model – comparison between known redshift samples and predictions on unknown redshift samples using histograms. The distribution of the redshift values for the predictions made on the unknown redshift samples, disaggregated by the ‘CLASS’ feature. Only classes with more than 50 samples are represented.

Table 5. Redshift prediction summary statistics.

Method	Known redshift samples			Unknown redshift samples		
	Mean prediction	Range	σ	Mean prediction	Range	σ
Frequentist model	0.559	0.04–1.99	0.372	0.455	0.07–1.77	0.258
Variational inference (flipout estimator)	0.581	0.027–2.11	0.382	0.415	0.0251–1.71	0.246
Variational inference (reparameterization estimator)	0.526	0.0004–1.82	0.332	0.393	0.0089–1.47	0.207

While it is evident that the predicted redshifts tend to be slightly lower than the actual values in many instances (a trend also observed in Coronado-Blázquez (2023), albeit with more scattered points), the incorporation of uncertainty and confidence levels addresses this issue. By utilizing a 3σ interval of the mean with a confidence level of 99.7 per cent, the majority of true values fall within this range – an analysis reveals that for all the samples with a known redshift, the true value falls within the 99.7 per cent confidence interval for 63 per cent of the samples using each method of variational inference. This enables astronomers to make informed decisions regarding the reliability of the algorithm’s predictions, considering the desired confidence level and width interval at any given point. Figs 6, 7, and 8 provide similar insights. Additionally, the figures highlight the algorithm’s limitations in regressing lower redshifts. However, due to the associated uncertainty and the range of predictions provided by variational inference, the lower redshifts are accounted for within the predicted range. This aspect of our proposed algorithm ensures that the true value is captured with a sufficiently high probability, depending on the allowed confidence level.

As illustrated in Figs 9, 10, and 11, the predictions made on the samples with an unknown redshift by the frequentist model, the flipout estimator model, and the reparameterization estimator model follow distributions similar to that of the predictions made on the known redshift samples with mean values of 0.455, 0.415, and 0.393, standard deviations of 0.258, 0.246, and 0.207, and redshift values ranging from 0.07–1.77, 0.0251–1.71, and 0.0089–1.47, respectively (Table 5).

Figs 12 and 13 display histograms corresponding to the data presented in Table 4. As evident from the figures, the predicted set of values for every redshift correspond to a Gaussian distribution, which confirms the inclination of the implemented algorithm with the theory and hence allows us to efficiently estimate the uncertainty associated with the range of predictions.

Also, as seen in Table 1 and Figs 9, 10, and 11, the predicted redshift class is mostly composed of BL Lacs and BCUs. These results are plausible because BL Lacs are strong gamma-ray emitters with weak or no emission lines, which makes estimating their redshifts very difficult. Similarly, the BCUs are unclassified sources

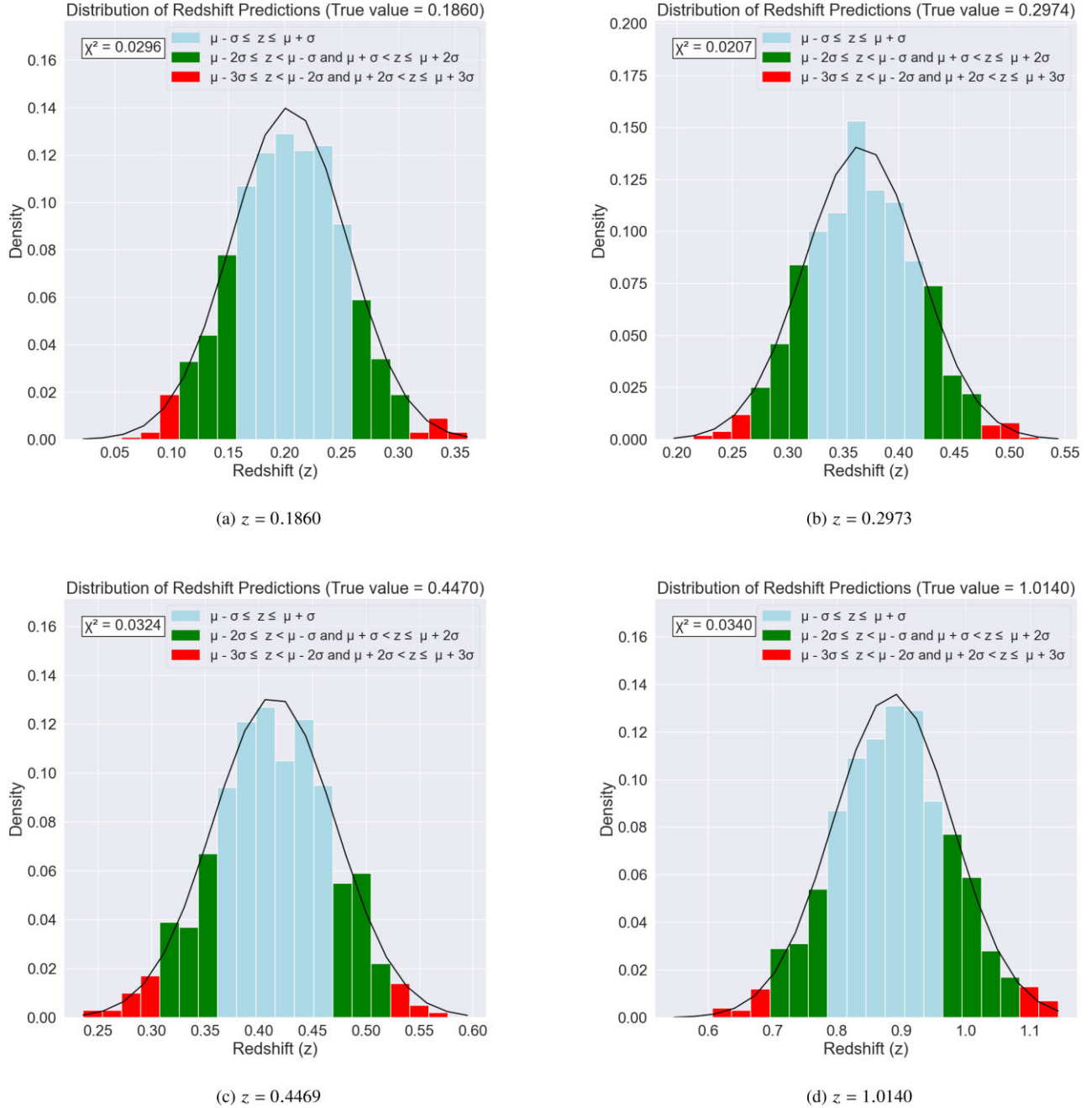


Figure 12. Evaluation of the uncertainty for variational inference using the flipout estimator. Redshift samples were evaluated 1000 times, and the resulting distribution for some of the known values is shown here. The distribution was fitted with a Gaussian PDF, and the values within 1σ , $1\sigma-2\sigma$, and $2\sigma-3\sigma$ from the mean were colour-coded as cyan, green, and red, respectively.

whose classification is challenging, as optical spectra or MWL observations required for a robust classification are not available. However, several studies based on machine learning predict that the majority of these sources are likely to be BL Lacs (see e. g. Kang et al. 2019; Agarwal 2023).

4 CONCLUSION

This study introduces a straightforward yet highly effective algorithm for redshift estimation using solely gamma-ray observa-

tions. The proposed algorithm shows improvements over existing methods, achieving significantly low RMSE values of 0.415, 0.406, and 0.438 in its frequentist, variational inference (flipout), and variational inference (reparametrization) variants, respectively. To further validate our results, we also employ the correlation coefficient as a complementary metric. Remarkably, we observe a substantial improvement in the correlation coefficient, with values increasing from 0.74 to 0.784, 0.777, and 0.778 for the respective algorithms, thus demonstrating the advantage of our proposed method. In addition to robust redshift regression, our algorithm

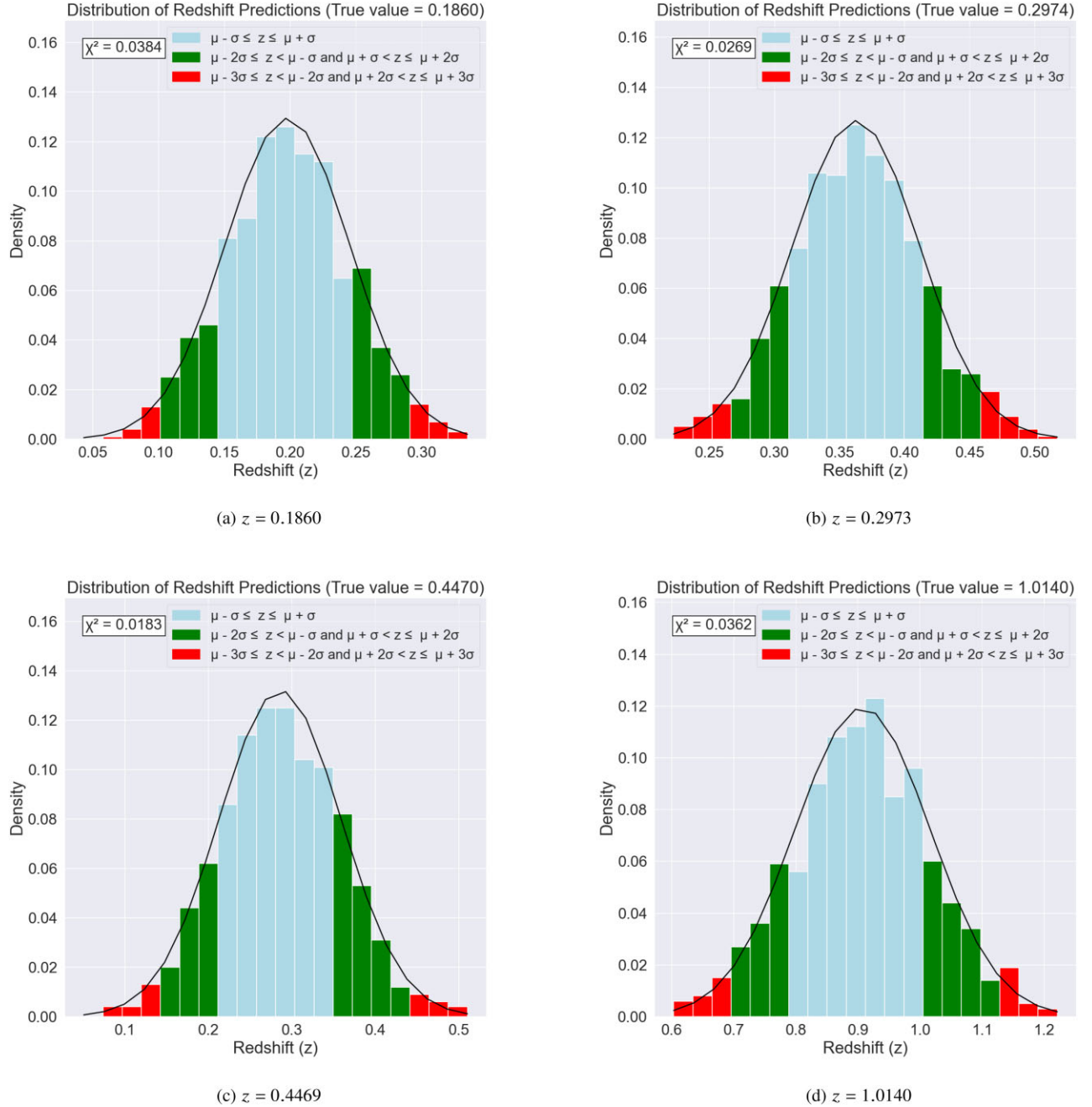


Figure 13. Evaluation of the uncertainty for variational inference using the reparameterization estimator. Redshift samples were evaluated 1000 times, and the resulting distribution for some of the known values is shown here. The distribution was fitted with a Gaussian PDF, and the values within 1σ , $1\sigma-2\sigma$, and $2\sigma-3\sigma$ from the mean were colour-coded as cyan, green, and red, respectively.

addresses the associated uncertainty by providing an estimated range of potential redshift values based on the desired confidence level. Notably, for highest confidence interval (99.7 per cent), the predictions of our algorithm encompass the true redshifts for the majority of the samples. This uncertainty quantification feature adds significant value to the algorithm’s predictions and helps users to make informed decisions based on their desired confidence level. Furthermore, we extend the application of our algorithm to predict unknown redshifts in the 4LAC-DR3 catalogue, utilizing variational inferences. This allows us to provide corresponding

uncertainties alongside the predicted redshifts, enhancing the reliability and applicability of our algorithm in real-world scenarios.

ACKNOWLEDGEMENTS

We thank the anonymous referee for a careful and thorough review of this paper, which helped us improve the quality of the work.

DATA AVAILABILITY

The data utilized in this paper can be accessed by the public through the Fermi Science Support Center (FSSC) of NASA's Goddard Space Flight Center. Furthermore, we have made both the code and the resulting data openly available on our [Github repository \(https://github.com/abhimanyu911/redshift-regression-with-uncertainty.git\)](https://github.com/abhimanyu911/redshift-regression-with-uncertainty.git) for public access.

REFERENCES

- Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, USENIX Association, Savannah, GA, USA, <https://www.tensorflow.org/>
- Acciari V. A. et al., 2019, *MNRAS*, 486, 4233
- Ackermann M. et al., 2012a, *Science*, 338, 1190
- Ackermann M. et al., 2012b, *ApJ*, 755, 164
- Agarwal A., 2023, *ApJ*, 946, 109
- Aharonian F. et al., 2023, *ApJ*, 950, L16
- Ajello M. et al., 2012, *ApJ*, 751, 108
- Ajello M. et al., 2022a, *ApJS*, 263, 24
- Ajello M. et al., 2022b, *ApJS*, 263, 24
- Atwood W. B. et al., 2009, *ApJ*, 697, 1071
- Baum E. B., 1988, *J. Complexity*, 4, 193
- Bhatta G., Dhital N., 2020, *ApJ*, 891, 120
- Bishop C. M., 2006, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg
- Buitinck L. et al., 2013, in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. p. 108
- Cai S., Shu Y., Chen G., Ooi B. C., Wang W., Zhang M., 2019, preprint (arXiv:1904.03392)
- Caruana R., Lawrence S., Giles L., 2001, *Advances in neural information processing systems*, p. 402
- Chiang J., Fichtel C. E., von Montigny C., Nolan P. L., Petrosian V., 1995, *ApJ*, 452, 156
- Chollet F. et al., 2015, Keras, <https://github.com/fchollet/keras>, Last Access: 10 November 2023
- Coronado-Blázquez J., 2023, *MNRAS*, 521, 4156
- Dainotti M. G. et al., 2021, *ApJ*, 920, 118
- Dillon J. V. et al., 2017, *CoRR*, abs/1711.10604
- Domínguez A. et al., 2019, *ApJ*, 885, 137
- Dubey S. R., Singh S. K., Chaudhuri B. B., 2022, *Neurocomputing*, 503, 92
- Dwek E., Krennrich F., 2013, *Astrophys. J.*, 43, 112
- Fermi-LAT Collaboration et al., 2018, *Science*, 362, 1031
- Finke J. D., Reyes L. C., Georganopoulos M., Reynolds K., Ajello M., Fegan S. J., McCann K., 2015, *ApJ*, 814, 20
- Fukushima K., 1975, *Biol. Cybern.*, 20, 121
- Harris C. R. et al., 2020, *Nature*, 585, 357
- Hodson T. O., 2022, *GMD*, 15, 5481
- Jospin L. V., Laga H., Boussaid F., Buntine W., Bennamoun M., 2022, *IEEE Comput. Intell. Mag.*, 17, 29
- Kang S.-J., Li E., Ou W., Zhu K., Fan J.-H., Wu Q., Yin Y., 2019, *ApJ*, 887, 134
- Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
- Kingma D. P., Welling M., 2013, preprint (arXiv:1312.6114)
- Lin W.-C., Tsai C.-F., 2020, *Artif. Intell. Rev.*, 53, 1487
- McKinney Wes, SciPy 2010, in *van der Walt Stéfan, Millman Jarrodeds, Proceedings of the 9th Python in Science Conference*. p. 56,
- Murtagh F., 1991, *Neurocomputing*, 2, 183
- Narendra A., Gibson S. J., Dainotti M. G., Bogdan M., Pollo A., Liodakis I., Poliszczuk A., Rinaldi E., 2022, *ApJS*, 259, 55
- Noriega L., 2005, *School of Computing, Staffordshire University*, 4, 444
- Pedregosa F. et al., 2011a, *JMLR*, 12, 2825
- Pedregosa F. et al., 2011b, *JMLR*, 12, 2825
- Rojas-Bravo C., Araya M., 2016, *MNRAS*, 463, 1068
- Shridhar K., Laumann F., Liwicki M., 2019, preprint (arXiv:1901.02731)
- Skinner G. K., 2010, preprint (arXiv:1009.2098)
- Srinivas S., Babu R. V., 2016, preprint (arXiv:1611.06791)
- Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, *JMLR*, 15, 1929
- Tavecchio F., Ghisellini G., Foschini L., Bonnoli G., Ghirlanda G., Coppi P., 2010, *MNRAS*, 406, L70
- Wen Y., Vicol P., Ba J., Tran D., Grosse R., 2018, preprint (arXiv:1803.04386)
- Zeng H., Yan D., 2019, *ApJ*, 882, 87

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.