Exploring the potential of neural networks to predict statistics of solar wind turbulence

Daniel Wrench[1], Tulasi N. Parashar[1], Ritesh K Singh[2], Marcus Frean[1], and Ramesh Rayudu[1]

[1]Victoria University of Wellington
[2]Indian Institute of Science, Education and Research

November 26, 2022

Abstract

Time series datasets often have missing or corrupted entries, which need to be ignored in subsequent data analysis. For example, in the context of space physics, calibration issues, satellite telemetry issues, and unexpected events can make parts of a time series unusable. Various approaches exist to tackle this problem, including mean/median imputation, linear interpolation, and autoregressive modeling. Here we study the utility of artificial neural networks (ANNs) to predict statistics, particularly second-order structure functions, of turbulent time series concerning the solar wind. Using a dataset with artificial gaps, a neural network is trained to predict second-order structure functions and then tested on an unseen dataset to quantify its performance. A small feedforward ANN, with only 20 hidden neurons, can predict the large-scale fluctuation amplitudes better than mean imputation or linear interpolation when the percentage of missing data is high. Although they perform worse than the other methods when it comes to capturing both the shape and fluctuation amplitude together, their performance is better in a statistical sense for large fractions of missing data. Caveats regarding their utility, the optimisation procedure, and potential future improvements are discussed.

Exploring the potential of neural networks to predict statistics of solar wind turbulence

Daniel Wrench[1], **Tulasi N. Parashar**[1], **Ritesh K. Singh**[2], **Marcus Frean**[1], **Ramesh Rayudu**[1]

[1]Victoria University of Wellington, Kelburn, Wellington, NZ 6012
[2]Department of Physical Sciences, Indian Institute of Science Education and Research Kolkata, Mohanpur, 741246, India

Key Points:

- Small artificial neural networks (ANNs) are good at predicting large scale values of structure functions.
- An ANN with only 20 hidden neurons statistically outperforms simple imputation techniques for large fractions of missing data.
- More work is needed to improve the ANN's performance in predicting both large and small scale values.

Corresponding author: Daniel Wrench, daniel.wrench@vuw.ac.nz

Abstract

Time series datasets often have missing or corrupted entries, which need to be ignored in subsequent data analysis. For example, in the context of space physics, calibration issues, satellite telemetry issues, and unexpected events can make parts of a time series unusable. Various approaches exist to tackle this problem, including mean/median imputation, linear interpolation, and autoregressive modeling. Here we study the utility of artificial neural networks (ANNs) to predict statistics, particularly second-order structure functions, of turbulent time series concerning the solar wind. Using a dataset with artificial gaps, a neural network is trained to predict second-order structure functions and then tested on an unseen dataset to quantify its performance. A small feedforward ANN, with only 20 hidden neurons, can predict the large-scale fluctuation amplitudes better than mean imputation or linear interpolation when the percentage of missing data is high. Although, they perform worse than the other methods when it comes to capturing both the shape and fluctuation amplitude together, their performance is better in a statistical sense for large fractions of missing data. Caveats regarding their utility, the optimisation procedure, and potential future improvements are discussed.

Plain Language Summary

We explore the utility of machine learning to predict statistics of a turbulent system such as the solar wind, in cases involving large data gaps. It is shown that simple artificial neural networks (ANNs) are good at estimating large-scale features of second-order structure functions even for very large amounts of missing data. However, these simple ANNs are limited in estimating other features of the structure functions, such as inner and outer scales, and the inertial range slope. More sophisticated methods are required to describe such features.

1 Introduction

Analyses of real-world time series are often hindered by incomplete datasets. This is very common for physiological, environmental, astronomical, and heliospheric time series. The instrumentation used to take measurements may be prone to failure, or variations in the environment itself may preclude data collection for certain periods. For example, time series of sea level and wave height based on radio signals are commonly incomplete due to radio interference, airborne seawater spray, and the loss of line-of-sight caused by large waves (Makarynskyy et al., 2005). In physiology, recordings of blood flow and other processes are often contaminated with artifacts due to movement of the subject and improper interfacing with sensors (Pavlova et al., 2019), and removal of these leaves gaps in the series. Ground-based astronomical observations are affected by cloud cover and the maintenance and malfunction of instruments. In the case of *in situ* measurements of the solar wind, incomplete time series result from calibration, instrumentation, and telemetry issues (Rehfeld et al., 2011). Telemetry is a particular issue for the two Voyager spacecraft, which must align their data transmissions with NASA's ground-based communication facilities, the Deep Space Network (Ludwig & Taylor, 2016; Gallana et al., 2016).

Discontinuity in time series data represents a loss of information, affecting the statistics and in turn polluting predictions. This includes significant effects on frequency-domain (spectral) and scale-domain analysis. An example is 'spectral inheritance' in which the gaps contaminate the rest of the data in the form of "spurious periodicities arising from the spectral properties of the sets of gaps" (Frick et al., 1998; Gallana et al., 2016). More generally, data gaps result in dirty spectra, which lead to poor estimation of power, particularly at high frequencies (Munteanu et al., 2016). In radio and gamma-ray astronomy, this causes issues for calculating the periodicity of stellar objects (VanderPlas, 2018).

In heliophysics it hinders our understanding of the spectral properties of turbulence (Gallana et al., 2016; Fraternale et al., 2019).

Many different methods have been explored to deal with this issue of spectral estimation from a time series that has gaps. A significant amount of literature has been dedicated to estimating the power spectra and periodicities of a gapped signal. We find that the methods of spectral estimation from sparse datasets can be grouped into two broad categories:

1. Interpolation of missing values, followed by spectral estimation from the reconstructed signal
2. Spectral estimation directly from the dataset with gaps

The first category of techniques is regularly used in the space plasma literature. Often segments that are relatively continuous are selected to avoid large gaps. For example, Wu et al. (2013) and C. Chen et al. (2020) removed gaps larger than 5% and 1% respectively. The remaining small gaps are typically filled using linear interpolation (Burlaga, 1991; Podesta et al., 2007). However, linear interpolation amounts to strong smoothing of part of the signal, which results in a loss of information at high frequencies (Frick et al., 1998). Because this effect becomes worse with increasing data loss, this technique is only feasible for relatively small gaps (Bavassano et al., 1982; Y. Chen et al., 2002). Furthermore, by excluding large segments of the data to avoid the gaps, a considerable amount of information about the system is lost. For this reason, interpolation methods that are more consistent with the spectral content of the observed data segments have also been used.

For example, interpolation of sparse signals has also been achieved by modelling the signal as a stochastic process (specifically, that of fractional Brownian motion), and then further defining the process as a multi-point "bridge" between the prescribed (observed) measurements (Friedrich et al., 2020). A strategy for identifying the optimal Hurst exponent required by the fractional Brownian motion algorithm was proposed and tested by reconstructing velocity field measurements from a superfluid helium experiment.

Singular spectrum analysis (SSA) is a non-parametric algorithm used for forecasting from gapped time series in a number of fields, including heliophysics (Schoellhamer, 2001; Kondrashov et al., 2010). SSA involves reconstructing a signal from its principal components, and its benefits are that it requires no prior knowledge of the periodicities in the data, and it accounts for noise in the signal. However, the technique is especially sensitive to increasing gap sizes: the root mean-squared error was shown to increase significantly in a study investigating data gaps in soil respiration data (Zhao et al., 2020). SSA has been used for gap-filling of solar wind data (Kondrashov et al., 2010). However, this study also made use of continuous measurements of geomagnetic indices, which could potentially improve the performance of this method.

ARIMA models are the standard models for forecasting time series, and these can be fitted to non-uniformly sampled data using a maximum likelihood technique (Harvey & Pierse, 1984; Broersen, 2006). This has been shown to result in much better estimation of time series parameters such as level, error variance, and slope of the time series, compared to simple mean imputation and linear interpolation (Velicer & Colby, 2005). A similar method of finding the best ARIMA model order based on maximising entropy has been applied to solar oscillation data (Brown & Christensen-Dalsgaard, 1990). Starting with some assumptions about the typical gap-lengths and the noise in the signal, the authors were able to reproduce unique spectral features.

Neural networks, a prominent algorithm from machine learning, have also been used to fill gaps in time series. Specifically, a simple feed-forward neural network was found to accurately reproduce simulated stochastic processes and fill gaps that matched the original power spectrum with up to 50% missing data (Comerford et al., 2015). Generative adversarial networks (Y. Luo et al., 2018) and convolutional neural networks (Jang et al., 2020) have also been used to impute missing intervals.

A comprehensive study of dealing with large data gaps in solar wind data used a combination of techniques to recover the spectrum from Voyager datasets (Gallana et al., 2016). This compared Fourier transforms of gap-free subsets; Fourier transforms of the correlation function of the data, with and without linear interpolation; maximum likelihood recovery; and compressed sensing spectral estimation. All of these methods, apart from compressed sensing, fall into the first category of gapped estimation techniques. Ultimately, this work was able to determine spectra over a very large range of frequencies and thereby extract information on various turbulent features.

Moving now to the second category of spectral estimation methods, a continuous wavelet transform method has been used to perform spectral estimation directly from a gapped signal (Frick et al., 1998). Of direct relevance to our work, this technique has been applied to magnetic field time series in the solar system by Magrini et al. (2017) and de Souza Echer et al. (2021). In the first study, the wavelet method was compared with two polynomial interpolation methods for spectral analysis of artificially-gapped OMNIWeb (near-Earth solar wind) data. It was found that all techniques perform satisfactorily for small gaps, but the wavelet method better estimates the energy of the signal at certain scales for large gaps. In the second study, the wavelet method was used to find the dominant periodicities of magnetic field fluctuations in the magnetosphere of Jupiter.

In an example of using neural networks in the second category of techniques, Randolph-Gips (2008) created a Cosine Neural Network, which is able to process and recognise missing data without any prior imputation. This addresses the issue of how to represent missing data to a neural network. It does this using 'weighted norms', parameters which reduce to 0 when the corresponding input feature is missing. This informs the network to ignore these features for that instance.

The present study further examines a machine learning approach to the second category of methods. Specifically, we investigate framing the estimation of high-quality statistics from a dataset with gaps as a supervised learning regression problem, bypassing interpolation and its attendant uncertainties entirely. Whereas the mapping from a complete dataset to its statistics is generally in the form of a simple function (e.g., the equation for the mean or standard deviation), we are interested in whether a neural network - the 'universal approximator' - can learn a mapping from an *incomplete* dataset to the 'clean' statistic that *would* have followed, had the complete dataset been available. This approach is taken because the primary goal is not to accurately reproduce the complete series itself, but rather the statistics calculated from the complete series. The focus here is not on understanding the relationship between input and output, but rather on finding an input-output mapping that achieves good performance on unseen gapped datasets. As a case study, this technique is applied to time series of the fluctuating interplanetary magnetic field, produced by the solar wind and measured by the NASA spacecraft Parker Solar Probe. The statistic we attempt to estimate is the *second-order structure function* (S^{fn} for brevity).

The nth order S^{fn} for a time-varying signal a is defined as (Batchelor, 1953; Biskamp, 2003)

$$S_a^{(n)}(\tau) = \langle |\delta a(t,\tau)|^n \rangle \tag{1}$$

where $a(t)$ is the scalar variable of interest, $\delta a(t,\tau) = a(t+\tau) - a(t)$ is the increment, n is the order, and $\langle\rangle$ denotes expectation over t. For a vector set of time series $\mathbf{a}(t) = (a_x(t),\, a_y(t),\, a_z(t))$, the $S^{fn}$is defined as

$$S_a^{(n)}(\tau) = \langle |\delta \mathbf{a}(t,\tau)|^n \rangle \tag{2}$$

where $\delta\mathbf{a}(t,\tau) = \mathbf{a}(t+\tau) - \mathbf{a}(t)$. The S^{fn}s of various orders follow power-law behaviour in the inertial range. The S^{fn}s, by themselves and in combinations, encode a significant amount of physics, such as the spread of energy across scales and the locality and intermittency of the turbulent structures (Biskamp, 2003; Panchev, 1971). In this paper, as a proof-of-concept, we stick to the second-order S^{fn}.

After the clean, 'true' structure functions are calculated for a series of continuous segments of the time series, the segments are artificially gapped in several different ways. These segments are provided as the input data to the model, with the original structure functions as the target outputs. The model predictions are then compared with structure functions calculated directly from the gapped intervals, and from gapped intervals with an interpolation technique applied. In this way we can compare the performance of each technique in approximating the true, 'clean' statistic of the original ungapped interval. The goal of estimating not just a single label or parameter but rather an array of values over a certain range also makes this approach unique.

2 Data Preparation & Training

The data used in this project were taken from the Parker Solar Probe's (PSP) (Fox et al., 2016) fluxgate magnetometer (FGM) instrument (Bale et al., 2016). PSP is a spacecraft launched in 2018 to study the physics of the inner heliosphere and the origins of the solar wind by flying very close to the Sun (as close as 9.9 solar radii during orbit 22 in 2024). The FGM measures magnetic fields at a native cadence of 256 samples/second. We use data from November 2018, during the first "encounter" (E1) of PSP (Bale et al., 2019; Kasper et al., 2019). Encounter data are typically at the highest resolution. A 17-day gap-free interval from 2018-11-01 to 2018-11-18 was selected, which contained no missing observations after performing down-sampling, justified in the following section. This final gap-free time series consisted of 1,950,000 points for each vector component *Bx, By, Bz*, all three of which were used here. This series was then split into 195 vector time series intervals of length 10,000.

2.1 Input preparation

The data needed to be prepared before training a neural network to make it easier for the ANN to process data from different sources and intervals. We start with a set of time series intervals with 100% of measurements available, following the data normalization described below. 80% of intervals were used to train and validate the ANN, and the remaining 20% (39) were used for testing.

Data normalization: The timescales and magnitudes of interest vary significantly from one system to another. For example, solar wind in the inner heliosphere has magnetic field amplitudes in the $\sim 100nT$ range and a correlation time of $\approx 600s$ (Parashar et al., 2020; C. Chen et al., 2020), whereas the solar wind at 1AU has magnetic field amplitudes in the $\approx\ 10nT$ range and a correlation time of $\approx\ 1hour$ (Isaacs et al., 2015; Jagarlamudi et al., 2019). On top of this variability, the time cadences of various instruments differ significantly. In order to train the ANN in a system-agnostic way, we normalized both the x-values (time series) and y-values (the fluctuation amplitudes) using the following methods. The time series were normalized by down-sampling to have 10,000 samples across $\approx 15t_{corr}$ so that the training series has a sampling rate of $\delta t \sim 1.5 \times 10^{-3} T_{corr}$. Therefore, the time cadence for each interval was chosen so as to have $\approx 15t_{corr}$ sampled by 10,000 points. The amplitudes were normalized by subtracting the mean value μ_a from each value of each interval and then dividing by the standard deviation σ_a:

$$a(t)_{norm} = \frac{a(t) - \mu_a}{\sigma_a} \tag{3}$$

Gapped series preparation: Each "good" time series of magnetic field is used to create 15 "bad" copies by removing random chunks of data. By having the same expected output for each of these copies, the aim here was to make the ANN indifferent to where and how large the gaps are, as well as giving us more data train on. Between 0% and 50% of the data were removed from each interval in between 3 and 20 segments at random locations in the time series. The total percentage of data to remove and the number of chunks in which to remove data were generated randomly, and 15 bad copies were created for each good copy. This provided us with 2340 series of bad data for training and validating the model. For the 39 test intervals, these were copied 5 times each and the gap percentage was expanded to be between 0% and 95% removed. This was done to test the algorithm's performance on missing data *in general*, rather than just the gapped percentages it was trained on. This also allows us to test its performance 'in the limit', i.e. right up to only 5% data remaining, which will help us assess overfitting of the model.

pics/ml_pipeline_schematic_v2.drawio.pdf

Figure 1. Diagram showing the workflow for adding artificial gaps and producing different structure function estimations from each interval.

ANN input: 95% of the 2223 training series were used for training the ANN and 5% (117) for validating the ANN during the training process. The input for the ANN training program was a 3x10,000 array representing all of the three vector components (Bx, By, Bz), with the missing values replaced with zeros after re-normalization (essentially mean-imputing the fluctuation series). This zero padding implies artificial information insertion, hence we tried informing the ANN of this by training it with input $(Bx, By, Bz, mask)$, where $mask$ is an array of ones with zeros at locations of the gaps, thus extending each input to a dimension of 4x10,000. (As discussed in the results, what was found was that the addition of this mask in fact degraded the performance of the network, therefore, the results presented are for unmasked data.)

ANN output: The corresponding expected output for the ANN is the second-order S^{fn} computed from the corresponding good time series. The S^{fn}s were computed up to a maximum lag of 20% of the data size ($n_{lag} = 2000$) (Matthaeus & Goldstein, 1982). The good and bad time series are arranged in random order to create the input matrix, with the appropriate expected outputs.

Benchmarking the results: The second-order S^{fn}s computed by the ANN are compared to the S^{fn}s computed three ways: i) ignoring the gaps, ii) mean imputing the gaps, and iii) linearly interpolating the gaps. Fig. 1 shows the workflow of training the network. Table 1 details the final datasets used for training and evaluating the model.

Data source	Purpose	Input lengths	Output lengths (n_{lag})	% of each input removed	No. of instances
PSP	Training set	30000	2000	0-50	2223
	Validation set	30000	2000	0-50	117
	Test set	30000	2000	0-95	195

Table 1. Dimensions of data used to build and evaluate the neural network model. Inputs lengths are correct for the training runs with no indicator vector - those with the vector have length 40,000 instead. 'No. of instances' refers to the count of intervals in the set after duplication of the original unique intervals.

3 Model training and testing

Using a feedforward neural network, a multi-output regression model was built in Python using the Tensorflow package (Abadi et al., 2015). The workflow to ensure a good model fit was the following:

1. Train the model until the early-stopping criterion is reached (see below)
2. Evaluate the model on the test data, checking for overfitting and underfitting by visual inspection of the predictions
3. Adjust the model hyperparameters (number of hidden layers and/or number of nodes)
4. Repeat 2-3 until a good fit is achieved
5. Compare final model predictions with output from other, non-ML methods

The loss function used to calculate the error for this network was the Mean Squared Error, or MSE. The overall error for one epoch of the network $MSE_{overall}$ is calculated as the MSE for a single instance MSE_i, averaged over all the instances (2223 for the PSP training set). (One epoch is one iteration through every instance in the training set.)

$$MSE_i = \frac{1}{n_{lag}} \sum_{j=1}^{n_{lag}} (S_{2ij,pred} - S_{2ij,true})^2 \quad (4)$$

where $S_{2ij,pred}$ is the predicted value of the second-order S^{fn} for the i^{th} interval at lag j, $S_{2ij,true}$ is the corresponding 'ground truth' value, and n_{lag} is the number of lags for which the S^{fn} has been computed. As is standard for neural networks, this value is minimised through the process of backpropagation of error via gradient descent, and each weight and bias is adjusted according to the learning rate and the weight's contribution to the overall error, calculated using partial derivatives.

For each epoch of training, the training loss and the validation loss were used to check whether the model is still improving. (A sustained increase in the validation loss indicates that the model is beginning to overfit.) Accordingly, training was stopped when the validation loss was reduced by no more than 0.01 over 10 epochs, which we call the EarlyStopping criterion.

The S^{fn} values decrease by a few orders of magnitude going from large to small lags. This could potentially bias the MSE on large-scale predictions. Hence, the mean squared error (MSE) on the test set was not be used in isolation to evaluate this particular regression problem. It was complemented with the mean absolute percentage error (MAPE) to quantify the model's performance.

$$MAPE_i = \frac{1}{n_{lag}} \sum_{j=1}^{n_{lag}} \left| \frac{S_{2ij,true} - S_{2ij,pred}}{S_{2ij,true}} \right| \quad (5)$$

where $S_{2ij,pred}$ is the predicted value of the second-order S^{fn} for the i^{th} interval at lag j, $S_{2ij,true}$ is the corresponding 'ground truth' value, and n_{lag} is the number of lags for which the S^{fn} has been computed.

3.1 Optimising the network: selection of hyperparameters

Part of the challenge of using neural networks is finding the optimal structure or 'architecture' so as to get the best performance on the test set. A step-wise trial-and-error approach was used to approximately find the best architecture instead of using an automated determination of an optimal architecture (R. Luo et al., 2018; Koza & Rice, 1991). This involved iterating through different numbers of both hidden layers and the nodes in each hidden layer. In principle a single hidden layer is sufficient to approximate any continuous function (a 'universal approximator' (Cybenko, 1989)), but the capacity of the network increases as the number of hidden layers and nodes increases. The number of layers and nodes was slowly increased until the model appeared to be overfitting. A Rectified Linear Unit (ReLU) activation function was used for each hidden layer, and linear activation was used for the output layer (Ramachandran et al., 2017). The Adam Optimizer, a common adaptive optimizer that automatically adjusts the learning, was used with the default value of 0.001. (Higher values of the initial value of this optimizer were found to degrade the performance of the model.) The number of training epochs was controlled by the EarlyStopping criteria, as described in the previous section.

4 Results

Given the requirement to fit the correct shape of the predicted outputs without regression to the mean curve, a visual inspection of the predicted curves against the expected curves for a sample of test intervals was required. After running several different iterations it was found that 10 or more layers always lead to regression to the mean, where virtually the same curve was predicted for every input interval (Bello, 1992). The best network configuration from those that were tried - i.e., that which produced smooth curves with shapes that at least partly matched the different shapes of the expected outputs - was one with 2 hidden layers, each with 10 nodes. A schematic diagram of this configuration is shown in Figure 2. We note that the input and the output layers require the largest number of parameters, 300,010 and 22,000, respectively in this case. Any additional hidden layers having order 10 neurons contribute order 100 parameters only to the list of trainable parameters. This final configuration was then applied to four different variations of the input data: mean-imputed and linearly interpolated gapped intervals, each with and without the additional missing data indicator vector. Ultimately, the linearly interpolated data with no missing data indicator gave the best results for this network configuration. This configuration was run three times to check the results, because slightly different results will occur each time due to different initial randomisation of the weights. The EarlyStopping criterion led to training being stopped after between 25 and 33 epochs. The mean test error on the PSP test intervals was 3.76. It is important to note the unusual machine learning workflow used here: this test error was not the best values out of all the network configurations attempted. The conventionally used measure of model performance($MSE_{overall}$) is a very limited measure because it can easily regress to the mean while trying to reduce the $MSE_{overall}$, given the range of variation in individual expected outputs. Further investigations into designing robust measures of model performance when predicting arrays of logarithmically spaced values is needed.

Once a good network configuration and input data structure were found, its approximation ability was compared with the alternative interpolation methods discussed previously, as well as naive calculation of the structure functions directly from the gapped intervals.. We start with a case study in Figure 3 of two versions of two unique intervals, with each version representing different gaps removed from the original interval.

In the first interval, 1a, with 10% data missing, the ANN output does not improve upon the S^{fn}s calculated from the mean-imputed or linearly-interpolated intervals, both of which stay very close to the true curve. As the amount of missing data increases to 43% in interval 1b, the S^{fn} calculated from the gapped interval diverges significantly from the true S^{fn}. However, the other estimations stay relatively close to the true curve, with the L-INT curve in particular remaining a very good estimation. The ANN prediction shows undesired fluctuations in log-log space at smaller lags, similar to the prediction for 1a.

Interval 2a, with 37% data missing, shows an interval with an ANN prediction that is significantly smoother than and superior to the S^{fn} calculated from the gapped interval, but is inferior to the simpler methods of imputation and interpolation, both of which have a shape that is closer to that of the true curve. In spite of this, the S^{fn} at large lags is better predicted by the ANN. As mentioned before, given that the S^{fn} decreases by a few orders of magnitude from large to small lags, small lags contribute very little to MSE. Hence, it is expected that such predictions could produce better MSE performance for the ANN when large amounts of data are missing. For the extreme data loss example in interval 2b, with 87% missing data, the ANN prediction is clearly the best at getting close to the true S^{fn}, especially at large lags. This foreshadows the ultimate conclusion on the utility of the neural network, which seems to perform best, relative to other methods, when dealing with large data loss.

The performance of each method, including the S^{fn}s calculated from the un-filled gapped intervals, was evaluated using the following measures:

- Average MSE ($MSE_{overall} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$) across all expected-observed S^{fn} pairs (recall this was the loss function used to train the neural network - see Equation 4).

pics/neural_net_diagram_final.png

Figure 2. Schematic diagram of the final ANN architecture. The inputs consist of 3 x 10,000 stacked vector components, and the outputs consist of 1 x 2,000 second-order structure functions. The number of hidden layers and nodes were chosen by iteratively increasing the size of each hyperparameter and visually inspecting the predictions made by the trained network for each architecture.

- Average MAPE ($MAPE_{overall} = \frac{1}{n}\sum_{i=1}^{n} MAPE_i$) across all expected-observed S^{fn} pairs. This loss function is easier to interpret than the MSE in terms of relative error and is scale-independent. It is also more stringent than MSE because it evaluates the method's performance in predicting both the large-scale and small-scale values. Both $MSE_{overall}$ and $MAPE_{overall}$ are given in Table 2.
- Scatterplots and corresponding linear regression lines of MSE and MAPE against % data missing of individual test intervals. This shows us the how each method performs with increasing data loss shown in Figures 4 and 5.

Before discussing the results, it is important to recall the pipeline of S^{fn} inputs and outputs. As shown in Figure 1, we calculate S^{fn}s from the original clean series (true S^{fn}), the gapped series, the mean-imputed series and the linear interpolated series us-pics/PSP_case_studies_plot_final.pdf

Figure 3. Results of the four different S^{fn}approximation methods for two unique (normalized) PSP intervals that have been gapped in two different ways. (For simplicity, only one of three vector components for each input interval has been shown, but this still illustrates the number and size of the gaps, which were consistent between components.) Note that the log-log plot emphasises differences between the curves at small lags (a.k.a., high frequencies).

ing Equation 2. In accordance with our model development, we also use the linearly interpolated series as input to the neural network model to produce our fourth estimate of S^{fn} to compare with the true S^{fn}.

The overall results shown in Table 2, suggest that, for overall average performance, the GAPS (calculation from gapped series) method is the worst S^{fn} approximation method across the board with $MSE \approx$ 6700 and $MAPE =$ 4.68. L-INT (calculation for linearly interpolated series) performs the best with $MSE =$ 2.98 and $MAPE =$ 0.28. The relative performance of M-IMP (calculation from mean-imputed series) and ANN vary between evaluation methods. When using MSE, ANN is the second best approximation method, but when using MAPE it is third best, behind M-IMP.

Spacecraft	Performance measure	SF calculated from GAPS	SF calculated from M-IMP	SF calculated from L-INT	SF estimated using ANN
PSP	MSE	6713.49	7.53	**2.98**	3.37
	MAPE	4.68	0.45	**0.28**	1.66

Table 2. Calculated performance measures for each S^{fn}approximation method. **Bolded figures are the lowest of each row.** GAPS: Gapped interval with no imputation. M-IMP: Gapped interval with mean (0) imputation. L-INT: Gapped interval with linear interpolation of gaps. ANN: Artificial neural network model.

pics/PSP_MSE_plot.pdf

Figure 4. Scatter plot of MSE against proportion of data removed for the PSP test interval with overlaid linear regression lines and confidence regions from other panels. The line for the GAPS method (no imputation) is not shown here as it quickly disappears from the plotting area, and Table 2 shows it is clearly inferior to the other methods.

However, these overall measures do not take into account the variation in their performance as a function of the degree of sparsity. Hence, we take a statistical approach to quantify the performance with increasing sparsity. Figure 4 shows the scatter plots of MSE versus percentage of missing data for each set of test intervals for each method, overlain with linear regression lines of best fit. As seen in the middle two panels, the MSE tends to increase with increasing sparsity for M-IMP and L-INT. This is to be expected: as the amount of data missing increases, the S^{fn} estimation gets worse for these methods. However, there is distinct funnelling on the plots, representing heteroskedasticity or unequal variance in MSE for different proportions missing. For low % missing values, around less than 20% missing, there is a very small range of MSE values for the imputation methods. This means that the accuracy of the S^{fn} estimations do not vary much for small percentage of missing data. On the other hand, as the amount of missing data increases, not only does the average error increase, but also the variation in error. This shows that intervals with large amounts of data missing can, in some cases, produce S^{fn}s quite similar to the true S^{fn}s, if imputation is performed. This is especially true for intervals that have been linearly interpolated - we can see in the L-INT scatterplot that there are intervals with up to 90% missing that have very low MSE values. This is due to the importance of not only the size of the gaps, but where the gaps are in the interval: if the removed data does not significantly depart from the overall trend, linear interpolation will result in a S^{fn} not very different from the expected curve. On the other hand, M-IMP does not show similarly low MSE values beyond about 45% missing, though there is still distinctly increasing variance.

In stark contrast to the M-IMP and L-INT methods, the MSE of the ANN model predictions are largely indifferent to the proportion of data missing. There is a constant band of MSE values across this scatterplot, and the Pearson correlation coefficient, a measure of the linear association between two variables, is very close to 0 for this method (-0.06).

As a way of establishing the comparative usefulness of each method, linear regression lines were fit to the data. Although fitting a linear model is inappropriate for this data due to the unequal variance present in the M-IMP and L-INT methods, it still provides a useful indicator of the typical values of MSE for different proportions missing. What really distinguishes these methods, as shown in Figure 4, are the slopes of the regression lines. MSE increases the fastest for the M-IMP method, and this association is also that with the highest correlation between MSE and % missing (0.87). Next is the L-INT method, and this association has the next highest correlation (0.70). Finally, the ANN has a very flat slope with a correlation close to 0. This result shows that the ability of the neural network to approximate the true S^{fn} is much less affected by increasing amounts of missing data than the other two methods. However, this does not make it the best S^{fn} approximation method for any bad dataset. What we can see in Figure 4 is that up to about 50% missing data, the simple imputation methods have lower typical values of MSE than the neural network. At values greater than 50%, the neural network, on average, produces the best estimations of the three approaches, according to the MSE metric. This, however, could be a result of the dominance of large lag values of the S^{fn} controlling the MSE.

To quantify the performance of a method to predict not only the large lag value but also the small lag values, we use the mean amplitude percent error (MAPE). The MAPE measure, shown in Figure 5, shows much less heteroskedasticity for the L-INT and M-IMP methods, with M-IMP in particular showing a much stronger linear relationship between % missing and MAPE, producing a correlation of 0.98. The ANN method has a small positive slope and a relatively constant band of scatter, with correlation of 0.04. Overlaying the regression lines in Figure 5, we see that the neural network linear regression line remains above that of both other methods for all gap percentages. Based on these findings, it seems reasonable to use L-INT for data gaps as large as $\sim 20\%$.

Overall, we find that the ANN is largely indifferent to the proportion of data missing in its approximations of the true S^{fn}. However, this means that the worst predictions for inputs with little or no data missing are as bad as those for inputs with at least 90% data missing.

pics/PSP_MAPE_plot.pdf

Figure 5. Scatter plot of MAPE against proportion of data removed for the PSP test intervals with overlaid linear regression lines and confidence regions. The line for the GAPS method (no imputation) is not shown here as it quickly disappears from the plotting area shows it is clearly inferior to the other methods.

5 Conclusions

Gaps are a common problem in almost all fields that deal with time series. The field is mature with many ways of filling the gaps, including mean-imputation, linear interpolation, maximum likelihood estimation of ARIMA models (Velicer & Colby, 2005), singular spectrum analysis (Schoellhamer, 2001), and artificial neural networks (ANNs) (Comerford et al., 2015). Our interest here is not in prediction, but in gleaning the best estimates of statistics of the system without having to reconstruct the time series. In particular, we studied the potential of simple feedforward artificial neural networks to predict turbulent statistics of solar wind magnetic field measurements. In space plasma physics, an accurate description of the S^{fn} in the inertial range of the structure function is desirable. This is particularly important to estimate not only the slope of the structure function in the inertial range, but also to estimate the inner and outer scales of turbulence.

Starting with "good" time series with 100% coverage, we created "bad" time series for which the second-order structure functions were estimated in four ways: i) direct computation ignoring gaps, ii) mean imputation of the gaps, iii) linear interpolation across the gap, and iv) a trained ANN.

ANNs do not seem to be the panacea that one might naively hope for in such a situation. As reflected by the error functions of MSE and MAPE, the ANN seems to somewhat learn to estimate the large-scale values of the structure function. This is not very surprising as the large lag S^{fn} values approach the mean-squared value of the fluctuation amplitudes. Given the trend in error with increasing data loss, the ANN is more useful for large portions of missing data, but over the entire range of data loss it tends to perform worse than simpler methods. However, it is worth noting that with only 20 neurons (and about 322,000 trainable parameters), ANN performs comparably to L-INT or M-IMP methods, and with MSE as the cost function it even outperforms these two methods for large gaps in the data. The case of MAPE as cost function is not very attractive but that may be improved by either changing the training strategy or the network architecture.

Our results indicate that to achieve a reasonable description of turbulent statistics for gapped time series, one needs to go beyond simple-minded feedforward ANNs. Possible improvements to ANNs could include grey-box modeling with turbulence physics incorporated into the input, and more advanced architectures such as LSTM networks or autoencoders.

It may also be the case that even the performance of this simple feedforward ANN structure could be improved through better optimisation of the model weights, biases, and hyperparameters (in particular, the number of hidden layers and nodes). To this end, a reliable method of avoiding over-fitting when trying to predict the shape of a curve, rather than a scalar output, is an important issue to address. This will be explored in follow-up studies, along with other ways of representing the missing data when feeding it into the network.

6 Author contributions

TNP came up with the project idea, DW performed the analysis and created the figures, RR, MF, and RKS provided guidance on ANNs. All authors discussed the results and contributed to manuscript writing.

Acknowledgments

This research was seeded by funding from a Summer Research Scholarship provided by Victoria University of Wellington. We would like to acknowledge the PSP instrument teams for high quality measurements in the inner heliosphere and the Space Physics Data Facility (SPDF) at the Goddard Space Flight Center for providing access to all the data used for this project. The analysis codes are available in a GitHub repository by request. The work of RKS is partially supported by SERB, DST, Government of India through the project EMR/2017/002778.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.*

Retrieved from https://www.tensorflow.org/
Bale, S., Badman, S., Bonnell, J., Bowen, T., Burgess, D., Case, A., ... others (2019). Highly structured slow solar wind emerging from an equatorial coronal hole. *Nature*, *576*(7786), 237–242.
Bale, S., Goetz, K., Harvey, P., Turin, P., Bonnell, J., Dudok de Wit, T., ... others (2016). The fields instrument suite for solar probe plus. *Space science reviews*, *204*(1), 49–82.
Batchelor, G. K. (1953). *The theory of homogeneous turbulence.* New York: Cambridge University Press.
Bavassano, B., Dobrowolny, M., Mariani, F., & Ness, N. F. (1982). Radial evolution of power spectra of interplanetary Aflvenic turbulence. *Journal of Geophysical Research*, *87*(A5), 3617-3622.
Bello, M. (1992). Enhanced training algorithms, and integrated training/architecture selection for multilayer perceptron networks. *IEEE Transactions on Neural Networks*, *3*(6), 864-875. doi: 10.1109/72.165589
Biskamp, D. (2003). *Magnetohydrodynamic turbulence.* Cambridge University Press.
Broersen, P. M. (2006). Automatic spectral analysis with missing data. *Digital Signal Processing: A Review Journal*, *16*, 754-766. doi: 10.1016/j.dsp.2006.01.001
Brown, T. M., & Christensen-Dalsgaard, J. (1990). A technique for estimating complicated power spectra from time series with gaps. *The Astrophysical Journal*, *349*, 667-674.
Burlaga, L. (1991). Intermittent turbulence in the solar wind. *Journal of Geophysical Research*, *96*(A4), 5847-5851.
Chen, C., Bale, S., Bonnell, J., Borovikov, D., Bowen, T., Burgess, D., ... others (2020). The evolution and role of solar wind turbulence in the inner heliosphere. *The Astrophysical Journal Supplement Series*, *246*(2), 53.
Chen, Y., Kopp, G. A., & Surry, D. (2002). Interpolation of wind-induced pressure time series with an artificial neural network. *Journal of Wind Engineering and Industrial Aerodynamics*, *90*, 589-615.
Comerford, L., Kougioumtzoglou, I. A., & Beer, M. (2015, 1). An artificial neural network approach for stochastic process power spectrum estimation subject to missing data. *Structural Safety*, *52*, 150-160. doi: 10.1016/j.strusafe.2014.10.001
Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*, 303-314.
de Souza Echer, M. P., Echer, E., Domingues, M. O., Mendes, O., Seo, R. T., & Gonzalez, W. (2021, 7). Wavelet analysis of low frequency magnetic field fluctuations in the Jupiter's magnetotail. *Advances in Space Research*, *68*, 246-258. doi: 10.1016/j.asr.2021.03.003
Fox, N., Velli, M., Bale, S., Decker, R., Driesman, A., Howard, R., ... others (2016). The solar probe plus mission: Humanity's first visit to our star. *Space Science Reviews*, *204*(1-4), 7–48.
Fraternale, F., Pogorelov, N. V., Richardson, J. D., & Tordella, D. (2019). Magnetic Turbulence Spectra and Intermittency in the Heliosheath and in the Local Interstellar Medium. *The Astrophysical Journal*, *872*(40).
Frick, P., Grossmann, A., & Tchamitchian, P. (1998). Wavelet analysis of signals with gaps. *Journal of Mathematical Physics*, *39*, 4091-4107. doi: 10.1063/1.532485
Friedrich, J., Gallon, S., Pumir, A., & Grauer, R. (2020). Stochastic interpolation of sparsely sampled time series via multipoint fractional brownian bridges. *Physical Review Letters*, *125*. doi: 10.1103/PhysRevLett.125.170602
Gallana, L., Fraternale, F., Iovieno, M., Fosson, S. M., Magli, E., Opher, M., ... Tordella, D. (2016). Voyager 2 solar plasma and magnetic field spectral analysis for intermediate data sparsity. *Journal of Geophysical Research: Space*

Physics, *121*, 3905–3919.
Harvey, A. C., & Pierse, R. G. (1984). Estimating missing observations in economic time series. *Source: Journal of the American Statistical Association*, *79*, 125-131.
Isaacs, J., Tessein, J., & Matthaeus, W. (2015). Systematic averaging interval effects on solar wind statistics. *Journal of Geophysical Research: Space Physics*, *120*(2), 868–879.
Jagarlamudi, V. K., de Wit, T. D., Krasnoselskikh, V., & Maksimovic, M. (2019). Inherentness of non-stationarity in solar wind. *The Astrophysical Journal*, *871*(1), 68.
Jang, J., Choi, K., Roh, H., Son, S., Hong, C., Kim, E., ... Yoon, D. (2020). Deep learning approach for imputation of missing values in actigraphy data: Algorithm development study. *JMIR Mhealth Uhealth*, *8*.
Kasper, J. C., Bale, S. D., Belcher, J. W., Berthomier, M., Case, A. W., Chandran, B. D., ... others (2019). Alfvénic velocity spikes and rotational flows in the near-sun solar wind. *Nature*, *576*(7786), 228–231.
Kondrashov, D., Shprits, Y., & Ghil, M. (2010, 8). Gap filling of solar wind data by singular spectrum analysis. *Geophysical Research Letters*, *37*. doi: 10.1029/2010GL044138
Koza, J., & Rice, J. (1991). Genetic generation of both the weights and architecture for a neural network. In *Ijcnn-91-seattle international joint conference on neural networks* (Vol. ii, p. 397-404 vol.2). doi: 10.1109/IJCNN.1991.155366
Ludwig, R., & Taylor, J. (2016). Voyager telecommunications. In *Deep space communications* (p. 37-77). John Wiley & Sons, Ltd. doi: https://doi.org/10.1002/9781119169079.ch3
Luo, R., Tian, F., Qin, T., Chen, E., & Liu, T.-Y. (2018). Neural architecture optimization. In *32nd conference on neural information processing systems.*
Luo, Y., Cai, X., Zhang, Y., Xu, J., & Xiaojie, Y. (2018). Multivariate time series imputation with generative adversarial networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31). Curran Associates, Inc.
Magrini, L. A., Domingues, M. O., & Mendes, O. (2017). On the effects of gaps and uses of approximation functions on the time-scale signal analysis: A case study based on space geophysical events. *Brazilian Journal of Physics*, *47*, 167-181.
Makarynskyy, O., Makarynska, D., E, R., & A, G. (2005). Filling gaps in wave records with artificial neural networks. *International Maritime Association of the Mediterranean. International Congress (12th : 2005 : Lisbon, Portugal).*
Matthaeus, W., & Goldstein, M. (1982). Measurement of the rugged invariants of magnetohydrodynamic turbulence in the solar wind. *Journal of Geophysical Research*, *87*(A8), 6011–6028.
Munteanu, C., Negrea, C., Echim, M., & Mursula, K. (2016, 4). Effect of data gaps: Comparison of different spectral analysis methods. *Annales Geophysicae*, *34*, 437-449. doi: 10.5194/angeo-34-437-2016
Panchev, S. (1971). *Random functions and turbulence.* Elsevier.
Parashar, T. N., Goldstein, M. L., Maruca, B. A., us, W. H. M., Ruffolo, D., Bandyopadhyay, R., ... Raouafi, N. (2020, feb). Measures of scale-dependent alfvénicity in the first PSP solar encounter. *The Astrophysical Journal Supplement Series*, *246*(2), 58. doi: 10.3847/1538-4365/ab64e6
Pavlova, O. N., Abdurashitov, A. S., Ulanova, M. V., Shushunova, N. A., & Pavlov, A. N. (2019, 1). Effects of missing data on characterization of complex dynamics from time series. *Communications in Nonlinear Science and Numerical Simulation*, *66*, 31-40. doi: 10.1016/j.cnsns.2018.06.002
Podesta, J., Roberts, D., & Goldstein, M. (2007). Spectral exponents of kinetic and magnetic energy spectra in solar wind turbulence. *The Astrophysical Journal*, *664*(1), 543.

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). *Searching for activation functions.*

Randolph-Gips, M. (2008). A new neural network to process missing data without imputation. In *2008 seventh international conference on machine learning and applications* (p. 756-762). doi: 10.1109/ICMLA.2008.89

Rehfeld, K., Marwan, N., Heitzig, J., & Kurths, J. (2011). Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, *18*, 389-404. doi: 10.5194/NPG-18-389-2011

Schoellhamer, D. H. (2001). Singular spectrum analysis for time series with missing data. *Geophysical Research Letters*, *0*.

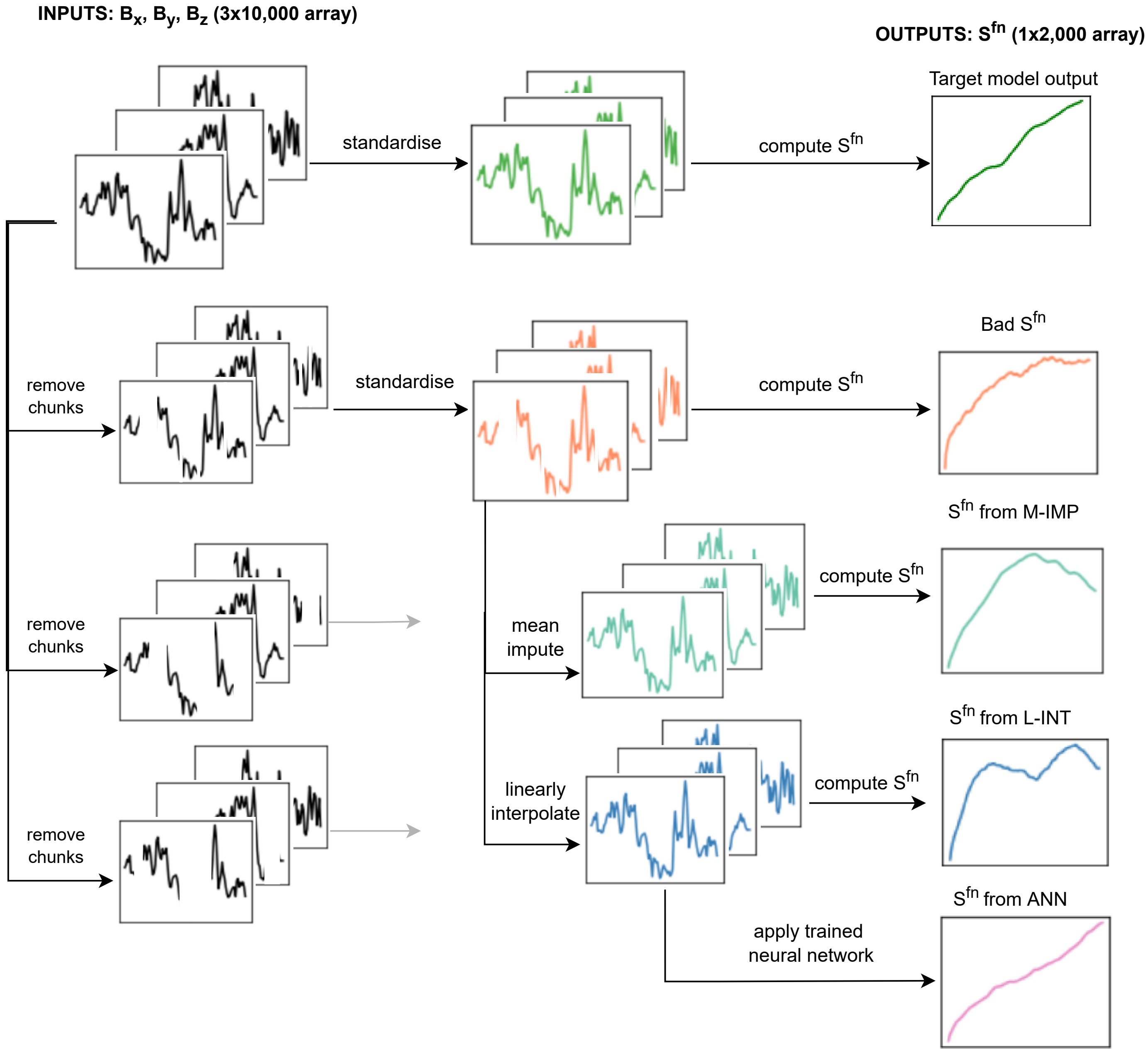
VanderPlas, J. T. (2018, 5). Understanding the lomb–scargle periodogram. *The Astrophysical Journal Supplement Series*, *236*, 16. doi: 10.3847/1538-4365/aab766

Velicer, W. F., & Colby, S. M. (2005, 8). A Comparison of Missing-Data Procedures for Arima Time-Series Analysis. *Educational and Psychological Measurement*, *65*, 596-615. doi: 10.1177/0013164404272502

Wu, P., Perri, S., Wan, M., Matthaeus, W. H., Shay, M. A., Goldstein, M. L., ... Chapman, S. (2013). Intermittent heating in solar wind and kinetic simulations. *The Astrophysical Journal Letters*, *763*.

Zhao, J., Lange, H., & Meissner, H. (2020, 5). Gap-filling continuously-measured soil respiration data: A highlight of time-series-based methods. *Agricultural and Forest Meteorology*, *285-286*. doi: 10.1016/J.AGRFORMET.2020.107912

Figure 1.



INPUTS: B_x, B_y, B_z (3x10,000 array)
OUTPUTS: S^{fn} (1x2,000 array)
standardise
compute S^{fn}
Target model output
remove chunks
standardise
compute S^{fn}
Bad S^{fn}
S^{fn} from M-IMP
mean impute
compute S^{fn}
remove chunks
S^{fn} from L-INT
linearly interpolate
compute S^{fn}
remove chunks
S^{fn} from ANN
apply trained neural network

Figure 2.

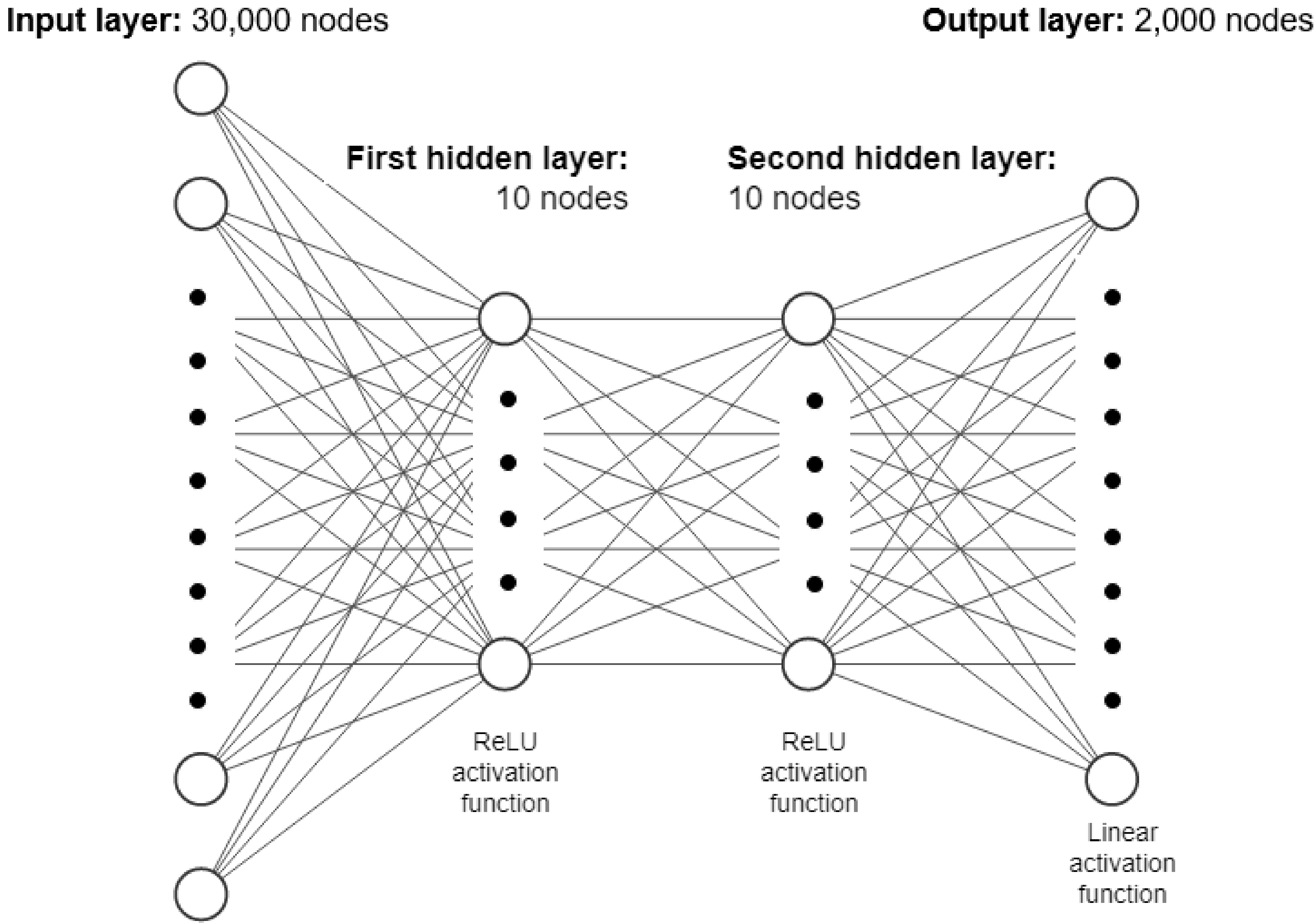
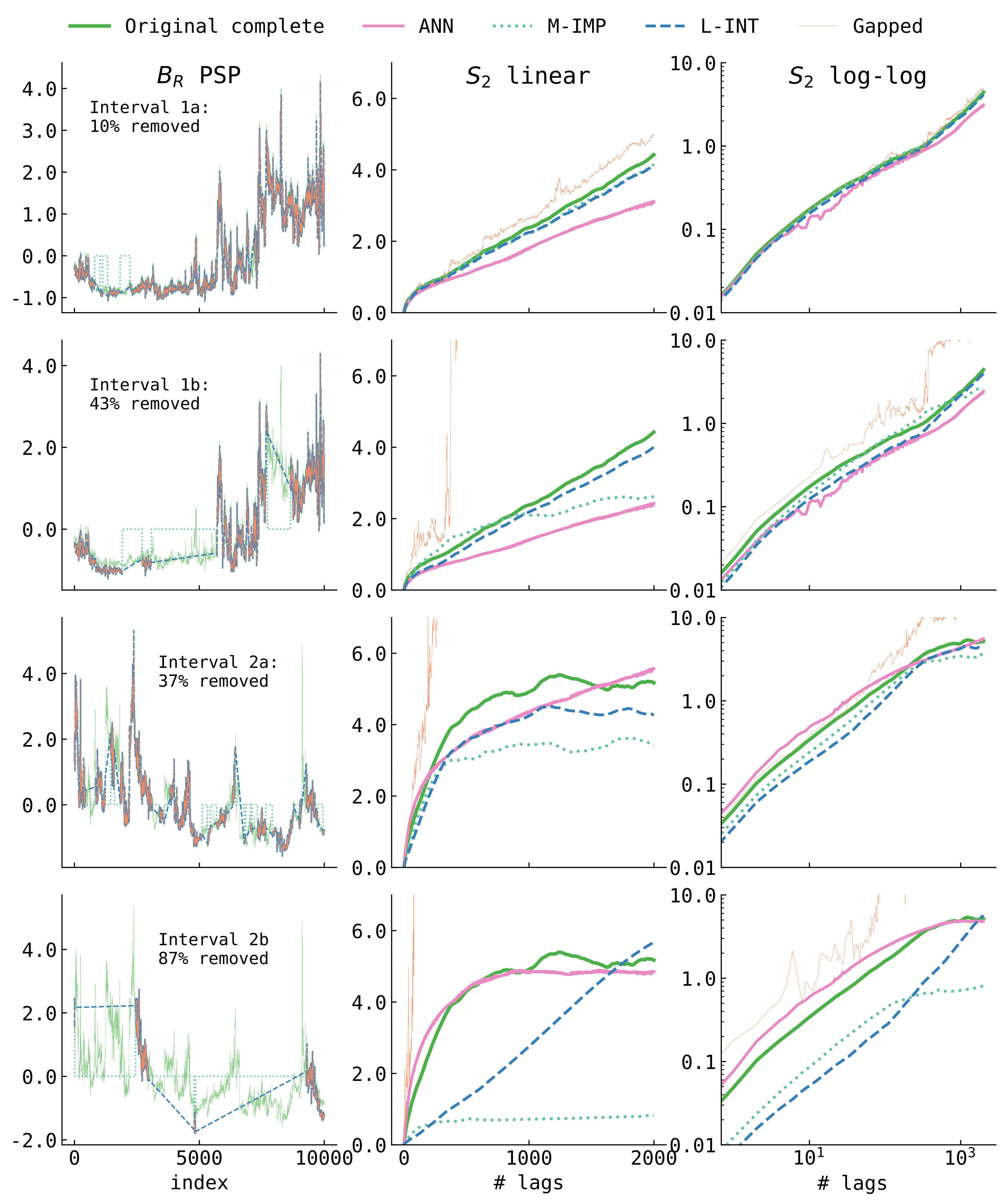

Input layer: 30,000 nodes
Output layer: 2,000 nodes
First hidden layer:
10 nodes
Second hidden layer:
10 nodes
ReLU activation function
ReLU activation function
Linear activation function

Figure 3.



Original complete
ANN
M-IMP
L-INT
Gapped
B_R PSP
S_2 linear
S_2 log-log
Interval 1a:
10% removed
Interval 1b:
43% removed
Interval 2a:
37% removed
Interval 2b
87% removed
index
lags
lags

Figure 4.

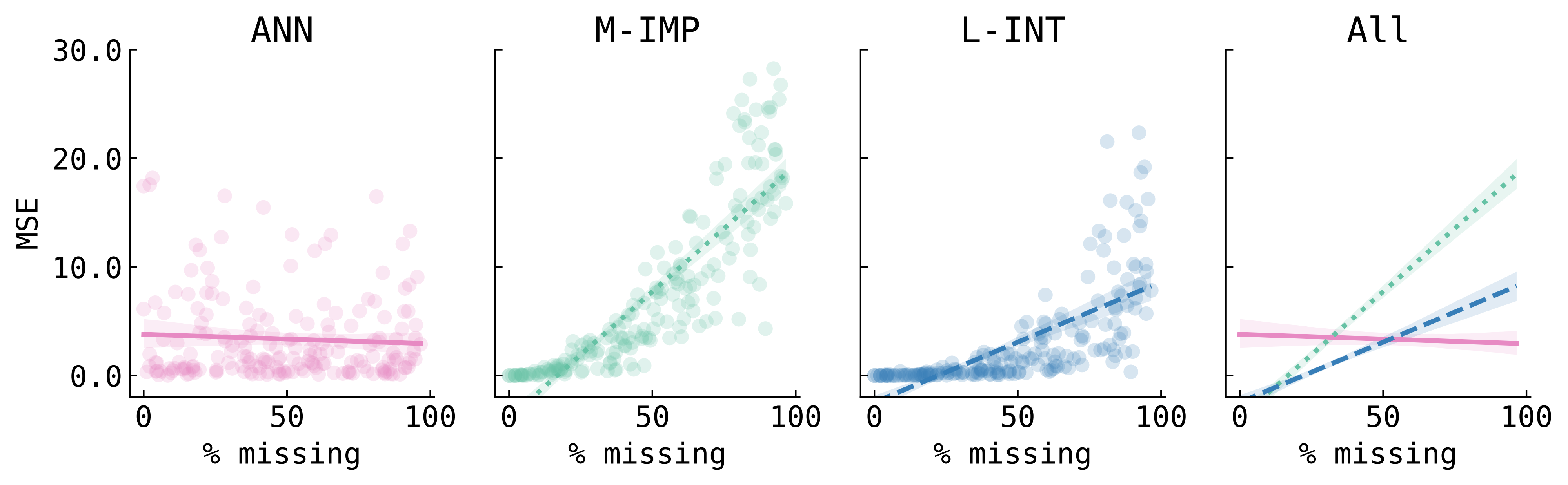
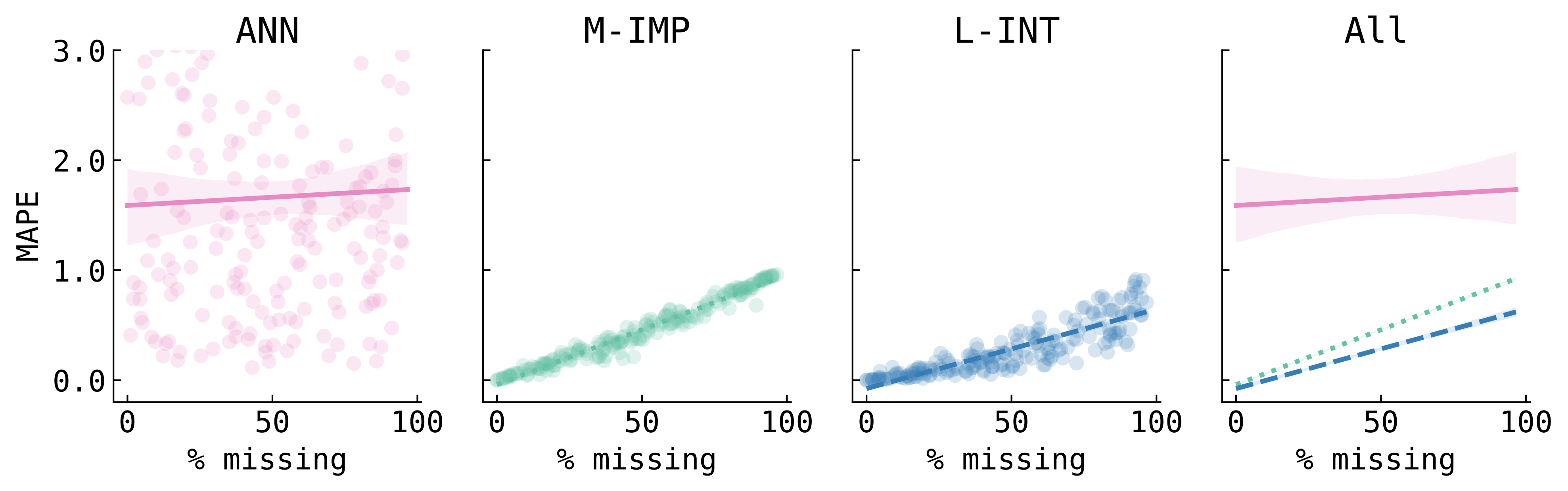

ANN
M-IMP
L-INT
All
MSE
30.0
20.0
10.0
0.0
0
50
100
% missing

Figure 5.



ANN
M-IMP
L-INT
All
MAPE
3.0
2.0
1.0
0.0
0
50
100
% missing