Automating the selection of preprocessing techniques for deep neural networks

Student: Marcus September

Supervisor: Francesco Sanna Passino

🏛 Department of Mathematics, Imperial College London

✉ marcus.september22@imperial.ac.uk

1. Problem and motivation

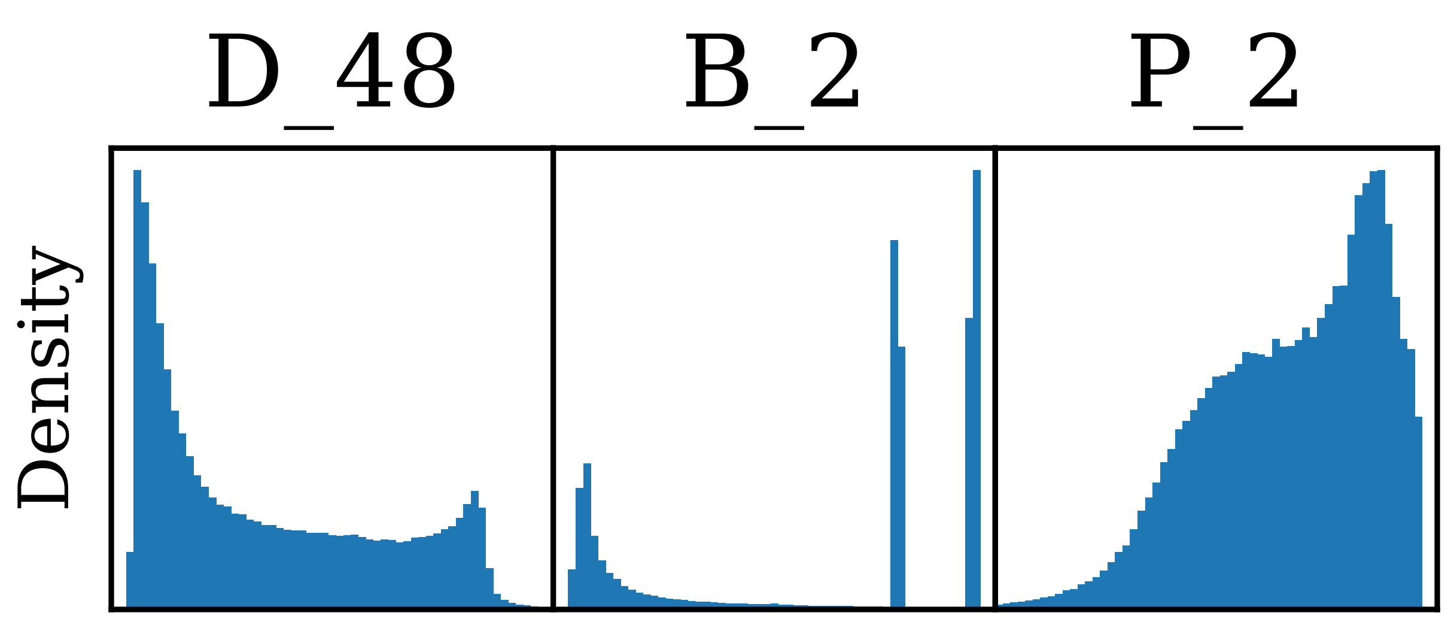
Deep learning sequence models, such as Recurrent Neural Networks (RNNs) and transformers, are sensitive to input variable distributions. Both training speed and performance can drop significantly for non-normal distributions, such as skewed distributions and those with outliers. Preprocessing includes all transformations applied to the data before feeding it into the neural network, and deciding the appropriate techniques is essential for optimising model performance. However, this is a time-consuming process. This project aims to automate this by automatically selecting the preprocessing methods to use for any given sequence dataset, increasing both model performance and training efficiency.

2. Default prediction dataset

The default prediction dataset, provided by American Express, contains a multivariate time-series from $N \approx 460\ 000$ different customers. Each time-series has $P = 188$ aggregated profile features recorded at up to $T = 13$ different credit card statement dates. For each multivariate time-series, the target label $y \in \{0, 1\}$ indicates whether the customer defaulted on their loan or not. The task is to predict the probability $\mathbb{P}(Y = 1)$ for each customer. Note that due to privacy concerns, the name of all the features have been anonymized. Additionally, a small amount of uniform noise has been added to all the numeric features.

3. Exploratory data analysis

Figure 1: ***Histogram of 3 of the 188 variables from the default prediction dataset.***



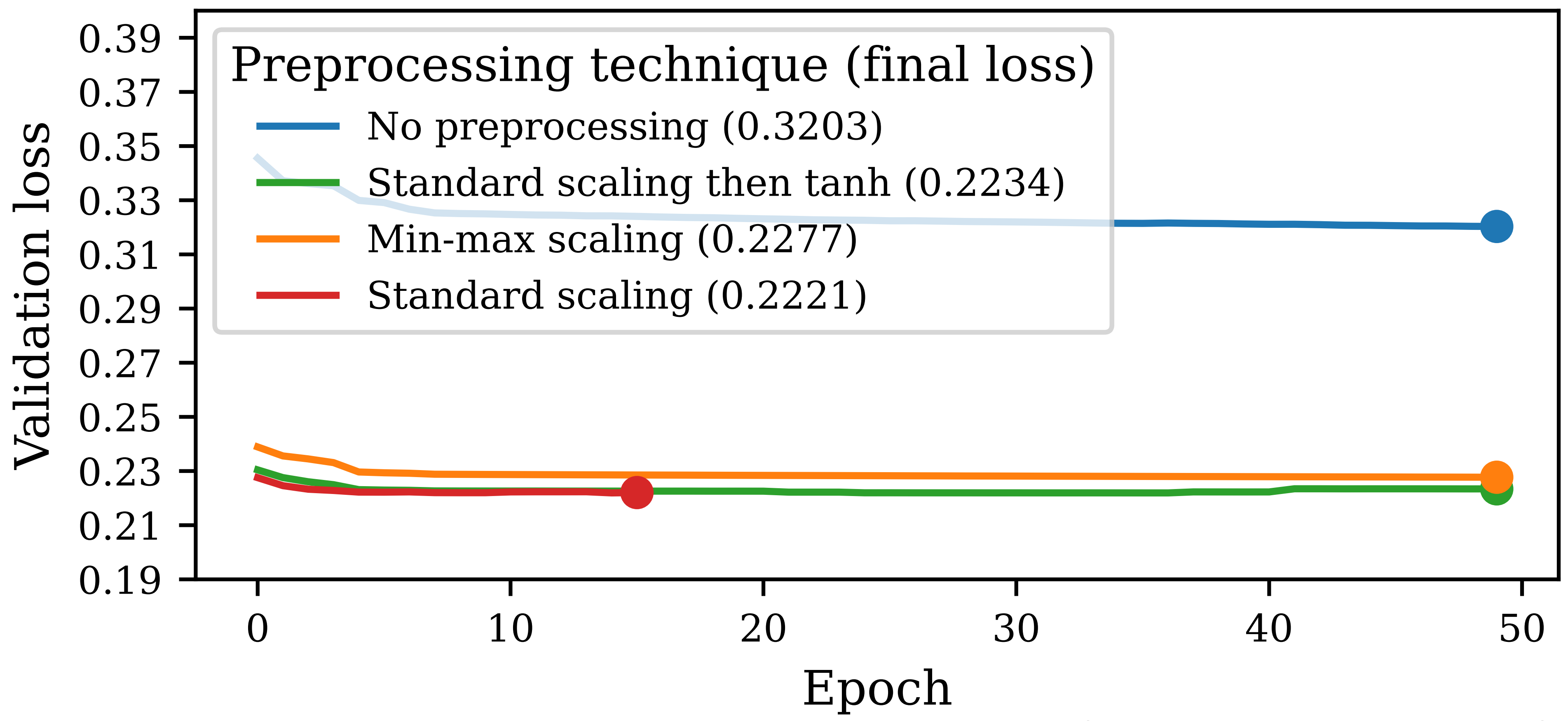
The default prediction dataset exhibits many traits of real-world datasets, such as very skewed distributions, multiple modes, unusual peaks and extreme values. Across the whole dataset, 8.50% of the numeric data points are missing. The 5 most incomplete variables are missing between 91.52% and 92.62% of the data points. Only 138 out of the 177 numeric variables have less than 1% missing values. All this makes the dataset ideal for evaluating how effective the proposed preprocessing techniques are on real-world data.

4. Synthetic data

With a custom-made synthetic data generation procedure, I can generate new labelled multivariate time-series where the variables follow arbitrary distributions based on the unnormalized PDFs I provide. This allows synthesizing data with real-world-like distributions in a controlled matter, which makes it easier and more efficiency to experiment and learn when each preprocessing method works best. This insight will then be used to automate the preprocessing step.

5. Preliminary result I: Preprocessing on real data

Figure 2: ***Average 5-fold cross-validation loss for different preprocessing techniques applied on the American Express default prediction dataset, using a GRU RNN model***



Applying an appropriate preprocessing technique before training can significantly improve performance. Additionally, with the right preprocessing technique the number of epochs required for convergence is reduced, as seen in figure 2.

6. Preliminary result II: Using synthetic data

Using the synthetic data generation procedure, two multivariate time-series datasets of dimensions $(N = 10\,000, T = 6, D = 2)$ were generated. The first one was configured to generate data with very skewed and irregular distributions, containing outliers, while the variables in the second dataset were all normally distributed.

Table 1: ***Performance metrics after training a GRU RNN model on synthetic data with specified variable distributions (Sample size 50, and results presented with a 95% CI)***

Variable distributions	Validation loss	AMEX metric	Binary accuracy
Non-normal	0.2209 ± 0.0183	0.7907 ± 0.0373	$90.53\% \pm 1.13\%$
Normal	0.1356 ± 0.0176	0.8683 ± 0.0242	$94.19\% \pm 1.12\%$

From table 1, we can conclude there exists suitable variable transformations that can be applied to the data to significantly increase performance.

7. Automating the preprocessing procedure

Figure 3: ***Proposition I: Automating preprocessing using heuristics and determining suitable static transformations for each variable based on these***

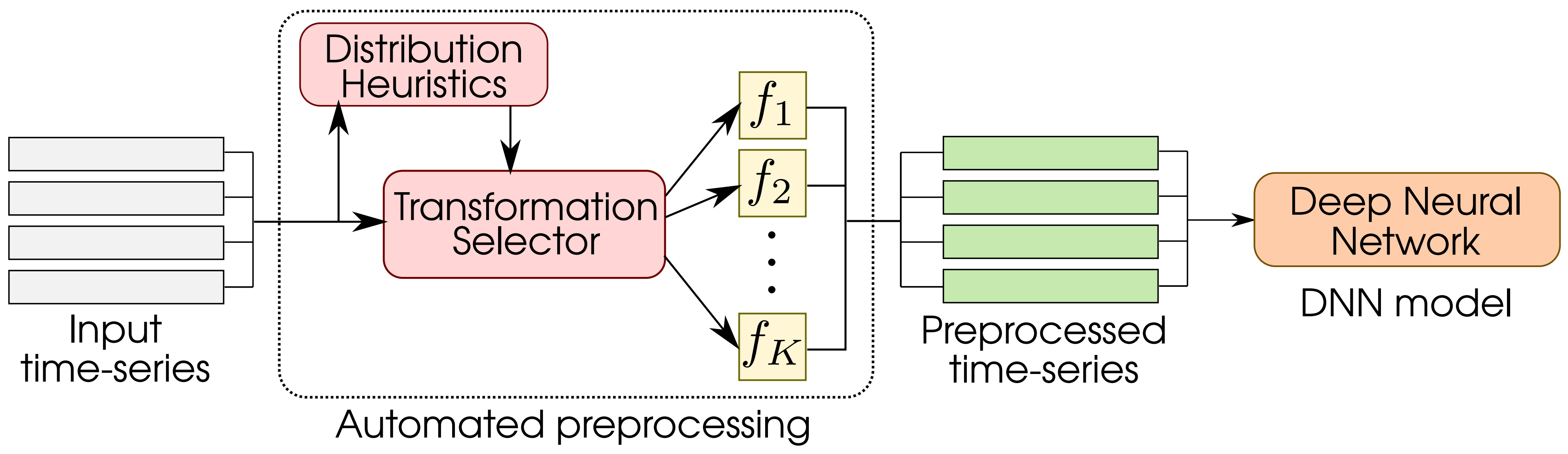
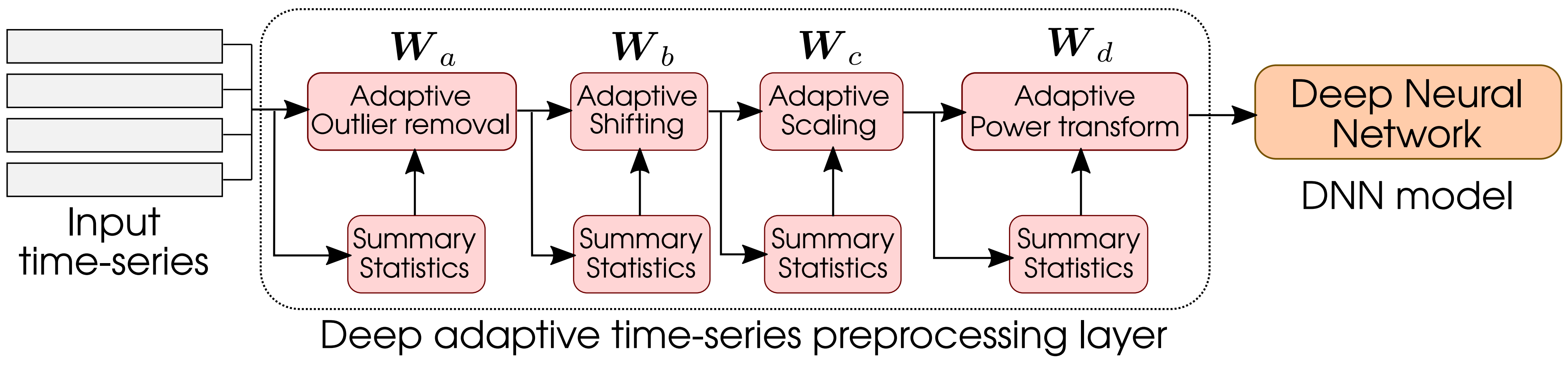


Figure 4: ***Proposition II: Automating preprocessing using an adaptive preprocessing layer with weights that are learned during training***



To automate the selection of preprocessing techniques, there are two methods I propose to investigate further. One option, illustrated in figure 3, is based on heuristics. With the second option, as shown in figure 4, the preprocessing methods are parameterized and learned as part of the training procedure.