Applying Transfer Learning to PhaseNet on a Small Continous Seismic Dataset from Iceland

ML-for-EPS III Report

Boyu Gao

Friday 28th June, 2024

Abstract

This report presents a study on the application of the PhaseNet deep neural network architecture for seismic event detection and prediction. Utilizing a continuous seismic dataset collected in Iceland, we explore the effectiveness of preprocessing techniques, particularly high-pass filtering when we know the range of ambient noise, and the impact of transfer learning from a pre-trained PhaseNet model. Our findings demonstrate that preprocessing significantly enhances the model's accuracy in predicting P-phase arrival times by suppressing ambient noise and highlighting seismic events. Additionally, we show that while the original PhaseNet model excels in detecting major seismic events with high certainty, it struggles to identify new potential events due to its training on a different dataset. Transfer learning emerges as a viable strategy to improve model performance on small datasets by adapting the pre-trained model to new regional data characteristics without the need for extensive retraining.

1 Methodology

1.1 PhaseNet

PhaseNet is a deep-neural-network-based arrival-time picking method which can automatically pick the arrival times of both P and S waves. The original PhaseNet was trained on the prodigious available dataset (>700,000 waveform samples) provided by analyst-labelled P and S arrival times from the Northern California Earthquake Data Center[1]. The prediction results of PhaseNet indicate that its performance surpasses traditional automated methods, especially in terms of the accuracy of its S-wave arrival time predictions, which has the potential to dramatically increase the number of S-wave observations compared to traditional methods.

Fig 1 displays the network architecture, which is very similar to the symmetric structure of U-Net, except that all convolutional layers use 1D convolutions since the continuous seismic data are 1D. Because all convolutional layers employ 1D convolutions, the total number of trainable parameters of this network is only 268,443, which allows fast training and the capability to train directly on the CPU. The input and output of this network both have a dimension of 3×3001, where 3 represents that the original dataset is three components, and 3001 represents that each unit of data is 3001 sample points long. The output provides predictions of the possibilities for each sample point for the P-phase, S-phase, and Noise (the sum of probabilities for these three is 1): For example, when a point is close to the actual arrival of the P phase, the probability of the P phase will increase significantly, while the probabilities of the S phase and noise will decrease simultaneously:

$$1 = \text{Prob(Noise)} + \text{Prob(P)} + \text{Prob(S)}$$

where the 'Prob' represents the probability of each class. Although the network will predict the probabilities of these three classes for each sample point, during the evaluation stage, we will extract the peaks of the probability distributions for the P-phase and S-phase as their accurate arrival times.

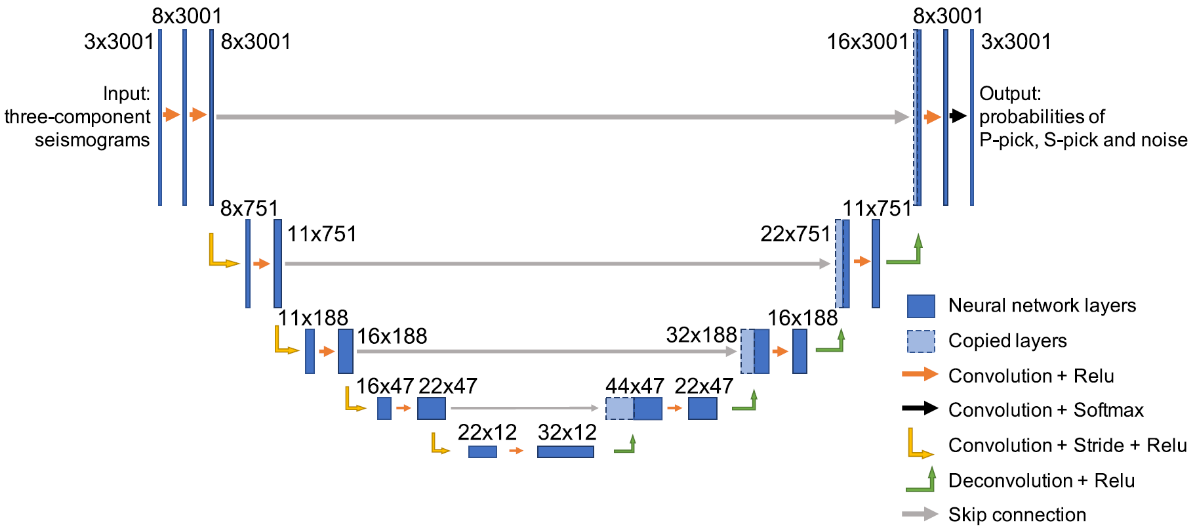


Fig 1. PhaseNet architecture [1]

This report utilizes `Seisbench`[2], a third-part Python library, for the building and training of PhaseNet. This library has already integrated a variety of commonly used models along with their weights trained on multiple publicly available large-scale datasets. In addition, this library also facilitates convenient dataset creation and preprocessing.

1.2 Transfer Learning

Since our new dataset from Iceland has a similar sampling rate to the dataset used to train the original PhaseNet (the original PhaseNet dataset sampling rate is 100 *Hz*, and our dataset is a mix of 100 *Hz* and 200 *Hz*), this report adopts the most commonly used transfer learning strategy[3, 4]: we use the entirety of the PhaseNet model and its pretrained weights as a starting point for training, and then fine-tune all model weights equally using just 111,836 seismograms. We observe whether transfer learning can improve the results over the original PhaseNet model, which was trained using 700,000 seismograms.

2 Dataset

The new dataset consists of three-component continuous seismic data collected in Iceland, totaling 111,836 records. This dataset documents microseismic events (with magnitude less than 2) and features a mixed sampling rate of 100 *Hz* and 200 *Hz*. Each data record is 80 seconds long, and in addition, each data record includes four important parameters:

- `trace_sampling_rate_hz`
- `trace_start_time`
- `trace_p_arrival_sample`
- `trace_s_arrival_sample`

There are other parameters, but they are irrelevant to transfer learning and PhaseNet retraining. Next, We first use `Obspy` to view the original waveforms and perform pre-processing (highpass filtering), and then use `Seisbench` to convert the raw dataset into a specialized dataset suitable for PhaseNet.

In conclusion, two distinct datasets are considered: one that has been preprocessed using a 3 *Hz* high-pass filter, and the other being the raw data. Our objective is to determine whether the high-pass filtering enhances the model's performance.

Table 1. Overview of Dataset Configurations

	Preprocessing	**Training Size**	**Validation Size**	**Test Size**
Dataset 1	3 *Hz* High-pass	89,468	11,184	11,184
Dataset 2	None	89,468	11,184	11,184

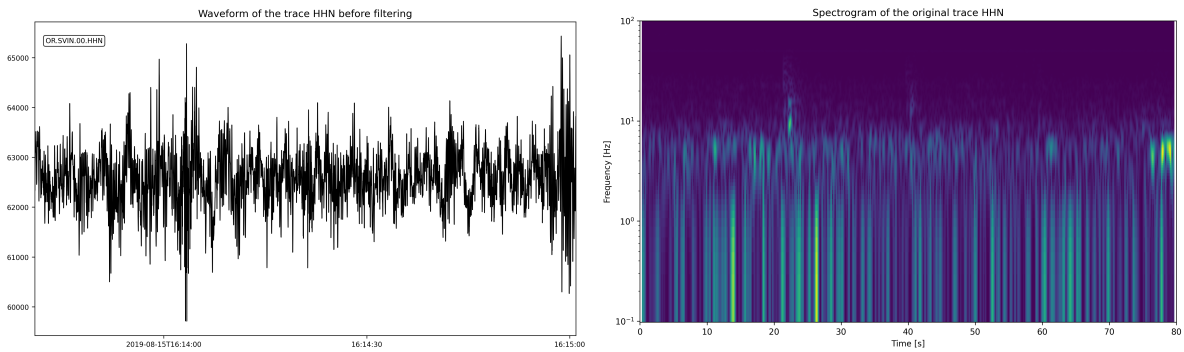


Fig 2. The left panel is the original waveform of a data; The right panel is its time-frequency spectrum.

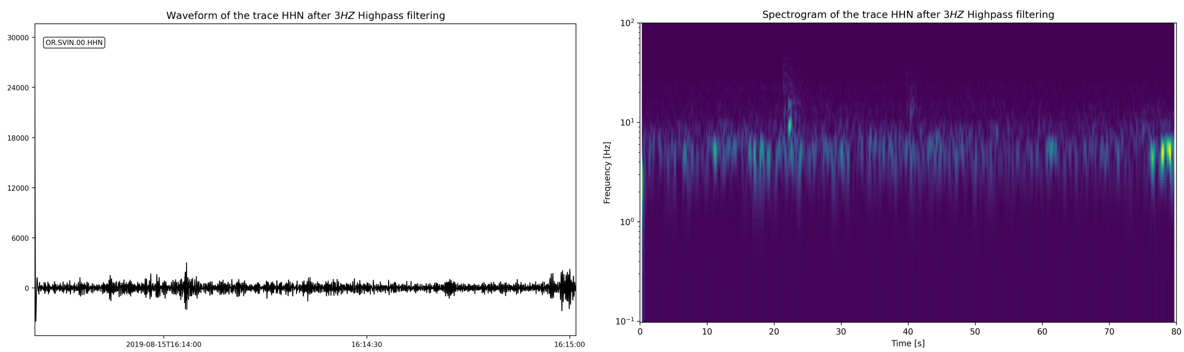


Fig 3. The left panel is the waveform after 3 *Hz* high-pass filtering; The right panel is the filtered time-frequency spectrum.

2.1 Dataset Preprocessing: High-pass filtering

The dataset is from Iceland and it is known that the primary source of ambient noise comes from the tidal movements of the surrounding seawater, with frequencies predominantly below 3 *Hz*. Therefore, the dataset can be preprocessed using high-pass filtering to eliminate ambient noise below 3 *Hz*.

Comparing the left panels in Fig. 2 and 3, we can find that the majority of the ambient noise has been filtered out after 3Hz high-pass filtering. Therefore, it can be expected that high-pass filtering is a key preprocessing method to improve dataset quality, and higher data quality can lead to better training results.

2.2 Dataset Preparation

The entire dataset is divided into training (89468), validation (11184), and test (11184) datasets in an 8: 1: 1 ratio. The training dataset is used for model training, while the validation dataset is used for accuracy assessment during the training process. After the model training is completed, the test dataset is used to calculate the final accuracy.

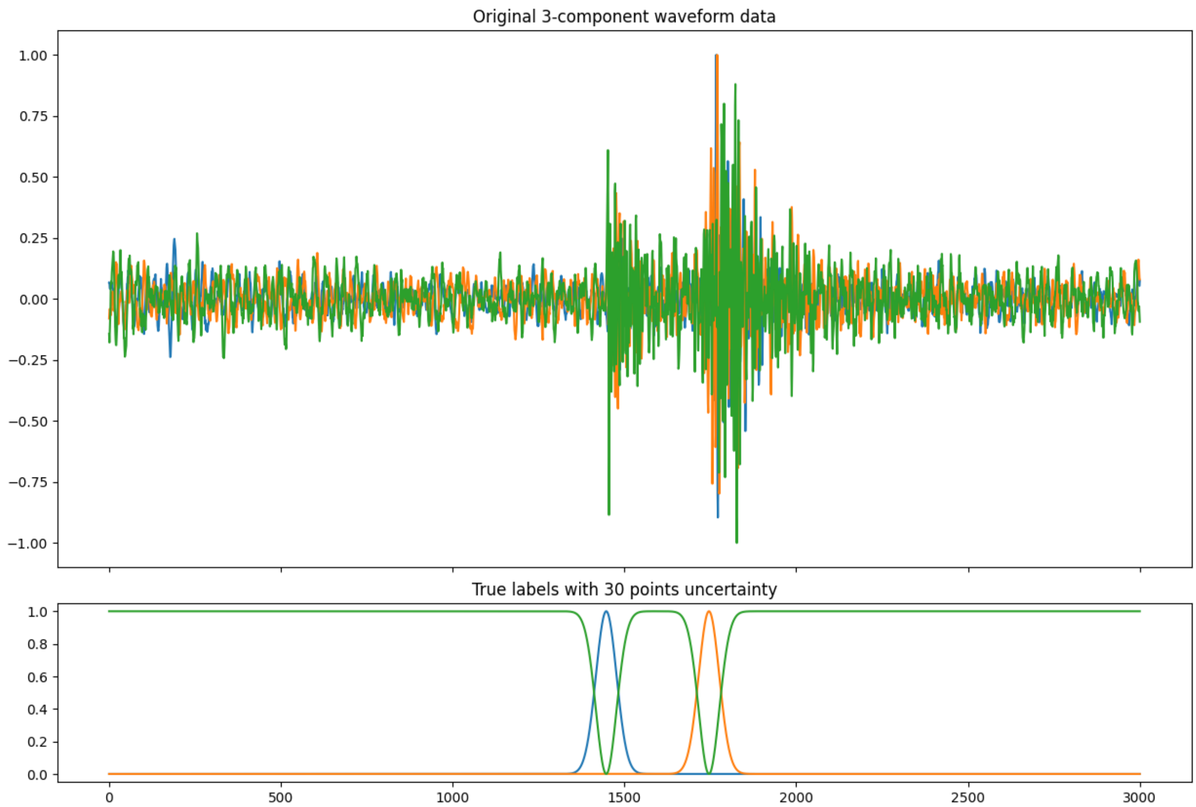


Fig 4. An example if a data after augmentation has a length of 3001 points, a size of 3×3001, and each label's uncertainty follows a normal distribution with a deviation of 30 points.

When setting up dataset augmentations, there are three functions to note. Fig. 4 illustrates a sample of the data used as input for PhaseNet.

- `sbg.WindowAroundSample`: Given that each data spans 80 seconds, which corresponds to 8000 data points at a 100 *Hz* sampling rate or 16000 data points at a 200 *Hz* sampling rate, not all of these points are necessary for training. Trimming the data appropriately is required, focusing only on the

P-phase and S-phase arrival points and a brief period before and after these points. In this function, the parameter `samples_before` indicates the number of data points preceding the first label (the P-phase arrival point) that mark the start point for data segmentation, while `windowlen` specifies how many data points from this starting position are included within the data segmentation range. For this paper, `samples_before` is set to 3000 and `windowlen` to 6000. As such, only 6000 data points per data segment are utilized.

- `sbg.RandomWindow`: Due to PhaseNet's requirement for an input data size of 3001, the parameter `windowlen` in this function must be configured to 3001. Furthermore, each dataset spans 6000 effective points, and the function will randomly shift within this 6000-point span to generate multiple segments, each 3001 points in length.
- `sbg.ProbabilisticLabeller`: This function introduces uncertainty to each label using a normal distribution, with the `sigma` parameter denoting the standard deviation of the said distribution. In this paper, it is assigned a value of 30.

3 Models

Table 2. Model Configurations and Training Details

Model	**Training Approach**	**Dataset**	**Epochs**
Model 1	PhaseNet structure only	Dataset 1	60
Model 2	PhaseNet structure only	Dataset 2	60
Model 3	Transfer learning with pre-trained (original) PhaseNet	Dataset 1	60
Model 4	Transfer learning with pre-trained (original) PhaseNet	Dataset 2	60
Model 5	Original PhaseNet (Base model, no training)	Evaluation on test datasets of Dataset 1 and Dataset 2	-

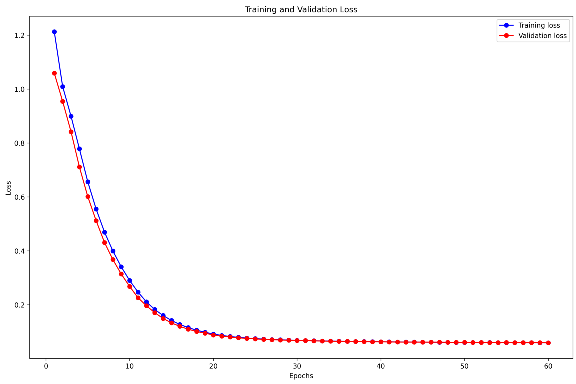
Note: All trainable models share the same hyperparameters: cross entropy loss, batch size of 256.

All models used in this report utilize the PhaseNet architecture and can be distinguished into five different models based on the final weights:

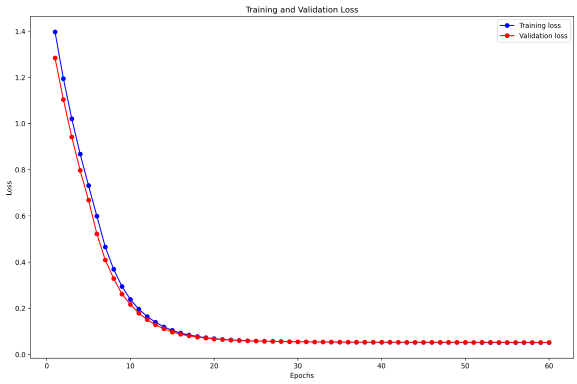
- Model 1: employs the PhaseNet structure and is trained from scratch on dataset 1 for 60 epochs.
- Model 2: also uses only the PhaseNet structure, starting from scratch and trained for 60 epochs on dataset 2.
- Model 3: utilizes a pre-trained PhaseNet (there are several pre-trained PhaseNet, in this report we only use the 'original') and continues training for 60 epochs on dataset 1 (transfer learning).
- Model 4: uses the same pre-trained original PhaseNet and continues training for 60 epochs on dataset 2.

- Model 5: the original PhaseNet; it undergoes no further training (using the original weights) and serves as the base model for final accuracy evaluation on the test datasets of dataset 1 and dataset 2.

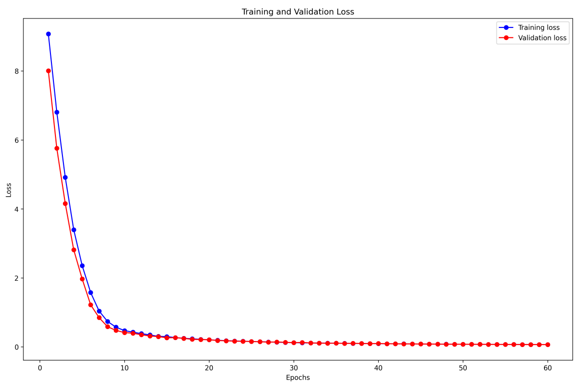
After 60 epochs of training, Models 1 to 4 have all converged (as shown in Fig 5, their loss functions eventually hover around 0.05). Next, we will verify whether preprocessing can improve model performance by comparing the prediction accuracies of Models 1 and 2, as well as Models 3 and 4, on their respective test datasets. We will also assess whether transfer learning performs better than training from scratch by comparing Models 1 and 3, as well as Models 2 and 4.



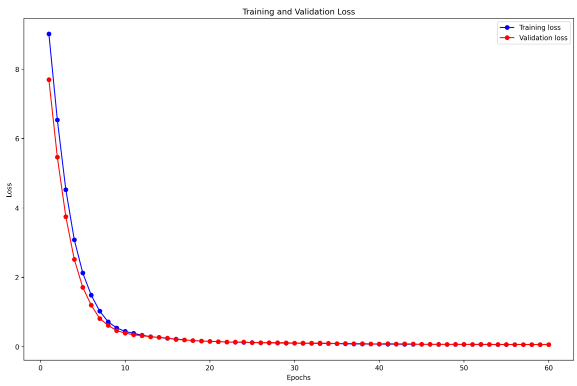
(a) Loss function of Model 1 over 60 epochs of Training



(b) Loss function of Model 2 over 60 epochs of Training



(c) Loss function of Model 3 over 60 epochs of Training



(d) Loss function of Model 4 over 60 epochs of Training

Fig 5. Loss functions of Model 1 to 4 over 60 epochs of Training using the same PhaseNet Architecture

4 Results

In this report, we have set a threshold of 0.5s for the P-phase and S-phase arrival times, meaning that an event is considered correctly predicted by the model if the predicted arrival time and the true label differ by no more than 0.5s. Figs. 6 to 9, respectively, display the accuracy statistics for Models 1 to 4 on their respective test datasets; Figs. 10 and 11 show the accuracy statistics for Model 5 (the base model).

For Figs 6 and 7, as well as 8 and 9, and 10 and 11, it is evident that using datasets that have undergone preprocessing allows the models to perform better, with this improvement primarily reflected in the predictions for the P-phase. For Models 1 and 2, which are trained from scratch, using the preprocessed dataset improved the prediction accuracy for the P-phase from 72.0% to 83.28%, while the improvement for S-phase predictions was limited, increasing from 91% to 93.12%. Similar results were observed for Models 3 and 4, which underwent transfer learning on the pre-trained PhaseNet model. For the base model: it performs very poorly when predicting on unprocessed datasets, effectively failing to identify accurately; however, when predicting on preprocessed data, the prediction accuracy significantly improves.

Comparing Fig 6 and 8, and Fig 7 and 9: Using transfer learning shows an improvement in the accuracy of S-phase predictions over training from scratch, even though PhaseNet itself is quite proficient at predicting the S-phase (as shown in Fig 11, the accuracy for the S-phase is 92.69%). With transfer learning, the prediction accuracy for the S-phase on new datasets can be further enhanced, reaching the highest accuracy of 95.47% for S-phase predictions as shown in Fig 9. However, it is also noted that if the new dataset is small, both training from scratch and using transfer learning do not perform as well as simply using the original PhaseNet. This is because the original PhaseNet was trained on a large-scale dataset and its ability to predict the relatively weak signals of P-phase arrival times is already very robust, making it unsuitable for further altering its internal weight values.

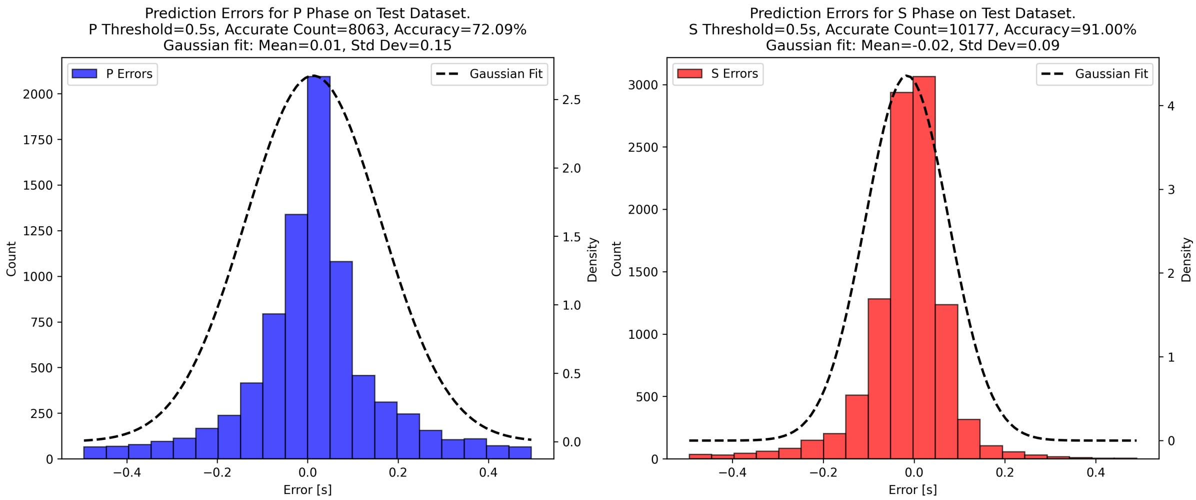


Fig 6. Accuracy Statistics for **Model 1** on Test Dataset of **Dataset 1**.

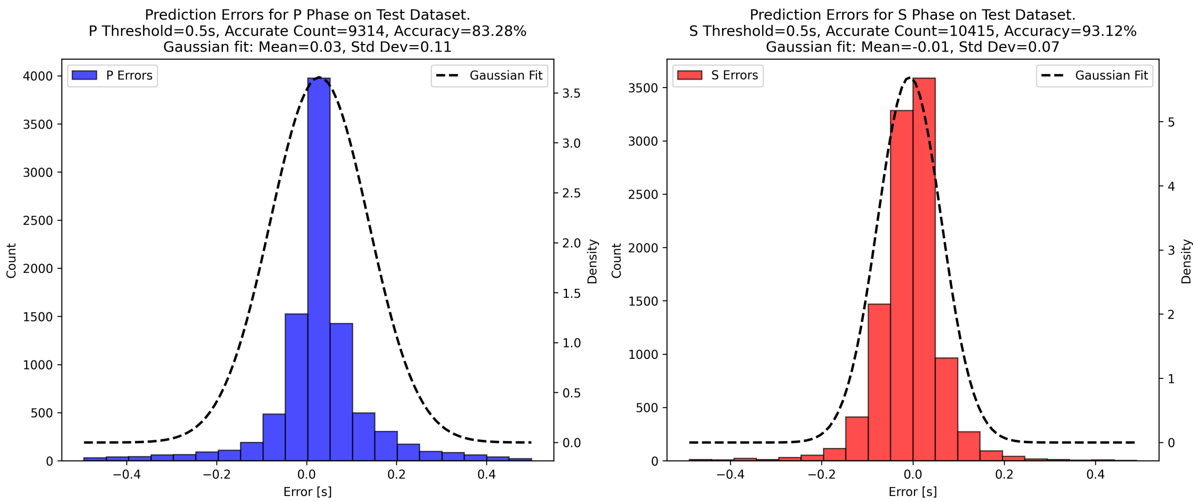


Fig 7. Accuracy Statistics for **Model 2** on Test Dataset of **Dataset 2**.

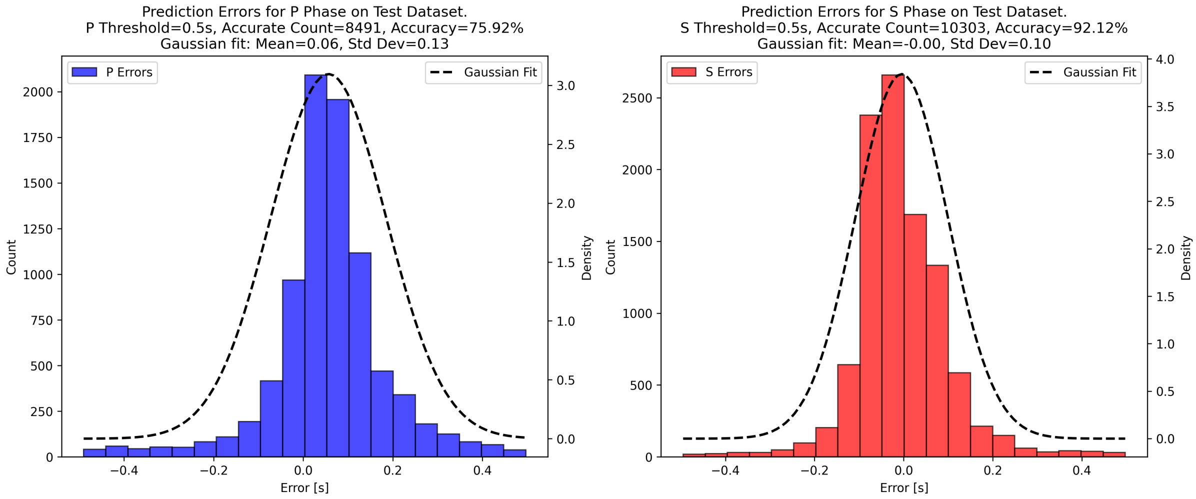


Fig 8. Accuracy Statistics for **Model 3** (transfer learning) on Test Dataset of **Dataset 1**.

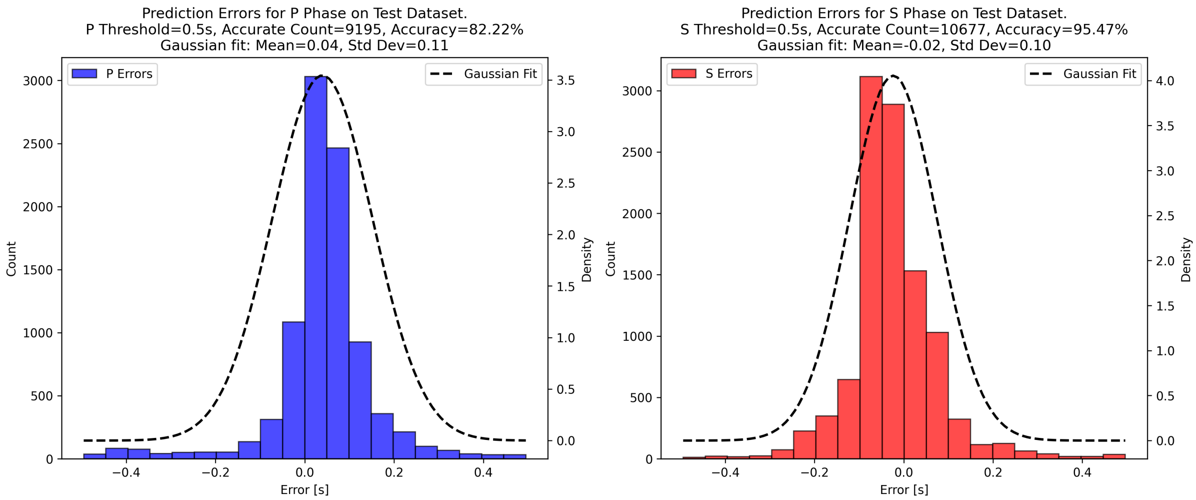


Fig 9. Accuracy Statistics for **Model 4** (transfer learning) on Test Dataset of **Dataset 2**.

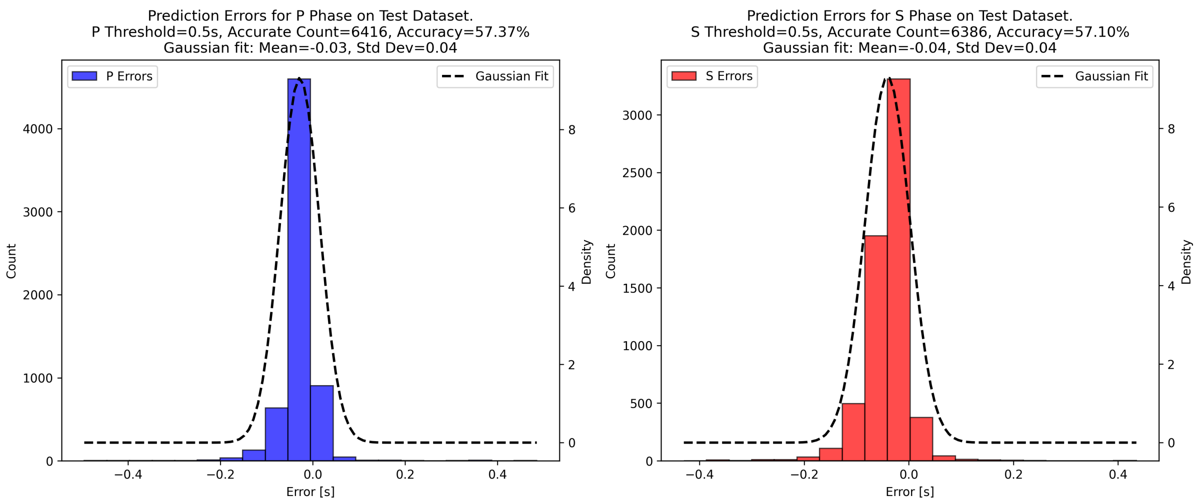


Fig 10. Accuracy Statistics for **Model 5** (base model) on Test Dataset of **Dataset 1**.

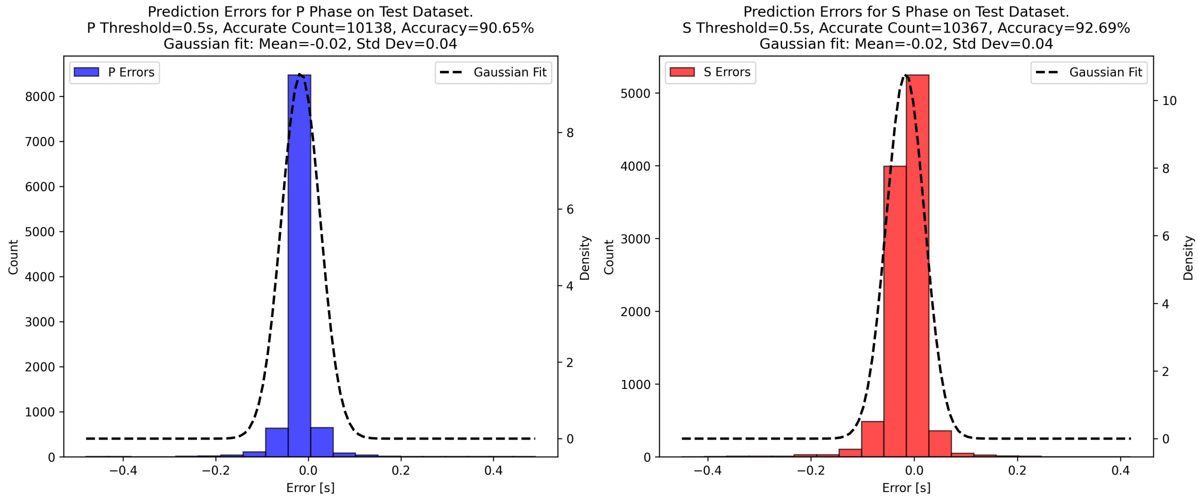


Fig 11. Accuracy Statistics for **Model 6** (base model) on Test Dataset of **Dataset 2**.

After the statistical analysis of accuracy, we further analyze the actual prediction performance of these five models on the same data. Regarding preprocessing: comparing Fig 12 and 13, it can be seen that preprocessing can suppress obvious erroneous predictions (such as the incorrect P-phase arrival time prediction around 80s in Fig 12); comparing Fig 14 and 15, it can be observed that preprocessing enhances the model's confidence in predicting potential events (such as the S-phase arrival time around 45s in Fig 15). Regarding transfer learning: comparing Fig 13 and 15, models trained with transfer learning indeed have a higher probability of detecting potential new events than models trained from scratch (again referring to the S-phase arrival time around 45s in Fig 15).

Although the base model accurately predicts major events with high certainty (the probability curve is very narrow as shown in Fig 16 and 17), it struggles to detect new potential events. This is because the base model has never encountered this new dataset, and its prediction experience is based solely on training with the original dataset. While the dataset used in this report is somewhat similar to the original dataset used by PhaseNet (such as the sampling rate), different datasets inevitably have their own unique detailed features, which are difficult to detect without having trained the model on them. Therefore, depending on the objective, different models can be selected: If the goal is to quickly annotate the arrival times of major events in a new dataset, then simply predicting using the original PhaseNet after preprocessing the data is sufficient; if the goal is to discover potential events, then the model needs to learn from this new set of data, but not from scratch: instead, opt for transfer learning based on a pre-trained model.

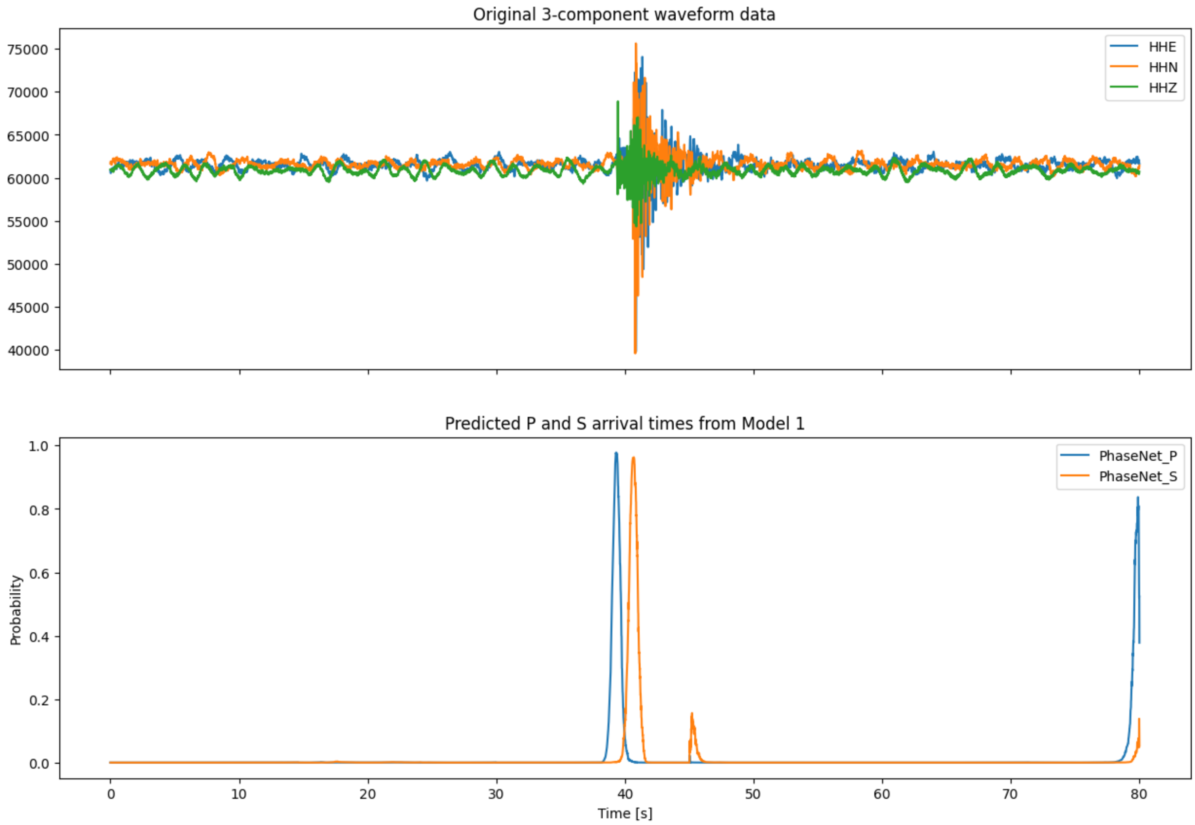


Fig 12. **Model 1** Prediction on a same Test Data (without preprocessing)

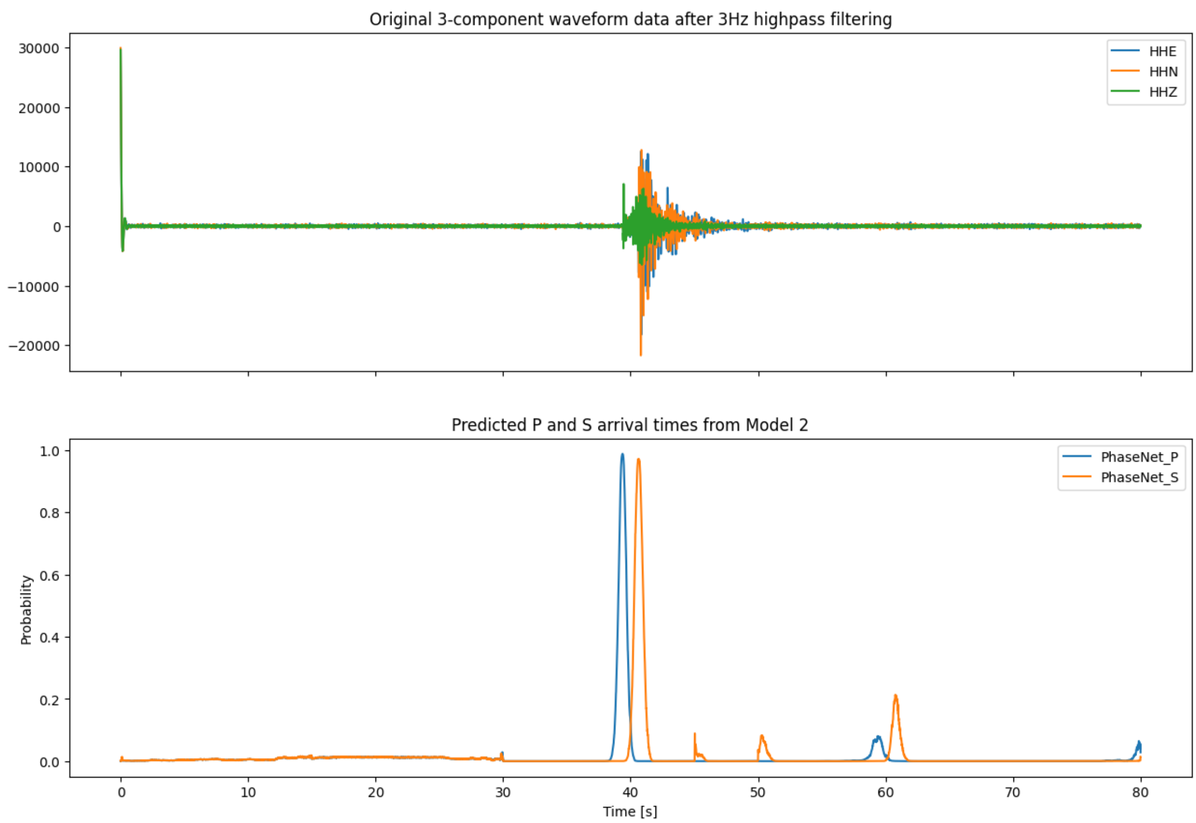


Fig 13. **Model 2** Prediction on a same Test Data (after 3 *Hz* highpass filtering)

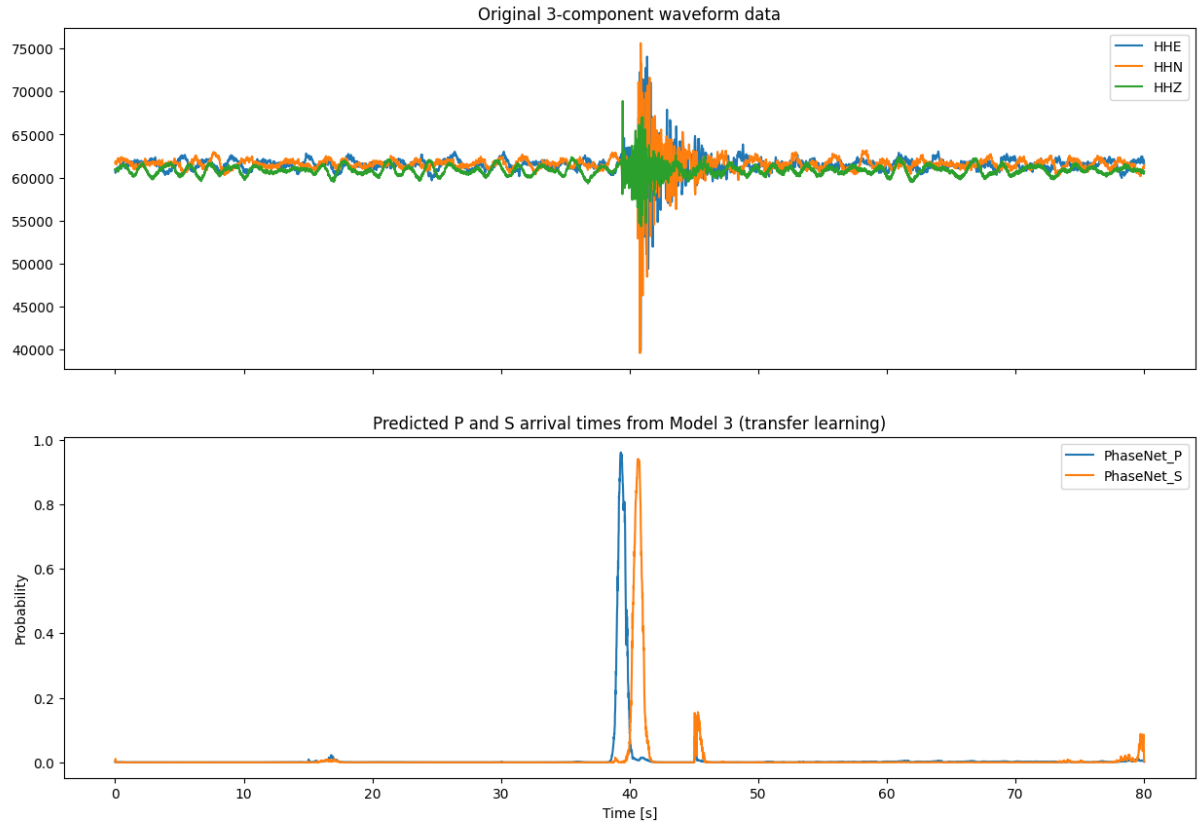


Fig 14. **Model 3** (transfer learning) Prediction on a same Test Data (without preprocessing)

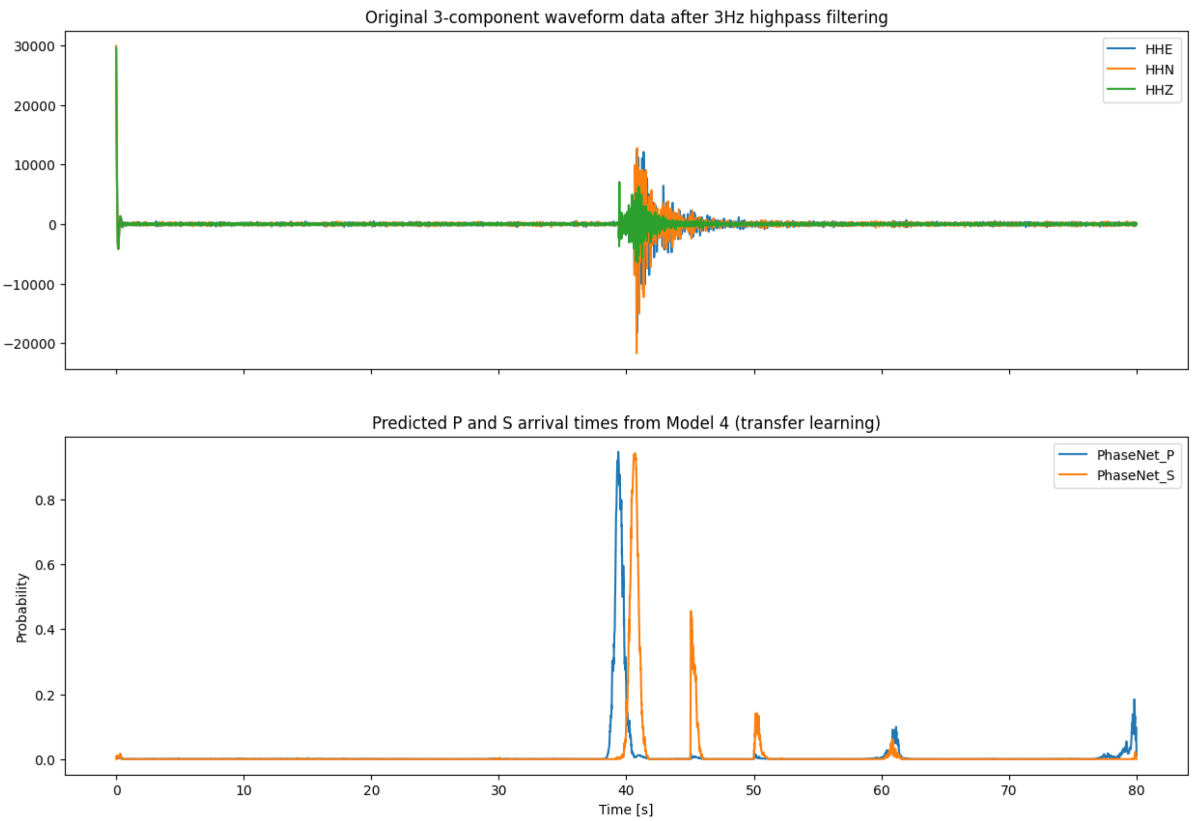


Fig 15. **Model 4** (transfer learning) Prediction on a same Test Data (after 3 *Hz* highpass filtering)

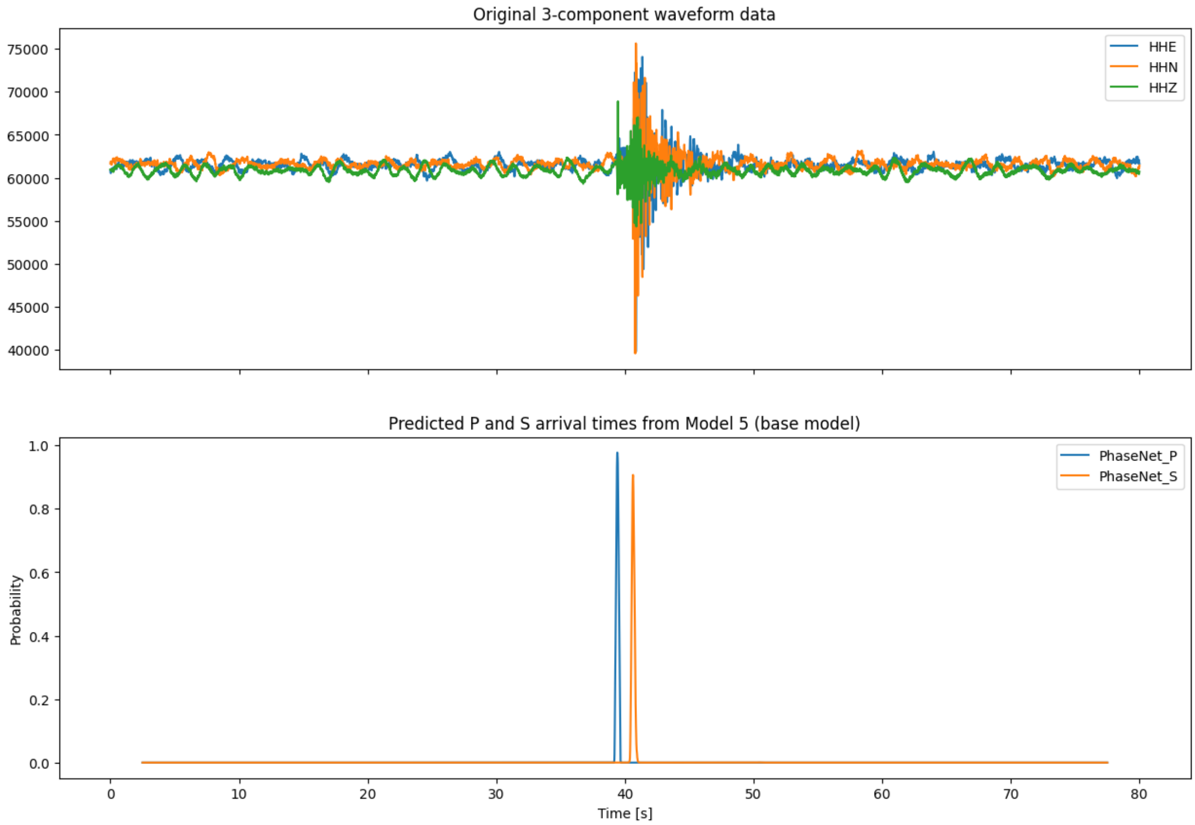


Fig 16. **Model 5** (base model) Prediction on a same Test Data (without preprocessing)

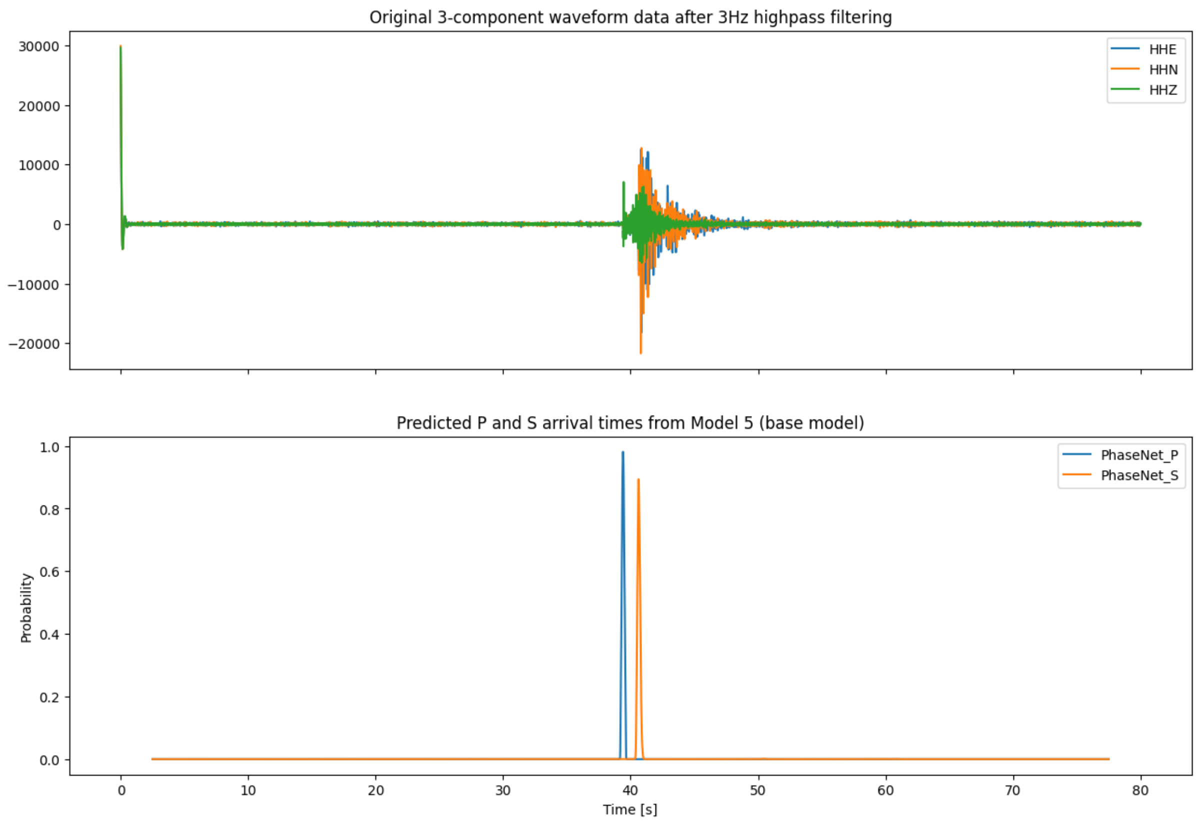


Fig 17. **Model 5** (base model) Prediction on a same Test Data (after 3 *Hz* highpass filtering)

5 Conclusions

The following conclusions can be drawn from the results.

1. Models using the PhaseNet architecture consistently show higher prediction accuracy for S-phase arrival times compared to P-phase arrival times. This is because the S-phase generally has a larger amplitude when it arrives, making it easier for the model to detect this variation.

2. For a new dataset, preprocessing (especially filtering if the approximate range of ambient noise is known) can effectively improve the model's prediction accuracy and performance.

3. If the new dataset is quite similar to the original dataset used for training the original PhaseNet, then directly using the original PhaseNet on the preprocessed new dataset can yield good results, especially for the prediction of P-phase arrival times, without the need for retraining from scratch or undergoing transfer learning based on a pretrained model.

4. If the new dataset is small-scale, transfer learning based on a pretrained model can slightly enhance the prediction accuracy for S-phase events and detect potential events (S-phase arrival times), but such a retrained model's accuracy for P-phase arrival times will be lower than that achieved by directly using the original PhaseNet.

References

[1] Zhu, Weiqiang, and Beroza, Gregory C. (2019). *PhaseNet: a deep-neural-network-based seismic arrival-time picking method*. Geophys. J. Int., 216, 261–273, doi: 10.1093/gji/ggy423.

[2] Woollam, J., Münchmeyer, J., Tilmann, F., Rietbrock, A., Lange, D., Bornstein, T., Diehl, T., Giunchi, C., Haslinger, F., Jozinović, D., et al. (2022). *SeisBench—A Toolbox for Machine Learning in Seismology*. Seismol. Res. Lett., 93, 1695–1709, doi: 10.1785/0220210324.

[3] Chai, C., Maceira, M., Santos-Villalobos, H. J., Venkatakrishnan, S. V., Schoenball, M., Zhu, W., et al. (2020). *Using a deep neural network and transfer learning to bridge scales for seismic phase picking*. Geophysical Research Letters, 47, e2020GL088651. `https://doi.org/10.1029/2020GL088651`

[4] Lapins, S., Goitom, B., Kendall, J.-M., Werner, M. J., Cashman, K. V., & Hammond, J. O. S. (2021). *A little data goes a long way: Automating seismic phase arrival picking at Nabro volcano with transfer learning*. Journal of Geophysical Research: Solid Earth, 126, e2021JB021910. `https://doi.org/10.1029/2021JB021910`