Transfer Learning in MCU Performance Screening

Transfer Learning in MCU Performance Screening

Nicolò Bellarmino*, Riccardo Cantoro*, Martin Huch†,
Tobias Kilian†‡, Ulf Schlichtmann‡ and Giovanni Squillero*

***Abstract*—In safety-critical applications, microcontrollers must meet performance standards, including the maximum operating frequency ($F_{\max}$). ML models can estimate $F_{\max}$ using data from on-chip ring oscillators (ROs). However, when new products are introduced, existing ML models may no longer be suitable and may require updating. Training a new model is challenging due to limited data availability and time needed to acquire $F_{\max}$. But data from legacy products, along with pre-trained models, may still be available. We propose using deep-learning models trained on a specific MCU product as feature extractors for new devices, to address the scarcity of labeled data, in a Transfer Learning fashion. Experimental results show that these models can extract useful general features for performance prediction even from new products. As a result, they achieve better performance with significantly less labeled data compared to traditional shallow learning approaches.**

I. Introduction

MCUs' performance screening aim at identifying devices that do not meet specified characteristics (maximum operating frequency, $F_{\max}$) [1]. Traditional speed binning involves executing functional tests at increasing clock frequencies. However, this approach is time-consuming, requires expensive Automatic Test Equipment (ATE), and only provides categorical binning results [1], [2].

Machine learning (ML) regression models have been proposed to predict the $F_{\max}$ of MCUs. Previous research has explored the use of on-chip ring oscillators, known as Speed Monitors (SMONs), as features to predict $F_{\max}$ [2], [3].

The accuracy of supervised ML models relies on high-quality and sufficient labeled data. While unlabeled data are readily available and inexpensive to obtain, acquiring $F_{\max}$ data is time-consuming and limits the amount of available labeled data. When introducing a new product, there is often a scarcity or absence of data. Constructing an ML model from scratch under these circumstances becomes challenging, as it requires obtaining an adequate training set, which is very time-consuming [4]. Additionally, using the very same model trained on previous-generation data may not be viable due to potential shifts in data distributions, resulting in significant prediction errors. However, data and models from previous-generation products remain accessible and can be utilized to build new ML models.

In this research, we propose leveraging the knowledge acquired from a legacy MCU product *(P1)* to develop models for a different new product class *(P2)* using Transfer Learning with deep neural networks as feature extractors.

* Politecnico di Torino (Turin, Italy). † Infineon Technologies AG (Munich, Germany). ‡ Technical University of Munich (Munich, Germany). Authors are listed in alphabetical order.

TABLE I
Results on P2 (average on 5 75%-25% training-test splits)

Model	nRMSE	Samples to 2% nRMSE	Samples to 1.51% nRMSE
Poly Ridge (P1)	33.93%	–	–
Poly Ridge (P2)	1.54%	121	812
PL-CNN	**1.51%**	**16**	**227**
AE-CNN	1.54%	24	746

II. Discussion

P1 and *P2* come from the same product family, with the same 27 SMONs equipped on board. The values of these are used as features by our ML models. The methodology was validated on 1015 *P2*-devices. We used two different 1-Dimensional Convolutional Neural Networks (an autoencoder and a supervised network, namely AE-CNN and PL-CNN), pre-trained on millions of *P1*-samples. We use these to extract features from *P2*. Then, just a linear regression was trained on *P2* devices only, to compute the $F_{\max}$ from the features extracted by the CNNs. The evaluation metric is the normalized Root Mean Square Error (nRMSE), i.e. RMSE normalized by the mean value of $F_{\max}$ in the test set: $nRMSE = RMSE(y_{true}, y_{pred})/mean(y_{true})$. Results are shown in Table I. Applying a shallow model trained on P1 (Poly Ridge) directly on P2 leads to enormous prediction errors. The coefficient of the model should be adapted to the new product. The re-training of the same on P2 is effective, since we can reach good accuracy (1.54%). The deep feature extractor can obtain not only better results (1.51%), but with a fraction of the labeled data: only 16 samples needed to obtain 2% nRMSE (121 with Poly Ridge, see Table I). With this approach, only hours of labeling are required (not days/months). The feature extraction step can be used in other scenarios in which pre-trained models are available. Future works aim at optimizing the deep models, by tuning the NNs architecture to achieve an additional reduction in the samples needed to build the final supervised models.

References

[1] J. Zeng *et al.*, "On correlating structural tests with functional tests for speed binning of high performance design," in *2004 International Conference on Test*, 2004.

[2] J. Chen *et al.*, "Data learning techniques and methodology for Fmax prediction," in *2009 ITC*, 2009.

[3] R. Cantoro *et al.*, "Machine Learning based Performance Prediction of Microcontrollers using Speed Monitors," in *2020 ITC*, 2020.

[4] N. Bellarmino *et al.*, "Semi-supervised deep learning for microcontroller performance screening," 2023.