In-Context Learning for Microcontroller Performance Screening Using Tabular Foundation Models

In-Context Learning for Microcontroller Performance Screening Using Tabular Foundation Models

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

*Politecnico di Torino
Torino, Italy

†Infineon Technologies AG
Munich, Germany

***Abstract*—Microcontroller (MCU) performance screening ensures that devices meet critical specifications, such as maximum operating frequency ($F_{\max}$). On-chip Speed Monitors (SMONs), implemented as ring oscillators, provide process-correlated signals that can be used to estimate $F_{\max}$ via machine learning (ML). However, traditional ML models require substantial domain expertise, extensive feature engineering, hyperparameter tuning, and dataset-specific training, limiting their scalability and generalization.**

In this preliminary study, we explore the use of TabPFN, a pre-trained Tabular Foundation Model (TabFM) based on In-Context Learning (ICL), for MCU performance prediction. TabPFN eliminates the need for task-specific training or tuning by conditioning directly on labeled examples provided at inference time, enabling few-shot and zero-shot learning.

We evaluate TabPFN on two distinct MCU datasets and compare its performance with conventional ML models, including tree-based and linear approaches. Our results show that TabPFN consistently achieves competitive accuracy with minimal human supervision, demonstrating its potential as a fast, generalizable, and low-maintenance alternative for performance screening in semiconductor manufacturing.

***Index Terms*—Fmax, Speed Monitors, Ring Oscillators, Speed Binning, Machine Learning, Device Testing, Manufacturing, Tabular Foundation Models, Prior-Fitted Networks**

I. Introduction

Microcontroller (MCU) performance screening aims to identify devices that fail to meet datasheet specifications, particularly the maximum operating frequency ($F_{\max}$). Recent studies have shown that Machine Learning (ML) models, when trained on relevant features correlated with $F_{\max}$, can accurately predict device performance and support screening processes [1]–[4]. A particularly effective approach involves using on-chip Ring Oscillators (ROs), commonly referred to as Speed MONitors (SMONs), as proxies for silicon speed [4]–[7]. By executing functional tests, these measurements are converted into tabular datasets used for supervised ML tasks.

Tabular data—structured data organized into rows and columns—is ubiquitous in manufacturing, healthcare, finance, and other industrial domains [8], [9]. Deep learning approaches have historically underperformed on tabular data when compared to conventional models, particularly tree-based algorithms such as XGBoost [10], Random Forests, and CatBoost [11], referred to as *shallow learning* models. These models, although effective, require substantial human involvement in the form of domain expertise, feature engineering, and hyperparameter tuning [12]. This limits their accessibility and slows down deployment, particularly in domains such as semiconductor testing, where ML expertise may not be readily available.

Recent advances in foundation models have introduced a promising paradigm shift: *In-Context Learning* (ICL) [13]. In ICL, models learn to generalize by conditioning on a set of input-output examples provided at inference time, without requiring retraining or gradient updates. This approach, successfully demonstrated in large language models like GPT-3 [14], has recently been extended to tabular data through the development of *Tabular Foundation Models* (TabFMs) [15].

One of the most prominent TabFMs is *TabPFN* [16], a transformer-based model trained on millions of synthetically generated tabular tasks. It embeds tabular learning into a probabilistic inference problem and performs prediction in a single forward pass. TabPFN requires minimal supervision: no hyperparameter tuning, no manual feature engineering, and no retraining on the target dataset. This makes it particularly suitable for industrial use cases where datasets are small, noisy, or collected under constrained testing environments—as is common in MCU characterization workflows.

This paper investigates the applicability of TabPFN for MCU performance screening. We assess its ability to generalize across two different MCU products and compare its performance with traditional ML models, including Ridge Regression, Polynomial Ridge, Random Forest, and XGBoost. Our goal is to evaluate whether TabPFN can serve as a drop-in, zero-shot predictor in early-stage production or characterization pipelines, potentially reducing the effort required to build performant ML solutions.

The remainder of the paper is organized as follows. Section II reviews prior literature on ML-based performance screening and tabular models. Section III provides founda-

Authors are listed in alphabetical order.

tional concepts, including MCU testing (Section III-A), dataset generation (Section III-B), SMON features (Section III-C), traditional ML pipelines (Section III-D), and TabPFN-specific methodology (Section III-E). Section IV outlines the rationale behind our study. Section V describes the experimental setup, while results and analysis are presented in Section VI. Conclusions and future directions are discussed in Section VII.

II. Related Work

Several techniques have been proposed over the years for predicting circuit performance parameters [17]–[20]. A prominent line of research involves the use of *indirect measurements*—also referred to in the literature as *alternate test*—to estimate critical device characteristics without direct testing. Originally developed for analog circuits, alternate test methodologies aim to learn mappings between low-cost, indirect features and target specifications [21]–[26]. These approaches can significantly reduce test time and cost while maintaining acceptable accuracy.

In the digital domain, and more specifically for microcontroller (MCU) performance screening, recent studies have demonstrated the effectiveness of machine learning (ML) methods in predicting the maximum operating frequency ($F_{\max}$). The work presented in [1], [3], [4], [27], [28] proposed supervised models trained on measurements from on-chip ring oscillators—known as Speed Monitors (SMONs)—to accurately infer $F_{\max}$. These studies confirmed the viability of using ML to exploit indirect structural information for performance screening and yield analysis.

As the demand for automated and data-driven methods has grown, interest in learning from structured (*tabular*) data has expanded considerably. Tabular data is widespread across many industrial domains, including manufacturing, healthcare, and finance, prompting extensive research into deep learning models tailored for this data type [8], [9], [27], [29]–[31]. One prominent direction involves transforming tabular datasets into image-like formats suitable for convolutional neural networks (CNNs) [32], [33]. In [29], for example, deep CNNs were applied to SMON-based datasets, enabling the development of semi-supervised pre-trained models for MCU performance prediction.

More recently, a new family of models known as *Tabular Foundation Models* (TabFMs) has emerged, offering a paradigm shift in how tabular data is processed and learned [15], [34]. These models leverage *in-context learning* (ICL) to perform zero-shot or few-shot inference on unseen tasks with minimal supervision. Among these, TabPFN [16] stands out as a transformer-based model trained entirely on synthetically generated data, allowing it to learn robust priors over tabular problems. Built upon the Prior-Data Fitted Networks (PFN) framework [35], TabPFN enables classification and regression through a single forward pass without requiring model retraining, feature engineering, or hyperparameter tuning.

III. Background

A. Device Testing

Testing is a fundamental step in the lifecycle of Integrated Circuits (ICs), ensuring that fabricated devices conform to datasheet specifications and functional requirements. It is performed at multiple stages, including pre-silicon validation, prototype characterization, high-volume production, and post-market deployment.

Following design verification, characterization tests are conducted on early prototypes to identify design or manufacturing anomalies. Once production begins, wafers are subjected to on-wafer testing prior to dicing and packaging. Subsequently, post-packaging tests verify the electrical and functional integrity of each IC. In some applications, additional in-field testing is performed to monitor behavior under real-world operating conditions.

Due to the complexity of modern digital systems, exhaustive functional testing is often impractical [36]. As a result, structural tests are commonly employed to detect manufacturing defects through indirect metrics rather than complete functional verification. These structural tests typically rely on automated test equipment (ATE), which can introduce significant costs due to hardware requirements and limited reusability across different chip designs.

To address these limitations, industry has adopted various Design-for-Testability (DfT) techniques [37], which embed dedicated test structures into the IC to facilitate faster and more cost-effective testing. Although DfT improves test coverage and automation, it may introduce trade-offs in terms of design effort and silicon area overhead.

B. Microcontroller Characterization Process

Training supervised ML models for performance prediction requires labeled datasets that correlate internal measurable features with external performance indicators. In the context of microcontrollers (MCUs), the performance metric of interest is the maximum operational frequency ($F_{\max}$), while the input features are derived from the frequencies of on-chip ring oscillators—known as Speed Monitors (SMONs).

SMON frequencies are readily available as part of the standard production test flow. These frequencies are measured with high precision through fast and stable test procedures, making them ideal candidates as input features. However, the corresponding $F_{\max}$ values (labels) must be acquired through a dedicated characterization process involving time-intensive functional testing. This process is typically applied to a limited subset of the production batch.

To determine $F_{\max}$, each device is mounted on a test board and subjected to progressively increasing clock frequencies while executing functional patterns. The frequency at which the device fails to operate correctly is recorded, with the last successful frequency taken as the $F_{\max}$ [2], [38]. This process is repeated across multiple test patterns, yielding a multi-label dataset. For performance prediction, the most stringent $F_{\max}$—i.e., the lowest value across all patterns—is generally considered the critical target.

Due to the cost and time constraints of this labeling process, datasets used for ML training typically contain only a few hundred to a few thousand labeled samples. These samples are often drawn from Split Lot wafers—special wafer batches produced under controlled variations of fabrication parameters (e.g., doping levels, oxide thickness, thermal conditions). This strategy ensures a wide representation of manufacturing-induced variability, enabling better generalization and robustness of the trained models.

C. Speed Monitors (SMONs)

Speed Monitors (SMONs) are specialized on-chip ring oscillators (ROs) used to monitor the effects of process variability on circuit performance. Their oscillation frequencies are sensitive to local and global variations in the manufacturing process, making them effective proxies for inferring device speed and reliability.

Each SMON is composed of a chain of logic cells arranged to produce an oscillating signal. The chain includes both generic ROs—constructed from standard cells such as inverters, NANDs, and NORs—and design-specific ROs, which replicate critical timing paths extracted directly from the functional circuitry of the MCU.

To capture spatial variability, SMONs are organized into modules and strategically distributed across the die. Each module includes a heterogeneous set of ROs, providing diverse sensitivity profiles. This spatial arrangement is essential for capturing Within-Die (WID) variations, which become increasingly relevant in advanced technology nodes [39].

In the designs analyzed in this work, up to 130 distinct SMONs are embedded in compact modules placed at key die locations. This rich and spatially distributed feature set enables robust modeling of device behavior under different process conditions, serving as the foundation for accurate $F_{\max}$ prediction through machine learning.

D. Machine Learning

Training an ML model involves learning a relationship between input features and target outputs based on labeled data. The model analyzes patterns within the data to make predictions or classifications. For instance, simple linear regression models use a weighted sum of input features to estimate an output, with these weights being optimized during training.

In the context of tabular datasets, ML tasks often require feature manipulation and engineering to improve model performance. This process includes selecting relevant features, transforming data, and handling missing values, which can be time-consuming and computationally demanding. Such preprocessing steps, while crucial for accuracy, can slow down model deployment in production environments. This approach necessitates domain expertise making it challenging to implement in fast-paced manufacturing environments. Recent advancements in deep learning and automated feature learning have introduced alternative methods, such as TabFMs, which aim to streamline the process by reducing the need for manual feature engineering while maintaining high predictive accuracy.

E. Tabular Deep Learning and Tabular Foundation Models

Traditional deep learning techniques have primarily excelled in domains like image and natural language processing, where unstructured data dominates [33], [40], [41]. However, applying deep learning to structured tabular datasets has been an ongoing challenge [15], [31], [33]. Researchers have explored various approaches to leverage neural networks' automatic feature extraction capabilities for tabular data, particularly in scenarios with large datasets where complex relationships exist between features and labels. While deep learning models can effectively capture these relationships, their effectiveness diminishes in environments with limited labeled data, a common constraint in many industrial applications [12].

One significant advantage of deep learning is the ability to pre-train models on a large dataset and adapt them to new, smaller datasets through transfer learning [42]. This technique is difficult to implement in classical machine learning models, such as decision trees or support vector machines, which generally require retraining from scratch when applied to new tasks. However, the success of deep learning models in tabular data remains inconsistent, often necessitating extensive feature engineering and data transformation. Some methods attempt to transform tabular data into image-like representations to leverage convolutional neural networks (CNNs), but this approach is computationally expensive and impractical in cases with scarce labeled data [32].

Due to these limitations, shallow learning models, such as linear regression and tree-based methods, remain the preferred choice for many tabular data applications [43]. Deep neural networks (DNNs) are not a one-size-fits-all solution, particularly in scenarios with high heterogeneity among datasets, missing or imbalanced data, and varying feature importance. However, recent advancements in prior-data-fitted foundation models are introducing new paradigms in tabular learning.

Foundation models [44] trained on diverse datasets have revolutionized fields such as natural language processing by enabling in-context learning (ICL). ICL [14] is a paradigm introduced by foundation models, where a model learns to perform a task simply by conditioning on examples provided in the input, without requiring task-specific fine-tuning. This capability has been widely demonstrated in large language models (LLMs), such as GPT-3 [14], which can generate coherent text, answer questions, or translate languages after being given a few demonstrations in the prompt. Inspired by this, researchers have developed TabPFN (Tabular Prior-Data Fitted Networks) [16], [45], a transformer-based model designed for tabular data classification. Unlike traditional supervised learning, which trains models per dataset, TabPFN is pre-trained on millions of synthetic tabular datasets, each representing a different prediction task. At inference time, the model processes an entire dataset in a single forward pass, eliminating the need for dataset-specific training, manual

feature engineering and pre-processing and thus with minimal human supervision or domain expertise.

The key to TabPFN's success is its synthetic dataset generation process. The authors leverage Bayesian Neural Networks (BNNs) and Structural Causal Models (SCMs) to create realistic training data. SCMs are particularly useful as they encode real-world causal relationships, ensuring that generated datasets reflect natural feature correlations. The synthetic datasets are designed to exhibit realistic feature interactions, varying levels of importance, and blockwise feature correlations, making the pre-trained model robust to diverse real-world scenarios.

Traditional ML models often struggle with out-of-distribution predictions and knowledge transfer between datasets. TabPFN addresses this by encoding a learned algorithm from its extensive synthetic training, enabling it to generalize across a broad range of tabular datasets without requiring fine-tuning. This shift from explicit algorithm design to learning from diverse input-output examples marks a significant advancement in tabular learning, suggesting a future where deep learning-based models can finally outperform traditional ML methods in tabular domains, reducing the need for manual feature engineering and dataset-specific optimizations.

IV. Motivation

As outlined in Section III-D, Machine Learning (ML) techniques have shown significant promise in semiconductor manufacturing, particularly for tasks such as MCU performance screening. However, traditional ML workflows typically involve substantial human intervention: from manual feature engineering to hyperparameter tuning and dataset-specific optimization. These requirements can slow down model development, limit scalability, and hinder adoption in industrial environments where domain expertise and labeled data are often scarce.

Moreover, conventional ML models often exhibit limited generalization capabilities across different manufacturing nodes or product generations. Transfer learning remains challenging due to the tight coupling between model parameters and the specific characteristics of the training dataset. In high-variability production contexts, this reduces the robustness and reusability of trained models, increasing time-to-market and operational costs.

Recent advances in foundation models, particularly Prior-Data Fitted Networks (PFNs) and In-Context Learning (ICL), offer a compelling alternative. In particular, TabPFN—a transformer-based foundation model for tabular data—has demonstrated the ability to make accurate predictions in a zero-shot or few-shot setting, relying on a learned prior derived from synthetically generated tasks. Unlike classical ML pipelines, TabPFN performs inference via a single forward pass without requiring dataset-specific retraining, hyperparameter optimization, or hand-crafted features.

These properties make TabPFN especially attractive for semiconductor applications, where data collection is expensive and engineering resources are constrained. Tabular foundation models like TabPFN require minimal human supervision and can operate directly on raw or lightly preprocessed features, significantly accelerating deployment and reducing dependency on ML expertise.

This study aims to evaluate the suitability of TabPFN for MCU performance screening by benchmarking it on two real-world semiconductor datasets and comparing its performance to traditional ML baselines. Our motivations are threefold:

- **Minimizing Human Supervision** – TabPFN eliminates the need for extensive manual feature engineering and model tuning, offering a plug-and-play alternative that reduces development overhead and expertise barriers.
- **Capturing Complex Feature Interactions** – Thanks to its transformer-based architecture, TabPFN is well-equipped to model intricate and nonlinear relationships often present in tabular semiconductor data, surpassing the limitations of tree-based or linear models.
- **Enabling Generalization Across Product Lines** – By leveraging prior-based meta-learning, TabPFN has the potential to generalize across manufacturing conditions, enabling performance screening with minimal additional data when transitioning between process technologies or product generations.

To further investigate its robustness, we also explore TabPFN's ability to generalize by unifying data from two distinct MCU products into a single predictive task. This experiment serves to assess the model's scalability and its potential to capture shared performance-relevant patterns across varying manufacturing contexts.

Thus, this work explores whether foundation models—originally designed for general-purpose learning—can offer a viable, low-effort, alternative to conventional ML pipelines in semiconductor performance prediction tasks.

V. Experimental Setup

The proposed methodology was validated on two real-world datasets, referred to as A and B, corresponding to two distinct MCU product families.

Dataset A comprises 1,148 labeled samples from product family A, which integrates six distinct SMON modules on-chip. Each module includes approximately 130 individual SMONs. Dataset B contains 2,002 labeled samples from product family B, incorporating five SMON modules per chip.

For both datasets, an 80%-20% train-test split was adopted. To ensure statistical robustness, all experiments were repeated six times using different random seeds. Final performance metrics are reported as the average across these six independent runs.

All experiments were conducted using Python on a computing server equipped with an Intel® Core™ i9-9900K CPU @3.60GHz (16 threads), 32GB RAM, and an Nvidia® 2080 TI GPU.

A. Preprocessing and Feature Scaling

As a preprocessing step, all input features were standardized using a *Standard Scaler*, which removes the mean and scales

each feature to unit variance. This transformation is applied independently to each training set to prevent data leakage.

B. Baseline Models

To assess the performance of TabPFN, we benchmarked it against established ML models previously proposed for similar tasks [4], [28], [46]. The baseline methods include:

- **Random Forest (RF)** [47] and **XGBoost** [10] — tree-based ensemble models, known for their robustness and ability to capture non-linear interactions.
- **Ridge Regression** — a linear model with L_2 regularization, used both with and without polynomial feature transformations.

Hyperparameter tuning for all models was performed via random search, with evaluation based on 5-fold cross-validation. For RF and XGBoost, 300 random hyperparameter configurations were tested. For Ridge Regression, the regularization coefficient α was selected from a grid of approximately 4,000 values.

Previous studies [4] showed that for product B, the best performance is achieved using a degree-2 polynomial expansion of the input features, referred to as *Polynomial Ridge* (or *Poly Ridge*). In contrast, for product A, a simple linear Ridge Regression model yielded the best results.

C. Cross-Dataset Generalization

To evaluate the generalization capability of TabPFN, we designed a cross-dataset experiment by combining datasets A and B into a unified training task. This setup aims to assess the model's ability to learn common patterns between different semiconductor products despite underlying distributional shifts.

To address distribution mismatch, we independently normalized the input features of each dataset using separate *Standard Scalers* fitted to their respective training partitions. This preserves the internal structure of each dataset while aligning feature ranges across sources. The target variable $F_{\max}$ was also scaled separately for each dataset to maintain consistency.

D. Evaluation Metrics

Model performance was assessed using the following regression metrics:

- **Normalized Root Mean Square Error (nRMSE)**: RMSE divided by the mean $F_{\max}$ of the test set.
- **Normalized Mean Absolute Error (nMAE)**: MAE divided by the mean $F_{\max}$ of the test set.
- **Coefficient of Determination (R^2)**: measures the proportion of variance in the target variable explained by the model [48]. An ideal model yields $R^2 = 1$, while a naive model predicting the mean yields $R^2 = 0$.

Learning curves were also analyzed to evaluate the impact of varying the number of labeled samples on model performance, particularly in data-scarce regimes.

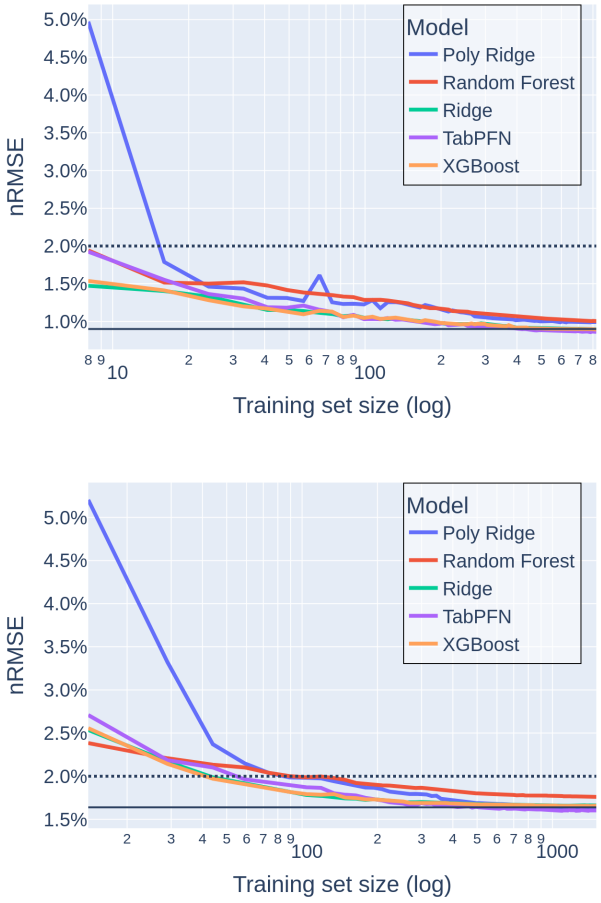

Fig. 1. Learning curves for different models on dataset A (upper) and B (lower). TabPFN exhibits stable and efficient learning, achieving competitive accuracy compared to other baselines.

VI. Results

A. Learning Curve Analysis

Figure 1 show the learning curves for Product A and Product B. These illustrate nRMSE as a function of the training set size (in log scale). For both product, Ridge regression, Random Forest, XGBoost, and TabPFN maintain relatively lower errors across all training sizes, with TabPFN achieving the lowest nRMSE at larger training sizes, outperforming all baseline models once the training set exceeded about 200 samples and achieving the lowest final nRMSE.

For product B, the best-performing baseline, Polynomial Ridge Regression, required at least 100 samples to reach a satisfactory nRMSE below 2%. Other models, such as Ridge Regression and XGBoost, achieved this 2% threshold with approximately 44 samples. Notably, XGBoost demonstrated the strongest ability to generalize with minimal data, reaching 1.66% nRMSE with only 44 samples, though still slightly worse than the 1.60% of TabPFN at full dataset capacity.

Interestingly, Random Forest showed the lowest nRMSE (2.38%) when trained on only 14 samples. However, its inability to scale effectively with larger datasets resulted in

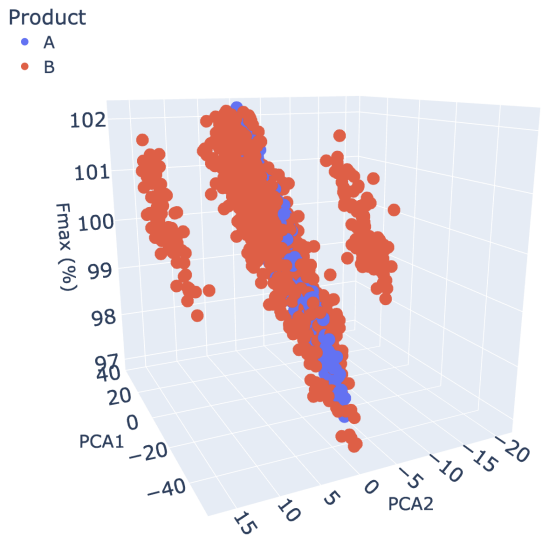

Fig. 2. Dataset consisting of both Product A and Product B, each normalized separately. The relationship between SMONs and $F_{\max}$ is distinctly pseudo-linear for both products. The grouping within Split-Lots is clearly visible for Product B.

the highest error among all models when trained on the full dataset (1,481 samples).

These results indicate that TabPFN performs competitively against well-established models, particularly in data-limited scenarios, while Ridge regression and XGBoost offer robust and stable generalization across different datasets.

In our cases, datasets are relatively "easy" to fit with linear models performing good. However, as the complexity of datasets growth, TabPFN could result in a good out-of-the-box alternative, as shown in recent benchmarks [16]

B. Composed Dataset

The resulting composite dataset is illustrated in Figure 2. The 3D scatter plot presents the first two principal components against $F_{\max}$, providing a visual representation of the data distribution across both product families.

After standardizing both features and labels, the relationship between SMONs and $F_{\max}$ appears approximately pseudo-linear for both products. Notably, the structure of Split-Lots is observable, with some wafers of Product B deviating from the central distribution.

For this task, only the SMONs common to both products were selected, resulting in approximately 380 features. All baseline models were trained on the composite dataset using SMON features, augmented with a binary artificial feature indicating the product family (A or B). This additional feature allows traditional models to differentiate between the two products while learning a common predictive function.

For the TabPFN approach, we evaluated two configurations: (i) a plain model trained without hyperparameter tuning and (ii) an optimized model, which performs an automated hyperparameter search and constructs an ensemble of high-performing configurations. The latter model was trained with a fixed time budget to ensure comparability.

Experimental results, summarized in Table I, indicate that while specialized models achieve the best performance when trained separately on individual products, TabPFN demonstrates superior generalization across both products in the composite dataset, achieving the lowest prediction error. TabPFN consistently outperforms traditional models in both datasets. Auto-TabPFN and the out-of-the-box model yielded comparable results, further confirming the robustness of the approach and the lack of necessity of performing extensive hyperparameter tuning. This highlights the potential of foundation models like TabPFN for developing a unified, general-purpose predictor capable of handling multiple product families without requiring extensive retraining.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON DATASETS A, B, AND ALL.

Method	Train Set	NRMSE	NMAE	R^2	Elapsed (s)
Dataset A					
AutTabPFN	A	0.8865	0.6909	96.46	1512.32
AutTabPFN	A+B	0.9469	0.7432	95.96	1335.44
TabPFN	A	**0.8824**	**0.6841**	**96.48**	9.26
TabPFN	A+B	0.9502	0.7470	95.93	28.40
Ridge	A	0.9151	0.7157	96.22	0.56
Ridge	A+B	1.2788	0.8177	91.97	2.10
Poly Ridge	A	1.0219	0.7917	95.29	0.18
Poly Ridge	A+B	1.0401	0.7816	95.13	1.03
R. Forest	A	1.0214	0.7901	95.31	165.53
R. Forest	A+B	1.0539	0.8180	95.00	452.76
XGBoost	A	0.9737	0.7584	95.72	109.12
XGBoost	A+B	1.1898	0.8203	92.90	381.31
Dataset B					
AutTabPFN	B	1.5631	1.2121	88.90	1512.32
AutTabPFN	A+B	**1.5621**	**1.2126**	**88.91**	1335.44
TabPFN	B	1.5682	1.2166	88.82	21.10
TabPFN	A+B	1.5654	1.2155	88.86	28.40
Ridge	B	1.6355	1.2904	87.88	1.52
Ridge	A+B	1.6525	1.2988	87.62	2.10
Poly Ridge	B	1.5904	1.2616	88.54	0.81
Poly Ridge	A+B	1.6053	1.2728	88.33	1.03
R. Forest	B	1.7290	1.3557	86.45	283.7
R. Forest	A+B	1.7519	1.3774	86.09	452.76
XGBoost	B	1.6385	1.2936	87.83	250.31
XGBoost	A+B	1.6806	1.3248	87.21	381.31
Dataset A+B					
AutTabPFN	A+B	1.2549	0.9472	99.45	1512.32
TabPFN	A+B	**1.2507**	**0.9480**	**99.45**	28.40
Ridge	A+B	1.4081	1.0354	99.31	2.10
Poly Ridge	A+B	1.3573	1.0347	99.36	1.03
R. Forest	A+B	1.3648	1.0479	99.35	452.76
XGBoost	A+B	1.3494	1.0303	99.36	381.31

In addition to predictive performance, computational efficiency is a crucial factor in model selection, especially when scaling to larger datasets. Auto TabPFN, exhibits the longest execution times, reaching over 1500 seconds for some experiments. Its predictive performance remains very close to the out-of-the-box TabPFN, which achieves similar R^2 scores while running significantly faster (only 28 seconds for the

largest dataset). Traditional methods like Ridge and Poly Ridge complete execution in under 3 seconds, but their accuracy is generally lower. Although execution times are measured in seconds in these experiments, as datasets grow—both in the number of products and sample size—runtime constraints will become more significant. TabPFN's out-of-the-box model provides a strong balance between speed and accuracy, making it a promising option when rapid experiments are required while still maintaining high generalization capability.

VII. Conclusions

In this preliminary study, we investigated the use of Tabular Foundation Models, specifically TabPFN, for MCU performance screening. We evaluated the generalization capability and computational efficiency of different machine learning models on two different MCU product and on a composite dataset comprising multiple products. Our results demonstrate that TabPFN, in its out-of-the-box configuration, consistently provides a strong balance between predictive performance, human supervision and computational efficiency in the development phase, without the need of extensive hyperparameter tuning.

As datasets grow in complexity—with more products, larger training sets, and increasing demands for hyperparameter optimization—TabPFN's efficiency makes it a compelling solution for real-world applications requiring robust generalization with minimal tuning efforts.

Future work could explore the impact of dataset heterogeneity on model performance and investigate additional optimization strategies for balancing accuracy and runtime in industrial-scale applications.

References

[1] N. Bellarmino *et al.*, "Microcontroller Performance Screening: Optimizing the Characterization in the Presence of Anomalous and Noisy Data," in *IEEE International Symposium on On-Line Testing and Robust System (IOLTS)*, 2022.

[2] R. Cantoro *et al.*, "Machine Learning based Performance Prediction of Microcontrollers using Speed Monitors," in *IEEE International Test Conference (ITC)*, 2020.

[3] N. Bellarmino *et al.*, "Exploiting active learning for microcontroller performance prediction," in *IEEE European Test Symposium (ETS)*, 2021.

[4] N. Bellarmino *et al.*, "A Multi-Label Active Learning Framework for Microcontroller Performance Screening," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2023.

[5] J. Chen *et al.*, "Data learning techniques and methodology for Fmax prediction," in *IEEE International Test Conference (ITC)*, 2009.

[6] J. Chen *et al.*, "Selecting the most relevant structural Fmax for system Fmax correlation," in *28th VLSI Test Symposium (VTS)*, 2010.

[7] M. Sadi *et al.*, "SoC Speed Binning Using Machine Learning and On-Chip Slack Sensors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2017.

[8] K. G. Liakos *et al.*, "Machine learning in agriculture: A review," *Sensors*, 2018.

[9] S. Zhang *et al.*, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, Feb. 2019.

[10] T. Chen *et al.*, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, ACM, Aug. 2016.

[11] L. Prokhorenkova *et al.*, *Catboost: Unbiased boosting with categorical features*, 2019.

[12] L. Grinsztajn *et al.*, *Why do tree-based models still outperform deep learning on tabular data?* 2022.

[13] Q. Dong *et al.*, *A survey on in-context learning*, 2024.

[14] T. B. Brown *et al.*, *Language models are few-shot learners*, 2020.

[15] B. van Breugel *et al.*, *Why tabular foundation models should be a research priority*, 2024.

[16] N. Hollmann *et al.*, *Tabpfn: A transformer that solves small tabular classification problems in a second*, 2023.

[17] K. von Arnim *et al.*, "An Effective Switching Current Methodology to Predict the Performance of Complex Digital Circuits," in *IEEE International Electron Devices Meeting (IEDM)*, 2007.

[18] G. Sannena *et al.*, "Low overhead warning flip-flop based on charge sharing for timing slack monitoring," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2018.

[19] T. B. Chan *et al.*, "DDRO: A novel performance monitoring methodology based on design-dependent ring oscillators," in *Thirteenth International Symposium on Quality Electronic Design (ISQED)*, May 2012.

[20] F. Angione *et al.*, "Test, reliability and functional safety trends for automotive system-on-chip," in *2022 IEEE European Test Symposium (ETS)*, 2022.

[21] H. Ayari *et al.*, "Making predictive analog/rf alternate test strategy independent of training set size," in *IEEE International Test Conference (ITC)*, 2012.

[22] P. Variyam *et al.*, "Prediction of analog performance parameters using fast transient testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2002.

[23] H.-G. Stratigopoulos *et al.*, "Error moderation in low-cost machine-learning-based analog/rf testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2008.

[24] J. Brockman *et al.*, "Predictive subset testing: Optimizing ic parametric performance testing for quality, cost, and yield," *IEEE Transactions on Semiconductor Manufacturing*, 1989.

[25] H.-G. Stratigopoulos *et al.*, "Defect filter for alternate rf test," in *2010 15th IEEE European Test Symposium*, 2010.

[26] H.-G. Stratigopoulos *et al.*, "Nonlinear decision boundaries for testing analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005.

[27] N. Bellarmino *et al.*, "Semi-Supervised Deep Learning for Microcontroller Performance Screening," in *IEEE European Test Symposium (ETS)*, 2023.

[28] N. Bellarmino *et al.*, "Feature Selection for Cost Reduction In MCU Performance Screening," in *IEEE 24th Latin American Test Symposium (LATS)*, 2023.

[29] N. Bellarmino *et al.*, "Deep learning strategies for labeling and accuracy optimization in microcontroller performance screening," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[30] N. Bellarmino *et al.*, *Covid-19 detection from exhaled breath*, 2023.

[31] N. Bellarmino *et al.*, "U-flex: Unsupervised feature learning with evolutionary exploration," in *Machine Learning, Optimization, and Data Science*, G. Nicosia *et al.*, Eds., Cham: Springer Nature Switzerland, 2024.

[32] Y. Zhu *et al.*, "Converting tabular data into images for deep learning with convolutional neural networks," *Scientific Reports*, May 2021.

[33] H.-J. Ye *et al.*, *A closer look at deep learning methods on tabular datasets*, 2025.

[34] S. B. Hoo *et al.*, *The tabular foundation model tabpfn outperforms specialized time series forecasting models based on simple features*, 2025.

[35] S. Müller *et al.*, *Transformers can do bayesian inference*, 2024.

[36] T. Ruokonen *et al.*, *Fault Detection, Supervision, and Safety for Technical Processes* (SAFEPROCESS'94 : IFAC Symposium, Helsinki University of Technology). International Federation of Automatic Control, 1994.

[37] G. D. Natale *et al.*, *Cross-Layer Reliability of Computing Systems*. Jan. 2020.

[38] R. McLaughlin *et al.*, "Automated Debug of Speed Path Failures Using Functional Tests," in *27th IEEE VLSI Test Symposium*, 2009.

[39] S. Asai, Ed., *VLSI Design and Test for Systems Dependability*. Springer Japan, 2019.

[40] Y. LeCun *et al.*, "Deep Learning," en, *Nature*, May 2015.

[41] N. Bellarmino *et al.*, "Investigating on Gradient Regularization for Testing Neural Networks," in *LOD 2024 : 10th International Confer-*

ence on machine Learning, Optimization and Data science, Riva de Sole, Italy, Sep. 2024.
[42] C. Tan *et al.*, "A Survey on Deep Transfer Learning," *27th International Conference on Artificial Neural Networks (ICANN)*, 2018.
[43] C. Yang *et al.*, *Gated convolutional networks with hybrid connectivity for image classification*, 2019.
[44] R. Bommasani *et al.*, "On the opportunities and risks of foundation models," *CoRR*, 2021.
[45] N. Hollmann *et al.*, "Accurate predictions on small data with a tabular foundation model," en, *Nature*, Jan. 2025.
[46] N. Bellarmino *et al.*, "Cosmo: Compressed sensing for models and logging optimization in mcu performance screening," *IEEE Transactions on Computers*, 2024.
[47] L. Breiman, "Random forests," en, *Machine Learning*, Oct. 2001.
[48] D. Chicco *et al.*, *The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation*, en, Jul. 2021.