Pre-trained Forecasting Models: Strong Zero-Shot Feature Extractors for Time Series Classification

Andreas Auer [1,2] **Daniel Klotz** [3] **Sebastian Böck** [1] **Sepp Hochreiter** [1,2]

[1]NXAI GmbH, Linz, Austria
[2]ELLIS Unit, LIT AI Lab, Institute for Machine Learning, JKU Linz, Austria
[3]Interdisciplinary Transformation University Austria, Linz, Austria

Abstract

Recent research on time series foundation models has primarily focused on forecasting, leaving it unclear how generalizable their learned representations are. In this study, we examine whether frozen pre-trained forecasting models can provide effective representations for classification. To this end, we compare different representation extraction strategies and introduce two model-agnostic embedding augmentations. Our experiments show that the best forecasting models achieve classification accuracy that matches or even surpasses that of state-of-the-art models pre-trained specifically for classification. Moreover, we observe a positive correlation between forecasting and classification performance. These findings challenge the assumption that task-specific pre-training is necessary, and suggest that learning to forecast may provide a powerful route toward constructing general-purpose time series foundation models.

1 Introduction

In time series forecasting, foundation models are becoming increasingly prominent. They are large models that are pre-trained on broad data, and therefore have the ability to generalize across unseen datasets [3, 33, 14, 13, 6]. New benchmarks with public leaderboards such as GiftEval [1] and BOOM [13] have accelerated advances in state-of-the-art methods. Apart from forecasting, Time Series Classification (TSC) is another key application in time series analysis.

Earlier general-purpose time series models [20, 19] evaluated multiple downstream tasks, but recent work shows that they have failed to reach state-of-the-art performance in either forecasting or classification [17, 16]. More recently, the majority of newly introduced foundation models [6, 13, 25, 32, 22] have been optimized specifically for forecasting, and only few have focused on classification [17, 24]. Some argue that pre-training objectives should be aligned with downstream applications, for example, contrastive objectives for classification or masked reconstruction for imputation [17]. This perspective suggests that task-specialized pre-training may be necessary for optimal performance, which is in contrast to language and vision foundation models, where a single pre-trained model often transfers effectively across many diverse tasks [11, 12].

This contrast motivates our central research question: **How well do representations from pre-trained forecasting models transfer to classification tasks?** To answer this question, we evaluate a diverse set of forecasting models as frozen feature extractors on TSC benchmarks, analyze key design choices for representation extraction, and investigate the role of model architectures. Beyond the direct application to classification, our study aims to provide broader insights into the generalizability of learned representations, which is a step toward developing true time series foundation models.

Our contributions are as follows: **(1)** We show that representations from pre-trained forecasting models yield classification accuracy on par with, and in some cases surpassing state-of-the-art

NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT^{2}S).

models pre-trained explicitly for classification. **(2)** We analyze design decisions for leveraging forecasting models in classification, providing practical guidance for future applications. **(3)** We propose two model-agnostic representation augmentations that incorporate absolute statistical features and differentiated series to further improve classification performance.

The remainder of the paper introduces the problem setup, details our methodology for using forecasting models as feature extractors (Section 2), presents the experimental setup (Section 3) and results (Section 4), and concludes with key findings (Section 5).

Problem Setup: Time Series Classification The TSC task is defined over a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where each sample consists of a time series $\mathbf{x}_i$ and its corresponding class label y_i. A time series $\mathbf{x}_i \in \mathbb{R}^{T \times V}$ is a sequence of T observations over V variates, and the label y_i belongs to one of K discrete classes. The objective is to learn a model that can accurately predict the label for a new, unseen time series.

2 Zero-Shot Forecasting Models as Classification Models

We leverage pre-trained time series forecasting models as feature extractors. Instead of training a classifier on the raw time series $\mathbf{x}_i$, we use a pre-trained model E to map $\mathbf{x}_i$ to a latent representation $\mathbf{z}_i = E(\mathbf{x}_i)$, which is then fed into a simple classifier C_L to output the final prediction $\hat{y}_i = C_L(\mathbf{z}_i)$. We refer to $\mathbf{z}_i$ also as *embedding* of $\mathbf{x}_i$.

We exclusively use a zero-shot protocol for these models, meaning the parameters of the pre-trained model E are frozen and never fine-tuned. For each TSC dataset, we only train a standard out-of-the-box classifier C_L on top of the embeddings produced by E. This approach allows us to isolate and evaluate the quality and generalizability of the representations learned by the forecasting models.

Embedding Extraction & Aggregation. Most state-of-the-art forecasting models do not specify a canonical method for extracting a single, fixed-size embedding for an entire time series. However, as the majority utilize a transformer(-like)[1], block-based architecture, we can extract hidden states at various points in the network. This presents two key design choices: how to aggregate information along (1) layer and (2) sequence dimensions. We hypothesize that simply using the output from the final token of the final layer is suboptimal. First, it is unclear which layer contains the best abstraction and transferable representation, as deeper layers often specialize to the original pre-training task, losing generalizability [34, 2]. Second, relying on the last sequence position may neglect important information contained earlier in the series.

We investigate different aggregation strategies in our ablations. For our main experiments, we apply mean pooling across the sequence dimension and concatenate these layer-wise representations. This sequence-pooling strategy also inherently handles datasets with variable-length time series, ensuring a fixed-size embedding dimension. The ablation study in Appendix C.2 confirms that aggregating across both dimensions is crucial.

Multivariate Data & Univariate Models. Most top-performing pre-trained forecasting models are univariate. For multivariate time-series classification, we adopt a proven forecasting technique: treating each variate independently [28, 6]. We therefore process each of the V variates independently through the frozen model E to yield V separate embeddings.

The subsequent design choice is how to aggregate these per-variate embeddings into a single representation. We hypothesize that pooling discards variate-specific information, while concatenation preserves it. Accordingly, we concatenate the per-variate embeddings in our main experiments, a choice empirically confirmed by our ablation studies (Appendix C.2), which show concatenation consistently outperforms pooling. We apply the same strategy to multivariate models that output per-variate embeddings.

2.1 Embeddings Augmentations

Absolute Sample Statistics. A common characteristic of pre-trained forecasting models is the use of instance normalization. While effective for forecasting, this removes all information regarding

[1] TiRex [6] uses xLSTM [9] instead of a Transformer but still employs a block-based architecture [8]

	Type	ZS	Univariate		Multivariate		Overall	
			No Aug	Stat+Diff	No Aug	Stat+Diff	No Aug	Stat+Diff
TiRex	Dec	yes	**0.80**	**0.81**	**0.74**	**0.74**	**0.79**	**0.80**
Chr. Bolt (Base)	EncDec	yes	0.77	0.79	0.72	**0.74**	0.76	0.78
Moirai (Large)	Enc	yes	0.79	0.80	0.70	0.70	0.78	0.78
TimesFM 2.0	Dec	yes	0.79	0.79	0.70	0.70	0.77	0.78
TimesFM 1.0	Dec	yes	0.74	0.75	0.71	0.72	0.73	0.74
Chronos (Base)	EncDec	yes	0.71	0.76	0.71	0.72	0.71	0.75
Toto	Dec	yes	0.71	0.74	0.71	0.70	0.71	0.73
Mantis	Enc	no	0.79		**0.74**		0.78	
NuTime	Enc	no	0.67		0.68		0.67	
Moment (Large)	Enc	no	0.63		0.57		0.62	
DTW	-	-	0.73		0.72		0.73	

Table 1: Classification accuracy of different models for the univariate, multivariate, and combined benchmark (Random Forest). "Stat+Diff" shows results with both proposed augmentations applied; "no Aug" utilizes the pure forecasting model representations. "ZS" indicates models that did not have access to the benchmarks training data during pre-training.

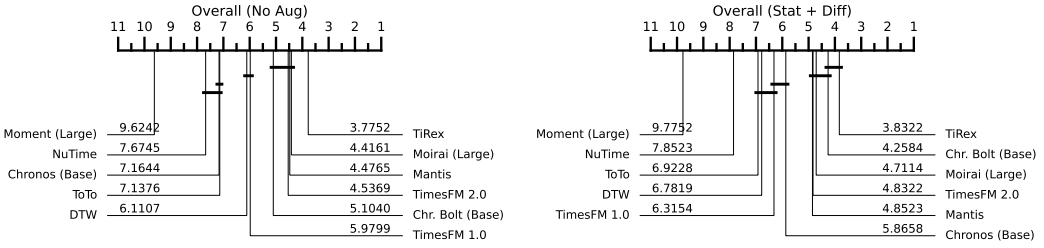


Figure 1: Critical difference plot of the average accuracy ranks for the evaluated models across the combined benchmark datasets (Random Forest). Left without augmentation; right with augmentations. Models connected by a bar are not significantly different (Wilcoxon signed-rank test).

the absolute values and scale of the time series. We hypothesize that for many classification tasks, this information might be an important discriminative signal. To recover it, we propose to augment the model's embedding with basic sample statistics. We divide the input time series $\mathbf{x}_i$ into k non-overlapping patches ($k = 8$ in our main experiments). For each patch, we calculate its mean, standard deviation, minimum, and maximum values. These statistics are then concatenated with the embedding $\mathbf{z}_i$ from the model to form the final representation. Using a fixed number of patches ensures the resulting feature vector has a consistent size.

Time Series Differencing. Time series may contain strong trends that can dominate the signal and mask more subtle patterns. To isolate these patterns, we propose to employ first-order differencing. We generate a new, differenced time series by taking the difference between consecutive time steps ($\mathbf{x}'_t = \mathbf{x}_t - \mathbf{x}_{t-1}$). This transformation, inspired by classical time series analysis, removes the local trend, making the resulting series more stationary and emphasizing step-to-step changes. The differenced series is then processed by the same pre-trained model to produce a second embedding, which is concatenated to the original embedding.

3 Experiments

Our evaluation uses the UCR [15] and UEA [7] archives, comprising 127 univariate and 30 multivariate classification datasets with predefined train/test splits. We excluded 5 datasets with sample lengths exceeding 2048 and 2 others due to processing problems. We evaluate a set of leading pre-trained forecasting models, including TiRex [6], Chronos (Bolt) [3], TimesFM [14], and Moirai [33] — including the newest and previous model generations and different sizes. These are compared against Moment [20], a "general" pre-trained model, the classification-specific pre-trained models NuTime [24] and Mantis [17], and Dynamic Time Warping (DTW) [10] as a baseline. For each pre-trained

model, we extract embeddings and train a Random Forest, a linear layer, and a kNN classifier on top — and evaluate accuracy. Details on the experiment setup are presented in Appendix B.

4 Results

This section reports results using the best-performing classifier (Random Forest) and the largest model size for each model. The main results are summarized in Table 1 and Figure 1. Full results for all classifiers, model sizes, and ablations are available in Appendix C. In the following, we discuss the individual aspects of our main findings.

Forecasting Models are Effective Zero-Shot Feature Extractors. The best forecasting models achieve accuracies competitive with or exceeding Mantis, a state-of-the-art model designed for this classification task. This result is particularly interesting because the forecasting models had no exposure to the classification benchmarks during their pre-training, unlike Mantis and NuTime, which were also pre-trained on the training split of the benchmarks. The results are robust across other classifier (Appendix C.1), metrics (Appendix C.4), and benchmark configuration (Appendix C.5). This suggests that pre-training towards forecasting tasks might be a viable path for generating general-purpose time series representations.

Forecasting and Classification Performance Correlate. We observe a positive correlation between a model's performance on the GiftEval forecasting benchmark [1] and its classification accuracy (Figure 2). The trend has considerable noise, with notable underperformance from Chronos (potentially due to missing patch processing) and Toto. However, overall this trend suggests that the features learned for accurate forecasting are transferable to classification tasks.

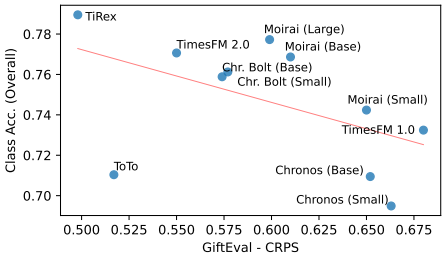


Figure 2: Classification accuracy versus forecasting performance (CRPS on GiftEval) of the evaluated models. The trend (red line) shows that better forecasting ability (lower CRPS) relates to higher classification accuracy.

Impact of Model Architecture. Results do not point to a superior architectural paradigm (Encoder, Decoder, Encoder-Decoder). Both, the top- and low-performing models, are diverse in that regard. Regarding base architecture, TiRex, as the only non-Transformer model, performs best. If the forecast advantage stems from its state-tracking capability, as prior work suggests [6], then this benefit seems to transfer to classification, implying a better general representation.

Efficacy of Augmentations. The proposed augmentations improve the results across most models — the significance regarding the signed rank test varies between the results. Detailed results including ablation of the individual augmentations and a qualitative analysis are presented in Appendix C.3.

5 Conclusion

This work demonstrates that pre-trained forecasting models are effective zero-shot feature extractors for time series classification. We found that representations from strong forecasting models match or even exceed the performance of specialized classification models — particularly noteworthy as the forecasting models did not pre-train with benchmark training data, while the classification-specific models did. This finding, combined with a positive correlation between forecasting and classification performance, questions the need for task-specific pre-training.

Limitations & Future Work The work focuses on a zero-shot evaluation protocol and does not include fine-tuning. This choice ensures a fair comparison of the base representations, as optimal fine-tuning strategies might be highly model-specific. The work also omits a direct comparison to task-specific and supervised classifiers; however, prior work [17, 24, 20] has already shown that the pre-trained classification models we evaluate are competitive with these. Future work could probe the generalizability of these representations on other tasks, such as anomaly detection.

Acknowledgments and Disclosure of Funding

The ELLIS Unit Linz, the LIT AI Lab, and the Institute for Machine Learning are supported by the Federal State Upper Austria.

References

[1] T. Aksu, G. Woo, J. Liu, X. Liu, C. Liu, S. Savarese, C. Xiong, and D. Sahoo. GIFT-eval: A benchmark for general time series forecasting model evaluation. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

[2] B. Alkin, L. Miklautz, S. Hochreiter, and J. Brandstetter. Mim-refiner: A contrastive learning boost from intermediate pre-trained representations. *ArXiv*, 2402.10093, 2024.

[3] A. F. Ansari, L. Stella, A. C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and B. Wang. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*, May 2024.

[4] A. F. Ansari, C. Turkmen, O. Shchur, and L. Stella. Fast and accurate zero-shot forecasting with Chronos-Bolt and AutoGluon. `https://aws.amazon.com/blogs/machine-learning/fast-and-accurate-zero-shot-forecasting-with-chronos-bolt-and-autogluon/`, 2024. AWS Machine Learning Blog.

[5] A. Auer, R. Parthipan, P. Mercado, A. F. Ansari, L. Stella, B. Wang, M. Bohlke-Schneider, and S. S. Rangapuram. Zero-shot time series forecasting with covariates via in-context learning. *ArXiv*, 2506.03128, 2025.

[6] A. Auer, P. Podest, D. Klotz, S. Böck, G. Klambauer, and S. Hochreiter. Tirex: Zero-shot forecasting across long and short horizons. In *1st ICML Workshop on Foundation Models for Structured Data*, 2025.

[7] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The UEA multivariate time series classification archive, 2018. *arXiv*, 1811.00075, 2018.

[8] M. Beck, K. Pöppel, P. Lippe, R. Kurle, P. M. Blies, G. Klambauer, S. Böck, and S. Hochreiter. xLSTM 7B: A Recurrent LLM for Fast and Efficient Inference. *International Conference on Machine Learning*, 2025.

[9] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. K. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. xLSTM: Extended Long Short-Term Memory. In *Advances in Neural Information Processing Systems*, Nov. 2024.

[10] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370, 1994.

[11] R. Bommasani. On the opportunities and risks of foundation models. *ArXiv*, 2108.07258, 2021.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[13] B. Cohen, E. Khwaja, Y. Doubli, S. Lemaachi, C. Lettieri, C. Masson, H. Miccinilli, E. Ramé, Q. Ren, A. Rostamizadeh, J. O. du Terrail, A.-M. Toon, K. Wang, S. Xie, Z. Xu, V. Zhukova, D. Asker, A. Talwalkar, and O. Abou-Amal. This time is different: An observability perspective on time series foundation models. *ArXiv*, 2505.14766, 2025.

[14] A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, pages 10148–10167. PMLR, July 2024.

[15] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML. The ucr time series classification archive. *arXiv*, 1810.07758, 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

[16] V. Ekambaram, A. Jati, P. Dayama, S. Mukherjee, N. H. Nguyen, W. M. Gifford, C. Reddy, and J. Kalagnanam. Tiny Time Mixers (TTMs): Fast Pre-trained Models for Enhanced Zero/Few-Shot Forecasting of Multivariate Time Series. *ArXiv*, 2401.03955, 2024.

[17] V. Feofanov, S. Wen, M. Alonso, R. Ilbert, H. Guo, M. Tiomoko, L. Pan, J. Zhang, and I. Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*, 2025.

[18] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems*, 2019.

[19] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik. UniTS: A unified multi-task time series model. In *Advances in Neural Information Processing Systems*, 2024.

[20] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.

[21] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023.

[22] S. B. Hoo, S. Müller, D. Salinas, and F. Hutter. The tabular foundation model tabpfn outperforms specialized time series forecasting models based on simple features. *ArXiv*, 2501.02945, 2025.

[23] Z. Li, Z. Rao, L. Pan, P. Wang, and Z. Xu. Ti-mae: Self-supervised masked time series autoencoders. *arXiv*, 2301.08871, 2023.

[24] C. Lin, X. Wen, W. Cao, C. Huang, J. Bian, S. Lin, and Z. Wu. Nutime: Numerically multi-scaled embedding for large-scale time-series pretraining. *Transactions on Machine Learning Research (TMLR)*, 2024.

[25] Y. Liu, G. Qin, Z. Shi, Z. Chen, C. Yang, X. Huang, J. Wang, and M. Long. Sundial: A family of highly capable time series foundation models. In *International Conference on Machine Learning*, 2025.

[26] M. Middlehurst, A. Ismail-Fawaz, A. Guillaume, C. Holder, D. Guijo-Rubio, G. Bulatova, L. Tsaprounis, L. Mentel, M. Walter, P. Schäfer, and A. Bagnall. Aeon: A python toolkit for learning from time series. *Journal of Machine Learning Research*, 25(289):1–10, 2024.

[27] M. Middlehurst, P. Schäfer, and A. Bagnall. Bake off redux: A review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4):1958–2031, 2024.

[28] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*, Sept. 2022.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[30] K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Biloš, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting. *ArXiv*, 2310.08278, 2024.

[31] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 35(2):401–449, 2021.

[32] X. Wang, T. Zhou, J. Gao, B. Ding, and J. Zhou. Output scaling: Yinglong-delayed chain of thought in a large pretrained time series forecasting model. *ArXiv*, 2506.11029, 2025.

[33] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified Training of Universal Time Series Forecasting Transformers. In *Forty-First International Conference on Machine Learning*, June 2024.

[34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27, 2014.

[35] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu. Ts2vec: Towards universal representation of time series. In *AAAI conference on artificial intelligence*, 2022.

[36] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[37] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, 2023.

Appendix

Table of Contents

A Related Work

Pre-trained foundation models have become popular in time series analysis. Early explorations adapted Large Language Model (LLM) for time series tasks [21], while more recent models typically only borrow the architecture from LLM's but pre-train with time series tasks and data. While there is a recent focus on forecasting [30, 33, 14, 3, 16, 6, 13, 25, 22, 5], other literature has explored models for a wider range of downstream tasks, including classification: General-purpose models like Moment [20], GPT4TS, [37], and UniTS [19] address classification alongside other tasks. More specialized models, like Mantis [17] and NuTime [24] focus specifically on pre-training for classification tasks. For our analysis, Moment, Mantis, and NuTime are particularly suitable as they allow feature extraction without task-specific fine-tuning. We note, however, that they are not "zero-shot" on our evaluated benchmark, as their pre-training corpora include the training split of the benchmark.

Distinct from the generalizable pre-training paradigm, another line of research involves task-specific unsupervised classification models. These methods are typically trained per-dataset. While some support limited transfer learning, they do not allow for zero-shot feature extraction with a single, fixed model. Notable examples include TLoss [18], TS2Vec[35], TF-C [36] and Ti-MAE [23].

Additionally, there is extensive literature on supervised classification models that are mostly not based on deep learning. These classical methods often rely on ensembles and heuristically engineered features. [27] and [31] provide a good overview of these methods.

B Experiment Details

B.1 Benchmark Data

For our evaluation we utilize the UCR [15] (127 univariate datasets) and the UEA [7] (30 multivariate datasets) classification benchmark datasets. The benchmark covers various types and domains of time series including for example sensor, audio, motion or health data. The train and test split is predefined by the benchmarks. We removed datasets with a sample length over 2048, which is the case for 5 datasets. Specifically, these are: *MotorImagery, HandOutlines, StandWalkJump, EigenWorms*, and *Rock*. Further, we removed *InsectWingbeat* and *PLAID* as these lead to processing problems across the majority of models in the classifier training, likely due to their size.

B.2 Pre-trained Models and Implementation Details

We evaluate a suite of prominent pre-trained forecasting models: TiRex [6], ToTo [13], Chronos [3], Chronos Bolt [4], TimesFM (1.0 and 2.0) [14], and Moirai[33]. When possible (e.g., for TimesFM and Chronos), we analyze both the newest and the previous generation of the models. This gives a better insight into how improvements in forecasting translate to gains in classification, i.e., how they reflect enhancements in the general, underlying representation. We compare these forecasting models to Moment [20], NuTime [24], and Mantis [17]. Moment is a "general" pre-trained time series models – NuTime and Mantis are classification-specific pre-trained models. These models support a feature extraction approach as introduced in Section 2 without fine-tuning. However, they are not really "zero-shot" as (parts) of the training data of the classification benchmark are utilized in pre-training. Additionally, we compare to Dynamic Time Warping (DTW) as a baseline. Implementation details are provided in the following:

- **TiRex** [6]: We utilize the official pre-trained weights from Hugginface and adapt the original source code from GitHub to extract hidden layer representations.
- **Chronos / Chronos Bolt** [3, 4]: For both Chronos and Chronos Bolt, we evaluate the small and base model size. This model family is unique in providing a dedicated API for embedding extraction. We utilize this API, which returns a single-layer representation, and therefore only perform aggregation along the sequence dimension.
- **TimesFM** [14]:For both versions 1.0 and 2.0, we use the official PyTorch weights from Hugging Face and modify the source code from GitHub to access hidden states from all decoder layers.
- **Moirai** [33]: We evaluate Moirai 1.1 in all model sizes (small, base, and large). We utilize the official pre-trained weights from Hugginface and adapt the original source code from GitHub to extract hidden layer representations. While inherently multivariate, Moirai's "variate flattening" fails on datasets with a very high number of variates due to memory constraints. In these cases, we apply the model in a univariate fashion to each variate and concatenate the resulting embeddings. This is the case for the following datasets:
  - Moirai Small: *FaceDetection, Heartbeat, MotorImagery, PEMS-SF, SpokenArabicDigits*
  - Moirai Base: *FaceDetection, Heartbeat, MotorImagery, PEMS-SF, PhonemeSpectra, SpokenArabicDigits*
  - Moirai Large: *FaceDetection, FingerMovements, Heartbeat, LSST, MotorImagery, NATOPS, PEMS-SF, PhonemeSpectra*
- **Mantis** [17]: We follow the official zero-shot feature extraction procedure from their Github repository, which includes interpolating all time series to a fixed length of 512 before embedding.
- **NuTime** [24]: Following the protocol in [17] we use the pre-trained weights provided in the respective GitHub repository, while utilizing the hyperparameters according to this configuration file. We use NuTime in zero-shot feature extraction mode, i.e., variates are embedded independently.
- **Moment** [20]: We use the official zero-shot feature extraction method as demonstrated in their GitHub repository and evaluate all size variants (small, base, and large).
- **Dynamic Time Warping (DTW)**: We use the implementation of the aeon library [26].

B.3 Failure Fallback: DTW

Certain model and dataset combinations result in computational failures (e.g., out-of-memory errors). To avoid skewing aggregate metrics by either dropping these results or assigning a score of zero, we adopt a fallback strategy: For any failed run, we substitute the model's result with the performance of our DTW baseline on that specific dataset. This approach ensures a complete comparison, mirroring a practical scenario. Fallbacks were utilized for the following model-dataset combinations:

- TimesFM 1.0: *Crop, FaceDetection*
- TimesFM 2.0: *FaceDetection, PEMS-SF, SpokenArabicDigits*
- Moirai (Large): *Crop, ElectricDevices, StarLightCurves, PenDigits, SpokenArabicDigits*
- Moment (Base & Large): *PEMS-SF*

B.4 Classifier Training & Hyperparameter

We evaluate three classifiers on the extracted embeddings: Random Forest (as suggested by [17] for Mantis), a linear model, and kNN as a baseline. This tests the linear and non-linear separability of the embeddings. Details are provided in the following:

- **Random Forest** implemented with scikit-learn [29]. Following the protocol from [17], we use "n_estimators=300"; keeping all other parameters at their default values.
- **Linear Model** implemented with PyTorch. It consists of a single linear layer trained with the AdamW optimizer (learning rate 10^{-4}, weight decay 10^{-2}). We use a 20% validation split from the training data for early stopping (patience of 100), with a maximum of 10,000 epochs.
- **kNN** implemented with scikit-learn [29]. We use "n_neighbors=1" (1-NN) with the cosine similarity as distance metric.

B.5 Critical Difference Plots

All critical difference plots in the paper show the average accuracy rank of each method (lower is better). A horizontal bar connects models with no statistically significant difference in performance. This significance is determined by a pairwise Wilcoxon signed-rank test with a Holm correction at a significance level of $\alpha = 0.1$.

	Type	ZS	Univariate		Multivariate		Overall	
			No Aug	Stat+Diff	No Aug	Stat+Diff	No Aug	Stat+Diff
TiRex	Dec	yes	**0.80**	**0.81**	**0.74**	**0.74**	**0.79**	**0.80**
Chr. Bolt (Base)	EncDec	yes	0.77	0.79	0.72	**0.74**	0.76	0.78
Chr. Bolt (Small)	EncDec	yes	0.77	0.79	0.73	**0.74**	0.76	0.78
Moirai (Large)	Enc	yes	0.79	0.80	0.70	0.70	0.78	0.78
Moirai (Base)	Enc	yes	0.79	0.79	0.69	0.71	0.77	0.78
Moirai (Small)	Enc	yes	0.75	0.77	0.69	0.73	0.74	0.77
TimesFM 2.0	Dec	yes	0.79	0.79	0.70	0.70	0.77	0.78
TimesFM 1.0	Dec	yes	0.74	0.75	0.71	0.72	0.73	0.74
Chronos (Base)	EncDec	yes	0.71	0.76	0.71	0.72	0.71	0.75
Chronos (Small)	EncDec	yes	0.70	0.75	0.70	0.72	0.70	0.75
ToTo	Dec	yes	0.71	0.74	0.71	0.70	0.71	0.73
Mantis	Enc	no	0.79		**0.74**		0.78	
NuTime	Enc	no	0.67		0.68		0.67	
Moment (Large)	Enc	no	0.63		0.57		0.62	
Moment (Base)	Enc	no	0.65		0.57		0.64	
Moment (Small)	Enc	no	0.63		0.56		0.62	
DTW (1-NN)	-	-	0.73		0.72		0.73	
DTW (3-NN)	-	-	0.71		0.71		0.71	

Table 2: Classification Accuracy of different models (and sizes) for the univariate, multivariate, and combined benchmark (Random Forest). "Stat+Diff" shows results with both proposed augmentations applied; "no Aug" utilizes the pure forecasting model representations. "ZS" indicates models that did not have access to the benchmarks training data during pre-training.

C Extended Results

This section provides extended results to Section 3. Extending Table 1, Table 2 shows the results for all evaluated model sizes. In almost all cases, larger models perform better, which aligns with the performance trend observed in forecasting. A notable exception is Moment, where the base model outperforms the large version.

The following subsections provide further analysis, including results for different classifiers (Section C.1), ablations of the aggregation methods (Section C.2), ablations and analysis of the embedding augmentation (Section C.3), and robustness checks using a different metric (Section C.4) and a dataset subset (Section C.5). The main results for each individual dataset are presented in Table 6 - 11.

C.1 Results for different Classifiers

This section complements the main paper's evaluation by presenting the results for the other two classifiers: the gradient-based trained linear model and the 1-NN baseline. The results are shown in Table 3.

The overall performance ranking of the models is largely consistent with the main evaluation, which uses a Random Forest. While there are minor shifts in relative performance — for example, with the linear classifier, the results for TiRex and Chronos-Bolt are not significantly different — key insights from our paper hold. The best forecasting models perform on par with pre-trained classification models and forecasting performance is correlated with classification accuracy. However, a difference is that when using the simplest classifier (1-NN), the forecasting models no longer outperform Mantis, the best pre-trained classification model.

We hypothesize that this discrepancy arises because less powerful classifiers, such as linear models or kNN, have a limited ability to transform the feature space. The embedding space of a model pre-trained on classification, like Mantis, might be already better aligned with the classification task. In contrast, a non-linear model like a Random Forest can better identify and exploit the relevant discriminative information, which we assume is present in the embeddings from both forecasting and classification models.

	Linear			1-NN		
	Univariate	Multivariate	Overall	Univariate	Multivariate	Overall
TiRex	0.78	0.72	**0.77**	0.75	0.67	0.74
Chr. Bolt (Base)	0.76	**0.73**	0.76	0.75	0.68	0.74
Chr. Bolt (Small)	0.76	**0.73**	0.75	0.75	0.68	0.74
Moirai (Large)	**0.79**	0.70	**0.77**	**0.77**	0.64	0.75
Moirai (Base)	0.78	0.70	0.76	0.76	0.65	0.74
Moirai (Small)	0.75	0.69	0.74	0.72	0.63	0.71
TimesFM 2.0	0.75	0.70	0.74	0.71	0.56	0.69
TimesFM 1.0	0.73	0.69	0.72	0.70	0.65	0.69
Chronos (Base)	0.71	0.72	0.71	0.67	0.66	0.67
Chronos (Small)	0.69	0.70	0.69	0.66	0.66	0.66
ToTo	0.70	0.71	0.70	0.65	0.63	0.65
Mantis	0.77	**0.73**	0.76	**0.77**	**0.72**	**0.76**
NuTime	0.59	0.63	0.59	0.60	0.61	0.60
Moment (Large)	0.58	0.44	0.55	0.61	0.55	0.60
Moment (Base)	0.58	0.48	0.56	0.56	0.50	0.55
Moment (Small)	0.54	0.47	0.53	0.53	0.48	0.52
DTW (1-NN)	0.73	0.72	0.73	0.73	**0.72**	0.73
DTW (3-NN)	0.71	0.71	0.71	0.71	0.71	0.71

Table 3: Classification accuracy of different models and classifiers (linear model and 1-NN) for the univariate, multivariate, and combined benchmark.

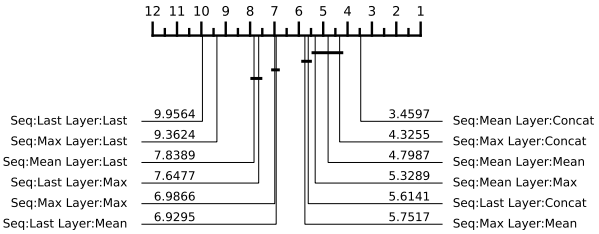
C.2 Ablation Analysis: Aggregation Methods

Sequence & Layer Aggregation We conduct an ablation study of the method to aggregate the hidden states across both the layer and sequence dimensions. For layer aggregation, we evaluated four strategies: concatenation of all layer representations, mean pooling, max pooling, and using only the representation from the last layer. For sequence aggregation, we considered mean pooling, max pooling, and using the last output. Concatenation is not a viable option for the sequence dimension, as it would result in variable-length embeddings dependent on the sample length. After the sequence aggregation and before the layer aggregation we normalize the embeddings as different layers might operate in different feature spaces. For the Chronos models, which have a predefined method for embedding extraction, we only ablated the sequence aggregation strategy.

Figures 3-11 present the results. Each figure presents a table with the mean accuracy over univariate, multivariate, and all datasets, complemented by a critical difference plot of mean ranks to visualize statistical significance. Across almost all models, the combination of mean pooling over the sequence dimension and concatenation over the layer dimension is the top-performing strategy. In no case any other strategy combination performs significantly better.

Seq	Layer	Uni	Multi	Comb
Mean	Concat	**0.80**	**0.74**	**0.79**
Mean	Mean	0.79	0.73	0.78
Max	Concat	0.79	0.73	0.78
Mean	Max	0.79	0.72	0.77
Max	Mean	0.77	0.73	0.77
Max	Max	0.76	0.72	0.75
Last	Concat	0.75	0.73	0.75
Mean	Last	0.75	0.71	0.75
Last	Mean	0.74	0.72	0.74
Last	Max	0.73	0.71	0.73
Max	Last	0.72	0.70	0.72
Last	Last	0.70	0.69	0.70

(a) Average Accuracy

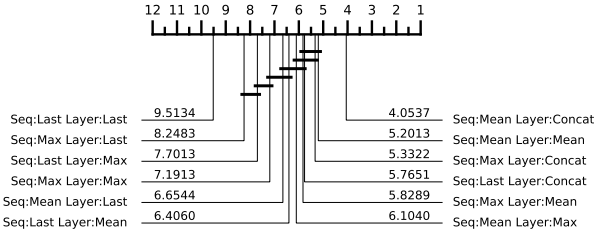


(b) Average Accuracy Rank (Overall benchmark)

Figure 3: Results for **TiRex** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

seq	L	Uni	Multi	Comb
Mean	Concat	**0.79**	0.69	**0.77**
Max	Concat	0.78	**0.70**	0.76
Mean	Mean	0.78	0.69	0.76
Max	Mean	0.77	0.69	0.76
Mean	Max	0.77	0.68	0.76
Last	Concat	0.77	0.69	0.76
Mean	Last	0.77	0.68	0.75
Last	Mean	0.76	0.68	0.75
Max	Max	0.76	0.68	0.75
Last	Max	0.75	0.67	0.74
Max	Last	0.75	0.67	0.73
Last	Last	0.73	0.64	0.71

(a) Average Accuracy

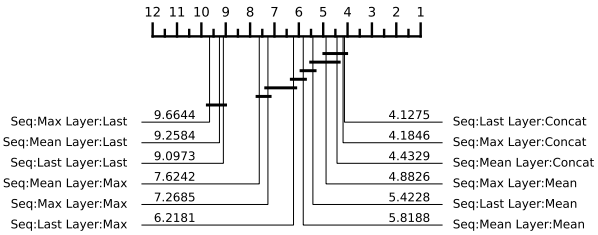


(b) Average Accuracy Rank (Overall benchmark)

Figure 4: Results for **Moirai 1.1 (Base)** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Layer	Uni	Multi	Comb
Mean	Concat	**0.79**	0.70	**0.77**
Max	Concat	0.78	0.71	**0.77**
Last	Concat	0.77	**0.74**	**0.77**
Max	Mean	0.77	0.70	0.76
Mean	Mean	0.77	0.69	0.76
Last	Mean	0.76	0.73	0.75
Last	Max	0.74	0.73	0.74
Mean	Max	0.74	0.68	0.73
Max	Max	0.73	0.69	0.73
Last	Last	0.71	0.69	0.70
Mean	Last	0.71	0.68	0.70
Max	Last	0.70	0.67	0.70

(a) Average Accuracy

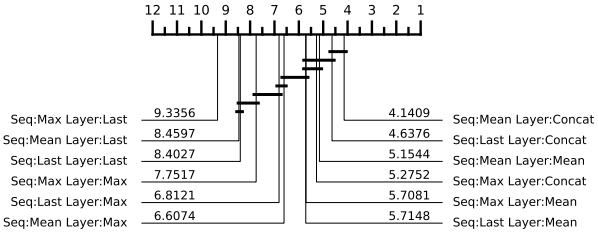


(b) Average Accuracy Rank (Overall benchmark)

Figure 5: Results for **TimesFM 2.0** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Layer	Uni	Multi	Comb
Mean	Concat	**0.74**	0.71	**0.73**
Last	Concat	0.73	**0.72**	**0.73**
Max	Concat	0.73	0.69	**0.73**
Mean	Mean	0.73	0.70	0.72
Max	Mean	0.73	0.69	0.72
Last	Mean	0.72	0.71	0.72
Mean	Max	0.72	0.68	0.71
Last	Max	0.71	0.69	0.71
Max	Max	0.71	0.66	0.71
Mean	Last	0.69	0.68	0.69
Last	Last	0.69	0.67	0.69
Max	Last	0.68	0.66	0.68

(a) Average Accuracy

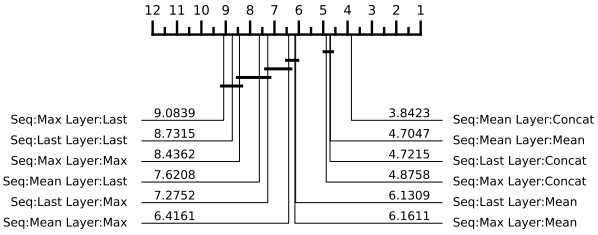


(b) Average Accuracy Rank (Overall benchmark)

Figure 6: Results for **TimesFM 1.0** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Layer	Uni	Multi	Comb
Mean	Concat	**0.71**	0.71	**0.71**
Mean	Mean	0.70	0.70	0.70
Max	Concat	0.69	**0.72**	0.69
Last	Concat	0.69	**0.72**	0.69
Mean	Max	0.68	0.70	0.68
Max	Mean	0.67	0.71	0.68
Last	Mean	0.67	0.71	0.67
Mean	Last	0.66	0.68	0.66
Last	Max	0.65	0.71	0.66
Max	Max	0.64	0.68	0.64
Last	Last	0.62	0.70	0.64
Max	Last	0.62	0.67	0.63

(a) Average Accuracy

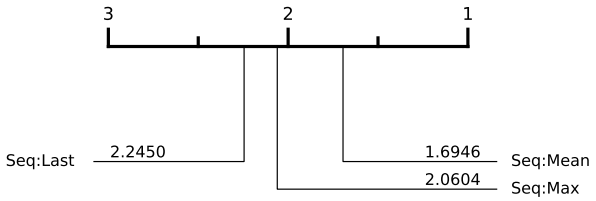


(b) Average Accuracy Rank (Overall benchmark)

Figure 7: Results for **ToTo** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Uni	Multi	Comb
Mean	**0.77**	**0.72**	**0.76**
Max	0.76	0.71	0.75
Last	0.74	**0.72**	0.74

(a) Average Accuracy

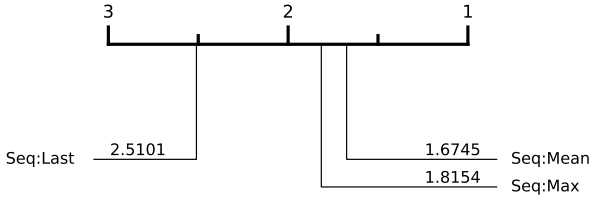


(b) Average Accuracy Rank (Overall benchmark)

Figure 8: Results for **Chronos Bolt (Base)** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Uni	Multi	Comb
Mean	**0.77**	**0.73**	**0.76**
Max	0.76	**0.73**	**0.76**
Last	0.72	0.71	0.72

(a) Average Accuracy

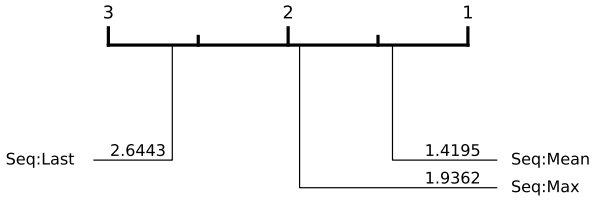


(b) Average Accuracy Rank (Overall benchmark)

Figure 9: Results for **Chronos Bolt (Small)** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Uni	Multi	Comb
Mean	**0.71**	**0.71**	**0.71**
Max	0.68	0.68	0.68
Last	0.59	0.65	0.60

(a) Average Accuracy

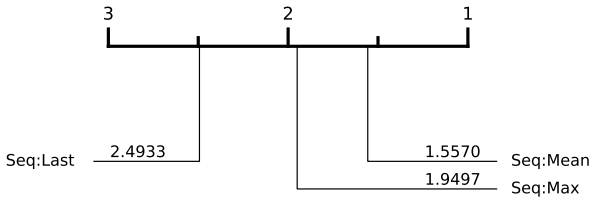


(b) Average Accuracy Rank (Overall benchmark)

Figure 10: Results for **Chronos (Base)** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Seq	Uni	Multi	Comb
Mean	**0.70**	**0.70**	**0.70**
Max	0.67	0.67	0.67
Last	0.60	0.64	0.61

(a) Average Accuracy



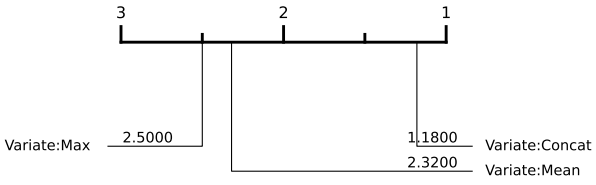
(b) Average Accuracy Rank (Overall benchmark)

Figure 11: Results for **Chronos (Small)** for the **layer and sequence aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Variate aggregation methods We conduct an ablation study of the method to aggregate per-variate embeddings into a single feature vector for a multivariate time series. This is necessary when applying a univariate model to each variate independently or when a multivariate model produces distinct per-variate outputs. We evaluate three strategies: mean pooling, max pooling, and concatenation. Figures 12-19 present the results. Each figure presents a table with the mean accuracy over univariate, multivariate, and all datasets, complemented by a critical difference plot of mean ranks to visualize statistical significance. Concatenation consistently outperforms both pooling methods across all tested models.

Var	Multi
Concat	**0.74**
Mean	0.67
Max	0.66

(a) Average Accuracy

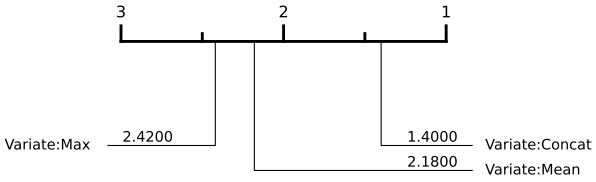


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 12: Results for **TiRex** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.72**
Mean	0.66
Max	0.65

(a) Average Accuracy

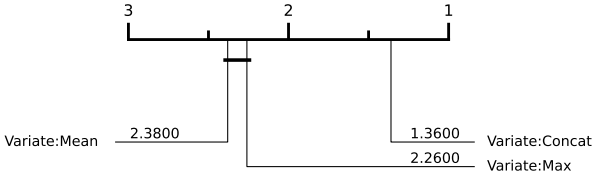


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 13: Results for **Chronos Bolt (Base)** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.73**
Max	0.66
Mean	0.65

(a) Average Accuracy

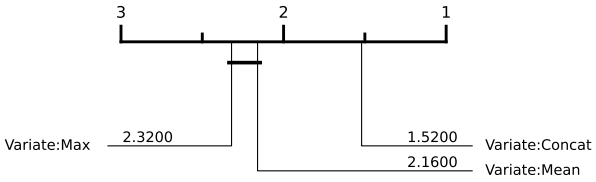


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 14: Results for **Chronos Bolt (Small)** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.71**
Mean	0.65
Max	0.65

(a) Average Accuracy

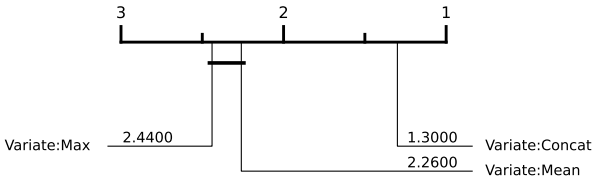


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 15: Results for **Chronos (Base)** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.70**
Mean	0.64
Max	0.63

(a) Average Accuracy

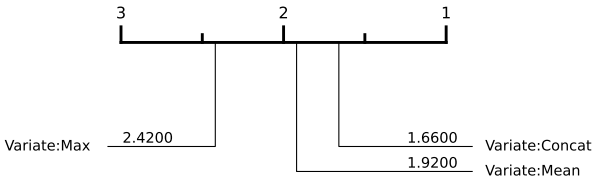


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 16: Results for **Chronos (Small)** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.71**
Mean	0.65
Max	0.64

(a) Average Accuracy

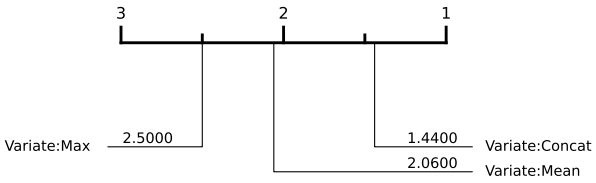


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 17: Results for **TimesFM 1.0** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.70**
Max	0.67
Mean	0.66

(a) Average Accuracy

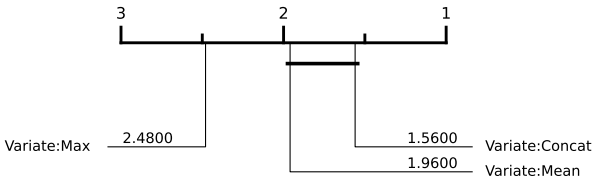


(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 18: Results for **TimesFM 2.0** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Var	Multi
Concat	**0.71**
Mean	0.68
Max	0.66

(a) Average Accuracy



(b) Average Accuracy Rank (Multivariate Benchmark)

Figure 19: Results for **ToTo** for the **variate aggregation ablation** experiments. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

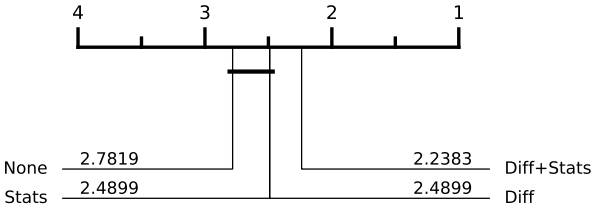
C.3 Full Results: Embedding Augmentation

In this section, we provide an ablation study of our proposed embedding augmentations. First, the impact augmentations, both individually and combined, are analyzed quantitatively for each model. Then we provide a hyperparameter ablation for the Absolute Sample Statistics Augmentation and a qualitative analysis of its impact.

Ablation on individual models We conduct an ablation study to evaluate the effectiveness of our two proposed embedding augmentations. Figures 20-28 present the results. Each figure presents a table with the mean accuracy over univariate, multivariate, and all datasets, complemented by a critical difference plot of mean ranks to visualize statistical significance. Both the statistics-based and the differencing-based augmentations individually improve performance for a majority of the models, although the statistical significance of these gains varies. The combination of both augmentations most often yields further improvements, resulting in the best overall performance.

Diff	Stats	Uni	Multi	Comb
True	True	**0.81**	**0.74**	**0.80**
True	False	**0.81**	0.73	0.79
False	True	0.80	0.73	0.79
False	False	0.80	**0.74**	0.79

(a) Average Accuracy

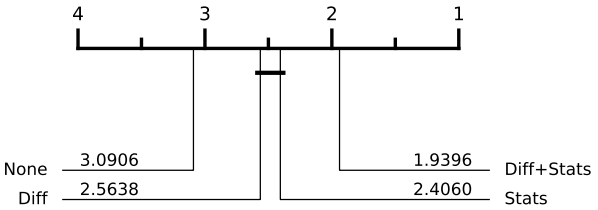


(b) Average Accuracy Rank (Overall benchmark)

Figure 20: Results for **TiRex** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.79**	**0.74**	**0.78**
True	False	**0.79**	0.72	0.77
False	True	0.78	**0.74**	0.77
False	False	0.77	0.72	0.76

(a) Average Accuracy

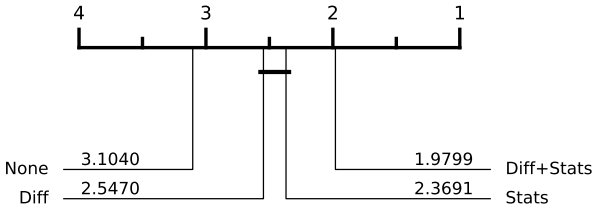


(b) Average Accuracy Rank (Overall benchmark)

Figure 21: Results for **Chronos Bolt (Base)** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.79**	**0.74**	**0.78**
False	True	0.78	**0.74**	0.77
True	False	0.78	0.72	0.77
False	False	0.77	0.73	0.76

(a) Average Accuracy

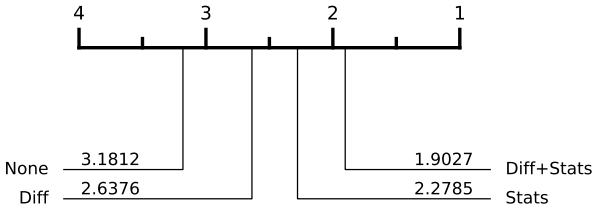


(b) Average Accuracy Rank (Overall benchmark)

Figure 22: Results for **Chronos Bolt (Small)** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.76**	0.72	**0.75**
False	True	0.75	**0.73**	**0.75**
True	False	0.74	0.71	0.73
False	False	0.71	0.71	0.71

(a) Average Accuracy

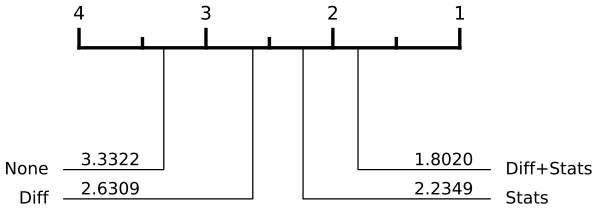


(b) Average Accuracy Rank (Overall benchmark)

Figure 23: Results for **Chronos (Base)** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.75**	0.72	**0.75**
False	True	0.74	**0.73**	0.74
True	False	0.72	0.71	0.72
False	False	0.70	0.70	0.70

(a) Average Accuracy

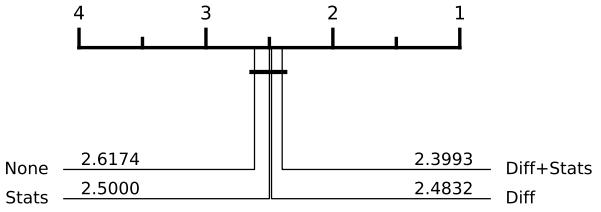


(b) Average Accuracy Rank (Overall benchmark)

Figure 24: Results for **Chronos (Small)** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.79**	0.70	**0.78**
False	True	**0.79**	**0.71**	**0.78**
True	False	**0.79**	0.70	**0.78**
False	False	**0.79**	0.70	0.77

(a) Average Accuracy

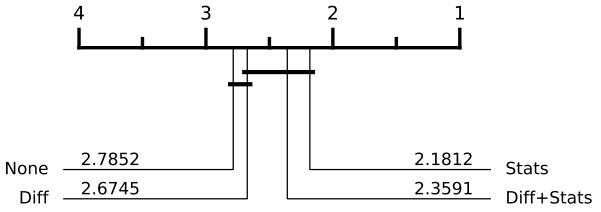


(b) Average Accuracy Rank (Overall benchmark)

Figure 25: Results for **TimesFM 2.0** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.75**	**0.72**	**0.74**
False	True	**0.75**	0.70	**0.74**
True	False	**0.75**	0.70	**0.74**
False	False	0.74	0.71	0.73

(a) Average Accuracy

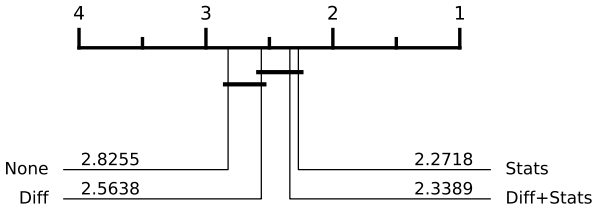


(b) Average Accuracy Rank (Overall benchmark)

Figure 26: Results for **TimesFM 1.0** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.79**	0.71	**0.78**
False	True	**0.79**	**0.72**	**0.78**
True	False	**0.79**	0.70	**0.78**
False	False	**0.79**	0.69	0.77

(a) Average Accuracy

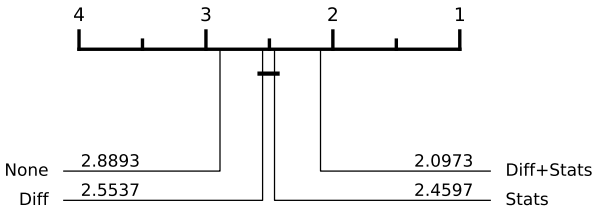


(b) Average Accuracy Rank (Overall benchmark)

Figure 27: Results for **Moirai 1.1 (Base)** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Diff	Stats	Uni	Multi	Comb
True	True	**0.74**	0.70	**0.73**
True	False	0.73	**0.71**	**0.73**
False	True	0.72	**0.71**	0.72
False	False	0.71	**0.71**	0.71

(a) Average Accuracy



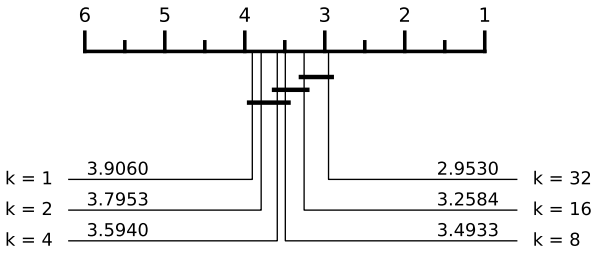
(b) Average Accuracy Rank (Overall benchmark)

Figure 28: Results for **ToTo** for the **embedding augmentation ablation** experiments. *Diff* and *Stats* indicate the application of the "differencing" and the "sample statistics" augmentations respectively. (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Absolute Sample Statistics Augmentation: Number of Patches The absolute sample statistics augmentation divides each time series into k non-overlapping patches. For the main experiments, we used a fixed value of $k = 8$. To analyze the impact of this choice, we conducted an ablation study on our best-performing model, TiRex, by evaluating $k \in \{1, 2, 4, 8, 16, 32\}$. The results are presented in Figure 29. While the average ranks suggest that a higher number of patches could marginally improve performance, the average accuracies remain very similar across all settings. This indicates that the procedure is generally robust to the choice of k, although we note that tuning this hyperparameter for specific datasets could be advantageous in a practical application.

k	Uni	Multi	Comb
32	**0.81**	0.73	**0.80**
16	**0.81**	**0.74**	**0.80**
4	**0.81**	0.73	**0.80**
8	**0.81**	**0.74**	**0.80**
2	0.80	**0.74**	0.79
1	**0.81**	0.72	0.79

(a) Average Accuracy



(b) Average Accuracy Rank (Overall benchmark)

Figure 29: Result of the ablation experiment regarding the number of patches for the absolute sample statistics augmentation (a) Average accuracy on univariate (Uni), multivariate (Multi), and overall (Comb) benchmark datasets. Sorted by overall accuracy. (b) Critical difference diagram of the average accuracy ranks.

Absolute Sample Statistics Augmentation: Qualitative Analysis As discussed in Section 2, instance normalization removes a signal's absolute scale information, such as its mean value. To visually demonstrate this effect and the efficacy of our statistics augmentation, we created a toy dataset composed of sine waves that differ only by their baseline value [20]. Each series is generated using the formula $y_t = sin(5t) + a$, where the baseline a is sampled uniquely for each of the 1024 examples. Figure 30 shows three such series.

We then generated embeddings for this dataset using TiRex and Chronos Bolt, once without and once with our statistics augmentation, and visualized the results using PCA. The projections in Figure 31 illustrate the outcome. Without the augmentation, the embeddings from the forecasting models (TiRex, Chronos Bolt) form a single, inseparable cluster. In contrast, the augmented embeddings show a gradient along the first principal component that directly corresponds to the baseline value a. Notably, the pre-trained classification models also cluster series with similar baselines, i.e., incorporate this property in their representation.

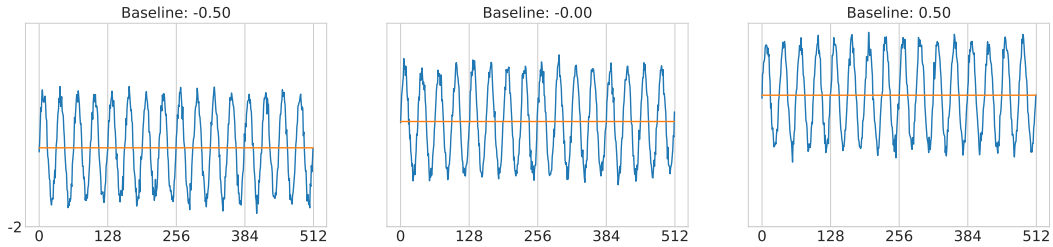


Figure 30: Illustration of three example time series from our synthetic toy dataset. For each series only the baseline value differs between them.

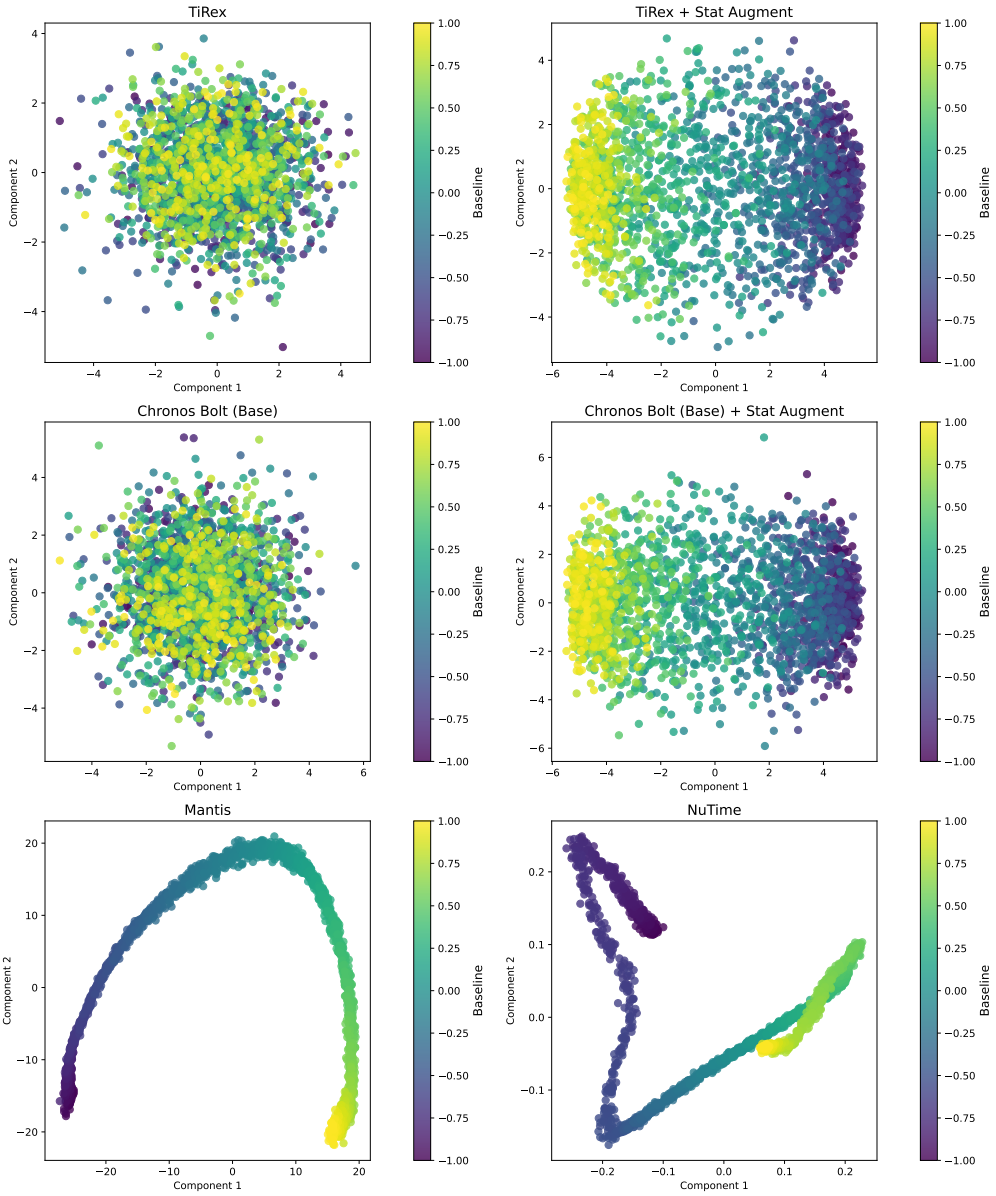


Figure 31: 2D PCA projections of embeddings from the baseline-shifted sine wave dataset. The left column of the top two row shows the original embeddings from each model, while the right column shows the same embeddings enhanced with our sample statistics augmentation. The bottom row shows the embeddings of the pre-trained classification models — which also allow for separation in terms of this property.

	Type	ZS	Univariate		Multivariate		Overall	
			No Aug	Stat+Diff	No Aug	Stat+Diff	No Aug	Stat+Diff
TiRex	Dec	yes	**0.78**	**0.78**	0.71	**0.72**	**0.77**	**0.77**
Chr. Bolt (Base)	EncDec	yes	0.75	0.77	0.70	**0.72**	0.74	0.76
Moirai (Large)	Enc	yes	0.77	**0.78**	0.68	0.68	0.76	0.76
TimesFM 2.0	Dec	yes	0.76	0.77	0.68	0.68	0.75	0.75
TimesFM 1.0	Dec	yes	0.72	0.72	0.68	0.69	0.71	0.72
Chronos (Base)	EncDec	yes	0.68	0.73	0.69	0.70	0.68	0.73
ToTo	Dec	yes	0.68	0.71	0.69	0.69	0.68	0.70
Mantis	Enc	no	0.76		**0.72**		0.76	
NuTime	Enc	no	0.64		0.66		0.65	
Moment (Large)	Enc	no	0.59		0.55		0.58	
DTW	-	-	0.72		0.71		0.72	

Table 4: **Balanced Accuracy** of different models for the univariate, multivariate, and combined benchmark with a Random Forest Classifier. "Stat+Diff" shows results with both proposed augmentations applied; "no Aug" utilizes the pure forecasting model representations. "ZS" indicates models that did not have access to the benchmark training data during pre-training.

	Type	ZS	Univariate		Multivariate		Overall	
			No Aug	Stat+Diff	No Aug	Stat+Diff	No Aug	Stat+Diff
TiRex	Dec	yes	**0.82**	**0.83**	0.77	0.77	**0.81**	**0.81**
Chr. Bolt (Base)	EncDec	yes	0.81	0.82	0.76	0.77	0.80	**0.81**
Moirai (Large)	Enc	yes	**0.82**	0.82	0.74	0.74	0.80	**0.81**
TimesFM 2.0	Dec	yes	0.81	0.81	0.74	0.73	0.80	0.80
TimesFM 1.0	Dec	yes	0.79	0.79	0.75	0.75	0.78	0.79
Chronos (Base)	EncDec	yes	0.74	0.79	0.74	0.76	0.74	0.78
ToTo	Dec	yes	0.73	0.76	0.74	0.74	0.73	0.75
Mantis	Enc	no	0.81		**0.78**		**0.81**	
NuTime	Enc	no	0.71		0.71		0.71	
Moment (Large)	Enc	no	0.68		0.60		0.66	
DTW	-	-	0.76		0.76		0.76	

Table 5: Classification Accuracy of different models for the univariate, multivariate, and combined benchmark with a Random Forest Classifier — **on the subset of datasets with a maximum length of 512**. "Stat+Diff" shows results with both proposed augmentations applied; "no Aug" utilizes the pure forecasting model representations. "ZS" indicates models that did not have access to the benchmark training data during pre-training.

C.4 Main Results: Balanced Accuracy

While accuracy is the primary metric in our main evaluation, for consistency with related literature, we also re-evaluated our main experiments using balanced accuracy to ensure the robustness of our findings. The results are presented in Table 4. The relative performance rankings of the models remain highly consistent across both metrics, with slight changes in the multivariate benchmark data, confirming the robustness of our conclusions.

C.5 Main Results: $\leq$ 512 length datasets

Several of the evaluated models were pre-trained with a maximum context length of 512, whereas our full benchmark includes datasets with series up to 2048 in length. To assess the impact of this context length discrepancy and to further test the robustness of our findings, we re-ran our main experiments on a subset of the benchmark containing only datasets with a series length of 512 or less. The results of this analysis are presented in Table 5. The relative performance rankings remain consistent with our primary results, with slight changes in the multivariate benchmark data — this confirms the robustness of our conclusions.

	TiRex	Chr. Bolt (Base)	Chr. Bolt (Small)	Moirai (Large)	Moirai (Base)	Moirai (Small)	TimesFM 2.0	TimesFM 1.0	Chronos (Base)
ACSF1	0.85	0.82	0.82	0.88	0.86	0.86	0.83	0.75	0.84
Adiac	0.78	0.79	0.80	0.79	0.79	0.78	0.78	0.79	0.71
ArrowHead	0.78	0.83	0.81	0.78	0.77	0.76	0.73	0.74	0.66
Beef	0.80	0.67	0.73	0.70	0.67	0.60	0.80	0.83	0.60
BeetleFly	0.90	0.90	0.90	0.95	0.95	0.85	0.95	0.85	0.75
BirdChicken	0.90	0.90	0.95	0.90	0.90	0.90	0.80	0.90	0.90
BME	0.99	1.00	1.00	0.99	0.96	0.95	0.95	0.95	0.98
Car	0.80	0.82	0.77	0.78	0.75	0.68	0.82	0.78	0.83
CBF	0.99	1.00	0.97	1.00	1.00	0.96	1.00	0.99	0.96
Chinatown	0.97	0.98	0.99	0.96	0.97	0.95	0.97	0.97	0.97
ChlorineConcentration	0.72	0.71	0.72	0.74	0.75	0.75	0.69	0.69	0.64
CinCECGTorso	0.99	0.85	0.90	0.84	0.83	0.75	0.98	0.96	0.97
Coffee	1.00	0.96	1.00	1.00	0.96	0.96	0.96	1.00	0.96
Computers	0.76	0.72	0.73	0.77	0.76	0.77	0.72	0.70	0.74
CricketX	0.71	0.71	0.69	0.69	0.68	0.62	0.69	0.64	0.62
CricketY	0.74	0.73	0.70	0.69	0.67	0.61	0.75	0.69	0.69
CricketZ	0.72	0.73	0.72	0.74	0.69	0.69	0.69	0.64	0.63
Crop	0.74	0.74	0.74	NaN	0.73	0.73	0.73	NaN	0.71
DiatomSizeReduction	0.86	0.90	0.92	0.87	0.82	0.85	0.81	0.85	0.88
DistalPhalanxOutlineCorrect	0.79	0.77	0.80	0.80	0.80	0.78	0.79	0.78	0.76
DistalPhalanxOutlineAgeGroup	0.76	0.76	0.75	0.78	0.74	0.71	0.75	0.76	0.76
DistalPhalanxTW	0.65	0.66	0.68	0.68	0.72	0.69	0.65	0.66	0.66
Earthquakes	0.75	0.76	0.76	0.72	0.74	0.75	0.74	0.76	0.74
ECG200	0.85	0.85	0.85	0.84	0.86	0.83	0.87	0.86	0.77
ECG5000	0.94	0.94	0.94	0.94	0.94	0.93	0.93	0.94	0.93
ECGFiveDays	0.83	0.86	0.90	0.92	0.84	0.81	0.91	0.75	0.77
ElectricDevices	0.70	0.70	0.69	NaN	0.72	0.70	0.70	0.66	0.74
EOGHorizontalSignal	0.54	0.54	0.56	0.60	0.57	0.48	0.66	0.21	0.36
EOGVerticalSignal	0.42	0.43	0.45	0.42	0.44	0.40	0.46	0.15	0.30
EthanolLevel	0.37	0.44	0.43	0.43	0.42	0.55	0.35	0.57	0.56
FaceAll	0.86	0.71	0.71	0.72	0.78	0.70	0.83	0.85	0.69
FaceFour	0.68	0.76	0.64	0.72	0.68	0.65	0.81	0.74	0.62
FacesUCR	0.81	0.75	0.76	0.76	0.73	0.72	0.73	0.77	0.73
FiftyWords	0.62	0.67	0.69	0.60	0.58	0.53	0.60	0.57	0.64
Fish	0.93	0.85	0.85	0.93	0.88	0.84	0.90	0.90	0.89
FordA	0.95	0.94	0.93	0.93	0.93	0.90	0.94	0.95	0.94
FordB	0.84	0.78	0.80	0.81	0.81	0.78	0.84	0.82	0.77
FreezerRegularTrain	0.93	0.92	0.91	0.97	0.97	0.91	0.88	0.85	0.97
FreezerSmallTrain	0.81	0.85	0.84	0.87	0.84	0.79	0.78	0.69	0.87
GunPoint	0.95	0.94	0.99	0.97	0.97	0.95	0.95	0.88	0.95
GunPointAgeSpan	0.98	0.99	0.98	0.98	0.97	0.97	0.95	0.94	0.98
GunPointMaleVersusFemale	1.00	1.00	1.00	0.99	1.00	0.99	1.00	0.99	0.99
GunPointOldVersusYoung	0.98	1.00	1.00	0.99	0.99	0.99	0.96	0.93	0.99
Ham	0.65	0.71	0.64	0.57	0.61	0.68	0.60	0.55	0.60
Haptics	0.52	0.51	0.51	0.51	0.57	0.49	0.54	0.51	0.51
Herring	0.59	0.67	0.67	0.59	0.59	0.62	0.61	0.53	0.59
HouseTwenty	0.97	0.89	0.93	0.97	0.96	0.94	0.87	0.60	0.72
InlineSkate	0.44	0.54	0.49	0.49	0.44	0.43	0.50	0.37	0.39
InsectEPGRegularTrain	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
InsectEPGSmallTrain	0.93	0.93	0.92	0.96	0.94	0.94	0.90	0.86	0.99

Table 6: Accuracy results of the different models with augmentations on the individual datasets with a Random Forest classifier. (Part 1/6)

	Chronos (Small)	ToTo	Mantis	NuTime	Moment (Large)	Moment (Base)	Moment (Large)	DTW (1-NN)	DTW (3-NN)
ACSF1	0.84	0.79	0.79	0.78	0.43	0.55	0.48	0.64	0.59
Adiac	0.72	0.55	0.74	0.70	0.08	0.12	0.08	0.60	0.56
ArrowHead	0.61	0.80	0.73	0.74	0.59	0.50	0.58	0.70	0.71
Beef	0.60	0.70	0.63	0.97	0.43	0.40	0.50	0.63	0.57
BeetleFly	0.75	0.95	0.85	0.70	0.85	0.95	0.90	0.70	0.70
BirdChicken	0.95	0.85	1.00	0.55	0.80	0.85	0.65	0.75	0.60
BME	0.98	0.97	0.92	0.93	0.79	0.81	0.88	0.89	0.85
Car	0.85	0.75	0.83	0.62	0.67	0.58	0.60	0.73	0.55
CBF	0.98	0.99	0.99	0.54	0.90	0.94	0.86	1.00	1.00
Chinatown	0.97	0.82	0.85	0.98	0.77	0.87	0.84	0.97	0.97
ChlorineConcentration	0.62	0.60	0.68	0.76	0.56	0.55	0.55	0.65	0.57
CinCECGTorso	0.95	0.90	0.67	0.58	0.57	0.68	0.68	0.65	0.50
Coffee	0.86	0.93	0.96	1.00	0.89	0.96	0.82	1.00	0.93
Computers	0.71	0.74	0.74	0.71	0.63	0.65	0.66	0.70	0.71
CricketX	0.68	0.59	0.75	0.24	0.59	0.64	0.65	0.75	0.74
CricketY	0.68	0.63	0.73	0.36	0.61	0.60	0.56	0.74	0.70
CricketZ	0.68	0.58	0.79	0.26	0.62	0.64	0.67	0.75	0.74
Crop	0.72	0.69	0.68	0.74	0.50	0.56	0.55	0.68	0.66
DiatomSizeReduction	0.87	0.83	0.87	0.87	0.50	0.50	0.56	0.97	0.93
DistalPhalanxOutlineCorrect	0.78	0.77	0.74	0.78	0.63	0.65	0.62	0.72	0.74
DistalPhalanxOutlineAgeGroup	0.75	0.74	0.79	0.77	0.63	0.67	0.65	0.77	0.73
DistalPhalanxTW	0.66	0.68	0.69	0.71	0.58	0.58	0.56	0.59	0.62
Earthquakes	0.75	0.75	0.75	0.75	0.75	0.74	0.75	0.72	0.74
ECG200	0.81	0.85	0.81	0.80	0.81	0.82	0.82	0.77	0.80
ECG5000	0.93	0.92	0.92	0.93	0.93	0.92	0.93	0.92	0.94
ECGFiveDays	0.82	0.60	0.93	0.76	0.65	0.88	0.74	0.77	0.62
ElectricDevices	0.73	0.68	0.73	0.65	0.59	0.59	0.59	0.60	0.61
EOGHorizontalSignal	0.45	0.49	0.58	0.33	0.07	0.10	0.11	0.44	0.43
EOGVerticalSignal	0.27	0.39	0.47	0.25	0.10	0.10	0.11	0.43	0.44
EthanolLevel	0.37	0.33	0.29	0.60	0.25	0.27	0.25	0.28	0.26
FaceAll	0.71	0.75	0.78	0.78	0.57	0.53	0.48	0.81	0.81
FaceFour	0.56	0.57	0.95	0.62	0.65	0.55	0.57	0.83	0.68
FacesUCR	0.82	0.63	0.83	0.67	0.54	0.48	0.47	0.90	0.88
FiftyWords	0.64	0.52	0.64	0.57	0.48	0.48	0.44	0.69	0.66
Fish	0.85	0.80	0.94	0.78	0.49	0.55	0.42	0.82	0.79
FordA	0.93	0.92	0.86	0.81	0.88	0.90	0.89	0.55	0.58
FordB	0.76	0.82	0.74	0.62	0.73	0.77	0.72	0.62	0.62
FreezerRegularTrain	0.96	0.91	0.94	0.99	0.78	0.78	0.77	0.90	0.88
FreezerSmallTrain	0.88	0.78	0.80	0.96	0.75	0.76	0.76	0.76	0.73
GunPoint	0.93	0.91	0.97	0.95	0.77	0.81	0.79	0.91	0.89
GunPointAgeSpan	0.97	0.91	0.99	0.88	0.88	0.86	0.85	0.98	0.99
GunPointMaleVersusFemale	1.00	0.96	1.00	0.97	0.94	0.95	0.96	0.98	0.97
GunPointOldVersusYoung	0.99	0.89	1.00	1.00	0.86	0.88	0.84	1.00	1.00
Ham	0.66	0.78	0.70	0.73	0.70	0.65	0.63	0.47	0.51
Haptics	0.49	0.50	0.49	0.45	0.40	0.44	0.41	0.38	0.43
Herring	0.67	0.59	0.66	0.59	0.58	0.58	0.59	0.53	0.48
HouseTwenty	0.71	0.97	0.95	0.65	0.55	0.65	0.62	0.84	0.85
InlineSkate	0.38	0.40	0.39	0.25	0.20	0.21	0.20	0.38	0.36
InsectEPGRegularTrain	1.00	1.00	1.00	0.82	0.89	0.92	0.90	1.00	1.00
InsectEPGSmallTrain	0.99	1.00	1.00	0.80	0.81	0.90	0.92	1.00	1.00

Table 7: Accuracy results of the different models with augmentations on the individual datasets with a Random Forest classifier. (Part 2/6)

	TiRex	Chr. Bolt (Base)	Chr. Bolt (Small)	Moirai (Large)	Moirai (Base)	Moirai (Small)	TimesFM 2.0	TimesFM 1.0	Chronos (Base)
InsectWingbeatSound	0.66	0.62	0.64	0.61	0.61	0.60	0.62	0.63	0.55
ItalyPowerDemand	0.96	0.95	0.95	0.95	0.95	0.96	0.97	0.97	0.92
LargeKitchenAppliances	0.79	0.75	0.72	0.79	0.83	0.75	0.77	0.67	0.76
Lightning2	0.75	0.75	0.72	0.75	0.70	0.67	0.66	0.69	0.70
Lightning7	0.70	0.71	0.77	0.63	0.64	0.63	0.58	0.67	0.67
Mallat	0.94	0.87	0.89	0.90	0.92	0.93	0.84	0.70	0.72
Meat	0.90	0.92	0.93	1.00	0.93	0.92	0.93	0.97	0.88
MedicalImages	0.72	0.72	0.72	0.72	0.71	0.70	0.74	0.75	0.69
MiddlePhalanxOutlineCorrect	0.85	0.84	0.83	0.85	0.86	0.84	0.86	0.87	0.81
MiddlePhalanxOutlineAgeGroup	0.60	0.58	0.58	0.59	0.58	0.59	0.60	0.58	0.56
MiddlePhalanxTW	0.55	0.58	0.53	0.54	0.55	0.56	0.53	0.54	0.55
MixedShapesRegularTrain	0.97	0.95	0.96	0.97	0.97	0.95	0.97	0.94	0.96
MixedShapesSmallTrain	0.94	0.93	0.93	0.94	0.96	0.93	0.95	0.91	0.92
MoteStrain	0.91	0.90	0.91	0.91	0.88	0.82	0.91	0.85	0.93
NonInvasiveFetalECGThorax1	0.92	0.89	0.89	0.91	0.89	0.88	0.90	0.76	0.84
NonInvasiveFetalECGThorax2	0.93	0.91	0.92	0.93	0.91	0.90	0.93	0.81	0.87
OliveOil	0.87	0.87	0.87	0.83	0.90	0.90	0.90	0.90	0.83
OSULeaf	0.96	0.90	0.92	0.95	0.95	0.88	0.96	0.84	0.93
PhalangesOutlinesCorrect	0.83	0.83	0.81	0.84	0.84	0.83	0.84	0.82	0.77
Phoneme	0.39	0.35	0.35	0.39	0.37	0.35	0.37	0.32	0.35
PigAirwayPressure	0.35	0.18	0.14	0.37	0.38	0.33	0.32	0.14	0.15
PigArtPressure	0.91	0.33	0.34	0.87	0.88	0.84	0.81	0.41	0.58
PigCVP	0.82	0.25	0.24	0.75	0.70	0.51	0.68	0.32	0.27
Plane	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PowerCons	0.89	0.88	0.90	0.93	0.91	0.90	0.90	0.91	0.93
ProximalPhalanxOutlineCorrect	0.89	0.85	0.85	0.89	0.89	0.90	0.89	0.87	0.85
ProximalPhalanxOutlineAgeGroup	0.86	0.85	0.87	0.87	0.86	0.84	0.87	0.86	0.85
ProximalPhalanxTW	0.83	0.82	0.82	0.80	0.81	0.82	0.81	0.81	0.80
RefrigerationDevices	0.58	0.57	0.58	0.52	0.53	0.55	0.55	0.51	0.59
ScreenType	0.51	0.49	0.46	0.51	0.51	0.38	0.54	0.48	0.46
SemgHandGenderCh2	0.87	0.90	0.90	0.89	0.89	0.88	0.92	0.68	0.80
SemgHandMovementCh2	0.66	0.67	0.71	0.59	0.58	0.59	0.64	0.39	0.61
SemgHandSubjectCh2	0.81	0.83	0.84	0.80	0.78	0.79	0.80	0.51	0.69
ShapeletSim	0.96	1.00	1.00	0.97	0.98	0.85	0.96	0.96	1.00
ShapesAll	0.86	0.81	0.83	0.85	0.85	0.82	0.85	0.81	0.83
SmallKitchenAppliances	0.82	0.81	0.82	0.83	0.78	0.81	0.83	0.79	0.81
SmoothSubspace	0.93	0.96	0.94	0.97	0.93	0.93	0.91	0.94	0.95
SonyAIBORobotSurface1	0.88	0.80	0.82	0.73	0.71	0.64	0.90	0.84	0.53
SonyAIBORobotSurface2	0.86	0.90	0.86	0.91	0.86	0.85	0.90	0.90	0.89
StarLightCurves	0.98	0.97	0.98	NaN	0.98	0.98	0.98	0.96	0.97
Strawberry	0.96	0.95	0.95	0.95	0.95	0.95	0.96	0.96	0.92
SwedishLeaf	0.94	0.92	0.94	0.96	0.95	0.93	0.95	0.95	0.93
Symbols	0.96	0.95	0.98	0.99	0.98	0.97	0.95	0.94	0.87
SyntheticControl	0.99	0.99	0.98	0.98	0.99	0.97	0.99	0.99	0.99
ToeSegmentation1	0.93	0.88	0.82	0.95	0.95	0.86	0.89	0.88	0.93
ToeSegmentation2	0.92	0.90	0.88	0.86	0.87	0.86	0.87	0.88	0.88
Trace	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TwoLeadECG	0.96	0.91	0.87	0.94	0.87	0.79	1.00	0.95	0.92
TwoPatterns	0.97	0.94	0.92	0.97	0.93	0.84	0.96	0.94	0.89
UMD	0.94	0.96	0.94	0.96	0.97	0.85	0.90	0.89	0.90

Table 8: Accuracy results of the different models with augmentations on the individual datasets with a Random Forest classifier. (Part 3/6)

	Chronos (Small)	ToTo	Mantis	NuTime	Moment (Large)	Moment (Base)	Moment (Large)	DTW (1-NN)	DTW (3-NN)
InsectWingbeatSound	0.56	0.61	0.51	0.62	0.55	0.52	0.48	0.36	0.35
ItalyPowerDemand	0.94	0.96	0.91	0.95	0.89	0.92	0.81	0.95	0.95
LargeKitchenAppliances	0.75	0.72	0.79	0.52	0.71	0.80	0.79	0.79	0.80
Lightning2	0.72	0.59	0.80	0.66	0.66	0.69	0.66	0.87	0.87
Lightning7	0.66	0.58	0.77	0.42	0.60	0.64	0.62	0.73	0.71
Mallat	0.76	0.75	0.90	0.88	0.50	0.55	0.53	0.93	0.93
Meat	0.87	0.87	0.93	0.92	0.40	0.35	0.35	0.93	0.93
MedicalImages	0.69	0.66	0.71	0.59	0.55	0.56	0.54	0.74	0.71
MiddlePhalanxOutlineCorrect	0.81	0.77	0.80	0.81	0.57	0.57	0.57	0.70	0.73
MiddlePhalanxOutlineAgeGroup	0.59	0.62	0.60	0.62	0.48	0.52	0.46	0.50	0.56
MiddlePhalanxTW	0.56	0.56	0.54	0.57	0.46	0.51	0.49	0.51	0.51
MixedShapesRegularTrain	0.95	0.96	0.94	0.89	0.78	0.82	0.80	0.84	0.83
MixedShapesSmallTrain	0.92	0.94	0.90	0.80	0.70	0.77	0.75	0.78	0.75
MoteStrain	0.88	0.88	0.92	0.86	0.85	0.85	0.79	0.83	0.81
NonInvasiveFetalECGThorax1	0.83	0.82	0.61	0.86	0.29	0.46	0.35	0.79	0.79
NonInvasiveFetalECGThorax2	0.87	0.86	0.68	0.90	0.35	0.53	0.42	0.86	0.86
OliveOil	0.83	0.47	0.93	0.90	0.37	0.40	0.43	0.83	0.87
OSULeaf	0.92	0.82	0.86	0.51	0.72	0.74	0.69	0.59	0.58
PhalangesOutlinesCorrect	0.75	0.74	0.77	0.81	0.62	0.64	0.63	0.73	0.76
Phoneme	0.35	0.36	0.33	0.15	0.28	0.28	0.26	0.23	0.21
PigAirwayPressure	0.12	0.22	0.50	0.04	0.05	0.06	0.06	0.18	0.12
PigArtPressure	0.49	0.50	0.91	0.09	0.22	0.37	0.31	0.48	0.36
PigCVP	0.23	0.44	0.77	0.13	0.12	0.25	0.18	0.33	0.23
Plane	0.99	0.98	1.00	0.99	0.91	0.97	0.91	1.00	1.00
PowerCons	0.94	0.94	0.91	0.88	0.82	0.85	0.77	0.92	0.86
ProximalPhalanxOutlineCorrect	0.80	0.80	0.80	0.90	0.69	0.73	0.68	0.78	0.83
ProximalPhalanxOutlineAgeGroup	0.86	0.86	0.85	0.86	0.80	0.80	0.80	0.80	0.81
ProximalPhalanxTW	0.80	0.81	0.78	0.80	0.60	0.67	0.59	0.76	0.77
RefrigerationDevices	0.53	0.58	0.51	0.46	0.51	0.56	0.54	0.46	0.46
ScreenType	0.47	0.43	0.44	0.43	0.39	0.47	0.47	0.40	0.39
SemgHandGenderCh2	0.82	0.83	0.90	0.78	0.66	0.67	0.68	0.92	0.91
SemgHandMovementCh2	0.61	0.52	0.73	0.38	0.24	0.27	0.32	0.78	0.76
SemgHandSubjectCh2	0.72	0.72	0.79	0.56	0.38	0.34	0.32	0.87	0.85
ShapeletSim	1.00	0.86	0.94	0.54	0.84	0.91	0.74	0.65	0.63
ShapesAll	0.84	0.76	0.83	0.71	0.68	0.68	0.64	0.77	0.71
SmallKitchenAppliances	0.82	0.79	0.81	0.78	0.70	0.72	0.74	0.64	0.67
SmoothSubspace	0.96	0.93	0.91	0.98	0.67	0.81	0.71	0.83	0.85
SonyAIBORobotSurface1	0.55	0.66	0.78	0.59	0.50	0.57	0.55	0.73	0.62
SonyAIBORobotSurface2	0.82	0.78	0.87	0.82	0.83	0.84	0.84	0.83	0.80
StarLightCurves	0.97	0.98	0.98	0.97	0.89	0.90	0.88	0.91	0.91
Strawberry	0.93	0.91	0.95	0.95	0.71	0.77	0.67	0.94	0.92
SwedishLeaf	0.92	0.87	0.92	0.90	0.67	0.70	0.65	0.79	0.77
Symbols	0.87	0.91	0.97	0.85	0.88	0.95	0.91	0.95	0.93
SyntheticControl	0.99	0.97	0.98	0.83	0.96	0.89	0.87	0.99	0.98
ToeSegmentation1	0.83	0.78	0.97	0.60	0.90	0.93	0.93	0.77	0.75
ToeSegmentation2	0.65	0.87	0.95	0.58	0.88	0.85	0.88	0.84	0.82
Trace	0.99	0.93	1.00	0.51	0.89	0.99	0.96	1.00	1.00
TwoLeadECG	0.98	0.79	1.00	0.69	0.63	0.70	0.69	0.90	0.85
TwoPatterns	0.80	0.87	0.88	0.57	0.86	0.83	0.76	1.00	1.00
UMD	0.81	0.91	0.97	0.81	0.83	0.88	0.85	0.88	0.85

Table 9: Accuracy results of the different models with augmentations on the individual datasets with a Random Forest classifier. (Part 4/6)

	TiRex	Chr. Bolt (Base)	Chr. Bolt (Small)	Moirai (Large)	Moirai (Base)	Moirai (Small)	TimesFM 2.0	TimesFM 1.0	Chronos (Base)
UWaveGestureLibraryAll	0.91	0.95	0.95	0.85	0.84	0.84	0.88	0.79	0.89
UWaveGestureLibraryX	0.81	0.80	0.82	0.79	0.78	0.77	0.73	0.70	0.81
UWaveGestureLibraryY	0.75	0.74	0.77	0.73	0.72	0.72	0.66	0.64	0.76
UWaveGestureLibraryZ	0.74	0.75	0.75	0.74	0.74	0.71	0.67	0.64	0.74
Wafer	1.00	1.00	0.99	0.99	0.99	0.99	1.00	1.00	1.00
Wine	0.72	0.78	0.61	0.72	0.70	0.81	0.85	0.76	0.54
WordSynonyms	0.54	0.53	0.58	0.48	0.47	0.45	0.49	0.49	0.52
Worms	0.82	0.68	0.70	0.81	0.83	0.75	0.77	0.69	0.69
WormsTwoClass	0.84	0.82	0.81	0.81	0.81	0.83	0.82	0.75	0.78
Yoga	0.80	0.84	0.85	0.83	0.77	0.80	0.80	0.77	0.82
AllGestureWiimoteX	0.60	0.62	0.61	0.62	0.60	0.54	0.67	0.64	0.53
AllGestureWiimoteY	0.70	0.66	0.70	0.66	0.65	0.62	0.72	0.68	0.57
AllGestureWiimoteZ	0.60	0.61	0.62	0.62	0.59	0.56	0.65	0.63	0.51
GestureMidAirD1	0.74	0.81	0.72	0.75	0.71	0.68	0.61	0.49	0.62
GestureMidAirD2	0.68	0.71	0.69	0.72	0.67	0.65	0.49	0.44	0.63
GestureMidAirD3	0.52	0.50	0.51	0.50	0.48	0.43	0.34	0.28	0.41
GesturePebbleZ1	0.86	0.86	0.86	0.84	0.87	0.87	0.85	0.83	0.76
GesturePebbleZ2	0.86	0.82	0.82	0.87	0.85	0.90	0.80	0.82	0.71
PickupGestureWiimoteZ	0.78	0.76	0.74	0.82	0.70	0.68	0.68	0.62	0.80
ShakeGestureWiimoteZ	0.86	0.88	0.86	0.84	0.82	0.82	0.92	0.90	0.88
DodgerLoopDay	0.47	0.52	0.57	0.49	0.43	0.49	0.48	0.42	0.48
DodgerLoopGame	0.69	0.84	0.76	0.75	0.80	0.70	0.62	0.65	0.70
DodgerLoopWeekend	0.90	0.92	0.94	0.88	0.94	0.90	0.98	0.97	0.97
MelbournePedestrian	0.92	0.93	0.93	0.90	0.90	0.91	0.90	0.90	0.93
ArticularyWordRecognition	0.99	1.00	1.00	0.97	0.99	0.98	0.98	0.98	0.97
AtrialFibrillation	0.27	0.33	0.33	0.07	0.33	0.33	0.07	0.47	0.20
BasicMotions	1.00	1.00	1.00	0.97	0.97	1.00	1.00	0.97	1.00
Cricket	1.00	0.99	1.00	0.96	0.94	0.94	1.00	0.69	0.93
Epilepsy	1.00	1.00	1.00	0.97	0.97	1.00	0.99	0.99	0.99
EthanolConcentration	0.37	0.37	0.37	0.36	0.33	0.47	0.37	0.54	0.54
ERing	0.97	0.96	0.99	0.88	0.93	0.93	0.97	0.93	0.95
FaceDetection	0.63	0.57	0.58	0.57	0.59	0.58	NaN	NaN	0.57
FingerMovements	0.48	0.54	0.52	0.56	0.48	0.51	0.52	0.53	0.47
HandMovementDirection	0.32	0.30	0.26	0.19	0.26	0.32	0.22	0.22	0.31
Handwriting	0.22	0.26	0.24	0.22	0.20	0.22	0.26	0.29	0.18
Heartbeat	0.73	0.74	0.76	0.74	0.73	0.75	0.73	0.73	0.73
Libras	0.87	0.88	0.88	0.73	0.73	0.81	0.84	0.84	0.87
LSST	0.59	0.61	0.61	0.60	0.60	0.60	0.58	0.56	0.58
NATOPS	0.81	0.82	0.89	0.86	0.82	0.84	0.80	0.83	0.82
PenDigits	0.97	0.96	0.96	NaN	0.95	0.95	0.97	0.96	0.95
PEMS-SF	1.00	0.95	0.99	0.99	1.00	0.99	NaN	1.00	0.98
PhonemeSpectra	0.26	0.25	0.24	0.27	0.25	0.22	0.27	0.24	0.26
RacketSports	0.83	0.86	0.86	0.80	0.81	0.75	0.84	0.90	0.84
SelfRegulationSCP1	0.84	0.79	0.82	0.75	0.77	0.77	0.82	0.76	0.75
SelfRegulationSCP2	0.57	0.54	0.53	0.56	0.54	0.49	0.50	0.47	0.43
UWaveGestureLibrary	0.88	0.86	0.86	0.81	0.83	0.88	0.68	0.68	0.81
CharacterTrajectories	0.96	0.97	0.98	0.93	0.93	0.96	0.95	0.95	0.96
JapaneseVowels	0.89	0.92	0.94	0.85	0.88	0.93	0.83	0.85	0.93
SpokenArabicDigits	0.96	0.97	0.97	NaN	0.96	0.94	NaN	0.96	0.97

Table 10: Accuracy results of the different models with augmentations on the individual datasets with a Random Forest classifier. (Part 5/6)

	Chronos (Small)	ToTo	Mantis	NuTime	Moment (Large)	Moment (Base)	Moment (Large)	DTW (1-NN)	DTW (3-NN)
UWaveGestureLibraryAll	0.90	0.88	0.85	0.88	0.73	0.68	0.66	0.89	0.90
UWaveGestureLibraryX	0.80	0.76	0.77	0.71	0.74	0.72	0.71	0.73	0.74
UWaveGestureLibraryY	0.75	0.68	0.69	0.65	0.65	0.64	0.62	0.63	0.63
UWaveGestureLibraryZ	0.72	0.71	0.73	0.65	0.66	0.67	0.66	0.66	0.67
Wafer	1.00	0.99	0.99	0.99	0.92	0.91	0.90	0.98	0.98
Wine	0.54	0.48	0.80	0.80	0.52	0.56	0.44	0.57	0.57
WordSynonyms	0.54	0.45	0.58	0.44	0.43	0.42	0.41	0.65	0.60
Worms	0.73	0.75	0.65	0.56	0.62	0.64	0.61	0.58	0.39
WormsTwoClass	0.81	0.83	0.79	0.60	0.73	0.77	0.77	0.62	0.55
Yoga	0.84	0.72	0.82	0.77	0.66	0.62	0.62	0.84	0.82
AllGestureWiimoteX	0.52	0.59	0.60	0.29	0.52	0.58	0.59	0.71	0.62
AllGestureWiimoteY	0.50	0.60	0.59	0.30	0.54	0.64	0.61	0.68	0.61
AllGestureWiimoteZ	0.51	0.56	0.63	0.28	0.48	0.56	0.53	0.70	0.64
GestureMidAirD1	0.67	0.58	0.58	0.54	0.66	0.66	0.58	0.45	0.39
GestureMidAirD2	0.65	0.55	0.65	0.47	0.60	0.58	0.63	0.32	0.33
GestureMidAirD3	0.41	0.32	0.33	0.33	0.37	0.39	0.34	0.18	0.15
GesturePebbleZ1	0.73	0.72	0.88	0.65	0.77	0.80	0.79	0.69	0.72
GesturePebbleZ2	0.72	0.66	0.86	0.60	0.77	0.74	0.76	0.67	0.69
PickupGestureWiimoteZ	0.70	0.70	0.88	0.32	0.66	0.64	0.54	0.74	0.68
ShakeGestureWiimoteZ	0.90	0.76	0.86	0.34	0.76	0.80	0.80	0.86	0.90
DodgerLoopDay	0.53	0.48	0.51	0.32	0.38	0.39	0.38	0.45	0.44
DodgerLoopGame	0.63	0.72	0.76	0.58	0.70	0.68	0.69	0.90	0.88
DodgerLoopWeekend	0.94	0.83	0.95	0.89	0.93	0.87	0.90	0.95	0.96
MelbournePedestrian	0.92	0.87	0.91	0.97	0.59	0.68	0.64	0.88	0.88
ArticularyWordRecognition	0.98	0.96	0.99	0.88	0.79	0.80	0.78	0.99	0.98
AtrialFibrillation	0.20	0.07	0.27	0.33	0.27	0.13	0.13	0.20	0.33
BasicMotions	1.00	1.00	1.00	0.85	0.97	0.95	0.95	0.97	0.85
Cricket	0.90	0.99	1.00	0.72	0.46	0.44	0.47	1.00	1.00
Epilepsy	0.99	0.99	1.00	0.88	0.98	0.98	0.98	0.96	0.95
EthanolConcentration	0.47	0.38	0.30	0.53	0.26	0.33	0.25	0.32	0.28
ERing	0.94	0.86	0.93	0.84	0.79	0.86	0.81	0.91	0.93
FaceDetection	0.56	0.58	0.52	0.55	0.52	0.51	0.51	0.53	0.54
FingerMovements	0.51	0.55	0.54	0.56	0.47	0.53	0.54	0.53	0.54
HandMovementDirection	0.28	0.28	0.28	0.26	0.18	0.24	0.24	0.19	0.20
Handwriting	0.17	0.15	0.33	0.13	0.16	0.15	0.12	0.61	0.50
Heartbeat	0.76	0.75	0.79	0.72	0.72	0.70	0.71	0.72	0.73
Libras	0.89	0.84	0.88	0.78	0.44	0.56	0.49	0.87	0.86
LSST	0.55	0.56	0.61	0.43	0.52	0.51	0.49	0.55	0.56
NATOPS	0.82	0.83	0.92	0.72	0.62	0.58	0.54	0.88	0.87
PenDigits	0.95	0.96	0.94	0.96	0.72	0.80	0.77	0.98	0.98
PEMS-SF	0.98	0.83	0.99	0.98	NaN	NaN	0.83	0.71	0.53
PhonemeSpectra	0.25	0.20	0.27	0.11	0.19	0.22	0.21	0.15	0.14
RacketSports	0.88	0.82	0.92	0.83	0.58	0.59	0.51	0.80	0.83
SelfRegulationSCP1	0.76	0.82	0.80	0.75	0.69	0.62	0.62	0.77	0.83
SelfRegulationSCP2	0.52	0.52	0.46	0.45	0.50	0.48	0.49	0.54	0.49
UWaveGestureLibrary	0.85	0.83	0.82	0.81	0.62	0.65	0.64	0.90	0.90
CharacterTrajectories	0.97	0.96	0.96	0.95	0.83	0.83	0.81	0.99	0.98
JapaneseVowels	0.94	0.93	0.96	0.95	0.39	0.39	0.32	0.96	0.96
SpokenArabicDigits	0.97	0.95	0.95	0.93	0.81	0.78	0.76	0.97	0.97

Table 11: Accuracy results of the different models with augmentations on the individual datasets with a Random Forest classifier. (Part 6/6)