TIMEE: TOWARDS END-TO-END TIME SERIES CLASSIFICATION VIA IN-CONTEXT LEARNING

Jaris Küken [1,2,*], **Shi Bin Hoo** [1,*], **Lennart Purucker** [3,1], **Frank Hutter** [3,4,1]
[1] University of Freiburg, [2] Zuse School ELIZA Darmstadt, [3] Prior Labs, [4] ELLIS Institute Tübingen
[*] Equal contribution
{kuekenj,hoos}@cs.uni-freiburg.de

ABSTRACT

We introduce TimEE[1], a 2M-parameter foundation model for end-to-end time series classification via in-context learning. Unlike prior works that rely on decoupled feature encoders and task-specific classifiers, TimEE utilizes a unified framework to directly approximate the conditional predictive distribution of a test sample given the training set. Concretely, it enables both temporal reasoning and classification within a single forward pass. Evaluated on 42 binary classification datasets from the UCR Time Series Archive, TimEE outperforms default linear-probing baselines and matches the performance of models up to 60× larger, while reducing runtime by up to an order of magnitude. Our results suggest that end-to-end trained foundation models are an effective and computationally efficient alternative to the two-stage paradigm for time series classification.

Track: Research

1 INTRODUCTION

Time series classification (TSC) is an essential predictive modeling task in domains ranging from healthcare (Wang et al., 2023; Esgalhado et al., 2021), to finance (Goldstein et al., 2021) and energy Elsevier (2018). While related fields like forecasting and tabular learning have seen the emergence of foundation models with the in-context learning paradigm (Ansari et al., 2025; Hollmann et al., 2025), TSC research remains largely centered on a two-stage paradigm: using a pre-trained encoder (e.g., MOMENT (Goswami et al., 2024) or Mantis (Feofanov et al., 2025) followed by a task-specific classifier like a Support Vector Machine (Cortes & Vapnik, 1995) or Random Forest.

This separation introduces practical and conceptual limitations, such as per-task classifier evaluation, hyperparameter tuning, and repeated training. Moreover, this often increases the risk of overfitting, especially in the low-data regimes typical of benchmarks such as the UCR Time Series Archive (Chen et al., 2015).

To address these limitations, we introduce TimEE, a foundation model designed for end-to-end TSC via in-context learning. By unifying feature extraction and classification into a single-forward pass, TimEE eliminates the need for task-specific training. Moreover, TimEE benefits from transfer learning across both feature extraction and classification. Despite its modest scale of only 2M parameters, TimEE achieves competitive performance against MOMENT on UCR binary classification tasks while providing a $10 - 30\times$ inference speedup.

2 BACKGROUND

Time Series Classification. Consider a univariate time-series classification problem with a labeled dataset:

$$\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{n},$$

[1] TimEE pronounced as 'Timmy'

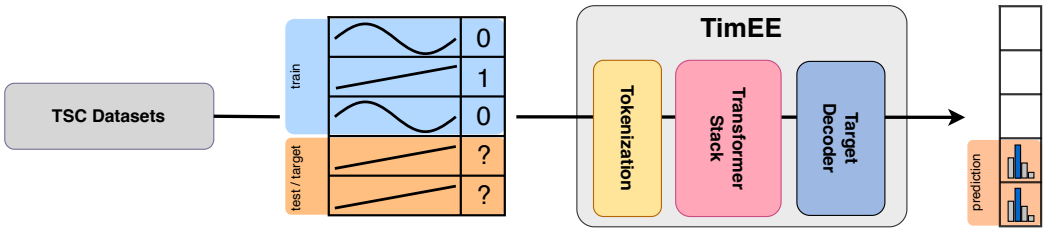

Figure 1: **TimEE pipeline.** We provide a high-level overview of the TimEE pipeline for time series classification. Given a task defined by a labeled support set and a query time series, TimEE operates in three stages. First, each input time series is tokenized into a sequence of fixed-dimensional embeddings. These tokens are then processed by a transformer stack, which jointly models interactions across the support and query examples. Finally, the resulting representations are decoded to produce task-specific predictions.

where $x_i \in \mathbb{R}^t$ and $y_i \in \mathcal{Y}$. Given a test observation x_{test}, we aim to model the predictive distribution $p(\cdot|x_{test}, \mathcal{D}_{train})$, effectively mapping the input sequence to a distribution over the label space $\mathcal{Y}$.

Two-Stage Paradigm. Many state-of-the-art approaches follow a decoupled, two-stage procedure. First, each time series is mapped to a $d-$dimensional embedding using a feature encoder:

$$F_\theta : \mathbb{R}^t \to \mathbb{R}^d, \quad z_i = F_\theta(x_i)$$

where F_θ is typically a pre-trained time-series foundation model (Feofanov et al., 2025; Goswami et al., 2024; Auer et al., 2025). Subsequently, a task-specific classifier G_ϕ is trained on these embeddings to produce predictions:

$$\hat{y}_i = G_\phi\big(F_\theta(x_i)\big).$$

Following Feofanov et al. (2025) and Goswami et al. (2024), G_ϕ is often implemented as a linear probe or a conventional classifier, such as a Support Vector Machine (SVM) or a Random Forest (RF). While effective, this paradigm decouples representation learning from the downstream task. As the encoder is optimized independently of the classification objective, the extracted features may be suboptimal for capturing discriminative information (Shi et al., 2022).

In-Context Learning (ICL). Originally popularized in large language modeling (Brown et al.), ICL has demonstrated state-of-the-art performance in tabular prediction (Hollmann et al.; 2025) and time-series forecasting (Ansari et al., 2025). In the context of time series classification, ICL models generate predictions by conditioning directly on a labeled support set of examples. To adapt ICL to this domain, Fang et al. (2026) proposes first pre-training a feature encoder, and subsequently pre-training an ICL head atop a feature encoder. While this framework achieves strong empirical results, it remains a two-stage paradigm with inherently limited interaction between the feature encoder and the classification objective.

3 Method

In contrast to the two-stage paradigm, we propose a unified framework that directly approximates the conditional predictive distribution $p(\cdot|x_{test}, \mathcal{D}_{train})$. This formulation eliminates the need for independent feature pre-training and task-specific classification heads. It enables end-to-end optimization of the entire predictive pipeline. Consequently, the model learns to perform joint temporal reasoning and contextual classification in a single pass, mitigating the suboptimality inherent in disjoint optimization stages.

3.1 Model Architecture

Taking inspiration from Chronos-2 (Ansari et al., 2025) and TabPFN (Hollmann et al., 2025), the architecture of TimEE aims to capture two distinct types of dependencies: intra-series temporal

dynamics (the "shape" of the sequence) and inter-series contextual relationships (the "comparison" between sequences).

Tokenization. Following Chronos-2, we first normalize the input time series and apply a $\sinh^{-1}$ transformation to handle varying scales and outliers. Then, each time series is split into non-overlapping patches of size P (Nie et al., 2023), which are projected into a d-dimensional embedding via a learnable linear projection layer. We use $P = 16$ and $d = 128$. The class labels (targets) are also projected into $d-$dimensional embedding using a separate projection layer, allowing labels and time-series patches to interact within a shared representation space.

Transformer Stack. TimEE utilizes the interweaving dual-axis attention to model dependencies across both time and samples. **(1) Temporal Axis:** We apply self-attention across all patches within an individual time series. We incorporate Rotary Positional Embeddings (RoPE) (Su et al., 2021) to preserve temporal order. Notably, we exclude RoPE from the target embeddings to maintain their identity as categorical indicators rather than sequential elements. **(2) Cross-Series Axis:** This allows each patch and target embedding to attend to representations from other samples in $\mathcal{D}_{train}$. This mechanism is the core of our in-context learning capability, as it enables the model to dynamically compare the test sample against the labeled support set.

Target Decoder. Finally, the processed embeddings for the test samples are passed through a decoding layer to directly model the conditional predictive distribution $p(\cdot|x_{test}, \mathcal{D}_{train})$, producing class probabilities in a single forward pass.

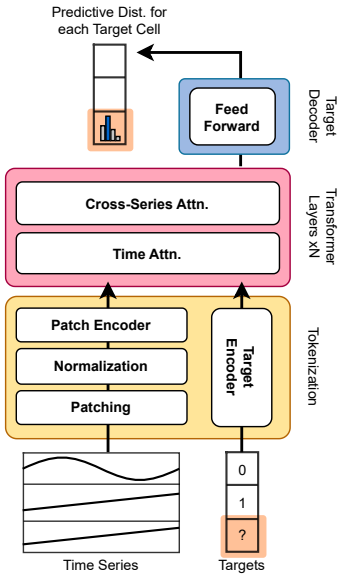

Figure 2: **Architecture of TimEE.** Each input time series is split into patches, normalized, and encoded into patch embeddings, while labels are embedded via a separate target encoder. The resulting embeddings are processed by a transformer with alternating temporal and sample-wise attention to capture intra- and inter-series dependencies. The updated target embeddings are decoded into class probabilities.

3.2 Data and Training Objective

To develop ICL capabilities in TimEE, like TabPFN, we employ a meta-learning strategy. At each iteration, we construct a task $\mathcal{T}_k$ by generating a synthetic dataset $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N}$. Each $\mathcal{D}_k$ is then partitioned into a train set $\mathcal{D}_{k,train}$ and a test set $\mathcal{D}_{k,test}$:

$$\mathcal{D}_{k,train} = \{(x_j, y_j)\}_{j=1}^{n_{train}}, \quad \mathcal{D}_{k,test} = \{(x_l, y_l)\}_{l=1}^{n_{test}}$$

The model is optimized to minimize the negative log-likelihood over the query set:

$$\mathcal{L}(\theta) = -\mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}\left[\sum_{(x,y)\in\mathcal{D}_{test}} \log p_\theta(y \mid x, \mathcal{D}_{train})\right]$$

To achieve this, we develop a pipeline the constructs augmented classification tasks from the train split of the UCR datasets. Concretely, we generate synthetic samples via weighted Mixup (Zhang et al., 2018), where multiple sequences within a dataset are linearly combined. Pseudo-labels are then assigned via a majority-weight thresholding criterion. See Appendix 1 for pseudo-code.

4 Experiments

Evaluation Setup. We pretrain TimEE on a single Nvidia L40s on approx. 700K augmented TSC datasets, and evaluate it on the UCR Time Series Archive (Chen et al., 2015), focusing on the 42

Table 1: **UCR Benchmark Results over 42 binary classification tasks.** We report mean accuracy and ROC AUC scores per model averaged over 42 binary classification tasks. Standard deviations are reported over 3 unique seeds for TimEE. TimEE significantly outperforms MOMENT-SVM variants while remains competitive against MOMENT-RF(S) (see Figure 4).

Model	Accuracy in (%)	ROC AUC
Moment(L) + RF	**85.57** ± **0.00**	**0.920** ± **0.000**
Moment(S) + RF	85.36 ± 0.00	0.911 ± 0.000
TimEE	81.71 ± 2.96	0.883 ± 0.022
Moment(L) + SVM	70.56 ± 0.00	0.645 ± 0.000
Moment(S) + SVM	65.59 ± 0.00	0.570 ± 0.000

binary classification tasks over 3 seeds. Following standard protocol, we use the official benchmark train/test splits. We emphasize that all UCR test splits were strictly excluded from our pre-training corpus to eliminate data leakage during evaluation. We compare our 2M parameter model against two variants of MOMENT (Goswami et al., 2024): MOMENT-Small (40M) and MOMENT-Large (125M). Consistent with the two-stage paradigm literature (Section 2), we evaluate these baselines by training two downstream classifiers—a Support Vector Machine (SVM) and a Random Forest (RF)—on the extracted embeddings. In contrast, TimEE performs inference in a single forward pass without any task-specific weight updates or tuning.

Results and Discussion. TimEE demonstrates that architectural alignment can match the performance of significantly larger, decoupled framework. As shown in Table 1, our 2M-parameter model significantly outperforms the MOMENT+SVM baselines, indicating that our end-to-end objective meta-learns classification that surpasses SVMs. The MOMENT+RF configuration serves as an upper bound: our small model does not yet rival a task-specific high-capacity RF.

We also compare computational efficiency in Figure 3. As the inference of TimEE is performed in a single forward pass, this leads to a significant speedup in prediction times, compared to the two-staged baselines. Overall, TimEE provides a strong balance between predictive performance and computational efficiency. We provide further evaluation details in Appendix B.

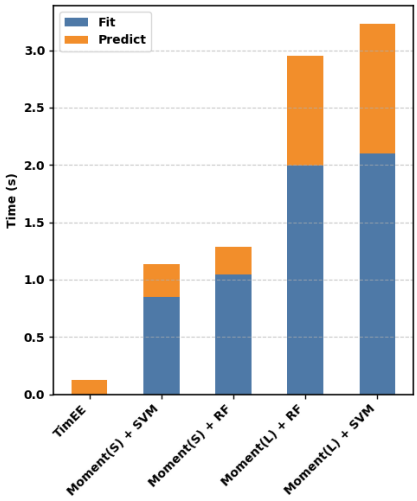

Figure 3: **Fit and predict times.** We report aggregated runtimes for training and inference across all evaluation datasets. TimEE is substantially more efficient than MOMENT, achieving on average a $10\times$ speedup over MOMENT small and up to a $30\times$ speedup over MOMENT large.

5 Conclusion

We present TimEE, an end-to-end foundation model for time series classification. By unifying representation learning and classification into a single objective, we demonstrate that a 2M-parameter model can match or exceed the performance of significantly larger, two-stage pipelines. Our findings suggest that: In contrast to existing approaches, which typically first train a feature encoder using contrastive learning and then train a task-specific model on top of that, we show that training a model to directly optimize the prediction task can be a viable alternative. Furthermore, we eliminate the need to choose the best task-specific classifier.

Limitations & Future Work. Our current pre-training relies on a limited distribution of data sampled and augmented from the UCR archive. Since it remains challenging to collect real-world TSC datasets, exploring synthetic TSC data offers a promising direction for future work. Additionally, while this work focused on univariate time series, extending the framework to multivariate time se-

ries or to multimodal settings that combine time series with additional inputs (e.g., textual metadata) represents other impactful yet unexplored areas. Finally, we aim to scale up our model in pretraining data, compute, and model size to fully unlock the potential of the end-to-end paradigm.

Acknowledgments J.K. is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD program Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research. L.P. acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under SFB 1597 (SmallData), grant number 499552394; F.H. acknowledges the financial support of the Hector Foundation. Finally, we thank the reviewers for their constructive feedback and contribution to improving the paper.

REFERENCES

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From Univariate to Universal Forecasting, October 2025. URL http://arxiv.org/abs/2510.15821. arXiv:2510.15821 [cs].

Andreas Auer, Daniel Klotz, Sebastian Böck, and Sepp Hochreiter. Pre-trained Forecasting Models: Strong Zero-Shot Feature Extractors for Time Series Classification. 2025.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. 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. pp. 1877–1901.

Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The UCR Time Series Classification Archive, July 2015.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Elsevier. Time-Series Classification Methods: Review and Applications to Power Systems Data. In *Big Data Application in Power Systems*, pp. 179–220. Elsevier, January 2018. doi: 10.1016/B978-0-12-811968-6.00009-7. URL https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780128119686000097.

Filipa Esgalhado, Beatriz Fernandes, Valentina Vassilenko, Arnaldo Batista, and Sara Russo. The Application of Deep Learning Algorithms for PPG Signal Processing and Classification. *Computers*, 10(12):158, December 2021. ISSN 2073-431X. doi: 10.3390/computers10120158. URL https://www.mdpi.com/2073-431X/10/12/158.

Juntao Fang, Shifeng Xie, Shengbin Nie, Yuhui Ling, Yuming Liu, Zijian Li, Keli Zhang, Lujia Pan, Themis Palpanas, and Ruichu Cai. Rethinking Zero-Shot Time Series Classification: From Task-specific Classifiers to In-Context Inference, January 2026. URL http://arxiv.org/abs/2602.00620. arXiv:2602.00620 [cs].

Vasilii Feofanov, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight Calibrated Foundation Model for User-Friendly Time Series Classification, February 2025. URL http://arxiv.org/abs/2502.15637. arXiv:2502.15637 [cs].

Itay Goldstein, Chester S Spatt, and Mao Ye. Big Data in Finance. *The Review of Financial Studies*, 34(7):3213–3225, July 2021. ISSN 0893-9454. doi: 10.1093/rfs/hhab038. URL https://doi.org/10.1093/rfs/hhab038.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: a family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 16115–16152, Vienna, Austria, July 2024. JMLR.org.

N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. TabPFN: A transformer that solves small tabular classification problems in a second.

N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A TIME SERIES IS WORTH 64 WORDS: LONG-TERM FORECASTING WITH TRANSFORMERS. 2023.

Zhenmei Shi, Jiefeng Chen, Kunyang Li, Jayaram Raghuram, Xi Wu, Yingyu Liang, and Somesh Jha. The Trade-off between Universality and Label Efficiency of Representations from Contrastive Learning. September 2022. URL https://openreview.net/forum?id=rvsbw2YthH_.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.

Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. Contrast Everything: A Hierarchical Contrastive Framework for Medical Time-Series. November 2023. URL https://openreview.net/forum?id=sOQBHlCmzp&noteId=ABYkEiBrAz.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

APPENDIX

A DATA GENERATION

Algorithm 1 Synthetic Time Series Mixture Generation

Require: Collection of UCR Datasets $\mathcal{D}$, Target Length T

$X_{out}, Y_{out} \leftarrow \emptyset$
Select random dataset $D_i \in \mathcal{D}$
Sample subset of time series $\{(x, y)\} \subset D_i$
Preprocessing:
$x \leftarrow \text{Normalize}(x)$
$x \leftarrow \text{Resize}(x, T)$ ▷ Interpolate or truncate
Mixture Generation:
Select k samples $\{x_j\}_{j=1}^{k}$ with labels $\{y_j\}_{j=1}^{k}$
Sample weights $\lambda \sim \text{Dirichlet}(\alpha)$
$x_{mix} \leftarrow \sum_{j=1}^{k} (\lambda_j \cdot x_j) + \mathcal{N}(0, \sigma_{noise})$
Pseudo-Label Assignment:
Assign y_{mix} based on λ and $\{y_j\}$ using threshold
Append (x_{mix}, y_{mix}) to (X_{out}, Y_{out})
return (X_{out}, Y_{out})

B EXPERIMENTS

B.1 CRITICAL DIFFERENCE DIAGRAM BASED ON ACCURACY

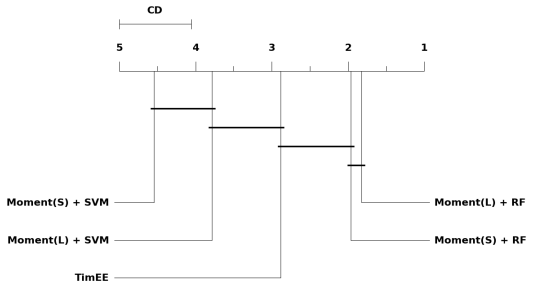

Figure 4: Critifcal Difference Diagram comparing TimEE against baselines. The diagram shows the average rank of each model across the 42 binary classification tasks. Thick horizontal lines indicate groups of models (cliques) between which there is no statistically significant difference in accuracy ($p > 0.05$).

B.2 Detailed Timing Results

Table 2: **Efficiency of all models**. We provide the runtime of the fit and predict operation as well as the combined runtime for all models. TimEE has the lowest fit-, predict- and total time amongst all models.

Model	Fit Time (in s)	Predict time (in s)	Total time (in s)
TimEE	0.000000	0.125197	0.125197
Moment (Small) + SVM	0.848109	0.283990	1.132099
Moment (Small) + RF	1.043201	0.242732	1.285933
Moment (Large) + RF	1.993732	0.956422	2.950155
Moment (Large) + SVM	2.103362	1.123350	3.226712

B.3 Per Dataset Results

Table 3: **UCR Benchmark Results over 42 binary classification tasks.** We report accuracy scores per model with corresponding standard deviations for each dataset.

Dataset	Moment(L) (RF)	Moment(L) (SVM)	Moment(S) (RF)	Moment(S) (SVM)	TimEE
BeetleFly	**100.00 ± 0.00**	90.00 ± 0.00	95.00 ± 0.00	90.00 ± 0.00	81.67 ± 7.64
BirdChicken	80.00 ± 0.00	75.00 ± 0.00	85.00 ± 0.00	70.00 ± 0.00	**86.67 ± 5.77**
Chinatown	97.38 ± 0.00	97.38 ± 0.00	**97.67 ± 0.00**	95.63 ± 0.00	81.05 ± 10.23
Coffee	**100.00 ± 0.00**	89.29 ± 0.00	96.43 ± 0.00	96.43 ± 0.00	84.52 ± 5.46
Computers	65.60 ± 0.00	65.60 ± 0.00	**67.60 ± 0.00**	56.00 ± 0.00	62.40 ± 2.80
DistalPhalanxOutlineCorrect	**78.62 ± 0.00**	58.70 ± 0.00	78.62 ± 0.00	58.33 ± 0.00	73.67 ± 1.46
DodgerLoopGame	76.38 ± 0.00	51.18 ± 0.00	70.08 ± 0.00	51.18 ± 0.00	**84.51 ± 2.41**
DodgerLoopWeekend	93.65 ± 0.00	26.98 ± 0.00	90.48 ± 0.00	26.98 ± 0.00	**97.35 ± 0.92**
ECG200	86.00 ± 0.00	64.00 ± 0.00	**89.00 ± 0.00**	64.00 ± 0.00	84.67 ± 2.31
ECGFiveDays	84.55 ± 0.00	49.71 ± 0.00	**92.33 ± 0.00**	49.71 ± 0.00	71.70 ± 2.73
Earthquakes	**75.54 ± 0.00**	74.82 ± 0.00	74.82 ± 0.00	74.82 ± 0.00	75.06 ± 1.10
FordA	89.77 ± 0.00	89.17 ± 0.00	**93.26 ± 0.00**	87.12 ± 0.00	91.82 ± 0.46
FordB	77.16 ± 0.00	76.79 ± 0.00	**79.01 ± 0.00**	65.31 ± 0.00	75.43 ± 1.84
FreezerRegularTrain	89.23 ± 0.00	76.00 ± 0.00	89.47 ± 0.00	66.81 ± 0.00	**94.14 ± 3.78**
FreezerSmallTrain	75.05 ± 0.00	75.93 ± 0.00	76.70 ± 0.00	74.81 ± 0.00	**77.93 ± 1.73**
GunPoint	**97.33 ± 0.00**	49.33 ± 0.00	96.00 ± 0.00	49.33 ± 0.00	85.78 ± 2.14
GunPointAgeSpan	**94.30 ± 0.00**	85.44 ± 0.00	93.99 ± 0.00	50.63 ± 0.00	93.78 ± 1.93
GunPointMaleVersusFemale	98.42 ± 0.00	92.41 ± 0.00	**98.73 ± 0.00**	58.23 ± 0.00	98.73 ± 0.32
GunPointOldVersusYoung	96.51 ± 0.00	81.59 ± 0.00	**97.46 ± 0.00**	52.38 ± 0.00	95.34 ± 1.20
Ham	69.52 ± 0.00	59.05 ± 0.00	**71.43 ± 0.00**	51.43 ± 0.00	64.76 ± 4.36
HandOutlines	**90.54 ± 0.00**	76.49 ± 0.00	87.84 ± 0.00	64.05 ± 0.00	85.50 ± 1.65
Herring	**60.94 ± 0.00**	59.38 ± 0.00	57.81 ± 0.00	59.38 ± 0.00	55.21 ± 3.93
HouseTwenty	**91.60 ± 0.00**	87.39 ± 0.00	89.92 ± 0.00	81.51 ± 0.00	87.11 ± 0.49
ItalyPowerDemand	**95.63 ± 0.00**	95.53 ± 0.00	95.14 ± 0.00	95.24 ± 0.00	91.90 ± 0.54
Lightning2	70.49 ± 0.00	54.10 ± 0.00	72.13 ± 0.00	54.10 ± 0.00	**79.23 ± 2.50**
MiddlePhalanxOutlineCorrect	**84.54 ± 0.00**	57.04 ± 0.00	80.76 ± 0.00	57.04 ± 0.00	68.73 ± 3.87
MoteStrain	**89.78 ± 0.00**	89.70 ± 0.00	86.10 ± 0.00	85.06 ± 0.00	84.03 ± 0.35
PhalangesOutlinesCorrect	**83.92 ± 0.00**	61.42 ± 0.00	81.82 ± 0.00	61.31 ± 0.00	69.77 ± 0.55
PowerCons	85.00 ± 0.00	81.67 ± 0.00	**89.44 ± 0.00**	65.56 ± 0.00	88.70 ± 2.63
ProximalPhalanxOutlineCorrect	**87.97 ± 0.00**	68.38 ± 0.00	85.57 ± 0.00	68.38 ± 0.00	77.09 ± 4.43
SemgHandGenderCh2	79.83 ± 0.00	71.83 ± 0.00	81.00 ± 0.00	70.33 ± 0.00	**91.78 ± 1.73**
ShapeletSim	**91.11 ± 0.00**	78.33 ± 0.00	78.89 ± 0.00	65.00 ± 0.00	86.30 ± 7.56
SonyAIBORobotSurface1	74.38 ± 0.00	42.93 ± 0.00	83.03 ± 0.00	42.93 ± 0.00	**84.75 ± 7.36**
SonyAIBORobotSurface2	**88.77 ± 0.00**	62.12 ± 0.00	87.72 ± 0.00	61.70 ± 0.00	81.25 ± 2.24
Strawberry	**94.59 ± 0.00**	64.32 ± 0.00	94.32 ± 0.00	64.32 ± 0.00	93.87 ± 1.28
ToeSegmentation1	**93.42 ± 0.00**	90.35 ± 0.00	92.54 ± 0.00	86.84 ± 0.00	77.63 ± 3.04
ToeSegmentation2	86.15 ± 0.00	**86.92 ± 0.00**	86.92 ± 0.00	83.08 ± 0.00	84.62 ± 1.33
TwoLeadECG	82.88 ± 0.00	49.96 ± 0.00	**94.29 ± 0.00**	49.96 ± 0.00	71.85 ± 4.70
Wafer	**98.96 ± 0.00**	95.23 ± 0.00	97.42 ± 0.00	89.21 ± 0.00	97.87 ± 0.29
Wine	**81.48 ± 0.00**	50.00 ± 0.00	64.81 ± 0.00	50.00 ± 0.00	60.49 ± 11.32
WormsTwoClass	72.73 ± 0.00	58.44 ± 0.00	**84.42 ± 0.00**	57.14 ± 0.00	75.76 ± 1.50
Yoga	**84.03 ± 0.00**	53.57 ± 0.00	79.97 ± 0.00	53.57 ± 0.00	77.46 ± 0.45

Table 4: **UCR Benchmark Results over 42 binary classification tasks.** We report ROC AUC scores per model with corresponding standard deviations for each dataset.

Dataset	Moment(L) (RF)	Moment(L) (SVM)	Moment(S) (RF)	Moment(S) (SVM)	TimEE
BeetleFly	**1.000 ± 0.000**	0.020 ± 0.000	0.990 ± 0.000	0.130 ± 0.000	0.983 ± 0.015
BirdChicken	0.940 ± 0.000	0.130 ± 0.000	0.930 ± 0.000	0.500 ± 0.000	**0.963 ± 0.047**
Chinatown	0.995 ± 0.000	**0.996 ± 0.000**	0.992 ± 0.000	0.011 ± 0.000	0.937 ± 0.042
Coffee	**1.000 ± 0.000**	0.026 ± 0.000	1.000 ± 0.000	0.500 ± 0.000	0.858 ± 0.061
Computers	0.722 ± 0.000	0.686 ± 0.000	**0.723 ± 0.000**	0.664 ± 0.000	0.703 ± 0.029
DistalPhalanxOutlineCorrect	**0.874 ± 0.000**	0.778 ± 0.000	0.866 ± 0.000	0.772 ± 0.000	0.819 ± 0.005
DodgerLoopGame	0.866 ± 0.000	0.133 ± 0.000	0.774 ± 0.000	0.393 ± 0.000	**0.915 ± 0.024**
DodgerLoopWeekend	**0.988 ± 0.000**	0.018 ± 0.000	0.973 ± 0.000	0.070 ± 0.000	0.987 ± 0.007
ECG200	0.940 ± 0.000	0.904 ± 0.000	**0.949 ± 0.000**	0.922 ± 0.000	0.905 ± 0.010
ECGFiveDays	0.967 ± 0.000	0.979 ± 0.000	**0.990 ± 0.000**	0.971 ± 0.000	0.879 ± 0.022
Earthquakes	0.635 ± 0.000	0.551 ± 0.000	0.606 ± 0.000	0.623 ± 0.000	**0.709 ± 0.025**
FordA	0.962 ± 0.000	0.962 ± 0.000	**0.977 ± 0.000**	0.950 ± 0.000	0.976 ± 0.001
FordB	0.846 ± 0.000	0.840 ± 0.000	**0.878 ± 0.000**	0.766 ± 0.000	0.833 ± 0.015
FreezerRegularTrain	0.964 ± 0.000	0.823 ± 0.000	**0.974 ± 0.000**	0.803 ± 0.000	0.967 ± 0.038
FreezerSmallTrain	0.855 ± 0.000	0.821 ± 0.000	0.869 ± 0.000	0.240 ± 0.000	**0.876 ± 0.036**
GunPoint	**0.997 ± 0.000**	0.044 ± 0.000	0.992 ± 0.000	0.116 ± 0.000	0.939 ± 0.023
GunPointAgeSpan	0.991 ± 0.000	0.946 ± 0.000	**0.992 ± 0.000**	0.777 ± 0.000	0.988 ± 0.003
GunPointMaleVersusFemale	0.999 ± 0.000	0.999 ± 0.000	**1.000 ± 0.000**	0.999 ± 0.000	0.999 ± 0.000
GunPointOldVersusYoung	0.993 ± 0.000	0.919 ± 0.000	**0.997 ± 0.000**	0.318 ± 0.000	0.988 ± 0.002
Ham	**0.771 ± 0.000**	0.667 ± 0.000	0.741 ± 0.000	0.271 ± 0.000	0.700 ± 0.015
HandOutlines	**0.945 ± 0.000**	0.923 ± 0.000	0.934 ± 0.000	0.830 ± 0.000	0.913 ± 0.021
Herring	0.643 ± 0.000	**0.683 ± 0.000**	0.622 ± 0.000	0.483 ± 0.000	0.532 ± 0.008
HouseTwenty	**0.975 ± 0.000**	0.954 ± 0.000	0.959 ± 0.000	0.083 ± 0.000	0.942 ± 0.002
ItalyPowerDemand	0.990 ± 0.000	**0.990 ± 0.000**	0.988 ± 0.000	0.988 ± 0.000	0.979 ± 0.008
Lightning2	0.845 ± 0.000	0.825 ± 0.000	0.799 ± 0.000	0.709 ± 0.000	**0.890 ± 0.025**
MiddlePhalanxOutlineCorrect	**0.921 ± 0.000**	0.793 ± 0.000	0.900 ± 0.000	0.640 ± 0.000	0.760 ± 0.038
MoteStrain	**0.967 ± 0.000**	0.034 ± 0.000	0.940 ± 0.000	0.086 ± 0.000	0.935 ± 0.004
PhalangesOutlinesCorrect	**0.905 ± 0.000**	0.725 ± 0.000	0.886 ± 0.000	0.655 ± 0.000	0.762 ± 0.014
PowerCons	0.964 ± 0.000	0.938 ± 0.000	0.962 ± 0.000	0.847 ± 0.000	**0.965 ± 0.009**
ProximalPhalanxOutlineCorrect	**0.929 ± 0.000**	0.879 ± 0.000	0.913 ± 0.000	0.878 ± 0.000	0.826 ± 0.041
SemgHandGenderCh2	0.869 ± 0.000	0.791 ± 0.000	0.869 ± 0.000	0.696 ± 0.000	**0.982 ± 0.002**
ShapeletSim	**0.962 ± 0.000**	0.090 ± 0.000	0.917 ± 0.000	0.160 ± 0.000	0.924 ± 0.067
SonyAIBORobotSurface1	0.978 ± 0.000	0.984 ± 0.000	**0.984 ± 0.000**	0.968 ± 0.000	0.949 ± 0.029
SonyAIBORobotSurface2	0.964 ± 0.000	0.947 ± 0.000	**0.965 ± 0.000**	0.948 ± 0.000	0.893 ± 0.028
Strawberry	0.985 ± 0.000	0.931 ± 0.000	**0.986 ± 0.000**	0.919 ± 0.000	0.984 ± 0.004
ToeSegmentation1	0.981 ± 0.000	0.036 ± 0.000	**0.988 ± 0.000**	0.088 ± 0.000	0.846 ± 0.007
ToeSegmentation2	**0.946 ± 0.000**	0.086 ± 0.000	0.896 ± 0.000	0.203 ± 0.000	0.844 ± 0.029
TwoLeadECG	0.914 ± 0.000	0.269 ± 0.000	**0.988 ± 0.000**	0.119 ± 0.000	0.845 ± 0.056
Wafer	**0.999 ± 0.000**	0.991 ± 0.000	0.995 ± 0.000	0.912 ± 0.000	0.995 ± 0.001
Wine	**0.931 ± 0.000**	0.500 ± 0.000	0.770 ± 0.000	0.500 ± 0.000	0.681 ± 0.084
WormsTwoClass	0.811 ± 0.000	0.805 ± 0.000	**0.895 ± 0.000**	0.804 ± 0.000	0.846 ± 0.017
Yoga	**0.923 ± 0.000**	0.689 ± 0.000	0.890 ± 0.000	0.629 ± 0.000	0.853 ± 0.003