FROM FOUNDATION MODEL TO TINY TIME SERIES CLASSIFIER THROUGH KNOWLEDGE DISTILLATION

Adilson Medronha, Diego Furtado Silva
Department of Computer Science
University of São Paulo, Brazil
{adilson.medronha,diegofsilva}@usp.br

ABSTRACT

Time series foundation models provide strong generalization but remain computationally expensive for deployment in resource-constrained settings. We introduce TinyCN[1], a compact convolutional model trained via knowledge distillation from a transformer-based foundation model (Mantis-8M). Our training procedure transitions from representation alignment to task-specific optimization, enabling effective transfer of foundation representations into a lightweight CNN. Across all 128 UCR datasets, TinyCN achieves statistically significant improvements over Hybrid InceptionTime (HIT), the ensemble state-of-the-art, while being over $40\times$ smaller than Mantis and $10\times$ smaller than HIT. These results demonstrate that foundation representations can be effectively compressed into simple CNNs architectures, achieving superior accuracy and efficiency for time series classification.

Track: Research

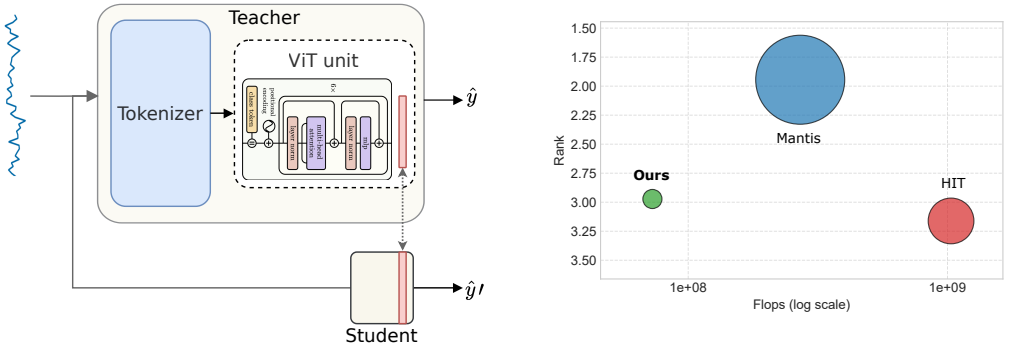
1 INTRODUCTION

Foundation models (FMs) improve classification by transferring representations learned from large-scale pretraining to downstream tasks (Bommasani et al., 2022). These models show strong generalization across tasks and data distributions (Brown et al., 2020). Most FM-based approaches are costly to deploy due to their memory footprint and attention-based inference, which limits their use in resource-constrained settings (Yang et al., 2025). Recent work extends FMs to time series classification using Transformer-based architectures pretrained on large signal collections (Liu et al., 2026; Liang et al., 2024). Among these, Mantis (Feofanov et al., 2025a) improves efficiency while achieving competitive performance with a smaller model size than Chronos (Ansari et al., 2024) and MOMENT (Goswami et al., 2024). In parallel, supervised time series classifiers (TSC) remain competitive, with ensemble convolutional models such as Hybrid InceptionTime (HIT) (Ismail-Fawaz et al., 2022) reaching state-of-the-art accuracy on standard benchmarks (Middlehurst et al., 2024).

However, a common limitation persists across both paradigms. High classification accuracy is typically achieved at the cost of substantial computational and memory requirements (Middlehurst et al., 2024). While foundation models rely on expensive attention mechanisms, supervised ensembles require concurrent inference across multiple deep networks, making both approaches impractical for deployment in scenarios with strict hardware or latency constraints.

To address this challenge, we propose an efficient distillation-based approach that leverages the generalization capabilities of foundation models while enabling lightweight inference. We introduce **TinyCN**, a compact convolutional neural network trained via knowledge distillation from a foundation model. Our adaptive distillation strategy progressively transitions from representation alignment to task-specific optimization, allowing the student to inherit high-level representations from the teacher while remaining computationally efficient. We evaluate our approach on all 128 datasets from the UCR archive, a standard benchmark for time series classification, where TinyCN achieves a favorable accuracy and efficiency trade-off, outperforming strong supervised baselines with substantially fewer parameters and floating-point operations, as shown in Figure 1b.

[1] https://github.com/adilsonmedronha/TinyCN



(a) Pipeline

(b) FLOPs vs Rank

Figure 1: (a) Feature-based distillation from Mantis foundation model (teacher) to a lightweight CNN (student). (b) Computational cost–performance trade-off: FLOPs against average rank on the 128 UCR datasets (lower is better), where bubble size is proportional to the number of parameters.

2 Proposed Method

Figure 1a illustrates the overall architecture of our proposed framework. The central procedure of our approach is knowledge distillation from a pre-trained time series foundation model (Mantis) into a compact convolutional student network (TinyCN). This procedure is performed through a teacher-student process using a composed (i) dynamic loss function designed to simultaneously align the student's representation space with the teacher's, and (ii) optimize the network for the specific downstream classification task. The source code will be made publicly available for reproducibility.

2.1 Teacher's Architecture

We adopt Mantis-8M as the teacher model, a Vision Transformer-based foundation model for time series (Feofanov et al., 2025b). Each time series is z-normalized, resampled to a fixed length of 512, tokenized, and processed by self-attention layers. In our experiments, Mantis is fine-tuned for 100 epochs for each dataset, using the same training configuration reported in the original paper.

We use the last-layer embedding as the teacher representation $z_t \in \mathbb{R}^{256}$, as deeper transformer layers capture increasingly task-specific semantic information (Dosovitskiy et al., 2021), aligning with the objective of transferring high-level representation in knowledge distillation (Hinton et al., 2015). In addition, Mantis-8M contains approximately 8 million parameters, which may limit its deployment in resource-constrained environments, motivating the use of a distilled student model.

2.2 Student's Architecture

The student model developed in this work, named TinyCN, is a lightweight network loosely inspired by the Fully Convolutional Network (FCN) (Wang et al., 2017). The network comprises three sequential convolutional blocks containing 64, 128, and 256 kernels, with sizes 8, 5, and 3, respectively. All the convolutional layers apply Batch Normalization and use a ReLU activation function. The dimensionality of the final layer ($d = 256$) is chosen to match the Mantis embedding dimension. Finally, Global Average Pooling (GAP) produces $z_s \in \mathbb{R}^{256}$, followed by a linear head.

It is important to note that while our architecture shares structural similarities with the standard FCN, we do not include the original architecture in our baseline comparisons. Recent comprehensive studies have demonstrated that HIT significantly outperforms the standard FCN (Middlehurst et al., 2024). Consequently, to ensure a rigorous evaluation against the strongest available benchmarks, we compare our proposal directly against the current state-of-the-art supervised model (HIT).

2.3 DATA PRE-PROCESSING

To ensure full compatibility with the teacher model's representation space, the input time series undergoes the same preprocessing protocol as Mantis. This pipeline consists of two essential stages: resizing and normalization. As Mantis is built on ViT architecture, it uses fixed positional embeddings that require a specific input resolution, which the authors set to 512. Consequently, consistent with the foundational model's pre-training configuration, all input series are interpolated to a fixed length of 512 observations. Also, each time series is independently normalized using z-scores, i.e., to have a mean of zero and unit variance, thereby addressing distribution shifts across samples.

2.4 ADAPTIVE LOSS FUNCTION

One of the central procedures of our proposal is the adaptive weighting of the objective function. The total loss $\mathcal{L}_{total}$ is defined as a linear combination of the distillation loss ($\mathcal{L}_{dist}$) and the task-specific loss ($\mathcal{L}_{task}$), controlled by a dynamic coefficient λ:

$$\mathcal{L}_{total} = \lambda \cdot \mathcal{L}_{dist}(z_s, z_t) + (1 - \lambda) \cdot \mathcal{L}_{task}(\hat{y}, y)$$

where $\mathcal{L}_{dist}$ is the Mean Squared Error (MSE) between the student's latent vector z_s and the teacher's embedding z_t, and $\mathcal{L}_{task}$ is the Cross-Entropy loss calculated between the predicted logits $\hat{y}$ and the ground truth labels y. The first component enforces the student to mimic the rich, generalizable representation space of the foundation model. The second one fits the head and fine-tunes the convolutional layers for the classification task.

To maximize performance, we implement a dynamic adaptive distillation schedule over 100 training epochs. The hyperparameter λ evolves as follows. In the first 50 epochs, we employ a cosine annealing schedule for λ, decaying from 1 to 0. In this phase, the network prioritizes learning the foundation model's structural representations, gradually shifting its focus toward the specific task. For the last 50 epochs, we fix $\lambda = 0$. In this final phase, the training procedure behaves as if it is undergoing fine-tuning, optimizing its weights solely to minimize classification error.

3 EXPERIMENTAL EVALUATION

3.1 EXPERIMENTAL SETUP

To validate the effectiveness of our proposal, we conducted an extensive experimental analysis using the UCR Time Series Archive (Dau et al., 2019). This repository is the standard benchmark for univariate time series classification, comprising 128 datasets from diverse application domains, including healthcare, finance, sensor data, and image outline classification. We applied the official train/test splits to ensure standardized, reproducible evaluation. We notice that the pretrained Mantis model does not use any data from the UCR test splits.

For each dataset, we fine-tune Mantis on the corresponding training split. The fine-tuned teacher is then frozen and used as a fixed feature extractor to generate target embeddings for distillation. The test split is never used during either fine-tuning or distillation.

For a rigorous comparative analysis, we evaluated TinyCN against a set of strong baselines representing different paradigms in the literature. Mantis is kept on the experiments, serving as the upper bound for performance in our distillation setup. We also added HIT to verify whether our distilled model can surpass the best models trained from scratch.

We added two additional methods to evaluate our proposal further. ROCKET (RandOm Convolutional KErnel Transform) (Dempster et al., 2020), a leading non-deep learning approach, is widely recognized for its high accuracy and computational speed, transforming time series using random convolutional kernels before applying a linear classifier. Also, to isolate the contribution of the convolutional inductive bias in our proposed architecture, we trained a Multi-Layer Perceptron (MLP) using the same distillation protocol as TinyCN. This MLP consists of two fully connected hidden layers, serving as an architectural baseline to determine if the performance gains stem solely from the distillation method or also from the specific design of the student network.

3.2 Results and Discussion

To summarize performance across 128 datasets, we report a Critical Difference diagram (Figure 2).

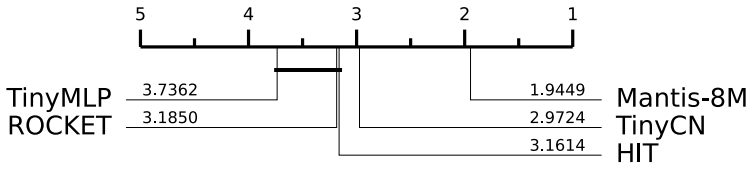


Figure 2: Critical Difference diagram computed using the Wilcoxon post-hoc test on UCR. Lower ranks indicate better performance; models horizontally line-connected are not statistically different.

The results support our main hypothesis. As expected, the teacher model (Mantis) achieves the best average rank, demonstrating the robustness of the FM. However, the most relevant result is TinyCN's positioning. Despite its substantially smaller size and lower computational cost, as shown in Table 1, TinyCN achieves a statistically significant improvement over HIT, the current ensemble supervised state-of-the-art, and also outperforms ROCKET, a random kernel-based approach. TinyCN Raw is the same model but without distillation. These findings indicate that the proposed adaptive distillation effectively transfers the knowledge from a FM to a compact student network.

Table 1: Model complexity comparison reporting model size (MB), number of parameters and floating point operations per forward pass (both in millions), computed for a fixed 512 input length.

Model	Params (M)	FLOPs (M)	Size (MB)
Mantis-8M	8.10	270.4	31
HIT	2.03	1032.3	8
FCN	0.27	135.8	1.2
TinyCN (ours)	**0.23**	**72.8**	**0.8**

Finally, the comparison with TinyMLP offers critical insight into the architectural requirements for this task. While TinyMLP followed the same training pipeline, its performance was significantly inferior to all other methods. This highlights that the distillation process alone does not represent a definitive solution. The convolutional inductive bias present in TinyCN is essential for effectively capturing the local temporal dependencies encoded in the teacher's representation. The TinyCN architecture, therefore, represents an optimal balance: it is complex enough to retain the teacher's knowledge but simple enough to maintain extreme computational efficiency.

4 Conclusion

This work demonstrates that representations from time series foundation models can be distilled into compact convolutional networks. Using Mantis as the teacher, **TinyCN** achieves competitive performance with only 0.23M parameters and 0.8 MB of memory, making it $40\times$ smaller than Mantis and $10\times$ smaller than HIT (the current state of the art). Despite requiring only 72.8M FLOPs per inference (compared to 1032.3M for HIT), TinyCN statistically outperforms it. These results suggest that distillation can produce efficient classifiers, offering a practical alternative to large models under computational constraints. Future work will extend this framework to other student architectures and foundation models, as well as to multivariate datasets and tasks such as regression and forecasting, while exploring improved trade-offs between distillation and task losses.

Acknowledgments

The authors would like to acknowledge the financial support by grants #2022/03176-1, #2024/09747-6, #2024/14856-9, #2025/18189-0 from São Paulo Research Foundation (FAPESP).

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Vasilii Feofanov, Marius Alonso, Songkang Wen, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. In *1st ICML Workshop on Foundation Models for Structured Data*, 2025a.

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, 2025b. URL https://arxiv.org/abs/2502.15637.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Ali Ismail-Fawaz, Maxime Devanne, Jonathan Weber, and Germain Forestier. Deep learning for time series classification using new hand-crafted convolution filters. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 972–981. IEEE, 2022.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6555–6565, 2024.

Zhen Liu, Yucheng Wang, Boyuan Li, Junhao Zheng, Emadeldeen Eldele, Min Wu, and Qianli Ma. A unified shape-aware foundation model for time series classification, 2026. URL https://arxiv.org/abs/2601.06429.

Matthew Middlehurst, Patrick Schäfer, and Anthony 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.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

Jianlei Yang, Jiacheng Liao, Fanding Lei, Meichen Liu, Lingkun Long, Junyi Chen, Han Wan, Bei Yu, and Weisheng Zhao. Tinyformer: Efficient transformer design and deployment on tiny devices, 2025. URL https://arxiv.org/abs/2311.01759.

A Appendix

TinyCN model was distilled from Mantis-8M. To isolate the effect of distillation, we compared our proposed model with its non-distilled counterpart (TinyCN Raw).

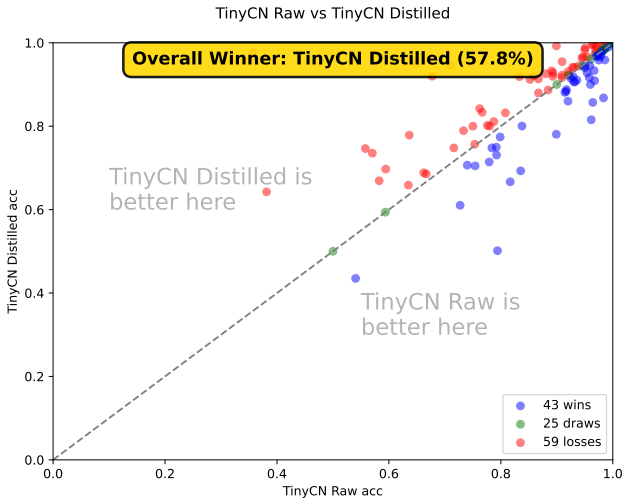


Figure 3: Win/Tie/Loss comparison of distilled and non-distilled (raw) TinyCN on UCR benchmark.