

Data Science in Factory: Multiscale Sensor Learning for Industrial Object Activity Recognition

Rui Feng
Wuhan University
Wuhan, China
fengr382@gmail.com

Yize Cai
HKUST (GZ)
Guangzhou, China
ycai186@connect.hkust-gz.edu.cn

Zhiqing Hong*
HKUST (GZ)
Guangzhou, China
zhiqinghong@hkust-gz.edu.cn

Abstract

Recognizing pallet activities from onboard sensors can improve warehouse process visibility without relying on camera infrastructure. However, high-frequency industrial signals are noisy, strongly overlapped, and difficult to generalize across experiments. We present a multiscale sensor-learning framework that converts each two-second, seven-channel sensor window into a compact tabular representation and applies pretrained TabICL for multi-label prediction. Following the Pal2Sim benchmark, we predict six major operational classes using the activity annotation at the final timestamp of each window. The multiscale representation preserves complementary motion information at 2,000, 400, and 100 Hz while reducing the difficulty of modeling long raw sequences. We evaluate the method on four real-world warehouse experiments using leave-one-experiment-out validation. The proposed framework achieves 90.61% exact-match accuracy, 83.04% macro-F1, and 86.88% MCC, outperforming all compared baselines. These results show that multiscale feature transformation and pretrained tabular learning provide a simple and effective approach to industrial object activity recognition.

Keywords

industrial object activity recognition, sensor data mining, multiscale sensor learning, tabular foundation models

1 Introduction

Pallets are involved in transportation, storage, lifting, and wrapping throughout warehouse operations. Automatically recognizing these activities can provide fine-grained process visibility and support scheduling, bottleneck analysis, and abnormal-operation detection [6]. Object-mounted sensors are attractive for this purpose because they can continuously record pallet motion without requiring extensive camera infrastructure or collecting visual information.

Most sensor-based activity recognition research focuses on human activities measured by wearable devices or smartphones [2, 12]. Industrial object activity recognition is less explored and differs from conventional HAR in both its sensing conditions and its evaluation setting. In the Pal2Sim task, pallet-mounted accelerometers, gyroscopes, and a barometric sensor are sampled at 2,000 Hz across four independent warehouse experiments. The resulting signals contain strong vibration and noise, while the limited number of experiments makes cross-experiment generalization difficult.

We follow the benchmark formulation and predict six major operational classes rather than the ten original fine-grained annotations. The six classes are the official evaluation targets and correspond to the principal warehouse states: *Driving (straight)*, *Driving (curve)*, *Lifting (raising)*, *Lifting (lowering)*, *Stationary processes*, and *Turntable wrapping*. Several minor labels describe brief preparatory, docking, or fork-interaction phases and are grouped into *Stationary processes*, avoiding an unnecessarily fragmented target space. Because driving and lifting can occur at the same time, the task remains multi-label.

Each sample contains the two seconds of sensor history preceding a prediction endpoint. Its target is the activity annotation at the final timestamp, rather than the majority activity within the window. This causal formulation is suitable for online monitoring: the input contains only recent measurements, while the target describes what the pallet is doing at the current time. Majority voting could retain the previous activity after a transition has already occurred. The two-second duration follows the challenge setup and provides sufficient context for short motion patterns while allowing frequent state updates; we do not claim that it is universally optimal.

These task choices create three practical challenges. First, each sample contains a 4000×7 high-frequency sequence, making direct raw-sequence modeling expensive and sensitive to noise. Second, neighboring windows strongly overlap, so the large number of samples does not represent equally diverse training evidence. Third, endpoint labeling, class imbalance, simultaneous activities, and experiment-level distribution shifts make transition and lifting cases difficult to recognize reliably.

To address these challenges, we propose a multiscale time-series-to-table framework. Each window is represented at 2,000, 400, and 100 Hz and summarized using compact statistical, temporal, and frequency-domain descriptors. We then decompose the six-label task into label-wise binary problems and adapt pretrained TabICL to the structured features. This design retains complementary motion information while avoiding the need to learn every temporal pattern directly from long raw sequences.

The main contributions of this work are as follows:

- We formulate high-frequency pallet activity recognition as a multiscale time-series-to-table learning problem and clarify the benchmark’s major-label, endpoint-label, and multi-label settings.
- We develop a simple framework that combines three-scale sensor descriptors with pretrained TabICL for label-wise activity prediction.
- We evaluate the framework on four real-world warehouse experiments using leave-one-experiment-out validation, where

*Corresponding author.

it achieves the best average exact-match accuracy, macro-F1, and MCC among the compared methods.

2 Background

2.1 Task Characteristics

This work studies object-centered activity recognition from high-frequency sensors mounted on a logistics pallet. Unlike conventional human activity recognition, the recorded signals describe pallet and forklift operations, including vehicle motion, lifting, contact events, and wrapping processes. The task is challenging because the 2,000 Hz signals contain vibration and noise, neighboring windows are strongly overlapped, and the signal distribution changes across experiments.

2.2 Label Construction

The original annotations contain ten fine-grained (minor) labels, whereas the benchmark evaluates six major operational classes: *Driving (straight)*, *Driving (curve)*, *Lifting (raising)*, *Lifting (lowering)*, *Stationary processes*, and *Turntable wrapping*. We use the major classes for three reasons. First, they follow the official task definition and make our results directly comparable with the challenge evaluation. Second, several minor labels describe brief auxiliary phases of the same broader stationary process, including docking, wrapping preparation, and fork entry or exit. Treating them as independent targets would fragment the available data and produce less stable cross-experiment estimates. Third, the major classes provide operational states that are easier to interpret for warehouse monitoring. The complete mapping is reported in Appendix 2.

Driving and lifting may be active at the same timestamp. The target is therefore a six-dimensional binary vector rather than one mutually exclusive class, and the data contain both single-label states and four valid driving–lifting combinations.

2.3 Window and Endpoint Construction

For a prediction endpoint at timestamp t_i , we use the preceding two seconds of measurements as the input window and assign the activity annotation at t_i as its target. The final-timestamp label is consistent with online recognition: past sensor measurements provide context for estimating the pallet’s current state, and no future observations are used. In contrast, a majority label over the full window can delay a state change when the earlier activity occupies most of a transition window.

The two-second duration follows the Pal2Sim setup [1]. It is long enough to include a short motion pattern rather than an isolated vibration, while remaining short enough for responsive monitoring. It is a fixed benchmark choice used consistently by all compared methods, rather than a universally optimal duration. At 2,000 Hz, each window contains 4,000 points per channel.

Window endpoints advance by 250 points, corresponding to 125 ms between consecutive predictions and 93.75% overlap. This retains dense temporal coverage while substantially reducing the extreme duplication of a one-point stride. Windows whose endpoint has an all-zero major-label vector are removed because they do not correspond to any target activity in the benchmark; they are not treated as an additional background class.

2.4 Motivation for Multiscale Features

Different activities can be reflected at different temporal scales. High-frequency signals preserve rapid contacts and changes in lifting dynamics, intermediate-scale signals describe local motion structure with reduced noise, and low-frequency signals emphasize smoother vibration and pressure trends. A representation constructed at only one scale may therefore miss useful evidence.

We use a multiscale time-series-to-table representation that summarizes each window at 2,000, 400, and 100 Hz. This produces a compact feature vector that exposes statistical, temporal, and frequency information directly to the classifier and makes pretrained tabular learning applicable to the sensor task.

2.5 Problem Definition

Let each two-second sensor window be denoted as $\mathbf{X}_i \in \mathbb{R}^{T \times C}$, where $T = 4000$ and $C = 7$. Each sample has a six-dimensional major-label vector $\mathbf{y}_i \in \{0, 1\}^6$ corresponding to *Driving (straight)*, *Driving (curve)*, *Lifting (raising)*, *Lifting (lowering)*, *Stationary processes*, and *Turntable wrapping*.

We transform each raw window into a multiscale tabular feature vector $\phi(\mathbf{X}_i)$ and decompose the prediction into six one-vs-rest binary classification problems. For label k , the corresponding classifier predicts $\hat{y}_i^{(k)} = f_k(\phi(\mathbf{X}_i))$, where $\hat{y}_i^{(k)} \in \{0, 1\}$. The six outputs are concatenated as $\hat{\mathbf{y}}_i = [\hat{y}_i^{(1)}, \dots, \hat{y}_i^{(6)}] \in \{0, 1\}^6$.

3 System Design

This section presents the proposed multiscale time-series-to-table framework for pallet activity recognition. As shown in Fig. 1, the framework first constructs endpoint-labeled sensor windows, then extracts multiscale features, and finally applies pretrained tabular learning for multi-label prediction.

3.1 Data Preparation and Task Formulation

Each experiment contains continuous signals collected from pallet-based warehouse operations and timestamp-level activity annotations. We use seven channels,

$$\mathbf{x}_t = [a_x, a_y, a_z, g_x, g_y, g_z, b_z],$$

where (a_x, a_y, a_z) are accelerometer signals, (g_x, g_y, g_z) are gyroscope signals, and b_z is the barometric signal. The original sampling rate is 2,000 Hz.

For a prediction endpoint t_i , we take the preceding two seconds of measurements to form a 4000×7 window. The window inherits the major-class annotation at t_i , ensuring that the target describes the current activity using only past sensor context. Endpoints are separated by 250 measurements, corresponding to 125 ms, and windows with an all-zero endpoint target are removed.

The original annotations contain ten fine-grained classes: *Standing*, *Driving (curve)*, *Driving (straight)*, *Wrapping*, *Lifting (lowering)*, *Forks (entering or leaving side)*, *Lifting (raising)*, *Forks (entering or leaving front)*, *Wrapping (preparation)*, and *Docking*. Following the benchmark, these annotations are mapped to six major classes: *Driving (curve)*, *Driving (straight)*, *Lifting (raising)*, *Lifting (lowering)*, *Stationary processes*, and *Turntable wrapping*. The mapping preserves the principal operational states while combining brief

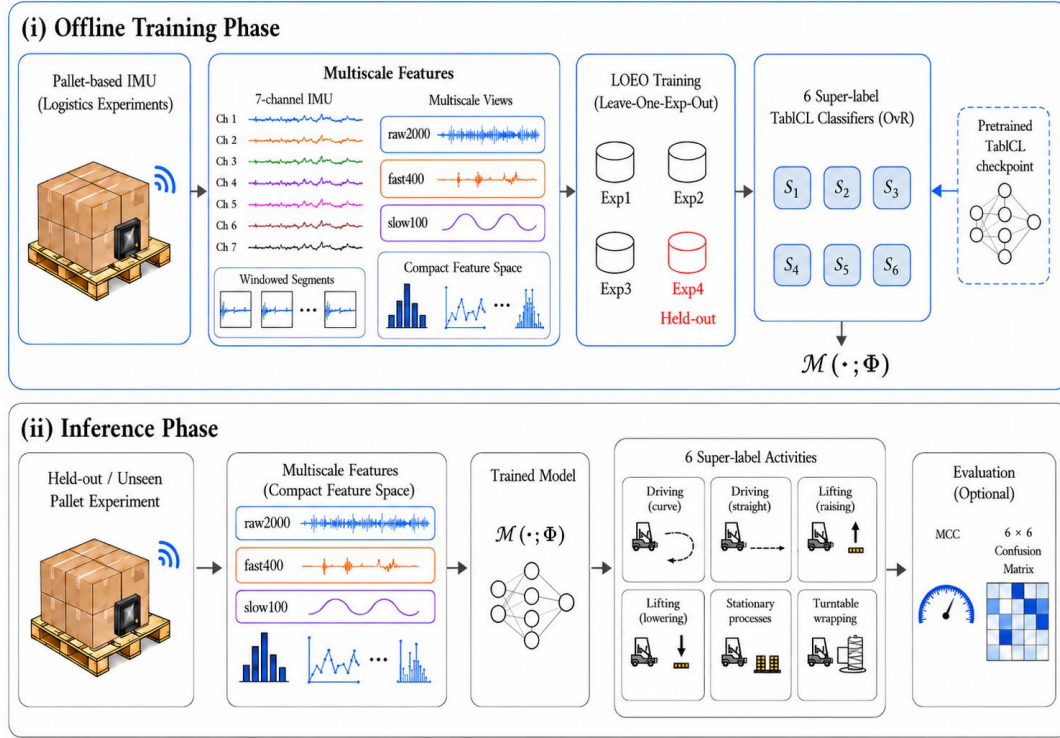


Figure 1: Overview of the proposed framework. Continuous pallet sensor streams are converted into endpoint-labeled windows, summarized as multiscale tabular features, and classified using pretrained TabICL.

auxiliary stationary phases into a more stable target. Appendix 2 lists the complete mapping.

The task is multi-label because driving and lifting can occur simultaneously at the prediction endpoint. Four double-label combinations appear in the data: *Driving (curve) + Lifting (lowering)*, *Driving (curve) + Lifting (raising)*, *Driving (straight) + Lifting (lowering)*, and *Driving (straight) + Lifting (raising)*.

3.2 Multiscale Feature Extraction

Instead of directly applying a sequence model to a 4000×7 raw window, we convert each sample into a compact tabular representation. Each window is represented at three temporal scales: the original 2,000 Hz scale, an intermediate 400 Hz scale, and a low-frequency 100 Hz scale.

The three views capture complementary information. The original view preserves rapid contacts and sensor changes. The 400 Hz view describes local motion with reduced noise, while the 100 Hz view emphasizes smoother motion and barometric trends. We also construct derived signals, including acceleration magnitude, gyroscope magnitude, jerk, angular acceleration, and barometric differences.

The extracted attributes include time-domain statistics, temporal transition descriptors, autocorrelation features, and frequency-domain energy features. Each sensor window is consequently represented as one row in a feature table, with each column corresponding to one multiscale descriptor.

3.3 Pretrained Tabular Learning

After feature extraction, the original sequence-recognition problem becomes a tabular classification problem. Given a training set

$$\mathcal{D}_{\text{train}} = \{(X_i, y_i)\}_{i=1}^N,$$

X_i denotes the multiscale feature vector of the i -th window and y_i denotes its six-dimensional target.

We use TabICL as the classifier. The multiscale features explicitly summarize temporal evidence, allowing TabICL to focus on relationships among structured numerical attributes rather than learning directly from long raw sequences.

The six-dimensional target is decomposed into six binary classification tasks. An independent TabICL classifier predicts the presence of each major activity, and the six outputs are combined to form the final multi-label vector.

4 Evaluation

4.1 Experimental Setup

4.1.1 Dataset and Protocol. We evaluate all methods on the Pal2Sim challenge data [1]. The dataset contains four independent warehouse experiments recorded with seven sensor channels at 2,000 Hz. Signals are segmented into two-second windows of 4,000 points, with neighboring endpoints separated by 250 points (125 ms). Each window inherits the major-class annotation at its final timestamp, and windows with an all-zero target vector are removed, yielding approximately 15.1K labeled samples.

Following the benchmark definition, the ten fine-grained annotations are mapped to six major operational classes. This mapping combines sparse preparatory and stationary sub-phases while retaining the main driving, lifting, stationary, and wrapping states. Since driving and lifting may occur simultaneously, every sample is represented by a six-dimensional binary label vector.

We adopt leave-one-experiment-out evaluation. In each fold, three complete experiments are used for model development and the remaining experiment is used for testing. This protocol measures whether the model can generalize to a warehouse recording that is not observed during training.

4.1.2 Preprocessing. The two-second window length follows the challenge setup and is used by all methods. It provides recent motion context while keeping the target tied to the current endpoint activity. The 250-point stride corresponds to a 125 ms interval between predictions and 93.75% overlap between adjacent windows. This preserves transition coverage while substantially reducing redundant samples compared with a one-point stride.

4.1.3 Baselines. We compare our framework with two categories of representative methods:

- **Machine-learning methods:** XGBoost [3], LightGBM [13], and CatBoost [14] are gradient-boosted tree models trained on structured sensor features. MiniRocket [4] is a fast time-series transformation method based on convolutional kernels.
- **Deep-learning methods:** InceptionTime [11] uses multiscale convolutional branches, TimesNet [17] models temporal variation through two-dimensional representations, and GenHAR [9] is designed for generalizable sensor-based activity recognition.

4.1.4 Metrics. We use exact-match accuracy (ACC), macro-F1, and Matthews correlation coefficient (MCC). Exact-match ACC requires the complete six-dimensional label vector to be correct. Macro-F1 and MCC are computed label-wise and then averaged across the six major activities. MCC is treated as the primary metric because it remains informative when positive and negative samples are strongly imbalanced.

4.1.5 Implementation Details. Experiments run on a 32 GB NVIDIA RTX 5090 GPU. We use Python 3.11 and PyTorch 2.9.1, with pretrained TabICL [15] as the backbone.

4.2 Main Results

Table 1 summarizes average performance over the four folds. Our method reaches 90.61% ACC, 83.04% macro-F1, and 86.88% MCC, ranking first on all three metrics. Compared with MiniRocket, the gains are 3.47 points in ACC, 6.02 in macro-F1, and 4.90 in MCC. These results support the combination of multiscale sensor features and pretrained tabular learning.

The tree-based baselines are also competitive, with average MCC scores between 80.52% and 80.81%. MiniRocket performs best among the non-pretrained baselines, confirming the value of transforming high-frequency windows into discriminative features. However, these methods remain below our approach on all three metrics.

The deep sequence baselines are less stable under leave-one-experiment-out evaluation. Although InceptionTime and TimesNet outperform GenHAR on average, they show a clear performance drop on Fold 4. This suggests that directly modeling raw sequences is more sensitive to experiment-level distribution changes in this setting.

4.3 Confusion Matrix Analysis

Figure 2 provides a more detailed view of the prediction behavior. In the six-major-class confusion matrix, the proposed method performs particularly well on *Stationary processes* and *Turntable wrapping*, suggesting that these states have relatively distinctive sensor patterns. The main confusion appears among driving and lifting activities, which can occur in similar temporal regions and can also be active simultaneously.

The fine-grained and combined-label confusion matrix further shows that double-label cases are more challenging than single-label cases. Samples containing both driving and lifting are sometimes predicted as only one component activity. This observation supports the multi-label formulation rather than forcing each window into one exclusive class.

4.4 Lessons Learned

The results suggest that high-frequency industrial IMU recognition does not necessarily require a large end-to-end sequence model. In the Pal2Sim setting, the data are sampled at 2,000 Hz and segmented into overlapping two-second windows, producing many strongly related samples. Under this condition, directly applying sequence models can be computationally expensive and may not fully exploit the structure of the data.

A key lesson is that explicit temporal abstraction is effective. By converting raw signals into multiscale representations, the proposed method reduces the complexity of dense sensor streams while retaining important motion information. The 2,000 Hz scale preserves rapid sensor changes, the 400 Hz scale captures intermediate motion patterns, and the 100 Hz scale provides a smoother view of low-frequency trends.

Another lesson is that tabular learning can be useful for sensor-based activity recognition when the representation is properly designed. The time-series-to-table transformation allows pretrained TabICL to operate on compact and discriminative features instead of dense raw sequences. This shows that the main advantage comes from combining suitable multiscale feature construction with a strong pretrained tabular classifier.

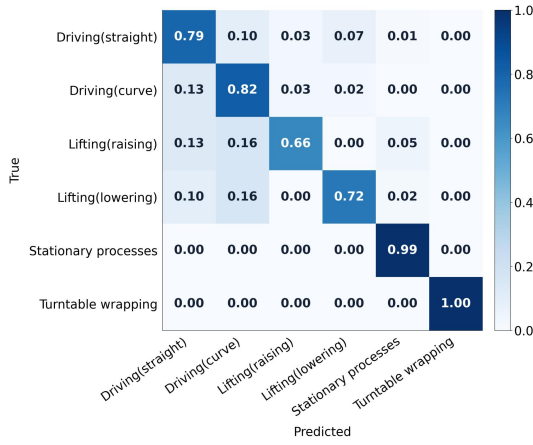
4.5 Limitations and Future Work

Although the proposed method achieves strong performance, several limitations remain. First, the current framework relies on hand-crafted multiscale features. While these features are compact and interpretable, their effectiveness may depend on the selected statistics, sensor channels, and frequency scales. Future work could explore automatic feature selection or learnable feature extraction to reduce this manual design effort.

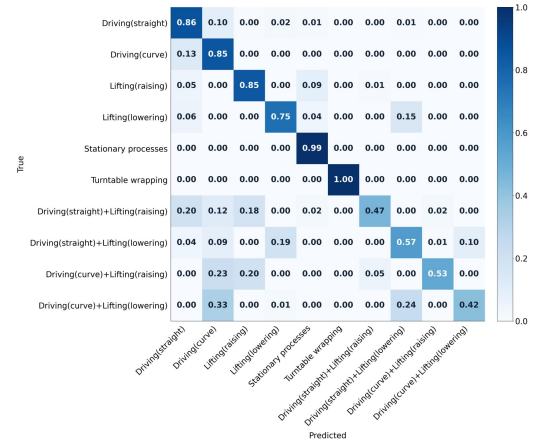
Second, the method is evaluated on one industrial forklift activity recognition dataset under the leave-one-experiment-out protocol. This setting provides a strict test of cross-experiment generalization,

Table 1: Overall results under the four leave-one-experiment-out folds. Each metric cell reports MCC / macro-F1 (%) / ACC (%). The bold represents the best results.

Group	Method	Fold 1	Fold 2	Fold 3	Fold 4	Average
ML	XGBoost	81.48 / 74.80 / 87.32	78.29 / 71.23 / 83.89	83.28 / 76.59 / 87.86	79.04 / 71.66 / 85.34	80.52 / 73.57 / 86.10
	LightGBM	82.03 / 74.62 / 87.75	78.88 / 72.33 / 84.33	82.87 / 75.54 / 87.55	79.47 / 71.64 / 85.67	80.81 / 73.53 / 86.32
	CatBoost	83.19 / 77.63 / 88.58	79.43 / 73.65 / 84.77	82.55 / 75.41 / 87.33	77.97 / 71.18 / 84.62	80.79 / 74.46 / 86.32
	MiniRocket	82.88 / 76.17 / 88.38	80.86 / 76.31 / 85.81	82.92 / 79.07 / 87.55	81.24 / 76.55 / 86.80	81.98 / 77.02 / 87.14
DL	GenHAR	71.28 / 66.94 / 80.21	69.88 / 65.14 / 77.42	74.22 / 70.23 / 81.17	67.86 / 63.61 / 77.16	70.81 / 66.48 / 78.99
	InceptionTime	79.43 / 75.40 / 85.74	79.75 / 75.23 / 84.99	81.08 / 77.59 / 86.14	58.19 / 50.12 / 69.22	74.61 / 69.59 / 81.52
	TimesNet	81.48 / 75.52 / 87.16	80.20 / 72.74 / 85.29	80.74 / 75.45 / 85.77	55.28 / 53.94 / 67.91	74.42 / 69.41 / 81.53
Ours	-	88.11 / 83.92 / 91.88	84.89 / 80.20 / 88.79	87.20 / 83.55 / 90.71	87.33 / 84.50 / 91.06	86.88 / 83.04 / 90.61



(a) Six-major-class confusion matrix.



(b) Fine-grained and combined-label confusion matrix.

Figure 2: Normalized confusion matrices of the proposed method.

but it is still necessary to validate the approach on more industrial sensing datasets, different machines, and different operating environments.

Third, the current model uses label-wise binary classification for the six major-label activities. This design is simple and suitable for overlapping activities, but it does not explicitly model label dependencies. Future work could investigate structured multilabel learning methods that better capture co-occurrence relationships among forklift activities.

Finally, although training-window subsampling reduces redundancy and improves efficiency, the choice of sampling stride is still empirical. Future studies could examine adaptive sampling strategies that preserve informative transition windows while further reducing redundant samples. Hybrid models that combine multi-scale tabular features with lightweight temporal neural modules may also be a promising direction, combining the efficiency of tabular learning with the representation capacity of sequence models.

5 Related Work

5.1 IMU-based Activity Recognition

Sensor-based activity recognition has been widely studied in mobile computing, wearable sensing, and cyber-physical systems. Most existing work focuses on human activity recognition (HAR), where accelerometer and gyroscope signals collected from smartphones or wearable devices are used to infer human actions. Early HAR methods usually rely on handcrafted statistical or frequency-domain features followed by conventional classifiers, while recent studies have increasingly adopted deep neural networks to learn temporal representations directly from raw sensor streams [2, 16]. Beyond standard closed-set recognition, recent HAR research has also investigated cross-subject, cross-device, and cross-dataset generalization. For example, CrossHAR studies hierarchical self-supervised pre-training for generalizable HAR across different datasets [10].

Compared with HAR, IMU-based activity recognition for industrial objects remains much less explored. In logistics environments, the target is not the human body but a load carrier, pallet, forklift, or other physical object whose motion reflects the state of an industrial process. Recent work on smart pallets has shown the

feasibility of detecting pallet-related logistics events using object-mounted IMUs [6]. The SPARL dataset further provides multimodal sensor recordings for pallet activity recognition in logistics [7]. The Pal2Sim project extends this direction by focusing on sensor-based activity recognition for load carriers in logistics supply chains [1]. These studies indicate that object-mounted IMUs are promising for industrial process monitoring, but this setting is still far less mature than conventional HAR in terms of datasets, model diversity, and benchmarked methods.

5.2 Time-Series Classification and Feature Transformation

Time-series classification can be addressed either by end-to-end neural models or by transforming raw sequences into discriminative feature representations. Deep architectures such as InceptionTime use multiscale convolutional modules to learn temporal patterns and have achieved strong performance on time-series classification benchmarks [11]. TimesNet further models temporal variations by transforming one-dimensional time series into two-dimensional representations and has shown strong performance across multiple time-series analysis tasks [17]. In HAR, GenHAR is designed to improve cross-domain generalization for human activity recognition [9]. These models provide strong deep-learning baselines for sequence modeling.

Direct sequence modeling is costly for high-frequency industrial signals, particularly when adjacent windows overlap heavily. In Pal2Sim, two-second windows are generated every 125 ms. The resulting redundancy increases training cost without providing equally diverse evidence. Our experiments also show that InceptionTime, TimesNet, and GenHAR do not outperform the proposed time-series-to-table framework, indicating that strong sequence architectures do not automatically transfer to this object-centered setting.

Feature-transformation methods provide an alternative path. Instead of feeding the entire raw sequence into a neural model, they summarize each time-series segment into a structured representation and then apply a classifier on top of the extracted features. MiniRocket is a representative transformation-based method that uses convolutional kernels to generate features for fast and accurate time-series classification [4]. Our method follows this feature-transformation perspective but extends it with a task-specific multiscale design. Each IMU window is represented at the original 2,000 Hz scale and two downsampled scales, allowing the final tabular features to capture high-frequency lifting dynamics, mid-scale motion patterns, and low-frequency vibration trends.

5.3 Tabular Learning and AutoML

Tabular data remains a central data format in practical machine learning. Gradient-boosted decision trees are strong and widely used baselines for tabular prediction. XGBoost improves tree boosting with a scalable system design, sparsity-aware learning, weighted quantile sketch, and cache-aware optimization [3]. LightGBM further improves the efficiency of gradient boosting decision trees through Gradient-based One-Side Sampling and Exclusive Feature Bundling [13]. CatBoost addresses target leakage and prediction

shift in boosting, especially for categorical features, through ordered boosting and ordered target statistics [14]. These models demonstrate that carefully engineered tabular learners can be highly competitive when informative features are available.

AutoML systems further automate the construction of strong tabular predictors. AutoGluon-Tabular emphasizes robust preprocessing, diverse model training, multi-layer stacking, and ensemble construction rather than only searching for a single best model and hyperparameter configuration [5]. This design is relevant to our setting because the transformed IMU windows form structured tables, where robust tabular learning and ensembling can improve stability across experimental folds.

Recent studies also show that deep learning models for tabular data are becoming increasingly competitive. TabM advances tabular deep learning by using parameter-efficient ensembling, where one MLP-based model produces multiple predictions per object and behaves like an efficient ensemble [8]. TabICL further explores in-context learning for large-scale tabular classification and is designed to handle larger training sets with a scalable tabular foundation model [15]. Our work is complementary to these studies: rather than designing a new tabular model, we focus on converting high-frequency IMU time series into a multiscale tabular representation that can be effectively handled by pretrained tabular learners.

6 Conclusion

This paper presents a multiscale time-series-to-table framework for IMU-based object activity recognition. By reducing redundancy through stride-based window subsampling, extracting compact multiscale sensor features, and adapting pretrained tabular knowledge in a label-wise binary classification setting, the proposed method achieves the best performance in the Pal2Sim challenge. These results show that multiscale feature transformation and pretrained tabular learning provide a practical and effective solution for high-frequency sensor-based activity recognition.

References

- [1] 2026. *Pal2Sim: IMU-Based Object Activity Recognition Competition*. <https://www.pal2sim.com/?tab=competition>
- [2] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–40.
- [3] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.
- [4] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. 2021. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 248–257.
- [5] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* (2020).
- [6] Sven Franke, Andrea Bommert, Marc Julian Brandt, Jean Lenard Kuhlmann, Marie-Claire Olivier, Kirsten Schorning, Christopher Reining, and Alice Kirchheim. 2024. Smart pallets: Towards event detection using imus. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 1–4.
- [7] Sven Franke, Andrea Bommert, Marc Julian Brandt, Jean Lenard Kuhlmann, Marie-Claire Olivier, Kirsten Schorning, Moritz Roidl, Christopher Reining, and Alice Kirchheim. 2024. Sensor-Based Pallet Activity Recognition in Logistics (SPARL Version 2): A Multi-Modal Dataset. Zenodo. doi:10.5281/zenodo.13318882
- [8] Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. 2025. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=Sd4wYYOhmY>

- [9] Zhiqing Hong, Zelong Li, Xiubin Fan, Guang Yang, Baoshen Guo, Haotian Wang, Tian He, and Desheng Zhang. 2026. GenHAR: Generalizing Cross-domain Human Activity Recognition for Last-mile Delivery. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 2208–2219.
- [10] Zhiqing Hong, Zelong Li, Shuxin Zhong, Wenjun Lyu, Haotian Wang, Yi Ding, Tian He, and Desheng Zhang. 2024. Crosshar: Generalizing cross-dataset human activity recognition via hierarchical self-supervised pretraining. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 2 (2024), 1–26.
- [11] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. Inceptiontime: Finding alexnet for time series classification. *Data mining and knowledge discovery* 34, 6 (2020), 1936–1962.
- [12] Charmi Jobanputra, Jatna Bavishi, and Nishant Doshi. 2019. Human activity recognition: A survey. *Procedia Computer Science* 155 (2019), 698–703.
- [13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
- [14] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems* 31 (2018).
- [15] Jingang QU, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2025. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *Forty-second International Conference on Machine Learning*. <https://openreview.net/forum?id=0VvD1PmNzM>
- [16] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters* 119 (2019), 3–11.
- [17] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=ju_Uqw384Oq

A Appendix

A.1 Fine-Grained to Major-Class Mapping

The Pal2Sim annotations contain ten fine-grained labels, while the evaluation uses six major operational classes. The mapping follows the benchmark definition. It keeps driving, lifting, and wrapping as distinct operational states and combines short docking, preparation, standing, and fork-interaction phases into *Stationary processes*. This reduces label fragmentation and provides a more stable target for cross-experiment evaluation.

Table 2: Mapping from fine-grained annotations to the six major classes.

Major Class	Fine-Grained Class(es)
Lifting (lowering)	Lifting (lowering)
Lifting (raising)	Lifting (raising)
Driving (curve)	Driving (curve)
Driving (straight)	Driving (straight)
Turntable wrapping	Wrapping
Stationary processes	Standing, Docking, Wrapping (preparation), Forks (entering or leaving front), Forks (entering or leaving side)

A.2 Double-Label Cases

Driving and lifting can be active at the same timestamp. The dataset contains four double-label combinations:

- Driving (curve) + Lifting (lowering)
- Driving (curve) + Lifting (raising)
- Driving (straight) + Lifting (lowering)
- Driving (straight) + Lifting (raising)

These cases motivate the multi-label formulation adopted in this paper.

A.3 Class-Duration Distribution

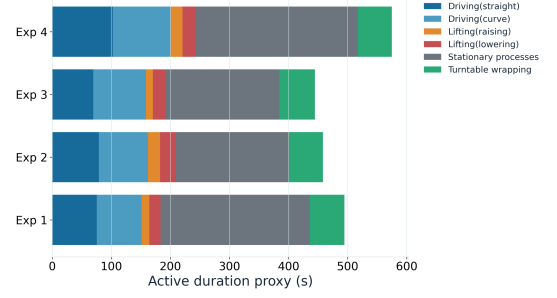


Figure 3: Class-duration distribution across the four experiments.