International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

Hybrid Machine Learning Framework for Personalized Spending Forecasting in E-Commerce Ecosystems Using the Digital Payment Method

Punam Prakash Patil[1], Dr. Netra Prashant Patil[2]
[1] Research Scholar, Savitribai Phule Pune University, Pune, India
[2]Research Guide, Savitribai Phule Pune University, Pune, India
Emails: *chaudhari.punam14@gmail.com[1], net.patil@gmail.com[2]*

Abstract
Online shoppers in India increasingly pay through the Unified Payments Interface (UPI), and payment apps hold detailed records of when and how people spend. Forecasting an individual's spending for the next month from such records can support budgeting tools, well-timed offers, and payment recommendations. This paper proposes a hybrid framework for personalized spending forecasting. It combines exponential smoothing, a gradient-boosted tree learner, and a neural network, where the last two model what smoothing leaves unexplained. A stacking layer then blends them using weights learned for each behavioral segment. A UPI context layer is specified for inputs such as transaction success, failure history, and reward intensity. The framework is evaluated on the public Online Retail II e-commerce log, covering 5,942 customers, using five forecast months and five random seeds. It lowers the scaled forecast error by about 25% compared with a seasonal naive benchmark (0.761 versus 1.011) and matches LightGBM, but simple exponential smoothing is slightly better overall (0.747). Learned components help most for customers with short or occasional purchase histories (0.50 versus 0.79 for new customers). Segment-specific blending gave no measurable gain, and nominal 80% prediction intervals covered only 62% of outcomes. UPI transaction data were not available, so the UPI layer is specified but not tested, and the paper makes no claim about UPI behavior.

Keywords: *spending forecasting; hybrid machine learning; personalization; UPI; e-commerce; gradient boosting; stacked generalization.*

1. Introduction
UPI has changed how many Indian consumers pay for goods and services. Each payment leaves a timestamped record with an amount, a counterparty, and an outcome. When shoppers pay for online orders this way, the payment app and the e-commerce platform together see a detailed picture of everyday spending. That picture can help consumers plan budgets, help platforms time offers and stock, and help payment apps suggest the most reliable way to pay. Forecasting spending for one person is harder than forecasting a total. Individual histories are short, many months contain no purchase at all, and a few large orders can dominate the totals. Festivals, salary dates, and sales add several seasonal patterns on top of each other. Statistical models handle trend and seasonality well but cannot use many extra inputs. Machine learning models can use those inputs, but they may struggle when a person has little history, and the evidence that one model family always wins is mixed (Makridakis et al., 2018; Grinsztajn et al., 2022). Combining models is a sensible response. Averaging forecasts has improved accuracy for decades (Bates & Granger, 1969), and the best entries in large forecasting competitions mixed statistical and learned parts (Smyl, 2020). Three gaps remain for payment and e-commerce data. Personalization is usually just another input column. Evaluation often lets future information leak into training, which makes results look better than they are (Cerqueira et al., 2020). And accuracy is rarely studied together with explanations that ordinary users could accept.
This paper makes four contributions:

1. A modular hybrid framework that separates a smoothing baseline, residual learning from tabular features, residual learning from recent sequences, and segment-aware blending.
2. A UPI context layer that defines payment-specific inputs and a cold-start path for new users.
3. A leakage-safe, rolling-origin evaluation with

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

significance tests and feature-level explanations.

4. An honest empirical test on a public e-commerce log, with unfavorable results reported alongside favorable ones.

Section 2 reviews related work in chronological order. Section 3 presents the framework. Section 4 describes data and setup. Section 5 gives results, Section 6 discusses them, Section 7 lists limitations, and Section 8 concludes.

2. Related Work

Foundations. Bates and Granger (1969) showed that a combination of forecasts can beat its members. Box and Jenkins (1970) set out the standard approach to modeling a single time series, and exponential smoothing and decomposition remain common baselines (Hyndman & Athanasopoulos, 2021). Davis (1989) explained why people adopt technology through perceived usefulness and ease of use, and Venkatesh et al. (2003) later unified competing acceptance models. These ideas matter here because a forecast helps only if people use the tool that shows it. Wolpert (1992) proposed stacking, in which a second-level model learns how to combine the predictions of first-level models.

Learning methods: Long short-term memory networks made it practical to learn from sequences (Hochreiter & Schmidhuber, 1997). Random forests (Breiman, 2001) and gradient boosting (Friedman, 2001) became the leading methods for tabular data, later made fast by XGBoost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017). Zhang (2003) combined a statistical model with a neural network that learns the leftover error, an idea that many hybrids still follow. Hyndman and Koehler (2006) argued for scale-free error measures such as MASE, which suit series of very different sizes. Koren et al. (2009) showed how latent factors drive personalized recommendation, and Ricci et al. (2015) collected the wider field in one handbook.

Evaluation practice: Bergmeir and Benítez (2012) examined cross-validation for time series, and Cerqueira et al. (2020) found that methods respecting time order give more trustworthy estimates. Gama et al. (2014) surveyed concept drift, the tendency of behavior to change over time. Dal Pozzolo et al. (2018) showed that payment data bring delayed feedback and drift, so careful evaluation is essential. Lundberg and Lee (2017) unified feature-attribution methods under SHAP, which allows predictions of tree models to be explained.

Recent evidence on hybrids and simple baselines: Makridakis et al. (2018) reported that statistical methods were often more accurate than several machine learning methods and that evaluation choices matter. Smyl (2020) combined exponential smoothing with recurrent networks and won the M4 competition, whose design is documented by Makridakis et al. (2020). Bandara et al. (2020) showed that training a network on groups of similar series improves accuracy. Hewamalage et al. (2021) reviewed recurrent networks for forecasting and stressed careful preprocessing. Grinsztajn et al. (2022) found that tree-based models often beat deep networks on tabular data, and Zeng et al. (2023) showed that simple linear models can rival complex transformers. Together these results suggest keeping simple baselines in every comparison, which this paper does.

Research gap: Few studies combine user-level personalization, a hybrid forecaster for intermittent spending, a leakage-safe evaluation, and a payment-context layer. This paper addresses these together and tests them on public data.

3. Proposed Framework

3.1.Problem

For user i and month t, let y(i,t) be total spending. At the end of month t, the task is to predict y(i,t+1) using only information available at that time. The forecast is written as a baseline plus a learned correction:

$$y_hat(i,t+1) = s(i,t+1) + r(i,t+1)$$

where s is the smoothing baseline and r is the blended correction. Spending is modeled on the log scale, log(1 + y), to reduce the effect of very large orders, and predictions are converted back and kept at zero or above.

3.2.Stage 1: Smoothing baseline

For each user, simple exponential smoothing is run on the log series with several smoothing rates, and the rate with the lowest past error is used. A small seasonal adjustment is added for each combination of

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

behavioral segment and calendar month, estimated from earlier forecast errors and shrunk toward zero when few cases are available.

3.3.Tree learner

A LightGBM model learns the baseline's error from tabular features: the last 12 months of spending, the same month one year earlier, rolling means and spread, the number of active months, months since the last purchase, order counts, calendar month, the baseline forecast, and the segment label.

3.4.Stage 3: Neural learner

A small feed-forward network reads the last 12 months of log spending, an indicator for missing months, calendar month, history length, and the segment label. It predicts the same error as the tree learner but from a different view of the data.

3.5.Stage 4: Segment-aware stacking

Users are grouped into four behavioral segments by clustering on average spending, volatility, share of active months, and time since the last purchase. For each segment, non-negative weights that sum to one are fitted to combine the two learners. The weights use only forecasts from earlier months whose outcomes are already known. Prediction intervals come from absolute errors of earlier forecasts within each segment.

3.6.UPI context layer

When UPI records are available, the framework adds inputs that describe how payments behave: share of payments by instrument or app, recent failure rate, share of payments completed in the first attempt, time of day, merchant category, and reward or cashback intensity. These inputs enter the tree and neural learners as extra features and also help to form the segments. The layer is specified here so that it can be applied directly to a consented UPI export, but it is not evaluated in this paper.

3.7.Cold-start path

New users have too little history for a personal baseline. For them, the tree and neural learners rely on segment-level patterns and calendar information, and personal history gradually takes over as more months are observed.

4. Data And Experimental Setup

4.1.Dataset used

The study uses the Online Retail II dataset (Chen, 2019) from the UCI Machine Learning Repository, which extends the retail data described by Chen et al. (2012). It holds 1,067,371 invoice lines from a UK-based online retailer between 1 December 2009 and 9 December 2011, released under a CC BY 4.0 licence. Fields include invoice number, product code, quantity, invoice date, unit price, and customer identifier. Spending is quantity multiplied by unit price, summed per customer and month, with cancellations netted against purchases. Rows without a customer identifier or with non-positive prices are removed. This leaves 5,942 customers over 24 months (December 2009 to November 2011; the partial month of December 2011 is dropped). Many customers of this retailer are wholesalers, which makes spending very uneven.

The copy used for the experiments came from a public code repository and matches the official row count and date range. Readers should download the file from UCI to repeat the study.

4.2.Why UPI data are not in the study

No public user-level dataset of real UPI transactions was found. Open UPI datasets on data-sharing sites are simulated or hold a single account's statement, and national statistics from the National Payments Corporation of India are aggregated by month. Simulated data cannot support claims about real behavior, so they were not used. The UPI layer in Section 3.6 therefore awaits a consented payment export, such as exported statements from participants who give informed consent under ethics approval.

4.3.Evaluation design

Forecasts are one month ahead from five rolling origins, with target months July to November 2011. Every model is refit at each origin using only targets known at that time, and nothing is tuned on the evaluation months. Customers with at least three active months form the main evaluation set (12,836 forecasts). Customers with fewer form a separate short-history set (14,005 forecasts). Segments are fitted on data from before the first evaluation origin. Five random seeds vary the clustering start, tree sampling, and network initialization.

4.4.Benchmarks, metrics, and tests

Benchmarks are last value, seasonal naive, a three-month moving average, simple exponential

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

smoothing (SES), LightGBM alone, the neural network alone, and a Zhang-style hybrid (SES plus a neural network on its error). Ablations remove or change parts of the framework: baseline only (A1), baseline plus trees (A2), baseline plus neural network (A3), one global blend instead of segment blends (A4), and random instead of behavioral segments (A7). Metrics are MAE, RMSE, sMAPE, and MASE. MASE divides the total absolute error by the total one-step naive error of each customer's own history, so a value below 1 beats the naive method. sMAPE is set to zero when forecast and actual are both zero. Paired Wilcoxon signed-rank tests compare customer-level absolute errors. Models use fixed settings (LightGBM: 300 trees, learning rate 0.05, 31 leaves; network: two hidden layers of 32 and 16 units) As shown in Table 1 Accuracy for customers with three or more active months (12,836 forecasts; mean ± standard deviation over five seeds; deterministic models show one value)

5. Results

5.1.Overall accuracy

Table 1 Accuracy for customers with three or more active months (12,836 forecasts; mean ± standard deviation over five seeds; deterministic models show one value)

Model	MAE (£)	RMSE (£)	sMAPE (%)	MASE
Last value	300.8	1395.6	77.5	0.997
Seasonal naive	304.9	1129.9	85.2	1.011
Moving average (3)	246.6	1055.6	102.6	0.817
SES	225.4	1184.3	184.7	0.747
LightGBM	229.5 ± 0.4	1217.5 ± 7.7	188.6 ± 0.05	0.761 ± 0.001
Neural network	231.4 ± 4.6	1220.6 ± 51.9	188.5 ± 0.3	0.767 ± 0.015
Zhang-style hybrid	233.1 ± 3.1	1258.3 ± 33.7	186.7 ± 0.5	0.773 ± 0.010
Proposed framework	229.5 ± 0.8	1226.2 ± 6.9	188.7 ± 0.2	0.761 ± 0.003

SES is the most accurate model overall (MASE 0.747). The proposed framework and LightGBM both score 0.761, and the neural network scores 0.767. Compared with seasonal naive (1.011), the framework lowers MASE by about 25%. Its edge over the Zhang-style hybrid (0.773) is small next to that model's seed-to-seed variation. Paired tests show SES beating the framework by £5.4 to £7.2 per forecast in every seed ($p < 0.001$), the framework beating seasonal naive by about £72 to £74 ($p < 0.001$), and no consistent difference from LightGBM, with gaps between −£0.9 and +£0.8 whose sign changes across seeds. Two cautions apply. The moving average has the lowest RMSE and the naive methods the lowest sMAPE. This reflects the data: many customers spend nothing in a month, which pushes sMAPE of any non-zero forecast toward 200%, and a few very large customers dominate RMSE. MASE is not distorted this way and is the main metric.

5.2.Segments and ablations

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

Table 2 MASE by segment (mean over five seeds; overall column shows mean ± standard deviation)

Variant	Regular high-value	Moderate	Occasional	Dormant	Short history	Overall
SES (reference)	1.013	0.751	0.642	0.415	0.790	0.747
A1 Baseline only	1.477	0.765	0.658	0.420	0.562	0.867
A2 Baseline + trees	1.114	0.751	0.618	0.422	0.496	0.766 ± 0.002
A3 Baseline + neural	1.088	0.748	0.619	0.421	0.496	0.759 ± 0.013
A4 Global blend	1.086	0.750	0.618	0.422	0.496	0.758 ± 0.005
A7 Random segments	1.056	0.755	0.618	0.421	0.496	0.753 ± 0.002
Full model	1.094	0.750	0.618	0.422	0.496	0.761 ± 0.003

The benefit of learning depends on the customer. For occasional customers the learned variants reach 0.618 against 0.642 for SES. For the short-history group they reach 0.496 against 0.790, about 37% lower. This fits the argument of Bandara et al. (2020): models trained across many customers can borrow knowledge about how new or sporadic customers behave. For regular high-value customers the pattern reverses. All learned variants (1.06 to 1.13) are worse than SES (1.013), and a three-month moving average does better (0.966), because a few large irregular orders dominate their errors. Dormant customers sit near 0.42 for every model because most outcomes are zero.

The baseline alone (A1, 0.867) is worse than plain SES (0.747), mainly because of the high-value group. The segment-by-month adjustment is estimated from about a year of errors and adds noise, which the learners largely repair. Segment-specific blending did not help: the global blend (0.758) and the full model (0.761) cannot be told apart given seed variation, and random segments (0.753) did slightly better than behavioral ones. The learned weights differ by segment, with the tree learner weighted 0.71 for high-value customers and 0.39 to 0.47 for the other groups, but this did not translate into better accuracy. As shown in Table 3 Importance of feature families in the tree learner at the final origin (sum of mean absolute SHAP values in log units, mean over five seeds).

5.3.Explanation and calibration

Table 3 Importance of feature families in the tree learner at the final origin (sum of mean absolute SHAP values in log units, mean over five seeds)

Feature family	Importance
Recent spending lags	0.67
Baseline forecast and segment	0.63
Activity and recency	0.46
Calendar and seasonal	0.35
Rolling statistics	0.34

Recent spending and the baseline carry the most weight, followed by activity and recency, such as the number of active months and months since the last purchase. Calendar variables matter, though less than behavior. The ranking was identical in every seed. Nominal 80% prediction intervals covered 61.7% of

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

outcomes on average, with a mean width of about £1,506. In the first run, coverage was 84.6% for regular high-value, 77.9% for moderate, 59.7% for occasional, and 43.1% for dormant customers, and 80.8% for the short-history set. The intervals work for regular buyers but poorly for customers whose outcome is often exactly zero.

5.4.Stability over time

No model was best in every month. SES led in July, September, and November, the framework in August, and LightGBM in October. Errors rose for all models toward the autumn peak. The seasonal naive method worsened steadily, from a MASE of 0.771 in July to 1.197 in November, which suggests that 2011 demand differed from 2010 more than last year's pattern allows. With only two seasonal cycles in the data, seasonality is weakly identified.

6. Discussion

What the evidence supports. On this e-commerce log, learned components add the most for customers with short or sporadic histories, where they cut error substantially compared with smoothing. A simple smoother remains hard to beat overall, in line with Zeng et al. (2023) and Grinsztajn et al. (2022), so added complexity should be justified group by group. Behavioral segmentation did not improve accuracy, and the intervals need rework for intermittent customers, for example with a model that first predicts whether a purchase will happen and then how large it will be. Implications for UPI-enabled e-commerce. Short-history customers are the group that gained most, and they are also the users a payment app meets first. A forecast shown to such users, such as a likely spending range for the coming month, could be delivered with the learned model, while regular high-value users may be served well by a simple smoother. For acceptance, forecasts should appear as a plain range with a one-line reason drawn from the feature explanations, since people adopt tools that seem useful and easy to use (Davis, 1989; Venkatesh et al., 2003). These are design suggestions drawn from the e-commerce results, not findings about UPI users. What UPI data would add. The UPI context layer assumes that payment outcomes and rewards carry information about future spending. This remains untested here. A consented export would allow the ablation that removes payment-context features and a direct check of whether it improves forecasts, particularly for the short-history group.

7. Limitations And Future Work

The dataset comes from one UK retailer between 2009 and 2011, with many wholesale customers, so results may differ for Indian consumers who pay through UPI. The study contains no UPI transactions. The neural learner is a feed-forward network on a 12-month window rather than a recurrent network, hyperparameters were fixed rather than tuned, and some benchmarks such as ARIMA, XGBoost, and transformer forecasters were not run. Only one forecast horizon was studied, and only about two seasonal cycles were available. Segmentation can carry historical bias if used for credit or pricing decisions, so fairness checks are needed before any consequential use, and all data use must follow consent rules and applicable law, such as India's Digital Personal Data Protection Act, 2023. Future work should repeat the study on consented UPI statements, add payment-context features, test recurrent and transformer learners, and model the purchase and amount decisions separately to improve intervals.

Conclusion

This paper presented a hybrid framework that blends exponential smoothing, tree learning, and neural learning with segment-specific weights, and it specified a UPI context layer for payment-related inputs. On a public e-commerce log with 5,942 customers, the framework beat a seasonal naive benchmark by about 25% and matched gradient boosting, but plain exponential smoothing was slightly better overall. Learned components helped most for new and occasional customers, while segment-specific blending added no measurable gain and prediction intervals were too narrow for intermittent buyers. The results show where learned models earn their complexity and where they do not. Testing the UPI layer on consented payment data is the essential next step.

References

[1]. Bates, J. M., & Granger, C. W. J. (1969). The combination of forecasts. Operational

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

Research Quarterly, 20(4), 451–468.
[2]. Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control. Holden-Day.
[3]. Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3), 319–340.
[4]. Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259.
[5]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
[6]. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
[7]. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.
[8]. Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425–478.
[9]. Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159–175.
[10]. Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), 679–688.
[11]. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37.
[12]. Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. Information Sciences, 191, 192–213.
[13]. Chen, D., Sain, S. L., & Guo, K. (2012). Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. Journal of Database Marketing & Customer Strategy Management, 19(3), 197–208.
[14]. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44.
[15]. Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2015). Recommender systems handbook (2nd ed.). Springer.
[16]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD (pp. 785–794).
[17]. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems, 30.
[18]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, 30.
[19]. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784–3797.
[20]. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. PLoS ONE, 13(3), e0194889.
[21]. Chen, D. (2019). Online Retail II [Data set]. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/502/online+retail+ii
[22]. Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series. Expert Systems with Applications, 140, 112896.
[23]. Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. Machine Learning, 109, 1997–2028.
[24]. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. International Journal of

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0402
e ISSN: 2584-2854
Volume: 04 Issue: 10
October 2026
Page No: 2853 - 2860

Forecasting, 36(1), 54–74.
[25]. Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. International Journal of Forecasting, 36(1), 75–85.
[26]. Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting, 37(1), 388–427.
[27]. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and practice (3rd ed.). OTexts.
[28]. Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In NeurIPS Datasets and Benchmarks Track.
[29]. Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence, 37(9), 11121–11128.