RAMC: A Regime-Adaptive Meta-Classifier for Explainable Trend Prediction

Hanxue Yao[0009−0001−1233−1558] and Irena Koprinska[0000−0001−9479−4187]

School of Computer Science, The University of Sydney, Sydney, NSW, 2006, Australia
hyao0763@uni.sydney.edu.au, irena.koprinska@sydney.edu.au

Abstract. This paper introduces RAMC, a novel Regime-Adaptive Meta-Classifier framework, for trend prediction in time series data. RAMC aims to provide model interpretability by design, without sacrificing predictive performance. It first leverages motif discovery and clustering to identify regime structures, and then uses a meta-classifier for trend prediction. Unlike conventional black-box ensembles, the meta-classifier is built from intrinsically interpretable logistic regression experts and traceable probability weights, to provide explainable decision-making. Extensive experiments on twelve datasets from various domains (temperature, electricity demand, wind speed, and finance) demonstrate that RAMC achieves significant predictive improvements over transparent and black-box baselines while maintaining interpretability.

Keywords: Time Series · Regime Identification · Trend Prediction · Explainable AI · Meta-Learning.

1 Introduction

Accurate trend prediction remains a challenge due to the complex, non-stationary nature of time series. Unlike numerical point forecasting, trend prediction identifies directional movements, e.g., an increasing or decreasing trend in the next time horizon, which is more valuable for anticipating regime changes and mitigating risk. This task is vital for many applications, e.g., to inform investment strategies in finance or schedule generators in electricity networks.

Despite its importance, trend prediction remains under-explored; existing studies are often limited to specific domains, such as finance [18,13] and predominantly rely on black-box architectures [13]. In addition, beyond specific trend prediction tasks, general time series classification methods typically fall into two categories, both with limitations. Traditional interpretable methods (e.g., Logistic Regression), struggle with sparse inter-feature correlations and yield limited accuracy. Conversely, black-box models (e.g., Deep Learning or XGBoost) achieve higher performance but lack the transparency and explainability essential for practitioner trust in high-stakes domains. Therefore, a framework providing interpretability without sacrificing predictive performance is highly desirable.

In response to these limitations, numerous studies have focused on post-hoc explanations for time series classification. In the specific context of financial trend

prediction, SHAP was employed to interpret gradient boosting models for market crash prediction, providing feature attributions for key events such as the March 2020 financial meltdown [9]. Similarly, [8] applied both SHAP and LIME to a deep learning model to identify the driving factors behind market trends.

Beyond these domain-specific applications, post-hoc explanation methods for general time series have also been applied. For instance, Panigutti et al. [11] introduced Doctor XAI, a model-agnostic technique for explaining clinical sequence classifiers by handling the complexities of sequential and ontology-linked healthcare data. An Explainable AI tool employing prototypes and counterfactuals for instance-based explanations was introduced in [5]. Ojha et al. [10] employed SHAP, GradCAM, and LIME for explaining deep learning model for ECG data classification. Papadeas et al. [12] investigated various time series segmentation algorithms to optimize SHAP's efficiency, finding that the number of segments significantly impacts explanation quality and presenting a novel length-based attribution normalization technique.

However, these external interpretation methods may not reliably reflect the internal decision-making process. Research into intrinsic interpretability — designing models that are transparent by design — remains a significant challenge in time series prediction. Early work, such as [6], introduced a transformation that extracts representative subsequences (shapelets). Building on this concept, [14] introduced a dual prototypical shapelet framework that provides model interpretation by combining representative global samples with local discriminative shapelets. Other methods, such as [4], leveraged prototypes learned from the latent space of deep neural networks to elucidate the decision-making process for two-dimensional time-series classification.

In this work, we propose RAMC, a novel forecasting framework that aims to achieve high interpretability and superior classification performance. RAMC utilizes intrinsically transparent building blocks throughout its architecture. Specifically, it uses motifs and clustering to identify distinct behaviors (regimes). The decision-making process is subsequently governed by a multi-level meta-classifier composed of Logistic Regression experts which are interpretable classifiers. In addition, unlike traditional black-box ensembles, where weights are often uninterpretable parameters, RAMC employs logistic-based meta-regime posterior probabilities as weights. This allows to decompose complex forecasts into a clear ensemble, where both the expert outputs and weighting mechanisms outputs are mathematically traceable and aligned with human intuition. The main contributions of this paper are:

- We propose RAMC, a novel Regime-Adaptive Meta-Classifier framework for time series trend prediction which aims to provide model transparency by design, while maintaining high predictive performance.
- We develop a method to identify interpretable regime structures by leveraging motifs and clustering.
- We introduce a multi-level meta-classifier architecture based on Logistic Regression, ensuring that both weights and experts are mathematically traceable and linearly decomposable.

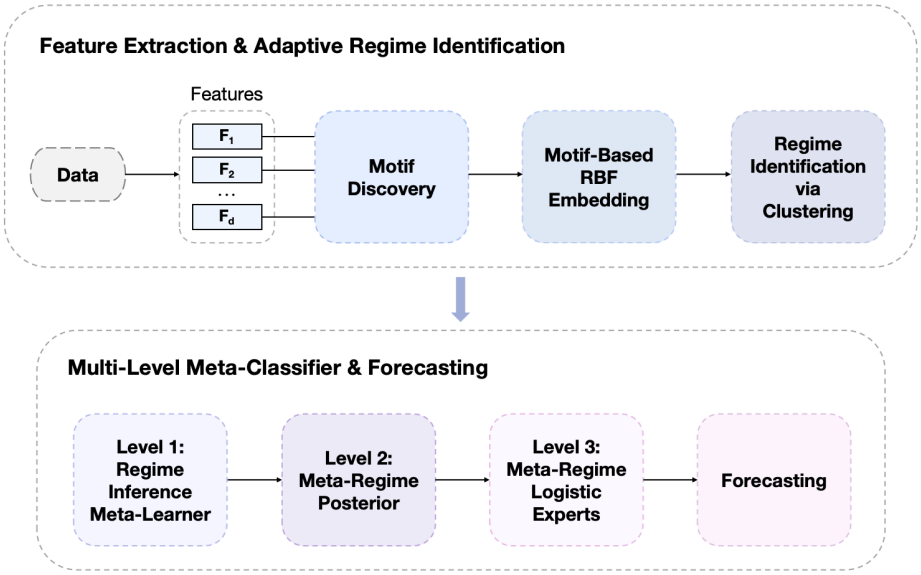

Fig. 1: The main stages of the RAMC framework

- We demonstrate RAMC's superiority across twelve diverse datasets, showing that it outperforms transparent and black-box baselines.

2 Proposed Methodology

2.1 Problem Definition

We consider an univariate time series $Y_1, \ldots, Y_t$. The objective is to predict the trend $L_t \in \{0, 1\}$ for the time step Y_{t+h} (with horizon $h = 1$ in our setting), where $L_t = 1$ denotes an increasing trend (up) and $L_t = 0$ – a non-increasing trend (not up).

The proposed RAMC framework for trend prediction is shown in Fig. 1.

2.2 Feature Extraction

The original univariate time series data is transformed into a set of features, summarized in Tables 1–4. For all application domains (temperature, electricity demand, wind speed, and finance), the raw observation at time t is denoted as Y_t. The feature design aims to capture key non-linear temporal characteristics of the data, including level differences, volatility, periodic patterns, lagged dependencies, with additional domain-specific features tailored to each application domain.

Table 1: Temperature datasets - summary of extracted features

Feature	Definition
Temperature Level (F_1)	Y_t (current temperature)
Temperature Difference (F_2)	$\Delta Y_t = Y_t - Y_{t-1}$ (1-hour)
Short-Term Deviation (F_3)	$Y_t - \overline{Y}_{t-11:t}$, where $\overline{Y}_{t-11:t} = \frac{1}{12}\sum_{i=0}^{11} Y_{t-i}$ (12-hour)
Volatility Indicator (F_4)	$\frac{\sigma_t}{\text{Range}_t}$, where $\sigma_t = \text{std}(Y_{t-23:t})$ and $\text{Range}_t = \max(Y_{t-23:t}) - \min(Y_{t-23:t})$ (24-hour)
Diurnal Pattern (F_5)	$[\sin(\frac{2\pi h}{24}), \cos(\frac{2\pi h}{24})]$, $h \in \{0, \ldots, 23\}$
Lag Difference (F_6)	$Y_t - Y_{t-24}$ (24-hour lag)

Table 2: Electricity demand datasets - summary of extracted features

Feature	Definition
Electricity Demand Level (F_1)	Y_t (current electricity demand)
Electricity Demand Difference (F_2)	$\Delta Y_t = Y_t - Y_{t-1}$ (5-min)
Electricity Demand Volatility (F_3)	$\sigma_t = \text{std}(Y_{t-11:t})$ (1-hour)
Diurnal Pattern (F_4)	$[\sin(\frac{2\pi h}{24}), \cos(\frac{2\pi h}{24})]$, $h \in \{0, \ldots, 23\}$
Lag Difference (F_5)	$Y_t - Y_{t-24}$ (2-hour lag)
Workday Indicator (F_6)	$\mathbb{I}[\text{day} \in \text{Workday}]$

2.3 Adaptive Regime Identification

This part includes motif discovery, motif-based RBF embedding, and regime identification via clustering.

Motif Discovery. *Input: Extracted features $\mathcal{F}$, Output: Motif M_k.*

The first step is *motif discovery* which aims to capture interpretable structural prototypes from the training data. Specifically, we split the extracted feature sequence $\mathcal{F} = \{F_1, F_2, \ldots, F_d\}$ into a series of sliding windows $\{x_{t-L+1:t}\}$ of length L. Each window is flattened into an $(L \times d)$-dimensional vector to serve as the input for K-means clustering. By clustering the training windows into K distinct groups, the algorithm identifies a set of representative motifs $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$.

Each motif M_k is defined by its cluster centroid, providing a compact and interpretable set of historical prototypes that characterize distinct states of the time series. These centroids serve as the basis for the subsequent embedding layer, where current behaviors are represented by their relative similarity to these discovered motifs. The hyperparameters $K \in \{3, 4, 5, 6, 7, 8\}$ and $L \in \{10, 20, 30, 40\}$ are determined using cross validation, see Section 3.2.

Motif-Based RBF Embedding. *Input: Motif M_k, Output: Motif-based RBF embedding $z(t)$.*

Table 3: Wind speed datasets - summary of extracted features

Feature	**Definition**
Wind Speed Level (F_1)	Y_t (current wind speed), where $Y_t = \sqrt{u_{10,t}^2 + v_{10,t}^2}$
Wind Speed Difference (F_2)	$\Delta Y_t = Y_t - Y_{t-3}$ (3-hour)
Wind Speed Volatility (F_3)	$\sigma_t = \text{std}(Y_{t-2:t})$ (3-hour)
Diurnal Pattern (F_4)	$[\sin(\frac{2\pi h}{24}), \cos(\frac{2\pi h}{24})]$, $h \in \{0, \ldots, 23\}$
Wind Speed Acceleration (F_5)	$(Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$
Wind Direction (F_6)	$\theta_t = \text{atan2}(u_{10,t}, v_{10,t})$

Table 4: Financial datasets - summary of extracted features

Feature	**Definition**
Price Level (F_1)	Y_t (current closing price)
Log Return (F_2)	$R_t = \ln(\frac{Y_t}{Y_{t-1}})$
Momentum (F_3)	$Y_t - Y_{t-5}$ (5-day)
Relative Strength Index (F_4)	$\text{RSI}_t = 100 - \frac{100}{1+\text{RS}_t}$, where $\text{RS}_t = \frac{\overline{\text{Gain}}_{t-13:t}}{\overline{\text{Loss}}_{t-13:t}}$ (14-day)
Volatility Change (F_5)	$\Delta\sigma_t = \sigma_t - \sigma_{t-1}$, where $\sigma_t = \text{std}(R_{t-9:t})$ (10-day)
Autocorrelation (F_6)	$\rho_t = \text{Corr}(R_{t-19:t}, R_{t-20:t-1})$ (20-day)
Volatility Regime (F_7)	$\mathbb{I}[\text{std}_{20}(R_t) > \text{median}(\text{std}_{20}(R))]$ (20-day)

The high-dimensional sliding windows are then projected into a lower-dimensional Z-space which captures the similarity between the current window and the identified motifs. This embedding is computed for both training and prediction purposes. For each window x_t at time step t, we compute its similarity to each motif M_k using an RBF kernel:

$$z_k(t) = \exp\left(-\frac{\|x_t - M_k\|_2}{2\sigma^2}\right), \quad k = 1, \ldots, K \tag{1}$$

where $\|\cdot\|_2$ is the Euclidean distance. The bandwidth parameter σ is selected from the mean, median, or 75th percentile of the pairwise distance distribution to ensure robust scaling across different datasets. To produce a probability representation, $z_k(t)$ is normalized where ϵ is a small constant for stability:

$$z(t) = \frac{[z_1(t), \ldots, z_K(t)]}{\sum_{j=1}^{K} z_j(t) + \epsilon} \in \mathbb{R}^K \tag{2}$$

The resulting vector $z(t)$ serves as the motif-based RBF embedding (Z-space representation), providing a normalized encoding that quantifies the current window's similarity to the discovered historical prototypes. This representation is utilized in two distinct phases: during training, the sequence $\{z(t)\}$ is used to fit the multi-level meta-classifier, and at prediction time, $z(t)$ is computed for each new incoming window to serve as the input for the pre-trained classifier. This representation is directly traceable to recognizable data patterns - the motifs.

Regime Identification via Clustering. *Input: Motif-based RBF embedding $z(t)$, Output: Regime label c_t for it.*

This clustering stage aggregates the motif-based RBF embeddings learned from the training set by grouping recurring local behaviors into representative states, called *regimes*. To do this, we apply a second-stage K-means clustering to the embeddings $\{z(t)\}$, partitioning the Z-space into R active regimes ($R \in \{2, 3\}$) and assigning each training time step a regime label $c_t \in \{1, \ldots, R\}$. By reducing the K-dimensional motif representations to R clusters on the training data, we transition from individual pattern matching to a higher-level structural understanding of time-series data.

2.4 Multi-Level Meta-Classifier

We employ a Mixture-of-Experts (MoE) framework for trend prediction. Unlike black-box ensemble methods [2,16], which lack transparency in weight allocation and expert selection, recent architectures such as N-BEATS-MOE [7] demonstrate that MoE can provide enhanced interpretability by explicitly identifying relevant experts via gating mechanisms.

Building on this principle, we introduce a three-level meta-classifier architecture based on logistic regression, ensuring that both MoE ensemble weights and experts are inherently interpretable. By employing a linear framework, the architecture avoids black-box decision-making and provides explicit attribution of expert contributions. This structure ensures that final results are directly convertible into human-interpretable justifications.

Level 1: Regime Inference Meta-Learner. *Input: Motif-based RBF embeddings for regimes $z(t)$ with their associated regime labels c_t, Output: Trained classifier on them, predicting the regime probability vector $\alpha(t)$ for a new example.*

The Level 1 meta-learner bridges the gap between the motif-based RBF embeddings $z(t)$ and the identified regime labels c_t from the training set, enabling to transform individual pattern matching into principled regime probabilities. To achieve this, we train a logistic regression meta-learner that learns the mapping between the motif-based RBF embedding $z(t)$ of each training set window and its associated regime label c_t. It produces a regime probability vector $\alpha(t) = \text{softmax}(Wz(t)+b) \in \mathbb{R}^R$, where $W \in \mathbb{R}^{R \times K}$ and $b \in \mathbb{R}^R$ are the learned weights and biases. Once trained, this meta-learner is used at prediction time to perform real-time regime inference for each new incoming window.

The component $\alpha_m(t)$ denotes the conditional probability of regime m given the embedding $z(t)$, where w_m is the m-th row of W:

$$\alpha_m(t) = \frac{\exp(w_m^\top z(t) + b_m)}{\sum_{m'=1}^{R} \exp(w_{m'}^\top z(t) + b_{m'})}, \quad m = 1, \ldots, R \tag{3}$$

The regime probability vector $\alpha(t) = [\alpha_1(t), \ldots, \alpha_R(t)]^\top$ resides on the $(R-1)$-simplex Δ^{R-1}, providing a soft regime assignment that satisfies $\sum_{m=1}^{R} \alpha_m(t) =$

1. The model hyperparameters, including the solvers $\in$ {'lbfgs', 'saga'} and maximum iterations $\in \{500, 1000, 2000\}$, are determined using grid search. The probabilistic output $\alpha(t)$ provides a granular and interpretable regime inference, serving as the input for the meta-regime posterior probability in Eq.(6).

Level 2: Meta-Regime Identification and Posterior. *Input: Motif-based RBF embeddings for regimes $z(t)$, regime probability $\alpha(t)$, Output: Meta-regimes and meta-regime posterior probabilities $\gamma(t)$ (ensemble weights for next stage).*

First, meta-regimes are identified by aggregating the individual regimes into S broader categories ($S \leq R$). To this end, we perform a second-level clustering on the regime centroids in Z-space to capture structural similarities among states.

Second, to obtain the meta-regime posterior probability $\gamma_s(t)$, we proceed as follows. For each individual regime m identified in step "Regime Identification via Clustering", we compute its centroid $\mu_m^{(z)}$:

$$\mu_m^{(z)} = \frac{1}{|T_m|} \sum_{t \in T_m} z(t), \quad m = 1, \ldots, R \tag{4}$$

where $T_m = \{t \mid c_t = m\}$ is the set of time steps assigned to individual regime m.

We apply K-means clustering to the R centroids to identify S meta-regime clusters and their centers $\{\nu_s^{(z)}\}_{s=1}^{S}$. The soft assignment $\beta_{m,s}$ between individual regime m and meta-regime s is calculated via a distance-based softmax:

$$\beta_{m,s} = \frac{\exp\left(-\|\mu_m^{(z)} - \nu_s^{(z)}\|_2\right)}{\sum_{s'=1}^{S} \exp\left(-\|\mu_m^{(z)} - \nu_{s'}^{(z)}\|_2\right)}, \quad m = 1, \ldots, R;\ s = 1, \ldots, S \tag{5}$$

where $\|\cdot\|_2$ is the Euclidean distance between regime and meta-regime centers.

We use the Level 1 regime probabilities $\alpha(t)$ to derive the meta-regime posterior probability $\gamma_s(t)$. Specifically, the probability for each meta-regime s is obtained by marginalizing over the individual regimes:

$$\gamma_s(t) = \sum_{m=1}^{R} \alpha_m(t) \cdot \beta_{m,s}, \quad s = 1, \ldots, S \tag{6}$$

The resulting probability vector $\gamma(t) = [\gamma_1(t), \ldots, \gamma_S(t)]^\top$ resides on the $(S-1)$-simplex Δ^{S-1}, serving as the *ensemble weights* for the Meta-Regime Logistic Experts in the next stage. These ensemble weights identify environmental transitions, selectively activating the corresponding experts to handle structural shifts in the underlying time-series dynamics (see Section 4.4).

Level 3: Meta-Regime Logistic Experts. *Input: Meta-regime clusters s and ensemble weights $\gamma(t)$, Output: Logistic regression expert classifiers for each cluster and MoE ensemble prediction $P(Y_{t+h} = 1 \mid X_t)$.*

We train an independent logistic regression expert for each meta-regime cluster s on the training data. The input is the original feature vector X_t, and the output is the regime-conditioned probability $p^{(s)}(Y_{t+h} = 1 \mid X_t)$ of each expert:

$$p^{(s)}(Y_{t+h} = 1 \mid X_t) = \sigma(\theta_s X_t + b_s) \tag{7}$$

where $Y_{t+h} \in \{0, 1\}$ is the binary trend label as defined in Eq. (9), $\sigma(\cdot)$ is the sigmoid function and $X_t = [F_1(t), \ldots, F_d(t)]$ is the input feature vector at time t. The learned parameters $\theta_s \in \mathbb{R}^d$ and $b_s \in \mathbb{R}$ are the feature weights and bias for the s-th meta-regime, respectively. By mapping each meta-regime to a specific configuration of θ_s, we can identify the distinct feature contributions that characterize each time series (details in Section 4.4).

The final MoE ensemble prediction is computed as a weighted sum of the predictions of these experts $p^{(s)}(Y_{t+h} = 1 \mid X_t)$ and their meta-regime posterior probabilities $\gamma_s(t)$ in Eq. (6):

$$P(Y_{t+h} = 1 \mid X_t) = \sum_{s=1}^{S} \gamma_s(t) \cdot p^{(s)}(Y_{t+h} = 1 \mid X_t) \tag{8}$$

This ensemble prediction $P(Y_{t+h} = 1 \mid X_t)$ is mapped to a binary trend prediction $\hat{L}_t \in \{0, 1\}$ through a decision threshold defined in Eq. (10).

2.5 Forecasting

We formulate the task as binary trend prediction. Specifically, the model predicts whether the observed univariate sequence will go up ($L_t = 1$) or not ($L_t = 0$) in the subsequent time step where Y_t represents the observed value at time t:

$$L_t = \begin{cases} 1, & \text{if } Y_{t+h} > Y_t \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The final trend forecast $\hat{L}_t$ is determined by applying a decision threshold θ to the ensemble prediction $P(Y_{t+h} = 1 \mid X_t)$ defined in in Eq. (8):

$$\hat{L}_t = \begin{cases} 1, & \text{if } P(Y_{t+h} = 1 \mid X_t) \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

To optimize prediction performance across various domains, we treat the decision threshold θ as a tunable hyperparameter evaluated over the candidate set $\Theta = \{0.4, 0.45, 0.5, 0.55, 0.6\}$.

3 Data and Experimental Setup

3.1 Data Description and Preprocessing

To assess the proposed RAMC framework, we evaluated its performance across twelve univariate datasets, categorized into four distinct domains: Temperature,

Electricity Demand, Wind Speed, and Finance. These datasets, sourced from established public repositories, encompass a wide range of real-world forecasting challenges. Fig. 2 shows the raw time series (feature F_1) for each dataset, highlighting their seasonal patterns and volatility. A summary of the datasets is presented in Table 5, with detailed descriptions provided below.

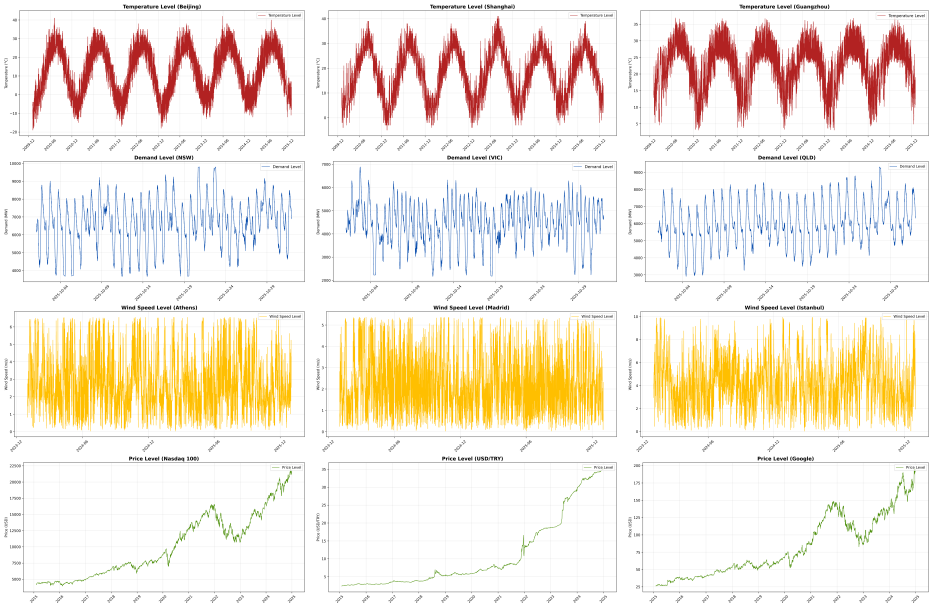

Fig. 2: Visualization of datasets (feature F_1) across twelve datasets. Rows correspond to domains: Temperature, Electricity Demand, Wind Speed, and Finance.

Table 5: Summary of twelve datasets. Length is per dataset.

Category	**Name of datasets**	**Frequency**	**Length**
Temperature	Beijing, Shanghai, Guangzhou	Hourly	52,584
Electricity Demand	NSW, VIC, QLD	5-min	8,928
Wind Speed	Athens, Madrid, Istanbul	Hourly	17,544
Finance	Nasdaq 100, Google USD/TRY	Daily Daily	2,494 2,584

Temperature: It contains hourly air temperature measurements for Beijing, Shanghai, and Guangzhou, China, for a six-year period from 1/1/2010 to

31/12/2015 [15]. We used the 'Temp" column. **Electricity Demand:** It features 5-minute electricity demand data for the states of NSW, VIC, and QLD in Australia, during October 2025 [1]. The 'Totaldemand' column is used for analysis. **Wind Speed:** It contains hourly wind speed records for Athens $(38.00°, 23.75°)$, Istanbul $(41.00°, 29.00°)$, and Madrid $(40.50°, -3.75°)$ from 1/1/2024 to 31/12/2025. The data consists of the 10m u-component ($u_{10,t}$) and v-component ($v_{10,t}$) of wind from the ERA5 dataset [3]. The Wind Speed Level Y_t is computed as the magnitude of the two-dimensional wind vector. **Finance:** It includes daily closing price for three key asset classes: stock indices (Nasdaq 100), foreign exchange (USD/TRY), and individual stocks (Google). It covers a ten-year period from 1/1/2015 to 31/12/2024 and is available from [17].

For all datasets, data preprocessing was performed. Missing observations were removed and remaining gaps were filled using second-order polynomial interpolation to maintain continuity. Outliers were filtered using the interquartile range (IQR) criterion. The processed dataset was then split into training (80%) and testing (20%) sets. All values were normalized to the range $[0, 1]$.

3.2 Baselines and Evaluation Metrics

We evaluate our proposed method against six baseline models categorized by their interpretability: (i) **Transparent Models** (Logistic Regression and Ridge Regression) — inherently interpretable baselines requiring no post-hoc explanation, (ii) **Black-box Models** from two groups — tree ensembles (XGBoost and LightGBM) and neural networks (CNN and LSTM), which require post-hoc interpretability techniques such as SHAP or other XAI techniques.

For comparison, the input window length L for all baselines is set to match the optimal value identified by our proposed method's grid search for each dataset (see Section 2.3). All hyperparameters in this work are tuned via 3-fold time-series cross-validation on the training set, maximizing the F1-score.

Logistic Regression: A linear classifier with balanced class weights, regularization parameter (C) = [0.01, 0.1, 1, 10], and maximum iterations = [1000, 2000]. **Ridge Regression:** A ridge classifier with balanced class weights, regularization strength (α) = [0.01, 0.1, 1, 10], and solver = ['auto', 'cholesky']. **XGBoost:** A gradient boosting ensemble with number of estimators = [150, 200], tree depth = [6, 8], and learning rate = [0.01, 0.05, 0.1]. **LightGBM:** A gradient boosting model with balanced class weights, number of estimators = [100, 200], number of leaves = [31, 63], and learning rate = [0.01, 0.05, 0.1]. **CNN:** A 1D-convolutional neural network with one convolutional layer with ReLU activation followed by a fully connected layer, number of filters = [16, 32], kernel size = [3, 5], number of epochs = [50, 100], batch size = [16, 32], and learning rate = [0.001, 0.005]. **LSTM:** A long short-term memory network with one hidden layer followed by a fully connected layer, number of hidden units = [32, 64], number of epochs = [50, 100], batch size = [16, 32], and learning rate = [0.001, 0.005].

All neural networks were trained using the Adam optimizer and weighted cross-entropy loss to handle class imbalance. The prediction performance of all models was measured using accuracy and F1-score.

4 Experimental Results and Discussion

4.1 Feature Correlation Analysis

Fig. 3 shows the feature correlations by selecting one dataset from each category: Beijing, NSW, Athens, and Nasdaq 100, highlighting that the correlations between input features remain generally low, with coefficients below 0.55. These low correlations reduce multicollinearity within the explainable latent state space, enabling the model to leverage diverse information with minimal redundancy. However, such sparse correlation structures imply that no single feature provides a dominant predictive signal. This suggests that the interpretability of the identified states depends not on simple linear associations, but on capturing the complex, non-linear interactions embedded within distinct regimes.

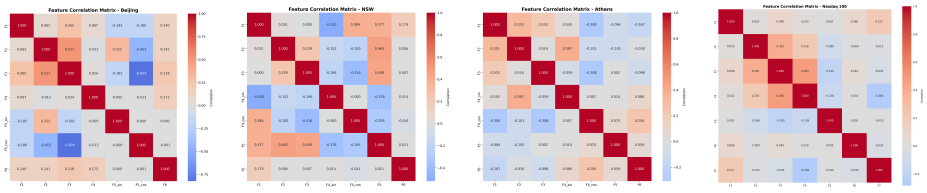

Fig. 3: Feature correlation matrices for Beijing, NSW, Athens, and Nasdaq 100

4.2 Overall Performance

The trend prediction performance of the proposed RAMC method, in comparison with baseline models across twelve datasets, is summarized in Table 6. RAMC uniquely achieves both superior predictive performance and enhanced interpretability. While existing models typically sacrifice one for the other (transparent models often fail to capture non-linear dependencies; black-box models lack sufficient explainability), RAMC consistently outperforms all baselines in terms of accuracy and F1-score while maintaining interpretability. This dual advantage is particularly valuable in trend prediction, where both reliability and transparency are essential for establishing trust.

Comparison with All Baselines: Our RAMC method significantly outperforms all six baselines across the twelve datasets. The baselines struggle to produce reliable predictions. Most baseline models achieve accuracy and F1-score values around 0.5, comparable to random guessing. While XGBoost and LSTM occasionally show marginal improvements on specific datasets such as USD/TRY or Nasdaq 100, their performance lacks stability. In contrast, RAMC delivers robust and consistent results across diverse domains, notably bridging the performance gap, particularly in the temperature and finance domains. These findings demonstrate RAMC's effectiveness in capturing predictive trend signals that other baselines fail to identify.

Analysis Against Transparent Models: While transparent baselines such as Logistic Regression and Ridge Regression offer high interpretability, they struggle to capture the complex and non-linear trend dependencies. This limitation is particularly evident in the USD/TRY dataset: our proposed RAMC model achieves an accuracy of 0.704 and an F1-score of 0.824, whereas Logistic and Ridge Regression yield significantly lower accuracies (0.386 and 0.366) and F1-scores (0.282 and 0.237).

The experimental results highlight a substantial performance gap, which can be attributed to the low inter-feature correlation visualized in Fig. 3. Linear models are typically effective only when individual features maintain strong, independent relationships with the target; however, they struggle to extract predictive trend signals embedded within a diverse set of weakly correlated variables. In contrast, our RAMC model effectively navigates this sparse correlation structure by identifying extracted features that capture latent, higher-order patterns that static linear weights are unable to represent. By extracting input features, the RAMC framework captures critical predictive information that traditional linear baselines overlook, while ensuring model explainability. This confirms that superior predictive performance and interpretability are not mutually exclusive.

Analysis Against Black-Box Models: XGBoost and LSTM demonstrate competitive performance on specific datasets; however, their results are often inconsistent across metrics. For instance, on the Shanghai dataset, XGBoost and LSTM achieve high accuracy scores of 0.715 and 0.749, yet their corresponding F1-scores are only 0.125 and 0.098. This discrepancy indicates a failure to maintain a balance between precision and recall, suggesting that these models may struggle with class imbalance or produce biased predictions despite high raw accuracy.

In certain cases, these baselines show strong performance on specific datasets, such as XGBoost on USD/TRY and LSTM on Nasdaq 100. However, they lack the inherent transparency required for financial applications, where trust and accountability are critically important. Unlike RAMC, which provides inherent interpretability, these models operate as black boxes, necessitating either supplementary post-hoc explanation methods or offering no clear mechanism to interpret the underlying decision-making process. Consequently, even when these models achieve high predictive performance, their utility is constrained in domains where explainability is as vital as predictive accuracy.

4.3 Statistical Test

A pairwise comparison for statistical significance was conducted using the Wilcoxon signed-rank test, following the methodology of [16], on both accuracy and F1-score across all datasets. All pairwise differences between RAMC and the baselines were statistically significant at $p \leq 0.05$, demonstrating consistent improvements in overall trend prediction.

Table 6: Performance evaluation of RAMC on all datasets

	Beijing		**Shanghai**		**Guangzhou**	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1
RAMC	**0.802**	**0.700**	**0.836**	**0.661**	**0.782**	**0.746**
Logistic Regression	0.510	0.379	0.500	0.300	0.519	0.469
Ridge Regression	0.509	0.379	0.499	0.301	0.517	0.466
XGBoost	0.501	0.138	0.715	0.125	0.488	0.202
LightGBM	0.458	0.180	0.521	0.236	0.444	0.322
CNN	0.494	0.258	0.529	0.245	0.402	0.572
LSTM	0.537	0.301	0.749	0.098	0.469	0.534

	NSW		**VIC**		**QLD**	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1
RAMC	**0.596**	**0.648**	**0.592**	**0.687**	**0.614**	**0.599**
Logistic Regression	0.488	0.489	0.493	0.435	0.479	0.484
Ridge Regression	0.491	0.493	0.497	0.439	0.477	0.481
XGBoost	0.515	0.469	0.495	0.444	0.497	0.515
LightGBM	0.503	0.536	0.487	0.463	0.493	0.481
CNN	0.490	0.450	0.525	0.368	0.467	0.429
LSTM	0.491	0.466	0.497	0.408	0.495	0.351

	Athens		**Madrid**		**Istanbul**	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1
RAMC	**0.608**	**0.613**	**0.577**	**0.687**	**0.593**	**0.684**
Logistic Regression	0.493	0.501	0.503	0.520	0.498	0.484
Ridge Regression	0.492	0.501	0.506	0.522	0.498	0.484
XGBoost	0.485	0.517	0.496	0.487	0.498	0.474
LightGBM	0.509	0.522	0.500	0.520	0.480	0.506
CNN	0.513	0.548	0.500	0.138	0.503	0.554
LSTM	0.503	0.585	0.495	0.484	0.483	0.524

	Nasdaq 100		**USD/TRY**		**Google**	
Model	Accuracy	F1	Accuracy	F1	Accuracy	F1
RAMC	**0.574**	**0.729**	**0.704**	**0.824**	**0.569**	**0.724**
Logistic Regression	0.497	0.548	0.386	0.282	0.460	0.353
Ridge Regression	0.495	0.545	0.366	0.237	0.452	0.343
XGBoost	0.486	0.602	0.551	0.663	0.483	0.562
LightGBM	0.512	0.596	0.419	0.447	0.501	0.510
CNN	0.446	0.414	0.543	0.650	0.450	0.409
LSTM	0.544	0.670	0.473	0.568	0.519	0.535

4.4 Interpretability Case Study

To demonstrate the interpretability of RAMC, we consider the NSW (electricity demand) dataset as a representative case, given its well-documented regime shifts driven by social and environmental factors. The interpretability is enhanced through the following two aspects.

Feature Weight Explanations: As defined in Eq. (7), each Meta-Regime Logistic Expert represents a fixed, interpretable linear configuration. Since all input features are standardized, the learned feature weights θ_s (Table 7) directly reflect the marginal contribution of each feature. Expert A (Steady-State Expert) exhibits a dominant positive reliance on F_5 (1.032), suggesting a strategy that prioritizes 2-hour lag momentum to capture stable, autocorrelated demand trends. In contrast, Expert B (Regime-Shift Expert) demonstrates a distinct decision logic with heavy negative coefficients on F_1 (-0.902) and F_4(Cos) (-0.540). This signals its activation during structural transition regimes where high current demand levels (F_1) are expected to undergo rapid reversals or be aligned with specific periodic cycles.

Table 7: Feature weights (θ_s) for each Meta-Regime Logistic Expert

Expert	F_1	F_2	F_3	F_4 **(Sin)**	F_4 **(Cos)**	F_5	F_6
Expert A	-0.480	0.099	-0.018	-0.230	0.111	1.032	0.089
Expert B	-0.902	0.094	0.113	-0.300	-0.540	0.464	0.188

Dynamic Ensemble Weights and Regime-Switching Analysis: The model identifies environmental transitions by dynamically shifting the ensemble weights $\gamma_s(t)$ defined in Eq. (6). As visualized in Fig. 4, Expert A serves as the primary choice for 76.2% of the steps, while Expert B is selectively activated (23.8%) to handle structural shifts. The 16 detected regime shifts at $t \in \{108, 127, \ldots, 1648\}$ mark precise moments where the model perceives a change in the underlying data-generating process; the exact timestamps for these shifts are detailed in Table 8. By mapping the time steps to the actual timeline, we observe that Expert B's activation periods align closely with physical operational contexts. For example, the recurring activations at $t \in \{442, 1020, 1308\}$ (approx. 04:00 PM) consistently coincide with the afternoon demand ramp-up on workdays, while a series of shifts at $t \in \{215, 498, 782, 1077, 1367, 1648\}$ (ranging from 08:15 PM to 09:00 PM) accurately mark the onset of the evening recession across multiple test days. Furthermore, the model successfully distinguishes between workdays and weekends, evidenced by the contrast between the sporadic switches handling short-term midday fluctuations on Sunday (26 Oct, $t \in [108, 147]$) and the prolonged, continuous activation of Expert B during the major Monday afternoon-to-evening transition (27 Oct, $t \in [442, 498]$).

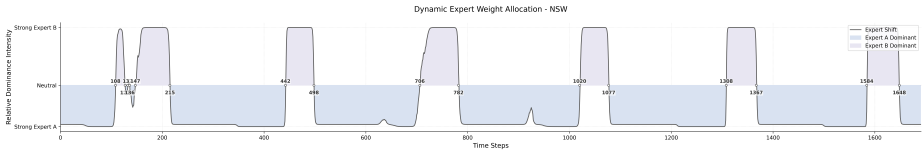

Fig. 4: Dynamic expert weight allocation $\gamma_s(t)$

Table 8: Exact timestamps of the 16 detected regime shifts

t	**108**	**127**	**132**	**136**	**147**	**215**	**442**	**498**
Time	26 Oct 12:05	26 Oct 13:40	26 Oct 14:05	26 Oct 14:25	26 Oct 15:20	26 Oct 21:00	27 Oct 15:55	27 Oct 20:35
t	**706**	**782**	**1020**	**1077**	**1308**	**1367**	**1584**	**1648**
Time	28 Oct 13:55	28 Oct 20:15	29 Oct 16:05	29 Oct 20:50	30 Oct 16:05	30 Oct 21:00	31 Oct 15:05	31 Oct 20:25

4.5 Limitations and Future Work

Despite the promising results and interpretability, several limitations remain. Currently, the framework is designed for binary trend prediction with a fixed horizon ($h = 1$). Future research could extend this to multi-class prediction (e.g., upward, downward, and stationary) and longer prediction horizons. Additionally, while this work focuses on univariate time series, generalizing the architecture to multivariate data would further broaden its practical applicability.

5 Conclusion

We propose RAMC, a novel Regime-Adaptive Meta-Classifier framework designed to bridge the gap between predictive accuracy and model interpretability in univariate trend prediction. RAMC first leverages motif and clustering to identify interpretable regimes. It then employs a logistic regression meta-learner to derive granular regime probabilities and aggregates structural similarities via meta-regime posterior probabilities. These posteriors serve as ensemble weights to combine interpretable logistic regression experts, producing a final ensemble prediction characterized by linearly decomposable and mathematically traceable decision logic. Extensive evaluations across twelve datasets from the temperature, electricity demand, wind speed, and finance domains demonstrate that RAMC significantly outperforms transparent and black-box baselines, offering a robust and intrinsically interpretable solution for high-stakes applications. Future work includes extending RAMC to longer prediction horizons, multi-class prediction and multivariate time series.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Australian Energy Market Operator (AEMO): Aggregated Price and Demand Data. https://www.aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/aggregated-data (2025)
2. Cerqueira, V., Torgo, L., Pinto, F., Soares, C.: Arbitrage of forecasting experts. Machine Learning **108**(6), 913–944 (2019)
3. European Centre for Medium-Range Weather Forecasts (ECMWF): ERA5 Hourly Time-Series Data on Single Levels from 1940 to Present. https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels-timeseries (2025)
4. Gee, A.H., Garcia-Olano, D., Ghosh, J., Paydarfar, D.: Explaining deep classification of time-series data with learned prototypes. In: IJCAI Workshop on Knowledge Discovery in Healthcare Data. vol. 2429 (2019)
5. Håvardstun, B., Ferri, C., Flikka, K., Telle, J.A.: XAI for time series classification: evaluating the benefits of model inspection for end-users. In: World Conference on Explainable Artificial Intelligence (xAI). pp. 439–453. Springer (2024)
6. Lines, J., Davis, L.M., Hills, J., Bagnall, A.: A shapelet transform for time series classification. In: International Conference on Knowledge Discovery and Data Mining (KDD). pp. 289–297 (2012)
7. Matos, R., Roque, L., Cerqueira, V.: N-BEATS-MOE: N-BEATS with a mixture-of-experts layer for heterogeneous time series forecasting. arXiv:2508.07490 (2025)
8. Muhammad, D., Ahmed, I., Naveed, K., Bendechache, M.: An explainable deep learning approach for stock market trend prediction. Heliyon **10**(21) (2024)
9. Ohana, J.J., Ohana, S., Benhamou, E., Saltiel, D., Guez, B.: Explainable AI (XAI) models applied to the multi-agent environment of financial markets. In: Workshop on Explainable and Transparent AI and Multi-Agent Systems. pp. 189–207 (2021)
10. Ojha, J., Haugerud, H., Yazidi, A., Lind, P.G.: Exploring interpretable AI methods for ECG data classification. In: Intelligent Cross-Data Analysis and Retrieval (ICDAR). pp. 11–18 (2024)
11. Panigutti, C., Perotti, A., Pedreschi, D.: Doctor XAI: an ontology-based approach to black-box sequential data classification explanations. In: Conference on Fairness, Accountability, and Transparency. pp. 629–639 (2020)
12. Papadeas, N., Serramazza, D.I., Abdallah, Z., Ifrim, G.: An empirical evaluation of factors affecting SHAP explanation of time series classification. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 458–473. Springer (2025)
13. Shen, J., Shafiq, M.O.: Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data **7**(1), 66 (2020)
14. Tang, W., Liu, L., Long, G.: Interpretable time-series classification on few-shot samples. In: Intern. Joint Conference on Neural Networks (IJCNN) (2020)
15. UCI Machine Learning Repository: PM2.5 Data of Five Chinese Cities. https://archive.ics.uci.edu/dataset/394/ (2017)
16. Wang, Z., Koprinska, I., Troncoso, A., Martínez-Álvarez, F.: Static and dynamic ensembles of neural networks for solar power forecasting. In: International Joint Conference on Neural Networks (IJCNN) (2018)
17. Yahoo Finance: Yahoo Finance. https://finance.yahoo.com/ (2024)
18. Zhang, J., Cui, S., Xu, Y., Li, Q., Li, T.: A novel data-driven stock price trend prediction system. Expert Systems with Applications **97**, 60–69 (2018)