Iranian Journal of Fuzzy Systems

Volume 23, Number 4, (2026), pp. 1-20

Original paper

University of Sistan and Baluchestan

Integrating fuzzy logic into transformer-based models for long-term multivariate time series forecasting: A novel approach to fuzzy positional encoding

M. Ahmadipor [1], M. Saeed [2] and M. Eftekhari [3]

[1,2,3] *Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran*

Mitraahmadi@eng.uk.ac.ir, msaeedmz@uk.ac.ir, m.eftekhari@uk.ac.ir

Abstract

Long-term multivariate time series forecasting is one of the most challenging problems in machine learning. Among the proposed solutions, deep learning networks, particularly transformer-based models, have demonstrated superior performance. However, these models are vulnerable to noise, uncertainty, and abrupt changes, and often lack interpretability. To address these limitations, this study introduces a novel hybrid architecture called FuzzyPE-KAN, which integrates fuzzy logic into the transformer framework. The proposed architecture incorporates: (1) a learnable Gaussian noise-based fuzzy attention mechanism that enhances robustness against noise; (2) a learnable fuzzy positional encoding relying on Gaussian membership functions and multilayer perceptrons to effectively model the inherently vague and graded nature of time; and (3) complete replacement of feed-forward layers with Kolmogorov-Arnold Networks to dramatically reduce the number of parameters and improve interpretability. The proposed architecture was applied to five state-of-the-art baseline models (Transformer, Informer, PatchTST, Crossformer, and iTransformer) and evaluated on eight standard benchmark datasets (ETTh1/2, ETTm1/2, Weather, Electricity, Traffic, and Exchange Rate). Results indicate that the proposed variants achieve an average improvement of 26-49% in Mean Squared Error and 17-29% in Mean Absolute Error across most scenarios compared to the baseline models. The most substantial gains were observed on the Exchange Rate dataset (78% improvement), Weather (71.28%), ETTh2 (76.41%), and ETTm2 (78.66%). This research demonstrates that the simultaneous integration of fuzzy logic and Kolmogorov-Arnold Networks within a transformer architecture not only enhances accuracy and robustness but also elevates model interpretability to a significant level, paving the way for real-world applications in the finance, energy, and healthcare domains.

Keywords: Multivariate time series forecasting, fuzzy logic, fuzzy positional encoding, transformer, Kolmogorov-Arnold networks.

1 Introduction

Multivariate time series, which sequentially track multiple variables over time, pose significant forecasting challenges [39]. Long-term forecasting is particularly demanding due to random noise, complex nonlinear patterns, temporal and cross-variable dependencies, sudden structural changes, and inherent uncertainties [9]. These forecasts have critical applications in finance, energy, healthcare, transportation, and meteorology, supporting strategic decision-making, cost reduction, and risk mitigation [25].

Despite significant advances, a deep analysis of existing architectures reveals two fundamental unresolved limitations preventing transformers from reaching their full potential in long-term multivariate time series forecasting.

The first gap concerns positional encoding. Literature on fuzzy positional encoding remains nascent and largely confined to vision transformers and medical diagnostics. Prior works include ViTAR [10] and PathoHR [24] (adding

Corresponding Author: M. Ahmadipor
Received: February 2026; Revised: May 2026; Accepted: June 2026.
https://doi.org/10.22111/ijfs.2026.54871.9725

uniform random noise to position coordinates), AutoLDT [37] (injecting Gaussian noise into positional embeddings), Morlier et al. (2025) [26] (perturbation and basic fuzzification), and FRPC [12] (fuzzy membership for patch relations). All existing methods-fixed sinusoidal or learnable vectors-represent each time instant with a static vector, failing to capture the inherently vague and graded nature of temporal intervals. No leading model has leveraged fuzzy positional encoding to model this uncertainty.

The second gap pertains to the feed-forward layer, which remains a conventional multilayer perceptron (MLP) with limitations in parameter efficiency and interpretability. Kolmogorov-Arnold Networks (KANs) offer a promising alternative [23]. However, prior works are limited: KAT [41] and ViKANformer [32] replace MLPs in ViT with KAN variants lacking hybrid components; Han et al. (2024) [13] and Lingyu et al. (2025) [16] substitute FFNs with adaptive KANs for multivariate time series forecasting; KAN2.0 [22] and ELKT [29] apply KANs in TFT or LSTM-Transformer contexts. None systematically integrates KANs into advanced long-term forecasting models.

More critically, no unified framework simultaneously addresses both gaps by re-engineering attention, positional encoding, and feed-forward layers with fuzzy logic and KAN paradigms.

The main contributions of this paper are summarized as follows:

- We propose a novel fuzzy positional encoding (FuzzyPE) based on Gaussian membership functions and multilayer perceptrons to explicitly model the vague and graded nature of temporal information in long-term multivariate time series.
- We introduce a learnable Gaussian noise-based fuzzy attention mechanism that significantly enhances the robustness of the transformer against noise and uncertainty.
- We completely replace the conventional feed-forward networks with Kolmogorov-Arnold Networks (KANs) to enhance nonlinear expressivity and interpretability while maintaining competitive computational efficiency.
- The proposed FuzzyPE-KAN architecture is systematically integrated into five state-of-the-art transformer-based models (Transformer, Informer, PatchTST, Crossformer, and iTransformer) and evaluated on eight widely-used benchmark datasets.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive review of related works on transformer-based time series forecasting, fuzzy logic applications, and Kolmogorov–Arnold Networks. Section 3 presents the necessary theoretical preliminaries, including the transformer architecture, fuzzy logic, and Kolmogorov-Arnold Networks. Section 4 details the proposed FuzzyPE-KAN architecture, describing the dynamic fuzzy attention mechanism, the novel fuzzy positional encoding module, the replacement of feed-forward layers with hybrid KANs, and a complexity analysis of the proposed components. Section 5 describes the experimental setup, benchmark datasets, evaluation metrics, and implementation details, followed by quantitative results, statistical tests, and visualizations for all baseline models, as well as a comparison with classical time series methods (ARIMA, GARCH, and splines). Finally, Section 6 concludes the paper and outlines directions for future research.

2 Related work

The evolution of forecasting methods began with classical statistical models like ARIMA, VAR, and exponential smoothing, which assumed linearity and stationarity. While adequate for simple, short-term series, they proved ineffective against nonlinear relationships, noise, and complex dependencies [1, 17, 18, 28, 34]. To address these, recurrent neural networks (RNNs) emerged [5]. Variants like LSTM (1997) and GRU (2014) mitigated vanishing gradients through memory gates, improving medium-range accuracy [3, 27]. However, sequential processing, limited parallelization, and O(L) time complexity hindered efficiency for long sequences [38].

Convolutional neural networks (CNNs) have alleviated the limitations of RNN [8]. Architectures such as Temporal Convolutional Networks (TCNs) employ dilated convolutions to capture long-range dependencies without relying on sequential processing, thereby enabling parallelization and faster training [6, 11]. Nonetheless, they struggled with arbitrary dependencies and complex cross-variable relationships in high-dimensional multivariate series [42].

The transformer architecture (2017) revolutionized sequence modeling using self-attention to capture direct dependencies between time points [35]. Its application to time series (2020) outperformed RNNs and CNNs, but quadratic $O(L^2)$ complexity limited long-horizon use [19]. Informer (2021) reduced complexity to O(L log L) via ProbSparse attention and distillation, though weak in cross-variable dependencies [46]. PatchTST (2023) improved multivariate accuracy through patching and channel-independent processing [43]. Crossformer (2023) captured cross-variable dependencies via two-stage attention and segmented embedding, increasing complexity in high dimensions [45]. iTransformer (2024) enhanced robustness via inverted structure (attention over variables) [21].

Despite advances, neural network-based methods inadequately handle qualitative uncertainties, leading to sensitivity to noise and shocks [14]. Fuzzy logic (Zadeh, 1965) addresses this by modeling vagueness through graded membership, offering interpretability and noise management without rigid assumptions [31, 33, 44]. Early fuzzy models like Fuzzy Time Series (1993) and Fuzzy-ARIMA integrated fuzziness with statistics [15, 40]. However, they rely on expert knowledge, limiting scalability for high-dimensional data [2].

Research then shifted to neuro-fuzzy hybrids, combining fuzzy interpretability with neural pattern learning [36]. ANFIS achieved success in short-term noisy forecasting [15]. More recently, fuzzy integration into transformers has included replacing softmax with fuzzy functions for noise-resistant attention [30] and dynamic fuzzy attention (FANTF, 2025) for improved accuracy and robustness [7].

Table 1: Comparison of transformer-based models for long-term multivariate time series forecasting.

Model	**Year**	**Key Innovation**	**Computational Complexity**	**Strengths and Limitations**
Transformer	2020	Standard scaled dot-product self-attention	$O(L^2 d)$	Powerful long-range dependency modeling High memory and computation cost
Informer	2021	ProbSparse Attention + Distillation mechanism	$O(L \log L)$	Efficient for very long sequences Relatively weak in modeling cross-variate dependencies
PatchTST	2023	Patching mechanism + Channel-independent processing	$O(P^2 \times N_{patch})$	High accuracy with low resource usage Sensitive to patch size choice
Crossformer	2023	Two-stage attention + Dimension-Segment-Wise (DSW) embedding	$\approx O(L \times D)$	Strong cross-variate and multi-scale modeling Higher complexity in very high-dimensional data
iTransformer	2024	Inverted attention (attention over variables instead of time steps)	$O(C^2 \times L)$	Excellent robustness and accuracy in multivariate settings Requires careful hyperparameter tuning

L: sequence length, d: model dimension, P: patch size, N_{patch}: number of patches, D or C: number of variates (channels). Complexities are reported approximately per layer.

3 Preliminaries

In this section, we present the essential theoretical foundations for understanding the proposed FuzzyPE-KAN method. These foundations include a focused overview of the core transformer architecture and its key components relevant to our modifications, an introduction to fuzzy logic, and the emerging concept of Kolmogorov–Arnold Networks.

3.1 Transformer architecture

The transformer architecture revolutionized sequence modeling by completely eliminating recurrent structures and relying solely on attention mechanisms. This design significantly enhanced parallel processing capabilities and improved the capture of long-range dependencies. The model typically employs an encoder-decoder structure, with each component consisting of stacked identical layers connected via residual connections and layer normalization. At the heart of the transformer lies the self-attention mechanism, which assesses the relative importance of data points within a sequence. The standard formulation, known as scaled dot-product attention, is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where Q, K, and V are the query, key, and value matrices, respectively, each with dimensionality d_k. The scaling factor $\sqrt{d_k}$ helps prevent gradient vanishing in high-dimensional settings [35]. To enhance expressive power, multi-head attention (typically $h = 8$) is employed, performing attention operations in parallel across different projection subspaces and concatenating the results. Since the attention mechanism is inherently permutation-invariant and lacks inherent sequential order information, positional encoding is added to the inputs [35]. In the original implementation, sinusoidal

positional encoding is used:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \tag{2}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \tag{3}$$

which effectively models relative positional relationships and facilitates generalization to longer sequences. Following the attention block in each layer, a position-wise feed-forward network (FFN) is applied. This consists of a simple two-layer multilayer perceptron (MLP) with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2. \tag{4}$$

This block introduces additional non-linearity but, as noted in previous literature, suffers from limitations in interpretability and computational efficiency due to its large parameter count and black-box nature. These limitations motivate its replacement in our proposed architecture.

3.2 Fuzzy logic

Fuzzy logic contrasts with binary logic by modeling vague and qualitative concepts through degrees of membership within the interval [0,1]. This paradigm primarily employs membership functions (e.g., triangular, trapezoidal, or Gaussian) to effectively manage uncertainties stemming from noise, abrupt changes, and imprecise relationships, while offering substantial interpretability. A fuzzy set extends the classical set by assigning each element a degree of membership, as determined by its membership function. This approach allows for the representation and processing of inherent uncertainties, which is crucial for robust time series analysis.

3.3 Kolmogorov-Arnold networks

Kolmogorov–Arnold Networks (KANs), introduced in 2024, serve as an elegant and powerful alternative to traditional multilayer perceptrons. Grounded in the Kolmogorov–Arnold theorem, KANs approximate multivariate functions via compositions of univariate functions. In contrast to MLPs, which feature fixed activation functions at nodes, KANs position learnable activation functions (such as B-splines) along edges. This novel design yields superior accuracy and enables visual interpretability through the analysis of learned functions, rendering it an apt substitute for the feed-forward blocks in transformer architectures.

4 Proposed method

This section comprehensively describes the proposed FuzzyPE-KAN architecture. This architecture addresses existing challenges in standard model frameworks through the intelligent integration of three key components: (1) the integration of a dynamic fuzzy attention mechanism (adapted from [7]) that incorporates controlled Gaussian noise to model uncertainties in token relationships; (2) the introduction of a learnable fuzzy positional encoding module to represent positional information in a graded and qualitative manner; and (3) the complete replacement of feed-forward layers with Kolmogorov–Arnold Networks to enhance expressiveness and improve interpretability.

These modifications yield a unified hybrid framework named FuzzyPE-KAN that enhances accuracy, noise robustness, and explainability, while intelligently modeling temporal uncertainties and elevating the capacity for nonlinear feature transformations, without altering the underlying transformer workflow. Each module described in detail in the following subsections.

4.1 Dynamic fuzzy attention mechanism

In standard transformers, the attention mechanism relies on deterministic score computations, modeling relationships between tokens without explicitly accounting for inherent uncertainty or noise. This limitation severely constrains performance on real-world time series data, which are frequently noisy, subject to sudden fluctuations, and characterized by qualitative uncertainties. Consequently, the model may assign disproportionate attention to outliers or noisy observations.

To mitigate this limitation, the fuzzy attention module proposed by Liu et al. (2025) in the FANTF paper [7] is employed. This module facilitates dynamic uncertainty modeling by adding learnable Gaussian noise to the attention score matrix. In contrast to static fuzzy methods (e.g., those using Gaussian or scaled sigmoid membership functions with fixed predefined parameters), the noise variance σ is here a learnable parameter optimized during training. Figure 1

shows the evolution of σ for all five backbones (from top to bottom: iTransformer, Transformer, Informer, PatchTST, Crossformer). In all cases, σ decreases rapidly in the initial epochs and converges to a stable value, indicating that the model learns to reduce uncertainty as it fits the training data.

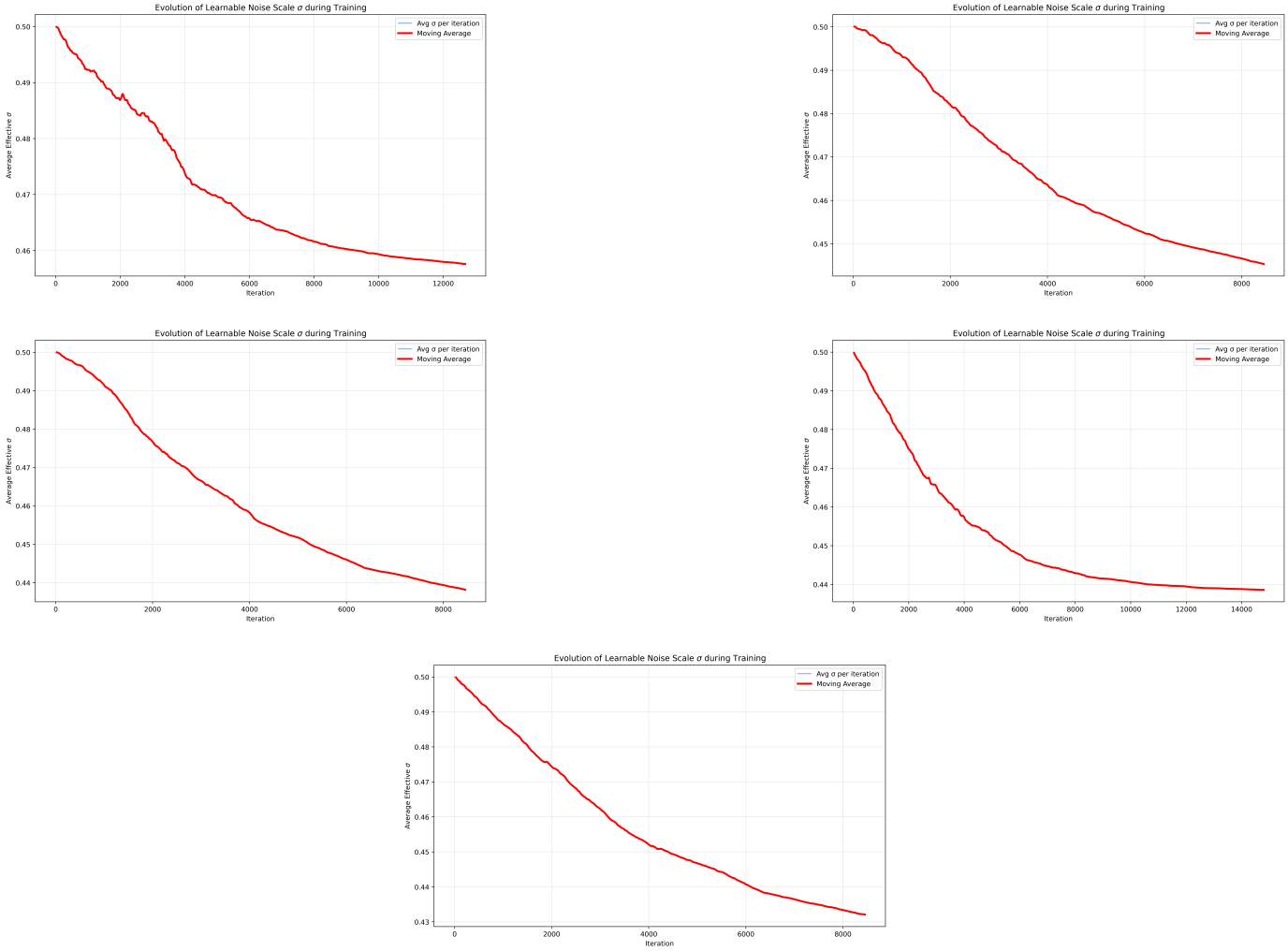

Figure 1: Evolution of σ during training for five backbone models on ETTh1.

The module operates in the following step-by-step manner: Linear projection of inputs: The input sequence X is mapped to query (Q), key (K), and value (V) matrices:

$$Q = XW^Q, \tag{5}$$

$$K = XW^K, \tag{6}$$

$$V = XW^V, \tag{7}$$

where W^Q, W^K, and W^V are learnable weight matrices.

Fuzzy attention score computation: The standard attention score is first calculated, then learnable Gaussian noise is added:

$$\text{Score} = \frac{QK^T}{\sqrt{d_k}} + \sigma \cdot \mathcal{N}(0,1), \tag{8}$$

where $\sigma \in \mathbb{R}^+$ is a learnable parameter, and $\mathcal{N}(0,1)$ denotes standard Gaussian noise with zero mean and unit variance.

Normalization and output computation: The fuzzy scores are normalized using the softmax function, then multiplied by the value matrix, yielding the attention output:

$$\text{Attention} = \text{softmax}(\text{Score})\, V. \tag{9}$$

Multi-head attention: The operations detailed above are performed in parallel across H attention heads. The individual head outputs are then concatenated and linearly projected:

$$\text{MultiHead} = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\, W^O. \tag{10}$$

This mechanism transforms attention from a deterministic process into a fuzzy one, enabling the model to dynamically adjust uncertainty levels in accordance with data characteristics. The parameter σ is updated via backpropagation, granting the model adaptability to varying noise levels across different datasets.

The advantages of this module include enhanced prediction accuracy, and greater robustness, greater interpretability (as the learned σ indicates the degree of uncertainty in attention distributions), and negligible additional computational cost, preserving the $O(L^2D)$ complexity of standard attention (where L is sequence length and D is model dimension).

4.2 Fuzzy positional encoding module

In standard transformer models, positional encoding is typically implemented using fixed sinusoidal functions or learnable vectors. While effective, these approaches represent positions in a rigid, deterministic manner and fail to capture the inherently vague, graded, and qualitative nature of time in real-world scenarios.

To overcome this fundamental limitation, a novel fuzzy positional encoding module is proposed. Rather than a static additive component, our proposed module acts as an adaptive framework.

The proposed module modulates the standard positional encoding with learnable fuzzy weights. A small multilayer perceptron is employed to generate parameters for Gaussian fuzzy sets, ultimately learning softer positional relationships that are more robust to noise and sensitive to qualitative uncertainties. The module operates step-by-step as follows:

Position normalization: Each position *pos* (e.g., time step t in a time series) is normalized to the range [0,1] to achieve sequence-length independence:

$$pos_{norm} = \frac{pos - pos_{min}}{pos_{max} - pos_{min}}. \tag{11}$$

Fuzzy parameter generation: A small multilayer perceptron (with output dimension $2K$) produces centers (μ_k) and widths (σ_k) for K Gaussian fuzzy sets for each normalized position:

$$(\mu_1, \sigma_1, \ldots, \mu_K, \sigma_K) = \text{MLP}(pos_{norm}). \tag{12}$$

Membership degree computation: The membership degree of the position in each fuzzy set is calculated using the Gaussian function:

$$\mu_k(pos) = \exp\left(-\frac{(pos_{norm} - \mu_k)^2}{2\sigma_k^2}\right). \tag{13}$$

Fuzzy weight extraction: A scalar fuzzy weight (in the range [0,1]) is obtained by averaging the membership degrees:

$$w = \frac{1}{K}\sum_{k=1}^{K} \mu_k(pos). \tag{14}$$

Positional encoding modulation: The fuzzy weight is applied to the standard positional encoding as follows:

$$PE_{fuzzy} = (1 + \alpha \cdot w) \cdot PE_{standard}, \tag{15}$$

where α is a learnable scalar parameter. It is initialized to 1.0 for the standard transformer and Informer models, to 3.0 for the PatchTST variant, and to 2.0 for the Crossformer segment-based variant. The parameter α is optimized during training through backpropagation.

The fuzzy weight w (computed via (14)) is a scalar value for each time step (or patch/segment). This scalar weight is then broadcast across the entire embedding dimension d_{model}. This design ensures minimal computational overhead while allowing the model to dynamically modulate the positional influence of each temporal unit according to its fuzzy membership strength. Finally, PE_{fuzzy} is added to the input embeddings.

In Transformer and Informer (which process full sequences), the module is directly applied to the embedding vectors, modulating the standard sinusoidal encoding with fuzzy weights for each time step.

In PatchTST (patch-based tokenization), the fuzzy weight is generated as a one-dimensional vector and aligned with the patching structure to perform modulation at the patch level.

In Crossformer (segment-based processing), fuzzy encodings are computed per segment (rather than per time point) and matched to the input dimensions to target inter-segment relationships.

In iTransformer, the module was not applied, as its inverted structure focuses attention on cross-variable relationships and eliminates explicit reliance on temporal order; introducing any form of positional encoding (even fuzzy) could

interfere with the model's core philosophy.

This modular and adaptive design represents a key strength of the study, demonstrating that a single concept (fuzzy positional modulation) can be effectively and consistently implemented across varied architectures through a deep understanding of their differences. No prior work has proposed the exact approach presented here—namely, generating learnable Gaussian fuzzy sets via a multilayer perceptron, averaging membership degrees to derive a scalar weight, and directly modulating the standard encoding without altering the underlying structure.

The module's advantages include soft and continuous positional weighting, reduced sensitivity to noise, strong generalization to long and noisy series, and capturing temporal ambiguity, all at minimal computational cost.

4.3 Replacement of the feed-forward block with Kolmogorov-Arnold networks

The proposed approach fully substitutes the feed-forward block with a Kolmogorov–Arnold Network (KAN). Rooted in the Kolmogorov–Arnold theorem, this network diverges from MLPs—where fixed activations reside on nodes—by instead positioning learnable univariate functions (parameterized via B-splines) along edges. The hybrid variant introduced here incorporates a straightforward linear pathway with SiLU activation to strike a balance between nonlinear expressivity and training stability.

In our implementation, we use B-splines of order 3 (`spline_order = 3`) with a grid size of 5 (`grid_size = 5`). This configuration results in $G + k = 8$ learnable coefficients per spline function, where G is the grid size and k is the spline order. The KAN layer operates in the following step-by-step manner:

Input normalization: The input x is scaled to the interval [0, grid_size] (default value: 5) to ensure scale-invariant basis computations. B_spline basis computation: For each input element, spline bases of degree 3 (spline_order = 3) are generated, forming a matrix of localized curvature functions. Application of spline weights: These bases are multiplied by learnable spline weights, producing the nonlinear component y_{spline}. Hybrid component: A parallel linear path, incorporating SiLU activation and a base weight, is computed to yield y_{base}. Layer output: The final output is the sum of the two components:

$$y = y_{\text{spline}} + y_{\text{base}}. \tag{16}$$

The overarching formulation of the proposed layer is:

$$y = \sum_{i,j} \phi_{i,j}(x_i) \cdot w_{i,j} + \text{SiLU}(x \cdot W_{\text{base}}), \tag{17}$$

where $\phi_{i,j}$ denote the learnable B-spline functions positioned on the edges.

This architecture reimagines the FFN as an intelligent operator. Although the hybrid KAN layer may have a similar or larger parameter count than an MLP for the same hidden dimension, its unique spline-based structure provides superior approximation capacity and interpretability per parameter. This allows for more aggressive dimensionality reduction, potentially leading to smaller overall models while maintaining or improving accuracy.

4.4 FuzzyPE-KAN

The FuzzyPE-KAN architecture seamlessly integrates the three proposed modules into the standard transformer framework, strategically enhancing its core blocks. The data processing pipeline is outlined as follows:

Input Embedding: The multivariate time series is initially projected into a fixed-dimensional feature space through a linear layer.

Fuzzy Positional Encoding: The proposed module (detailed in the previous section) is added to the embeddings to model temporal ordering while incorporating its inherently graded and qualitative aspects.

Attention Layers: The dynamic fuzzy attention mechanism (featuring learnable Gaussian noise) is applied to capture temporal and cross-variable dependencies, accounting for inherent uncertainties.

KAN-Based Feed-Forward Layer: Features are processed via the Kolmogorov–Arnold Network to achieve a more efficient, parameter-frugal, and interpretable nonlinear transformation.

Normalization and Residual Connections: The output of each sublayer is added to its input, followed by a layer normalization, to maintain stable gradient propagation.

Final Output: After passing through N modified transformer blocks, a linear layer maps the representations to the target prediction dimensions (forecast horizon and number of variables).

This modular design serves as an extension applicable to the baseline models (Informer, PatchTST, Crossformer, and iTransformer), with the primary variation being the adaptation of the fuzzy positional encoding module to align with each model's tokenization scheme (as previously described).

Figure 2 illustrates the overall data flow of the proposed FuzzyPE-KAN architecture. The pipeline proceeds as follows: (1) the raw multivariate time series is input, (2) it undergoes initial embedding and linear projection to the feature space, (3) fuzzy positional encoding is added to incorporate graded temporal information, (4) the sequence passes through N modified transformer blocks where dynamic fuzzy attention captures dependencies while explicitly modeling uncertainty, (5) the KAN-based feed-forward layers perform efficient and interpretable nonlinear transformations, and (6) a final linear layer maps the representations to the target forecast horizon and variables. The two primary innovations—learnable fuzzy positional encoding and replacement of feed-forward layers with hybrid KANs—are introduced at steps 3 and 5, respectively. Figure 3 provides a schematic overview of the complete FuzzyPE-KAN architecture, clearly showing the unified integration of the dynamic fuzzy attention mechanism, fuzzy positional encoding module, and KAN feed-forward blocks within the standard transformer structure. This cohesive and modular design simultaneously improves forecasting accuracy, robustness to noise, computational efficiency, and model interpretability, all while preserving the core transformer workflow and ensuring broad compatibility with existing architectures.

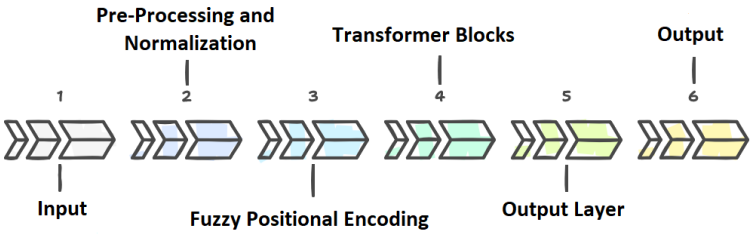

Figure 2: Overall data flow of the proposed FuzzyPE-KAN architecture, illustrating the step-by-step processing pipeline from raw input to final forecast (steps 1–6).

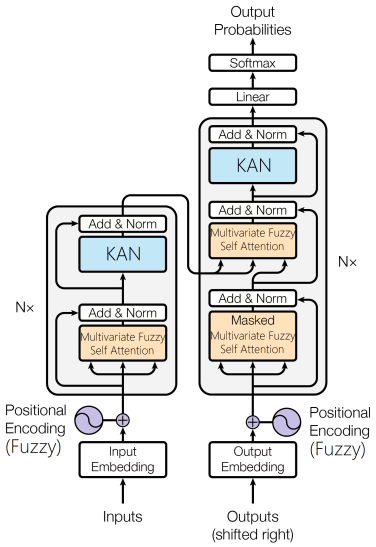

Figure 3: Schematic diagram of the FuzzyPE-KAN architecture, highlighting the integration of dynamic fuzzy attention, learnable fuzzy positional encoding, and hybrid KAN feed-forward blocks within the transformer structure.

4.5 Complexity analysis

To address the claim of negligible computational overhead and provide a transparent comparison, we analyze both theoretical complexity and empirical training time of the proposed FuzzyPE-KAN architecture.

The dominant computational cost in transformer-based models remains the self-attention mechanism with $O(L^2 d)$ complexity per layer, which is unchanged in our approach. The fuzzy positional encoding module introduces a small MLP with linear complexity $O(L \cdot d_{\text{MLP}})$, which is negligible compared to attention.

The main modification lies in the feed-forward network. We replace the standard MLP with a hybrid KAN layer. Table 2 compares the feed-forward blocks.

To provide a more concrete measure of computational cost, we estimate the floating point operations (FLOPs) per

Table 2: Comparison of the feed-forward block between standard MLP and hybrid KAN ($d_{\text{model}} = 128$, $d_{\text{ff}} = 256$).

Component	**Parameters**	**Complexity (per layer)**
Standard MLP (FFN)	65,536	$O(L \cdot d \cdot 2d_{\text{ff}})$
Hybrid KAN (ours)	393,216	$O(L \cdot d \cdot 15)$

layer for a sequence length of $L = 96$. For the standard MLP, the computational cost is approximately:

$$\text{FLOPs}_{\text{MLP}} = 2 \times L \times d_{\text{model}} \times d_{\text{ff}} \approx 2 \times 96 \times 128 \times 256 = 6.29 \times 10^6.$$

For the hybrid KAN, the cost includes spline basis evaluation and linear projection:

$$\text{FLOPs}_{\text{KAN}} \approx L \times d_{\text{model}} \times (d_{\text{ff}} \times (G + k) \times 2 + d_{\text{ff}}) \approx 96 \times 128 \times (256 \times 8 \times 2 + 256) \approx 15.7 \times 10^6.$$

Thus, the proposed KAN layer incurs approximately 2.5× higher computational cost than the standard MLP for this configuration, which is offset by the significant gains in forecasting accuracy reported in Section 5.

As discussed in Section 4.3, the hybrid KAN offers superior approximation capacity and interpretability per parameter despite its larger parameter count.

From the empirical perspective, Table 3 reports the average training time per epoch on the Exchange Rate dataset using the Transformer backbone.

Table 3: Empirical training time per epoch (seconds) on Exchange Rate dataset (Transformer backbone).

Model	**Time per epoch (s)**
Transformer (baseline)	56.5
+ Fuzzy PE only	55.5
+ KAN only	72.5
+ FuzzyPE-KAN (proposed)	71.0

The integration of the hybrid KAN increases the training time by approximately 26%, primarily due to the evaluation of B-spline basis functions. However, the fuzzy positional encoding module adds almost no computational overhead. Given the substantial improvements in forecasting accuracy (up to 78% reduction in MSE on some datasets), this moderate increase in computational cost is considered acceptable.

5 Experimental results

The proposed FuzzyPE-KAN framework was evaluated on eight standard multivariate long-term time series forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, and Exchange Rate. These real-world datasets span energy, meteorology, traffic, and finance domains and feature natural noise, complex seasonality, and abrupt changes that test model robustness. Performance was measured using Mean Squared Error (MSE), sensitive to large errors and suitable for peak-sensitive applications, and Mean Absolute Error (MAE), robust and interpretable on the original scale.

The framework was applied to five state-of-the-art transformer baselines: Transformer, Informer, PatchTST, Crossformer, and iTransformer. Experiments used sequential non-overlapping splits to prevent leakage, input/prediction length of 96 steps, MSE loss, Adam optimizer (lr=0.0001), batch size 4 (due to GPU memory limits), max 100 epochs, and early stopping (patience=3). Implementation was in PyTorch on two NVIDIA Tesla T4 GPUs with mixed-precision and DataParallel. Final metrics are average MSE/MAE across all variables and test samples.

Statistical significance was assessed with the non-parametric Friedman test on MAE ranks. When significant ($p < 0.05$), post-hoc Li test identified pairwise differences [4].

Results are presented separately for each baseline model. For every baseline, we report the MSE and MAE tables (with percentage improvements), Friedman test mean ranks, post-hoc Li test results, and visualizations including scatter plots (with R^2 and adjusted R^2), error heatmaps, and actual-vs-predicted time series comparisons.

The ablation study includes four variants. The first variant incorporates only the dynamic fuzzy attention mechanism from (FANTF, [7]). Adding the learnable fuzzy positional encoding produces the Fuzzy PE variant. Replacing the feed-forward layers with Kolmogorov–Arnold Networks produces the KAN variant. The full integration of all three components constitutes the proposed FuzzyPE-KAN. This stepwise design allows transparent assessment of individual and combined effects.

We also compare the proposed variants with classical methods (ARIMA, GARCH, and splines) in Subsection 5.6.

5.1 Transformer model

Table 4 presents the MSE and MAE values for the baseline Transformer model and its enhanced variants. FuzzyPE-KAN yields average improvements of 38% in MSE and 25% in MAE across all datasets. KAN achieves > 60% gains on ETTh2 and ETTm2, demonstrating its enhanced nonlinear capacity. Fuzzy PE reaches up to 89.58% improvement on ETTm2, highlighting its robustness to seasonal patterns. Degradations on Electricity and Traffic are attributed to the curse of dimensionality, noisy gradients from small batch size, and potential overfitting of the Fuzzy PE MLP in high-dimensional settings.

Table 4: Results of the Proposed Modules on the Transformer Model

Dataset	**Metric**	**Value**				**Improvement (%) vs. Baseline**		
		Baseline Transformer	Fuzzy PE	KAN	FuzzyPE-KAN	Fuzzy PE	KAN	FuzzyPE-KAN
ETTh1	MSE	0.96	0.59	0.89	0.89	38.28	6.79	6.86
	MAE	0.78	0.57	0.76	0.77	27.42	3.05	1.15
ETTh2	MSE	4.49	1.42	1.70	2.19	68.35	62.18	51.31
	MAE	1.68	0.93	1.05	1.17	44.89	37.24	30.33
ETTm1	MSE	0.95	0.43	0.62	0.56	54.53	35.12	41.39
	MAE	0.74	0.46	0.57	0.52	38.19	23.13	29.79
ETTm2	MSE	1.51	0.16	0.51	0.32	89.58	66.30	78.66
	MAE	0.88	0.24	0.53	0.43	72.69	39.00	51.29
Weather	MSE	0.61	0.16	0.35	0.19	74.05	41.28	68.73
	MAE	0.56	0.24	0.40	0.26	57.10	27.35	53.24
Electricity	MSE	0.27	0.29	0.26	0.27	−7.92	1.31	−2.47
	MAE	0.36	0.38	0.35	0.35	−3.78	2.53	2.09
Traffic	MSE	0.67	0.66	0.67	0.66	1.07	0.00	1.47
	MAE	0.36	0.37	0.36	0.36	−0.55	1.49	2.08
ExchangeRate	MSE	1.35	0.71	0.87	0.61	47.34	35.62	55.11
	MAE	0.89	0.67	0.76	0.62	24.39	13.81	29.84

Friedman test ranks (Table 5) place Fuzzy PE first (1.875), followed by FuzzyPE-KAN (2.0625), KAN (2.4375), and baseline last (3.625). The post-hoc Li test (Table 6) confirms that the baseline is significantly worse than all variants ($p < 0.05$), while no significant differences exist among the enhanced variants ($p > 0.05$).

Table 5: Mean Ranking of Transformer Models Based on the Friedman Test

Model	**Mean Rank**
Baseline	3.625
Fuzzy PE	1.875
KAN	2.4375
FuzzyPE-KAN	2.0625

Table 6: Post-Hoc Comparison Results for Transformer Algorithms

Model	$Z = (\textbf{Rank Difference})/\textbf{SE}$	p**-value**	Li
Baseline	2.711 088	0.006 706	0.012 029
KAN	0.871 421	0.383 524	0.012 029
FuzzyPE-KAN	0.290 474	0.771 454	0.050 000

Visualizations on the Weather dataset (representative for complex multivariate cases) provide deeper insight:

The scatter plot (Figure 4) shows tight clustering around the $y = x$ line with adjusted $R^2 = 0.7274$, indicating that the model explains approximately 73% of the variance.

The error heatmap (Figure 5) exhibits no recurring vertical bands across the prediction horizon, confirming the absence of systematic seasonal or periodic errors. Rows in yellow, orange, or light green indicate near-zero errors for variables with regular patterns; green-dominated rows suggest systematic negative bias (consistent underestimation); and red or orange rows indicate positive bias (consistent overestimation).

The actual-vs-predicted plot (Figure 6) demonstrates close alignment between predicted and actual curves, particularly during rapid changes and peaks. The model adopts a conservative approach at extremes—slightly underestimating high peaks and overestimating deep troughs—which mitigates the impact of outliers.

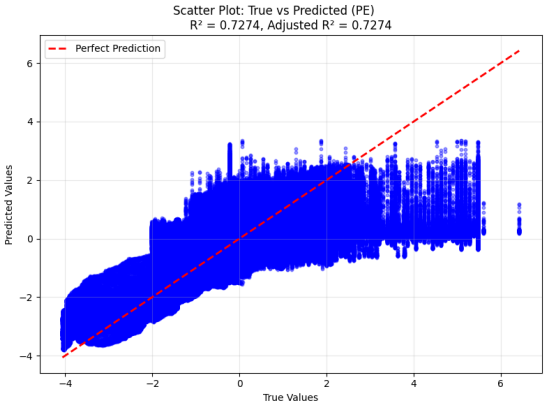

Figure 4: Scatter plot for Fuzzy PE on Weather (adj. $R^2 = 0.7274$).

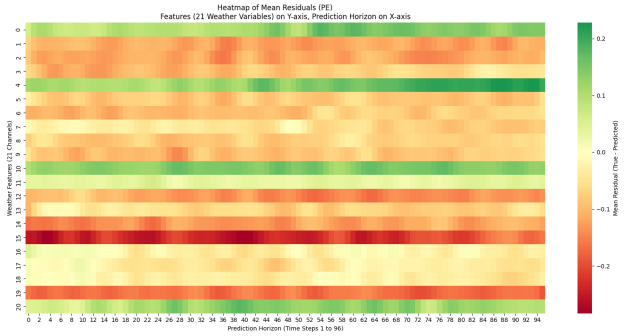

Figure 5: Error heatmap for Fuzzy PE on Weather.

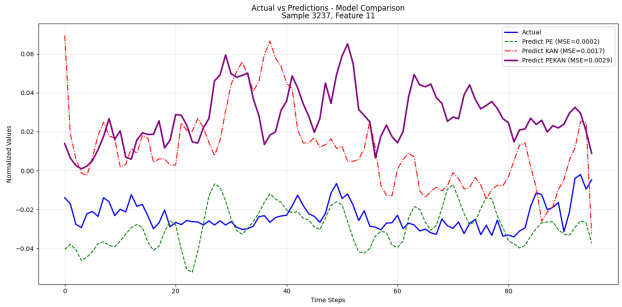

Figure 6: Actual vs. predicted time series for Fuzzy PE on Weather.

5.2 Informer model

Table 7 presents the results. FuzzyPE-KAN achieves the highest average improvement (42% MSE, 29% MAE). The largest reductions occur on Weather (71.28% MSE) and ETTm2 (72.15% MSE). Similar to the Transformer model, limited improvements on Traffic and Electricity reflect challenges with high-dimensional data.

Friedman test ranks (Table 8) rank FuzzyPE-KAN first (1.75), followed by Fuzzy PE (1.9375), KAN (2.5625), and baseline last (3.75). The post-hoc Li test (Table 9) confirms the baseline is significantly worse than all variants ($p < 0.05$), with no significant differences among the enhanced models.

Visualizations on the Weather dataset show patterns consistent with the Transformer model. The scatter plot(Figure 7) yields $R^2 = 0.6843$, and the heatmap(Figure 8) confirms no systematic periodic errors. The actual-vs-predicted plot(Figure 9) demonstrates strong alignment with the true series, accurately capturing overall trends and local fluctuations.

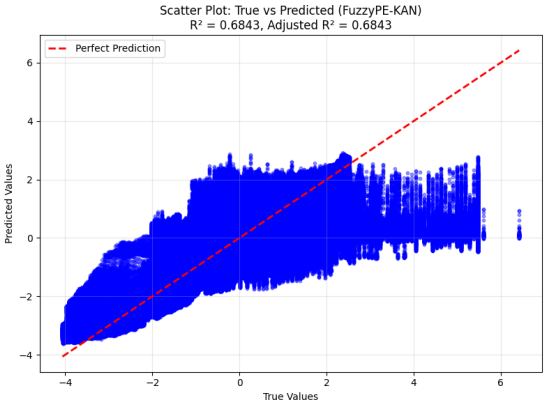

Figure 7: Scatter plot for FuzzyPE-KAN on Weather (adj. $R^2 = 0.6843$).

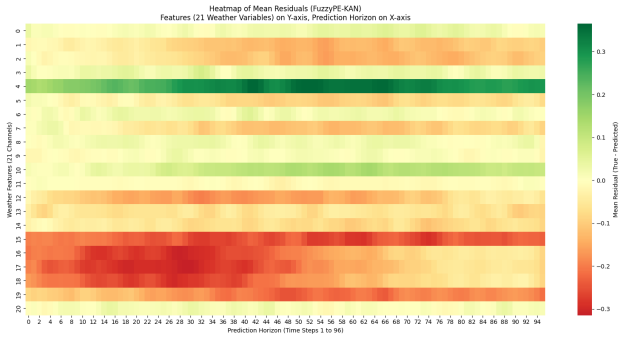

Figure 8: Error heatmap for FuzzyPE-KAN on Weather.

Table 7: Results of the Proposed Modules on the Informer Model

Dataset	Metric	Value				Improvement (%) vs. Baseline		
		Baseline Informer	Fuzzy PE	KAN	FuzzyPE-KAN	Fuzzy PE	KAN	FuzzyPE-KAN
ETTh1	MSE	0.96	0.65	0.88	0.64	32.27	8.40	33.21
	MAE	0.78	0.61	0.75	0.59	21.48	4.32	24.00
ETTh2	MSE	3.25	1.29	1.37	1.90	60.30	57.72	41.39
	MAE	1.44	0.91	0.94	1.04	36.79	34.64	27.79
ETTm1	MSE	0.85	0.46	0.60	0.44	45.74	30.30	48.46
	MAE	0.70	0.46	0.58	0.46	34.48	18.08	34.64
ETTm2	MSE	1.55	0.40	0.41	0.43	74.16	73.87	72.15
	MAE	0.87	0.46	0.47	0.49	47.26	45.99	44.00
Weather	MSE	0.63	0.17	0.30	0.18	73.11	51.90	71.28
	MAE	0.55	0.26	0.37	0.26	53.47	32.70	53.37
Electricity	MSE	0.28	0.27	0.28	0.28	2.17	−0.90	−1.64
	MAE	0.37	0.36	0.36	0.36	2.75	2.59	3.80
Traffic	MSE	0.66	0.66	0.68	0.65	−0.74	−3.70	1.82
	MAE	0.36	0.37	0.37	0.35	−3.10	−3.47	1.29
ExchangeRate	MSE	1.44	0.68	0.67	0.51	52.96	53.15	64.54
	MAE	0.93	0.66	0.64	0.58	28.98	31.58	37.18

Table 8: Mean Ranking of Informer Models Based on the Friedman Test

Model	Mean Rank
Baseline	3.75
Fuzzy PE	1.9375
KAN	2.5625
FuzzyPE-KAN	1.75

Table 9: Post-Hoc Comparison Results for Informer Algorithms

Model	$Z = (\textbf{Rank Difference})/\textbf{SE}$	p-value	Li
Baseline	3.098 387	0.001 946	0.012 029
KAN	1.258 470	0.208 132	0.012 029
FuzzyPE	0.290 474	0.771 454	0.050 000

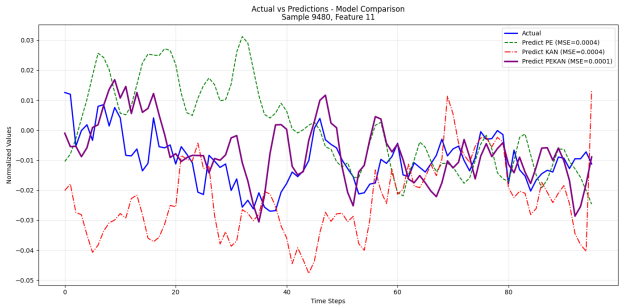

Figure 9: Actual vs. predicted time series for FuzzyPE-KAN on Weather.

5.3 PatchTST model

Table 10 presents the results. Improvements are consistent across all datasets, with the highest reductions on Exchange Rate (78% MSE). Unlike the previous two models, PatchTST shows substantial improvements on Electricity and Traffic.

Friedman test ranks (Table 11) rank FuzzyPE-KAN first (1.75), followed by KAN (1.9375), Fuzzy PE (2.3125), and baseline last (4.00). The post-hoc Li test (Table 12) confirms that all variants significantly outperform the baseline.

Visualizations on the Exchange Rate dataset(Figure 10) shows an extremely high $R^2 = 0.9513$ in the scatter plot, and the heatmap(Figure 11) reveals distinct vertical bands only in the initial prediction steps, indicating rapid adaptation to shocks. After the midpoint, errors dissipate, confirming the model's strong recovery capability.The actual-vs-predicted plot(Figure 12) exhibits substantial overlap between curves.

Table 10: Results of the Proposed Modules on the PatchTST Model

Dataset	**Metric**	**Value**				**Improvement (%) vs. Baseline**		
		Baseline PatchTST	Fuzzy PE	KAN	FuzzyPE-KAN	Fuzzy PE	KAN	FuzzyPE-KAN
ETTh1	MSE	0.46	0.38	0.39	0.37	17.39	13.51	17.68
	MAE	0.45	0.40	0.40	0.40	12.12	10.25	12.14
ETTh2	MSE	0.39	0.30	0.30	0.29	24.35	23.21	25.66
	MAE	0.42	0.35	0.35	0.34	16.41	15.72	17.70
ETTm1	MSE	0.39	0.32	0.33	0.33	17.62	15.81	16.45
	MAE	0.41	0.36	0.37	0.36	11.07	9.87	10.20
ETTm2	MSE	0.29	0.18	0.18	0.18	39.08	35.22	38.86
	MAE	0.34	0.26	0.26	0.26	22.42	22.35	22.63
Weather	MSE	0.26	0.18	0.17	0.17	31.28	33.81	31.69
	MAE	0.28	0.22	0.21	0.22	21.88	23.07	22.02
Electricity	MSE	0.21	0.18	0.18	0.18	12.33	15.55	14.52
	MAE	0.30	0.27	0.47	0.48	9.94	10.81	11.08
Traffic	MSE	0.54	0.50	0.47	0.48	6.98	12.11	10.89
	MAE	0.35	0.33	0.31	0.31	5.92	12.17	10.06
ExchangeRate	MSE	0.38	0.09	0.09	0.08	76.57	77.64	78.00
	MAE	0.41	0.21	0.20	0.20	50.18	50.87	50.97

Table 11: Mean Ranking of PatchTST Models Based on the Friedman Test

Model	**Mean Rank**
Baseline	4
Fuzzy PE	2.3125
KAN	1.9375
FuzzyPE-KAN	1.75

Table 12: Post-Hoc Comparison Results for PatchTST Algorithms

Model	$Z = (\textbf{Rank Difference})/\textbf{SE}$	p**-value**	Li
Baseline	3.485 685	0.000 491	0.012 029
FuzzyPE	0.871 421	0.383 524	0.012 029
KAN	0.290 474	0.771 454	0.050 000

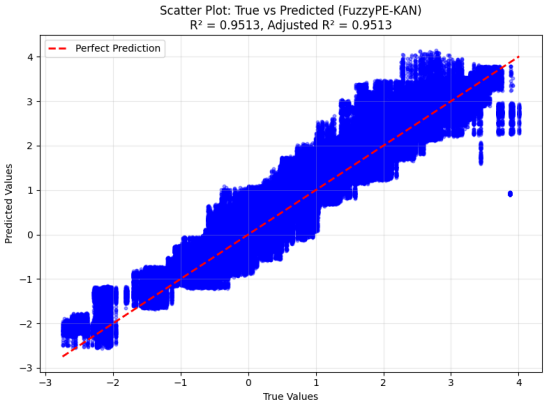

Figure 10: Scatter plot for FuzzyPE-KAN on Exchange Rate (adj. $R^2 = 0.9513$).

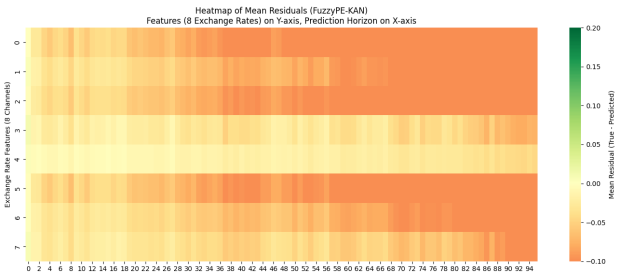

Figure 11: Error heatmap for FuzzyPE-KAN on Exchange Rate.

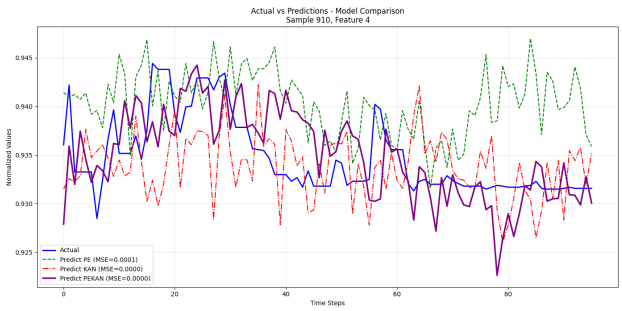

Figure 12: Actual vs. predicted time series for FuzzyPE-KAN on Exchange Rate.

5.4 Crossformer model

Table 13 presents the results. Crossformer shows the largest average improvement among all backbones (45% MSE), with peak reductions of 79.87% on ETTh2. This strong performance is attributable to its effective modeling of long-term dependencies and cross-variable interactions, which synergizes well with proposed modules.

Table 13: Results of the Proposed Modules on the Crossformer Model

Dataset	**Metric**	**Value**				**Improvement (%) vs. Baseline**		
		Baseline Crossformer	Fuzzy PE	KAN	FuzzyPE-KAN	Fuzzy PE	KAN	FuzzyPE-KAN
ETTh1	MSE	0.55	0.41	0.40	0.43	25.76	26.88	22.83
	MAE	0.53	0.44	0.44	0.44	17.81	17.42	16.24
ETTh2	MSE	2.29	0.46	0.62	0.54	79.87	72.96	76.41
	MAE	1.19	0.50	0.58	0.57	57.90	51.01	51.66
ETTm1	MSE	0.66	0.38	0.48	0.44	42.35	27.29	33.60
	MAE	0.60	0.41	0.48	0.46	31.67	19.25	22.96
ETTm2	MSE	1.21	0.28	0.25	0.31	76.74	79.64	74.50
	MAE	0.73	0.37	0.33	0.40	49.23	54.17	45.60
Weather	MSE	0.26	0.16	0.16	0.17	37.97	39.14	35.79
	MAE	0.31	0.23	0.23	0.23	25.36	25.64	25.09
Electricity	MSE	0.18	0.15	0.15	0.15	17.28	18.64	14.82
	MAE	0.28	0.25	0.25	0.26	10.01	12.14	8.32
Traffic	MSE	0.56	0.54	0.53	0.53	4.17	5.08	5.28
	MAE	0.31	0.30	0.29	0.29	2.43	4.20	3.80
ExchangeRate	MSE	0.88	0.26	0.25	0.22	70.61	71.69	75.20
	MAE	0.72	0.39	0.36	0.35	45.77	49.13	5.89

Friedman test ranks (Table 14) rank Fuzzy PE first (1.9375), followed by KAN (2.00), FuzzyPE-KAN (2.0628), and baseline last (4.00). The post-hoc Li test(Table 15) confirms the baseline is significantly worse than all variants.

Table 14: Mean Ranking of Crossformer Models Based on the Friedman Test

Model	**Mean Rank**
Baseline	4
Fuzzy PE	1.9375
KAN	2
FuzzyPE-KAN	2.0628

Table 15: Post-Hoc Comparison Results for Crossformer Algorithms

Model	$Z = (\textbf{Rank Difference})/\textbf{SE}$	p**-value**	Li
Baseline	3.195 211	0.001 397	0.004 060
FuzzyPE-KAN	0.193 649	0.846 451	0.004 060
KAN	0.096 825	0.922 866	0.050 000

Visualizations on the Electricity dataset show a dense scatter plot(Figure 13 with $R^2 = 0.88$ and a heatmap(Figure14) with no recurring vertical bands, confirming accurate daily and weekly pattern capture across 321 variables. The actual-vs-predicted plot(Figure15) shows close tracking of the true series, with minor errors only at extremely sharp peaks.

5.5 iTransformer model

Table 16 presents the results. Note that the fuzzy positional encoding module is incompatible with iTransformer's inverted architecture (which focuses on cross-variable relationships without explicit temporal order). Therefore, Fuzzy PE results match the baseline, and FuzzyPE-KAN results are identical to KAN. The KAN module outperforms the baseline on 7 out of 8 datasets, with average improvements of 27% MSE and 18% MAE. The largest gain occurs on Exchange Rate (75.53% MSE). Slight degradation on Traffic (-1.42% MSE) may be due to additional nonlinearity from KAN disrupting the inverted architecture's balance on high-dimensional noisy data.

Friedman test ranks (Table 17) rank KAN and FuzzyPE-KAN jointly first (1.75), followed by baseline and Fuzzy PE (both 3.25). The post-hoc Li test (Table 18) confirms that KAN and FuzzyPE-KAN significantly outperform the baseline and Fuzzy PE ($p < 0.05$).

Visualizations on the Exchange Rate dataset show that the scatter plot(Figure 16) exhibits extremely tight clustering

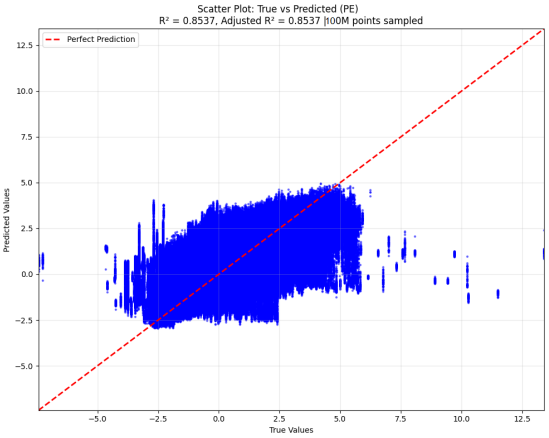

Figure 13: Scatter plot for Fuzzy PE on Electricity (adj. $R^2 = 0.88$).

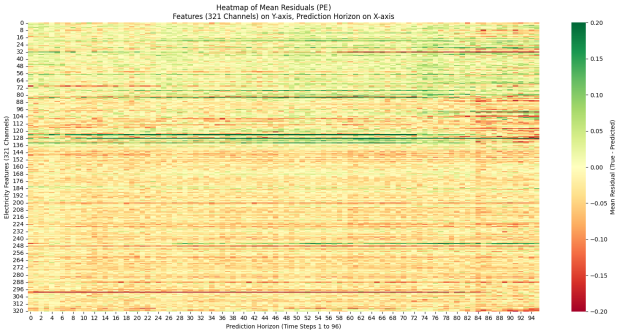

Figure 14: Error heatmap for Fuzzy PE on Electricity.

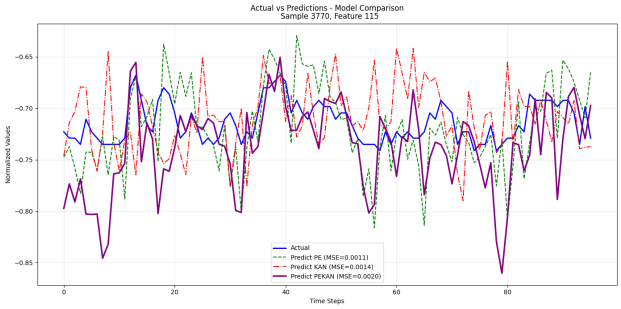

Figure 15: Actual vs. predicted time series for Fuzzy PE on Electricity.

Table 16: Results of the Proposed Modules on the iTransformer Model

Dataset	**Metric**	**Value**				**Improvement (%) vs. Baseline**		
		Baseline iTransformer	Fuzzy PE	KAN	FuzzyPE-KAN	Fuzzy PE	KAN	FuzzyPE-KAN
ETTh1	MSE	0.46	0.46	0.39	0.39	0.00	14.73	14.73
	MAE	0.45	0.45	0.41	0.41	0.00	8.50	8.50
ETTh2	MSE	0.39	0.39	0.30	0.30	0.00	23.06	23.06
	MAE	0.41	0.41	0.35	0.35	0.00	14.14	14.14
ETTm1	MSE	0.41	0.41	0.33	0.33	0.00	18.58	18.58
	MAE	0.41	0.41	0.37	0.37	0.00	9.58	9.58
ETTm2	MSE	0.29	0.29	0.18	0.18	0.00	37.28	37.28
	MAE	0.33	0.33	0.26	0.26	0.00	21.07	21.07
Weather	MSE	0.25	0.25	0.17	0.17	0.00	32.42	32.42
	MAE	0.28	0.28	0.21	0.21	0.00	24.45	24.45
Electricity	MSE	0.18	0.18	0.16	0.16	0.00	11.31	11.31
	MAE	0.27	0.27	0.26	0.26	0.00	6.17	6.17
Traffic	MSE	0.44	0.44	0.45	0.45	0.00	−1.42	−1.42
	MAE	0.29	0.29	0.31	0.31	0.00	−3.87	−3.87
ExchangeRate	MSE	0.36	0.36	0.09	0.09	0.00	75.53	75.53
	MAE	0.40	0.40	0.21	0.21	0.00	48.17	48.17

Table 17: Mean Ranking of iTransformer Models Based on the Friedman Test

Model	**Mean Rank**
Baseline	3.25
Fuzzy PE	3.25
KAN	1.75
FuzzyPE-KAN	1.75

($R^2 = 0.9490$), and the heatmap(Figure17) contains no red or green tones, indicating balanced predictions. Limited vertical patterns appear only in the initial horizon but dissipate rapidly, confirming sustained stability. The actual-vs-

Table 18: Post-Hoc Comparison Results for iTransformer Algorithms

Model	$Z = (\textbf{Rank Difference})/\textbf{SE}$	p**-value**	Li
Baseline	2.323 790	0.020 137	0.000 000
FuzzyPE	2.323 790	0.020 137	0.000 000
KAN	0.000 000	1.000 000	0.050 000

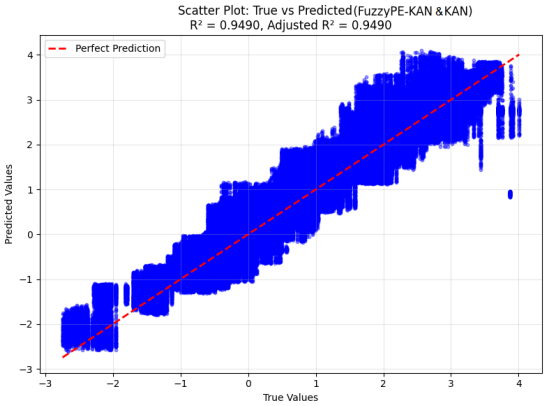

Figure 16: Scatter plot for KAN/FuzzyPE-KAN on Exchange Rate (adj. $\mathbf{R^2} = 0.9490$).

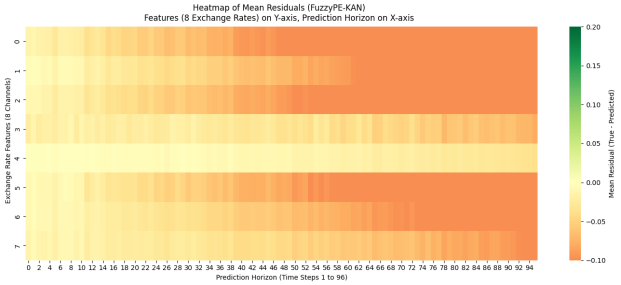

Figure 17: Error heatmap for KAN/FuzzyPE-KAN on Exchange Rate.

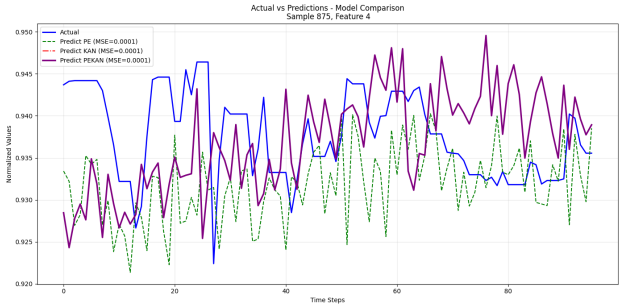

Figure 18: Actual vs. predicted time series for KAN/FuzzyPE-KAN on Exchange Rate.

predicted plot(Figure 18) further confirms the strong alignment between predicted and actual values.

5.6 Comparison with classical methods

To provide a comprehensive evaluation, we compare the proposed FuzzyPE-KAN framework against traditional time series forecasting methods: ARIMA, GARCH(1,1), and a spline-based regression model with B-spline basis functions and Ridge regularization. All classical models were implemented using the same data preprocessing and train/test split protocol as our proposed method (70% training, 20% testing) on the Exchange Rate dataset with a prediction horizon of 96 steps.

Table 19 presents the results. The proposed FuzzyPE-KAN variants consistently and substantially outperform all classical methods. The best-performing variant (PatchTST-FuzzyPE-KAN) achieves an MSE of 0.08 and MAE of 0.20, compared to ARIMA (MSE=1.91, MAE=1.24) and Spline (MSE=1.42, MAE=1.05). This represents an improvement of approximately 95.8% in MSE over ARIMA. The poor performance of GARCH (MSE=7.34) confirms that volatility-focused models are not suitable for direct point forecasting of multivariate time series. These results validate the necessity of the proposed hybrid deep learning approach for capturing the complex, nonlinear, and uncertain patterns inherent in long-term multivariate time series forecasting.

Table 19: Performance comparison between the proposed FuzzyPE-KAN framework and classical time series forecasting methods on the Exchange Rate dataset. The prediction horizon is 96 steps.

Model	**MSE**	**MAE**
ARIMA	1.914	1.240
GARCH(1,1)	7.337	2.464
Spline (B-spline + Ridge)	1.418	1.055
Transformer-FuzzyPE-KAN	0.61	0.62
Informer-FuzzyPE-KAN	0.51	0.58
PatchTST-FuzzyPE-KAN	**0.08**	**0.20**
Crossformer-FuzzyPE-KAN	0.22	0.35
iTransformer-FuzzyPE-KAN	0.09	0.21

6 Conclusions and future work

Long-term multivariate time series forecasting is a highly challenging task due to inherent noise, uncertainty, complex nonlinear patterns, and intricate temporal and cross-variable dependencies. While transformer-based models and their variants have established state-of-the-art performance since 2021, they continue to suffer from high sensitivity to noise and abrupt changes, limited interpretability, and large parameter counts, which restrict their effectiveness in real-world applications such as finance, energy management, and healthcare.

To address these limitations, this study proposes FuzzyPE-KAN, a novel integrated framework that enhances the transformer architecture through three complementary components:

- **Learnable Fuzzy Positional Encoding**: A dynamic module based on Gaussian membership functions and a small multilayer perceptron that generates adaptive fuzzy weights. This approach models the inherently vague, graded, and qualitative nature of time, enabling the model to automatically adjust the positional influence of each time step according to its contextual importance (e.g., proximity to trends, seasonal phases, or abrupt changes).
- **Dynamic Fuzzy Attention Mechanism**: Adapted from recent advances, this mechanism replaces deterministic attention scores with scores augmented by learnable Gaussian noise. It allows the model to dynamically capture and adapt to varying levels of uncertainty, thereby improving robustness against outliers and noisy observations.
- **Hybrid Kolmogorov–Arnold Networks (KANs) for Feed-Forward Layers**: The conventional position-wise feed-forward networks are fully replaced with hybrid KAN layers combining learnable B-spline functions and a parallel SiLU-activated linear path. This substitution enhances nonlinear expressivity and provides exceptional interpretability through visual analysis of the learned univariate functions.

The proposed framework was systematically applied to five state-of-the-art transformer-based baselines—Transformer, Informer, PatchTST, Crossformer, and iTransformer—and evaluated on eight widely used multivariate time series benchmark datasets (ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, and Exchange Rate) using a 96-step prediction horizon. Experimental results demonstrate average improvements of 26–49% in Mean Squared Error (MSE) and 17–29% in Mean Absolute Error (MAE) across most scenarios compared to the respective baselines. The most significant gains were achieved on datasets exhibiting sharp fluctuations and regular seasonal patterns, such as Exchange Rate (up to 78% MSE improvement), Weather (71.28%), ETTh2 (76.41%), and ETTm2 (78.66%). On highly noisy and high-dimensional datasets (e.g., Traffic and Electricity), performance gains were more modest due to computational constraints (limited batch size) and the curse of dimensionality, yet the proposed variants consistently outperformed or matched the baselines.

Beyond accuracy and robustness, the framework offers notable reductions in model parameters and enable the extraction of human-interpretable insights from the trained KAN functions. These results highlight the value of integrating fuzzy logic principles with Kolmogorov–Arnold representation theory to advance transformer-based forecasting toward more reliable, efficient, and interpretable solutions for real-world applications.

6.1 Future work

Several promising directions remain for extending this work:

- Developing lighter and faster variants of FuzzyPE-KAN by incorporating recent efficient KAN implementations such as FastKAN or Efficient-KAN, enabling deployment in real-time systems and resource-constrained edge devices [20].

- Integrating the framework with large language models (e.g., Llama-3, Qwen2, or GPT-4o) to generate natural language explanations of predictions, thereby further enhancing interpretability in fuzzy-enhanced deep models.
- Extending the architecture to multimodal forecasting tasks that combine time series with textual (e.g., news) or visual (e.g., satellite imagery, sensor heatmaps) data, and evaluating its performance in such hybrid settings.
- Applying the proposed fuzzy positional encoding and hybrid KAN components to other domains, including natural language processing, computer vision, and scientific computing, to explore their broader generalizability.

References

[1] L. T. Abdullah, *Forecasting time series using vector autoregressive model*, International Journal of Nonlinear Analysis and Applications, **13**(1) (2022), 499-511. https://doi.org/10.22075/ijnaa.2022.5521

[2] R. Al-Hmouz, W. Pedrycz, M. Mansouri, A. Al-Hmouz, *Dimensionality-based evaluation of fuzzy models developed for high-dimensional data*, International Conference on Artificial Intelligence and Soft Computing, **15948** (2026), 231-242. https://doi.org/10.1007/978-3-032-03705-3_20

[3] K. Albeladi, B. Zafar, A. Mueen, *Time series forecasting using LSTM and ARIMA*, International Journal of Advanced Computer Science and Applications, **14**(1) (2023), 313-320. https://doi.org/10.14569/IJACSA.2023.0140133

[4] J. Alcalá-Fdez, et al., *KEEL: A software tool to assess evolutionary algorithms for data mining problems*, Soft Computing, **13**(3) (2009), 307-318. https://doi.org/10.1007/s00500-008-0323-y

[5] I. Amalou, N. Mouhni, A. Abdali, *Multivariate time series prediction by RNN architectures for energy consumption forecasting*, Energy Reports, **8**(9) (2022), 1084-1091. https://doi.org/10.1016/j.egyr.2022.07.139

[6] S. Bai, *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*, arXiv, (2018). https://doi.org/10.48550/arXiv.1803.01271

[7] S. Chakraborty, F. Heintz, *Enhancing time series forecasting with fuzzy attention-integrated transformers*, arXiv, (2025). https://doi.org/10.48550/arXiv.2504.00070

[8] X. Chen, L. Lai, M. Luo, *FDACNet: Enhancing time-series classification with fuzzy feature and integrated self-attention and temporal convolution*, International Journal of Approximate Reasoning, **186** (2025). https://doi.org/10.1016/j.ijar.2025.109521

[9] P. Diggle, E. Giorgi, *Time series: A biostatistical introduction*, Oxford University Press, 2025. https://doi.org/10.1093/oso/9780198714835.001.0001

[10] Q. Fan, et al., *Vitar: Vision transformer with any resolution*, arXiv, (2024). https://doi.org/10.48550/arXiv.2403.18361

[11] S. S. W. Fatima, A. Rahimi, *A review of time-series forecasting algorithms for industrial manufacturing systems*, Machines, **12**(6) (2024), 380. https://doi.org/10.3390/machines12060380

[12] Y. Guo, et al., *A novel fuzzy relative-position-coding transformer for breast cancer diagnosis using ultrasonography*, Healthcare, **11**(18) (2023). https://doi.org/10.3390/healthcare11182530

[13] X. Han, et al., *Are KANs effective for multivariate time series forecasting?*, arXiv, (2024). https://doi.org/10.48550/arXiv.2408.11306

[14] W. He, J. Zhe, T. Xiao, Z. Xu, Y. Li, *A survey on uncertainty quantification methods for deep learning*, ACM Computing Surveys, **58**(7) (2026), 1-35. https://doi.org/10.1145/3786319

[15] J. S. R. Jang, *ANFIS: Adaptive-network-based fuzzy inference system*, IEEE Transactions on Systems, Man, and Cybernetics, **23**(3) (1993), 665-685. https://doi.org/10.1109/21.256541

[16] L. Jiang, et al., *KANMixer: Can KAN serve as a new modeling core for long-term time series forecasting?*, arXiv, (2025). https://doi.org/10.48550/arXiv.2508.01575

[17] M. Khodarahmi, V. Maihami, *A review on Kalman filter models*, Archives of Computational Methods in Engineering, **30**(1) (2023), 727-747. https://doi.org/10.1007/s11831-022-09815-7

[18] V. I. Kontopoulou, et al., *A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks*, Future Internet, **15**(8) (2023), 255. https://doi.org/10.3390/fi15080255

[19] S. Li, *Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting*, Proceedings of the 33rd International Conference on Neural Information Processing Systems, (2019), 5243-5253. https://doi.org/10.48550/arXiv.1907.00235

[20] Z. Li, *Kolmogorov-Arnold networks are radial basis function networks*, arXiv, (2024). https://doi.org/10.48550/arXiv.2405.06721

[21] Y. Liu, et al., *iTransformer: Inverted transformers are effective for time series forecasting*, arXiv, (2023). https://doi.org/10.48550/arXiv.2310.06625

[22] Z. Liu, et al., *Kan 2.0: Kolmogorov-Arnold networks meet science*, arXiv, (2024). https://doi.org/10.48550/arXiv.2408.10205

[23] Z. Liu, Y. Wang, et al., *Kan: Kolmogorov-Arnold networks*, arXiv, (2024). https://doi.org/10.48550/arXiv.2404.19756

[24] Y. Luo, et al., *Pathohr: Breast cancer survival prediction on high-resolution pathological images*, arXiv, (2025). https://doi.org/10.48550/arXiv.2503.17970

[25] R. Mohammadi Farsani, E. Pazouki, *A transformer self-attention model for time series forecasting*, Journal of Electrical and Computer Engineering Innovations, (2021). https://doi.org/10.22061/jecei.2020.7426.391

[26] J. Morlier, M. Léonardon, V. Gripon, *Input resolution downsizing as a compression technique for vision deep learning systems*, arXiv, (2025). https://doi.org/10.48550/arXiv.2504.03749

[27] M. Pirani, et al., *A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting*, 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), (2022), 1-6. https://doi.org/10.1109/ICDCECE53908.2022.9793213

[28] M. B. A. Rabbani, et al., *A comparison between seasonal autoregressive integrated moving average (SARIMA) and exponential smoothing (ES) based on time series model for forecasting road accidents*, Arabian Journal for Science and Engineering, **46**(11) (2021), 11113-11138. https://doi.org/10.1007/s13369-021-05650-3

[29] D. Ren, Q. Hu, T. Zhang, *EKLT: Kolmogorov-Arnold attention-driven LSTM with transformer model for river water level prediction*, Journal of Hydrology, **649** (2025). https://doi.org/10.1016/j.jhydrol.2024.132430

[30] L. Ren, T. Zhao, H. Wang, *FDformer: A fuzzy dynamic transformer-based network for efficient industrial time series prediction*, IEEE Transactions on Fuzzy Systems, **33**(7) (2025). https://doi.org/10.1109/TFUZZ.2025.3549920

[31] X. Shi, J. Wang, B. Zhang, *A fuzzy time series forecasting model with both accuracy and interpretability is used to forecast wind power*, Applied Energy, **353** (2024). https://doi.org/10.1016/j.apenergy.2023.122015

[32] S. Shreyas, M. Akshath, *ViKANformer: Embedding kolmogorov arnold networks in vision transformers for pattern-based learning*, arXiv, (2025). https://doi.org/10.48550/arXiv.2503.01124

[33] S. Singh, *Neuro-fuzzy architectures for interpretable AI: A comprehensive survey and research outlook*, Journal of Machine Learning Research, **1**(11) (2025). https://doi.org/10.20944/preprints202506.1173.v1

[34] W. Sulandari, S. Suhartono, S. S. Saleh, P. C. Rodrigues, *Exponential smoothing on modeling and forecasting multiple seasonal time series: An overview*, Fluctuation and Noise Letters, **20**(04) (2021). https://doi.org/10.1142/S0219477521300032

[35] A. Vaswani, et al., *Attention is all you need*, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, (2017), 6000-6010. https://doi.org/10.48550/arXiv.1706.03762

[36] W. Wang, J. Shao, H. Jumahong, *Fuzzy inference-based LSTM for long-term time series prediction*, Scientific Reports, **13**(1) (2023). https://doi.org/10.1038/s41598-023-47812-3

[37] P. Wang, K. Wang, Y. Song, X. Wang, *AutoLDT: A lightweight spatio-temporal decoupling transformer framework with AutoML method for time series classification*, Scientific Reports, (2024). https://dx.doi.org/10.2139/ssrn.4884435

[38] M. Waqas, U. W. Humphries, *A critical review of RNN and LSTM variants in hydrological time series predictions*, MethodsX, **13** (2024). https://doi.org/10.1016/j.mex.2024.102946

[39] H. Wu, J. Xu, J. Wang, M. Long, *Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting*, NIPS'21: Proceedings of the 35th International Conference on Neural Information Processing Systems, (2021), 22419-22430. https://doi.org/10.48550/arXiv.2106.13008

[40] Y. Xie, P. Zhang, Y. Chen, *A fuzzy ARIMA correction model for transport volume forecast*, Mathematical Problems in Engineering, **2021**(1) (2021). https://doi.org/10.1155/2021/6655102

[41] X. Yang, X. Wang, *Kolmogorov-Arnold transformer*, arXiv, (2024). https://doi.org/10.48550/arXiv.2409.10594

[42] P. Yu, H. Kong, Z. Li, *Wavelet-enhanced transformer for adaptive multi-period time series forecasting*, Applied Sciences, **15**(23) (2025), 12698. https://doi.org/10.3390/app152312698

[43] N. Yuqi, et al., *A time series is worth 64Words: Long-term forecasting with transformers*, arXiv, (2022). https://doi.org/10.48550/arXiv.2211.14730

[44] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8**(3) (1965), 338-353. https://dx.doi.org/10.1016/S0019-9958(65)90241-X

[45] Y. Zhang, J. Yan, *Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting*, International Conference on Learning Representations, (2023). https://api.semanticscholar.org/CorpusID:259298223

[46] H. Zhou, et al., *Informer: Beyond efficient transformer for long sequence time-series forecasting*, Proceedings of the AAAI Conference on Artificial Intelligence, **35** (2021), 11106-11115. https://doi.org/10.48550/arXiv.2012.07436