Computing, Telecommunication and Control, 2025, Vol. 18, No. 4, Pp. 20–29.
Информатика, телекоммуникации и управление. 2025. Том 18, № 4. С. 20–29.

Research article
DOI: https://doi.org/10.18721/JCSTCS.18402
UDC 004.032.26

A STUDY OF THE APPLICABILITY OF THE KOLMOGOROV-ARNOLD NETWORK ARCHITECTURE FOR TIME SERIES FORECASTING

O.G. Maleev ✉ , *O.A. Kovaleva*

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

✉ olegmg@bk.ru

Abstract. The recently proposed Kolmogorov–Arnold Network (KAN) architecture emerges as a promising alternative to traditional neural networks based on the Multilayer Perceptron (MLP). By leveraging the Kolmogorov–Arnold representation theorem, KAN represents multidimensional functions as combinations of univariate functions, thereby offering potentially higher accuracy and model interpretability through its inherently simpler structure. This paper investigates the applicability of KAN to time series forecasting using the well-known hourly electricity consumption dataset as a benchmark. Meteorological observation data are selected as an additional testbed. A comparative analysis is conducted between KAN networks and traditional MLPs, as well as implementations of recurrent architectures based on KAN (TKAN variants) versus established designs such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). Experimental results demonstrate the superiority of the KAN architecture over MLPs in temporal prediction tasks. The proposed recurrent architecture, TKAN1, achieves the highest coefficient of determination (R^2 = 0.3483) among TKAN variants, with a Root Mean Squared Error (RMSE) of 0.1010 in energy demand forecasting.

Keywords: time series, time series forecasting, Kolmogorov–Arnold network, multilayer perceptron, recurrent neural network

Citation: Maleev O.G., Kovaleva O.A. A study of the applicability of the Kolmogorov–Arnold network architecture for time series forecasting. Computing, Telecommunications and Control, 2025, Vol. 18, No. 4, Pp. 20–29. DOI: 10.18721/JCSTCS.18402

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.18402
УДК 004.032.26

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ АРХИТЕКТУРЫ НЕЙРОННЫХ СЕТЕЙ КОЛМОГОРОВА-АРНОЛЬДА (KAN) К ЗАДАЧЕ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

О.Г. Малеев ✉ , *О.А. Ковалева*

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

✉ olegmg@bk.ru

Аннотация. Недавно предложенная архитектура нейросетей Колмогорова–Арнольда (Kolmogorov–Arnold Networks, KAN) является перспективной альтернативой традиционным нейронным сетям на основе многослойного персептрона (Multilayer Perceptron, MLP). Благодаря использованию теоремы Колмогорова–Арнольда, KAN представляет многомерные функции в виде комбинации одномерных, обеспечивая потенциально более высокую точность и интерпретируемость модели. В данной статье исследуется применимость KAN и ее рекуррентного расширения – Temporal Kolmogorov–Arnold Networks (TKAN) – к задаче прогнозирования временных рядов на примере известного набора данных почасового потребления электроэнергии. В качестве дополнительного набора данных выбраны данные метеорологических наблюдений. Проведен сравнительный анализ сетей KAN с традиционными MLP, а также реализации архитектуры рекуррентной нейросети на основе архитектуры KAN с широко известными архитектурами долгой краткосрочной памяти (Long Short-Term Memory, LSTM) и управляемого рекуррентного блока (Gated Recurrent Units, GRU). Экспериментальные результаты демонстрируют превосходство архитектуры KAN над MLP в задачах временного прогнозирования. Предложенная в статье рекуррентная архитектура TKAN1 демонстрирует лучший среди TKAN коэффициент детерминации R^2 = 0,3483 при RMSE 0,1010 в задаче прогнозирования энергопотребления.

Ключевые слова: временные ряды, прогнозирование временных рядов, нейросеть Колмогорова–Арнольда, многослойный персептрон, рекуррентная нейронная сеть

Для цитирования: Maleev O.G., Kovaleva O.A. A study of the applicability of the Kolmogorov–Arnold network architecture for time series forecasting // Computing, Telecommunications and Control. 2025. Т. 18, № 4. С. 20–29. DOI: 10.18721/JCSTCS.18402

Introduction

Forecasting time series is a critical task across diverse industries, from economics and transportation to meteorology and medicine. Achieving highly accurate forecasts is essential for maintaining business competitiveness, minimizing risks, optimizing resources and justifying significant decision-making processes.

Established approaches address time series forecasting using classical statistical methods [1–3]. However, with increased computational power and the daily generation of vast volumes of temporal data, deep neural networks have proven effective for forecasting. These networks can learn complex data representations, eliminating or reducing the need for manual feature engineering [4, 5]. Despite notable advancements in time series forecasting, several challenges persist. Models based on the Multilayer Perceptron (MLP) architecture require larger statistical datasets for training due to the lack of prior knowledge [6]. Furthermore, black-box models exhibit reduced interpretability and explain ability compared to statistical methods.

A recently proposed fundamentally novel neural network architecture, the Kolmogorov–Arnold Network (KAN) [7], presents a promising alternative to the MLP and opens new avenues for advancing deep learning models.

Related works

MLP networks are effective approximators of nonlinear functions due to the underlying universal approximation theorem (Cybenko's theorem) [8], which asserts, that a feedforward network with a single hidden layer containing a finite number of neurons can approximate any continuous multivariate function to arbitrary accuracy, provided the hidden layer contains a sufficient number of neurons and the network parameters are appropriately chosen. The KAN, in turn, is grounded in the Kolmogorov–Arnold representation theorem, which states that any multivariate continuous function can be expressed as a composition of univariate functions and addition operations:

$$f(x_1,\dots,x_n)=\sum_{q=1}^{2n+1}\Phi_q\left(\sum_{p=1}^{n}\phi_{q,p}\left(x_p\right)\right),$$

where $\phi_{q,p}:[0,1]\to\mathbb{R},\ \Phi_q:\mathbb{R}\to\mathbb{R}$.

Each univariate function is parameterized as a B-spline curve with trainable coefficients c_i of local B-spline basis functions $B_i(x)$, and is represented as a weighted sum with trainable weights w_b and w_s:

$$\phi(x)=w_b b(x)+w_s spline(x),\quad spline(x)=\sum_i c_i B_i(x),$$

where $b(x)$ is analogous to a residual connection.

Thus, l-th layer of KAN is defined by a matrix of functions:

$$\Phi_l=\left\{\phi_{l,q,p}\right\},\ \ l=0,\ \dots,\ L-1,\ \ p=1,\ \dots,\ n_{in},\ \ q=1,\ \dots,\ n_{out},$$

enabling KAN to extend the Kolmogorov–Arnold representation theorem to arbitrary width and depth [9].

The general structure of KAN comprises a composition of L layers, with dimensions specified by the array $[n_0,\ \dots,\ n_{L-1}]$. For an input vector $X\in\mathbb{R}^{n_0}$, the KAN output is expressed as:

$$KAN(X)=\left(\mathbf{\Phi}_{L-1}\circ\mathbf{\Phi}_{L-2}\circ\dots\circ\mathbf{\Phi}_0\right)X,$$

where each layer $\mathbf{\Phi}_l$ transforms its input through learnable univariate functions parameterized via B-spline basis expansions.

Thus, in KAN, activation functions are moved to the edges of the computational graph: each weight is replaced by a univariate activation function parameterized as a spline, while neurons themselves perform only summation of incoming activations.

KAN combines the strengths of MLPs and splines, featuring internal and external levels of degrees of freedom. At the external level, KAN learns the compositional structure of the target function through its MLP-like architecture, while at the internal level, it approximates univariate functions with high precision via spline-like parameterization. Architectural complexity in KAN involves not only adding more layers but also refining the spline grids.

The KAN approach to representing multivariate functions aligns with structural properties of time series, such as trends and seasonality. Embedding prior knowledge about data structure directly into the neural network architecture suggests KAN's potential effectiveness in time series forecasting [10].

Transition to the recurrent TKAN architecture

To extend the capabilities of KAN for time series forecasting, a natural direction is to integrate the KAN architecture with other widely adopted deep learning methods for this task. Recurrent Neural Network (RNN) models have demonstrated themselves as effective tools for forecasting in practical scenarios of varying complexity [11].

For this study, we use a combination of a Recurrent Kolmogorov–Arnold Network (RKAN) and a modified Long Short-Term Memory (LSTM) block, forming the Temporal Kolmogorov–Arnold Network (TKAN) architecture [12]. This approach enables capturing complex nonlinearities through RKAN's learnable activation functions while efficiently managing memory over extended periods via the LSTM cell architecture.

KAN layers retain short-term memory of previous network states, and the gating mechanism regulates information flow by determining which information should be preserved or forgotten over time. The schematic of the TKAN cell is illustrated in Fig. 1.

Following the analogy of hidden state updates in RNNs, the dependence of the current hidden state on its prior value introduces temporal dynamics into each activation function $\phi_{l,j,i}$:

$$x_{l+1,j}\left(t\right)=\sum_{i=1}^{n_l}\tilde{x}_{l,j,i}\left(t\right)=\sum_{i=1}^{n_l}\phi_{l,j,i}\left(x_{l,i}\left(t\right),h_{l,i}\left(t\right)\right),\quad j=1,\ \ldots,\ n_{l+1},$$

where $h_{l,i}\left(t\right)$ is the memory state function for the i-th neuron of the l-th layer at time t.

By analogy with the LSTM cell, information flow in the TKAN cell is governed by a forget gate, input gate and output gate:

$$f_t=\sigma\left(W_f x_t+U_t h_{t-1}+b_f\right),\ i_t=\sigma\left(W_i x_t+U_i h_{t-1}+b_i\right),$$
$$o_t=\sigma\left(RKAN\left(X,t\right)\right).$$

The hidden state h_t is computed as the output of the cell:

$$h_t=o_t\odot\tanh\left(c_t\right),$$

where c_t represents the long-term memory of the cell, updated according to:

$$c_t=f_t\odot c_{t-1}+i_t\odot\tilde{c}_t,$$

where $\tilde{c}_t=\sigma\left(W_c x_t+U_c h_{t-1}+b_c\right)$.

The final predicted value $\hat{y}_t$ is derived via a linear layer:

$$\hat{y}_t=W_{hy}h_t+b_y.$$

This formulation aligns with standard LSTM-based memory mechanisms, where the hidden state h_t acts as a compressed representation of temporal dependencies, and the cell state c_t retains long-term memory through gated updates.

Numerical experiments

To evaluate the applicability of KAN to time series forecasting, we conduct a comparative analysis of KAN against MLP and TKAN against classical recurrent architectures – LSTM [13] and GRU [14].

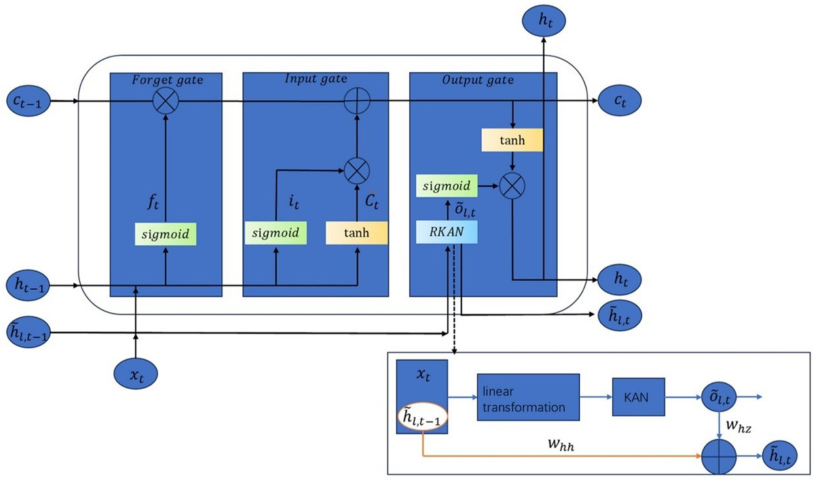

Fig. 1. Architecture of the TKAN Cell

We frame multivariate time series forecasting as a supervised learning task. The input to the model is a sequence of historical time steps $X = [X_1, \ldots, X_H] \in \mathbb{R}^{H \times D}$, where H denotes the historical window size used for forecasting and D represents the number of variables. The forecasting task involves generating an output sequence $Y = [X_{H+1}, \ldots, X_{H+F}] \in \mathbb{R}^{F \times D}$, where F is the forecasting horizon.

We evaluate the considered models on widely used benchmark datasets (Table 1):

1. Electricity Dataset: This dataset contains hourly electricity consumption data from 321 clients between 2012 and 2014, measured in kilowatts[1].

2. Weather Dataset: This dataset includes meteorological observations recorded at 10-minute intervals near Beutenberg (Germany) from 2021 to 2023[2]. Hourly data was obtained by sampling the first observation of each hour.

Table 1

Characteristics of the used datasets

	Dimensions	**Series length**	**Granularuty**	**Split**
Electricity	321	26304	1 hour	[7:1:2]
Weather	20	26280	1 hour	[8:0.5:1.5]

Data preprocessing involves MinMax scaling, which maps values to the [0,1] interval while preserving the distribution shape. Validation follows a simple strategy: datasets are split into training, validation and testing subsets in chronological order according to the predefined ratios.

We compare MLP and KAN on the Electricity dataset, analyzing the dependence of model performance on a key KAN parameter – the spline grid size G. Identical network configurations are considered, with varying numbers of hidden layers and neurons. The historical window size is fixed at $H = 24$, and the forecasting horizon is set to $F = 6$. Network configurations are detailed in Table 2. The loss function is Mean Squared Error (MSE), and training employs the Adam optimizer with an initial learning rate of lr = 0.001. The MSE and Mean Absolute Error (MAE) are used as evaluation metrics for models on test data. Results are summarized in Table 2.

[1] Electricity Hourly Dataset, Available: https://zenodo.org/records/4656140 (Accessed 08.04.2025)

[2] Max-Planck-Institut fuer Biogeochemie – Wetterdaten, Available: https://www.bgc-jena.mpg.de/wetter/weather_data.html (Accessed 08.04.2025)

Table 2

Comparison of MLP and KAN evaluation results

	Configuration	MSE	MAE	Number of parameters
KAN	[24, 5, 6], G=3	0.00471	0.04299	1.2 K
	[24, 10, 6], G=3	0.00451	0.04215	2.4 K
	[24, 20, 6], G=3	0.00442	0.04180	4.8 K
	[24, 10, 10, 6], G=3	0.00464	0.04354	3.2 K
	[24, 10, 6], G=10	0.00419	**0.04060**	4.5 K
	[24, 10, 10, 6], G=10	**0.00416**	0.04081	6.0 K
MLP	[24, 5, 6]	0.00561	0.04801	161
	[24, 10, 6]	0.00498	0.04484	316
	[24, 20, 6]	0.00460	0.04265	626
	[24, 10, 10, 6]	0.00522	0.04655	426
	[24, 20, 20, 20, 6]	0.00480	0.04440	1.5 K

For TKAN, LSTM and GRU, we adopt a simple unified architecture: an input recurrent layer returning full sequences and an intermediate layer returning only the final hidden state, both with identical hidden state dimensions. The output layer is fully connected with linear activation. Hidden state sizes h = [50, 100] , KAN layer sizes h_k = 20, spline grid size G = 3 and spline order k = 3 are tested. Forecasts are generated for F = [12, 24, 48, 96, 168] with H = 48. The loss function remains MSE, and evaluation metrics include RMSE, MAE and the coefficient of determination R^2. Results are reported in Tables 3–6, with bold indicating the best performance per metric and forecasting horizon.

Table 3

Evaluation results of recurrent models with hidden state size *h* = 100 on the electricity dataset

F	TKAN			GRU			LSTM		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
12	0.1316	**0.0766**	**0.0593**	0.2228	0.0797	0.0621	**0.3497**	0.0767	0.0596
24	0.2073	0.0767	0.0590	0.3592	**0.0756**	**0.0585**	**0.3668**	0.0777	0.0605
48	0.2292	0.0873	0.0677	**0.3204**	**0.0776**	**0.0597**	0.2436	0.0808	0.0624
96	**0.2869**	0.0829	0.0640	0.2790	0.0829	0.0639	0.1663	**0.0822**	**0.0632**

Table 4

Evaluation results of recurrent models with hidden state size *h* = 100 on the weather dataset

F	TKAN			GRU			LSTM		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
12	0.6822	0.0738	0.0506	**0.7029**	**0.0697**	**0.0469**	0.6676	0.0747	0.0514
24	0.5893	0.0884	0.0633	0.6070	0.0858	0.0611	**0.6133**	**0.0842**	**0.0590**
48	0.4781	0.1027	0.0748	**0.5774**	**0.0895**	**0.0632**	0.5416	0.0944	0.0674
96	0.4666	0.1043	0.0751	**0.5078**	**0.0989**	**0.0708**	0.4904	0.1012	0.0726
168	**0.4453**	**0.1068**	**0.0767**	0.3738	0.1143	0.0843	0.4168	0.1100	0.0800

Table 5

Evaluation results of recurrent models with hidden state size *h* = 50 on the electricity dataset

F	TKAN			GRU			LSTM		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
12	**0.2513**	0.0809	0.0631	0.1929	**0.0756**	**0.0583**	0.2130	0.1063	0.0840
24	**0.3033**	0.0940	0.0744	0.1615	0.0768	0.0593	0.1344	**0.0756**	**0.0581**
48	0.3425	0.1006	0.0791	0.0974	0.0768	0.0588	**0.3700**	**0.0763**	**0.0583**
96	**0.3733**	0.0881	0.0683	0.1886	**0.0817**	**0.0631**	0.3227	0.0915	0.0716

Table 6

Evaluation results of recurrent models with hidden state size *h* = 50 on the weather dataset

F	TKAN			GRU			LSTM		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
12	0.6563	0.0773	0.0547	0.6879	0.0724	0.0497	**0.6929**	**0.0717**	**0.0484**
24	0.5880	0.0889	0.0640	0.6081	0.0858	0.0611	**0.6319**	**0.0820**	**0.0566**
48	0.4703	0.1040	0.0760	**0.5187**	**0.0981**	**0.0712**	0.4840	0.1021	0.0745
96	0.4703	0.1041	0.0748	0.4368	0.1077	0.0791	**0.4888**	**0.1016**	**0.0726**
168	0.4393	0.1075	0.0771	0.4121	0.1107	0.0807	**0.4459**	**0.1069**	**0.0767**

The experiments use an optimized KAN implementation[3], which addresses performance limitations of the original KAN framework while achieving speed comparable to MLP.

Based on Table 2, it can be concluded that KAN outperforms MLP on the selected dataset. Bold values indicate the best results, while underlined values correspond to approximately the same number of parameters. The simplest KAN configuration with a single hidden layer of 5 neurons achieves accuracy comparable to an MLP with a significantly larger number of parameters and a more complex configuration – three hidden layers with 20 neurons each. Adding 10 neurons to a single hidden layer in KAN (third row) improved accuracy more effectively than adding an additional hidden layer with 10 neurons (fourth row). Notably, the two lowest error rates were achieved with a refined spline grid. This suggests that KAN's internal degrees of freedom, implemented via splines, grant the architecture greater expressive power in approximating data dependencies.

In experiments using recurrent models (Tables 3 and 5), for energy consumption data, R^2 increases with the forecasting horizon for TKAN, while it decreases for weather data (Tables 4 and 6) and other models. At a hidden state size of $h = 50$, TKAN consistently achieves the highest R^2 across most experiments, indicating its potential for optimal energy consumption forecasting when further optimizing configurations.

To improve generalization, regularization techniques will be applied next. As shown in loss curves (e.g., $F = 96$, $h = 100$ in Fig. 2), TKAN demonstrates potential for further training, whereas LSTM and GRU exhibit overfitting, evidenced by diverging training and validation loss curves.

On weather data, TKAN demonstrates superior performance for the longest forecasting horizon when the hidden state size is set to $h = 100$. Unlike other models, the decline in the coefficient of determination (R^2) slows down starting from $F = 48$, indicating improved stability in long-term predictions. This suggests that TKAN retains greater explanatory power as the forecasting horizon increases, making

[3] GitHub – Blealtan/efficient-kan: An efficient pure-PyTorch implementation of Kolmogorov-Arnold Network (KAN), Available: https://github.com/Blealtan/efficient-kan (Accessed 08.04.2025)

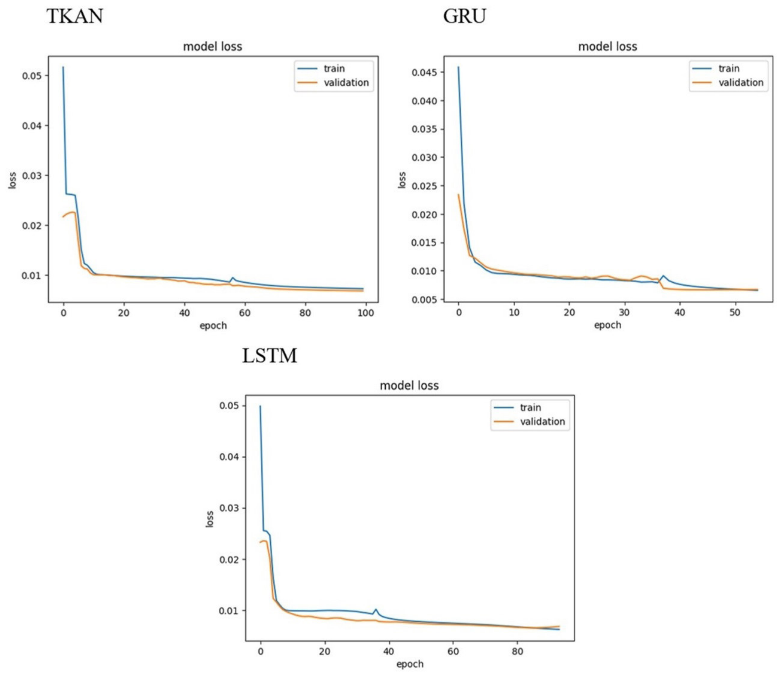

Fig. 2. Loss Functions for Recurrent Models (h = 100)

it a viable alternative to traditional architectures like LSTM and GRU for long-term time series forecasting tasks.

Enhancement through dropout integration in TKAN architecture

To mitigate overfitting and improve model robustness in time series forecasting, we propose a refined architectural framework by incorporating dropout regularization into the TKAN architecture. Dropout, a well-established technique for preventing co-adaptation of neurons during training [15], introduces stochastic deactivation of neurons, thereby enhancing generalization by reducing dependency on specific pathways.

Three variants of the modified architecture were evaluated:

1. TKAN1: A dual-dropout configuration with hidden state size h = 50, featuring dropout layers (rate = 0.1) after each recurrent layer.
2. TKAN2: A single-dropout variant with h = 40 and a dropout rate of 0.2 applied after the first recurrent layer.
3. TKAN3: A dual-dropout design with h = 40, KAN layer output size h_k = 15, and dropout rate = = 0.1 after each recurrent layer.

Experimental results validate the efficacy of this approach (Table 7), TKAN1 achieves the best performance on the longest forecasting horizon, outperforming other models. Additionally, TKAN3 demonstrates the highest R^2 for shorter-term forecasts, highlighting its adaptability to varying temporal dependencies.

This systematic integration of dropout layers underscores TKAN's capacity to harmonize structural complexity with regularization, positioning it as a competitive alternative to traditional recurrent architectures in multi-horizon forecasting tasks.

Conclusion

The empirical analysis presented on this paper demonstrates that KAN serve as a promising alternative to MLP in time series forecasting. When extended to recurrent architectures via TKAN, the

Table 7

Evaluation results of TKAN with regularization on the electricity dataset

F	TKAN			GRU			LSTM		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
12	**0.3801**	0.0949	0.0729	0.3088	0.1025	0.0802	0.2775	**0.0818**	**0.0634**
24	0.3344	0.0978	**0.0755**	**0.3540**	**0.0976**	0.0762	0.3480	0.1013	0.0795
48	**0.3709**	**0.0988**	0.0770	0.3220	0.1019	0.0807	0.3693	0.0989	**0.0767**
96	**0.3483**	**0.1010**	**0.0791**	0.2877	0.1047	0.0834	0.3441	0.1012	**0.0791**

integration of LSTM-inspired gating mechanisms with spline-based parameterization exhibits superior accuracy for extended forecasting horizons while maintaining reduced architectural complexity. Specifically, proposed in the paper modification TKAN1, augmented with dual Dropout layers (dropout rate = 0.1) and a hidden state size $h = 50$, achieves the highest $R^2 = 0.3483$ and lowest RMSE 0.1010 on the Electricity dataset for extended forecasting horizons $F = 96$.

REFERENCES

1. **Holt C.C.** Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 2004, Vol. 20, No. 1, Pp. 5–10. DOI: 10.1016/j.ijforecast.2003.09.015

2. **Winters P.R.** Forecasting sales by exponentially weighted moving averages. *Management Science*, 1960, Vol. 6, No. 3, Pp. 324–342. DOI: 10.1287/mnsc.6.3.324

3. **Box G.E.P., Jenkins G.M., Reinsel G.C., Ljung G.M.** *Time series analysis: Forecasting and control*, 5th ed. New Jersey: John Wiley & Sons, 2015.

4. **Lim B., Zohren S.** Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2021, Vol. 379, No. 2194, Art. no. 20200209. DOI: 10.1098/rsta.2020.0209

5. **Chigasova M.A., Maleev O.G.** Prognozirovanie kotirovok aktsii s ispol'zovaniem neironnoi seti [Stock Price Forecasting Using Neural Networks]. *Sovremennye tekhnologii v teorii i praktike programmirovaniia* [*Modern technologies in the theory and practice of programming*], 2022. S. 180–182.

6. **Bachmann G., Anagnostidis S., Hofmann T.** Scaling MLPs: A tale of inductive bias. *arXiv:2306.13575*, 2023. DOI: 10.48550/arXiv.2306.13575

7. **Liu Z., Wang Y., Vaidya S., Ruehle F., Halverson J., Soljačić M., Hou T.Y., Tegmark M.** KAN: Kolmogorov-Arnold networks. *arXiv:2404.19756*, 2024. DOI: 10.48550/arXiv.2404.19756

8. **Cybenko G.** Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 1989, Vol. 2, Pp. 303–314. DOI: 10.1007/BF02551274

9. **Fakhoury D., Fakhoury E., Speleers H.** ExSpliNet: An interpretable and expressive spline-based neural network. *Neural Networks*, 2022, Vol. 152, Pp. 332–346. DOI: 10.1016/j.neunet.2022.04.029

10. **Han X., Zhang X., Wu Y., Zhang Z., Wu Z.** Are KANs effective for multivariate time series forecasting? arXiv:2408.11306, 2024. DOI: 10.48550/arXiv.2408.11306

11. **Hewamalage H., Bergmeir C., Bandara K.** Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting, 2021, Vol. 37, No. 1, Pp. 388–427. DOI: 10.1016/j.ijforecast.2020.06.008

12. **Genet R., Inzirillo H.** TKAN: Temporal Kolmogorov-Arnold networks. *arXiv:2405.07344*, 2024. DOI: 10.48550/arXiv.2405.07344

13. **Hochreiter S., Schmidhuber J.** Long short-term memory. *Neural Computation*, 1997, Vol. 9, No. 8, Pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735

14. **Cho K., van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y.** Learning phrase representations using RNN encoder–decoder for statistical machine translation. *arXiv:1406.1078*, 2014. DOI: 10.48550/arXiv.1406.1078

15. **Gal Y., Ghahramani Z.** A theoretically grounded application of dropout in recurrent neural networks. *arXiv:1512.05287*, 2015. DOI: 10.48550/arXiv.1512.05287

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Oleg G. Maleev
Малеев Олег Геннадьевич
E-mail: olegmg@bk.ru
ORCID: https://orcid.org/0000-0001-7810-7973

Olga A. Kovaleva
Ковалева Ольга Анатольевна
E-mail: kovaleva.oa@edu.spbstu.ru

Submitted: 01.07.2025; Approved: 25.11.2025; Accepted: 26.11.2025.
Поступила: 01.07.2025; Одобрена: 25.11.2025; Принята: 26.11.2025.