

Kolmogorov-Arnold Networks (KANs) for Time Series Analysis

Cristian J. Vaca-Rubio, Luis Blanco, Roberto Pereira, Màrius Caus

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Barcelona, Spain, 08860.

Emails: {cvaca, lblanco, rpereira, mcaus}@cttc.es

Abstract—This paper introduces a novel application of Kolmogorov-Arnold Networks (KANs) to time series forecasting, leveraging their adaptive activation functions for enhanced predictive modeling. Inspired by the Kolmogorov-Arnold representation theorem, KANs replace traditional linear weights with spline-parametrized univariate functions, allowing them to learn activation patterns dynamically. We demonstrate that KANs outperforms conventional Multi-Layer Perceptrons (MLPs) in a real-world satellite traffic forecasting task, providing more accurate results with considerably fewer number of learnable parameters. We also provide an ablation study of KAN-specific parameters impact on performance. The proposed approach opens new avenues for adaptive forecasting models, emphasizing the potential of KANs as a powerful tool in predictive analytics.

Index terms— Kolmogorov-Arnold Networks, ML, Time series, Satellite

I. INTRODUCTION

Time series forecasting is a traditional problem that plays a key role in a wide range of fields, driving critical decision-making processes in finance, economics, medicine, meteorology, and biology, reflecting the wide applicability and its significance across many domains [1]–[4]. It involves predicting future values based on the previously observed data points. With this goal in mind, understanding the dynamics of time-dependent phenomena is essential and requires unveiling the patterns, trends and dependencies hidden with the historical data. While conventional approaches have been traditionally centered on parametric models grounded in domain-specific knowledge, such as autoregressive (AR), exponential smoothing, or structural time series models, contemporary Machine Learning (ML) techniques offered a pathway to discern temporal patterns solely from data-driven insights.

Non-ML methods traditionally tackle the time series forecasting problem and often rely on statistical methods to predict future values based on previously observed data. One of the most well-known techniques is the AutoRegressive Integrated Moving Average (ARIMA) model, which combines autoregression, integration, and moving averages to forecast data.

This work is part of the project SOFIA PID2023-147305OB-C32 funded by MICIU/AEI/10.13039/501100011033 and by FEDER/UE, in part by 5G-STARDUST project which has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096573. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI), and in part by the grant CHIST-ERA-20-SICT-004 (SONATA) by PCI2021-122043-2A/AEI/10.13039/501100011033.

The authors in [5] detailed this approach, providing a comprehensive methodology foundational for subsequent statistical forecasting methods. Extensions of ARIMA, like Seasonal ARIMA (SARIMA), adapt the model to handle seasonality in data series, particularly useful in fields like retail and climatology [6]. Exponential Smoothing techniques constitute another popular set of traditional (non-ML-based) forecasting methods. They are characterized by their simplicity and effectiveness in handling data with trends and seasonality. An exponent of this family of techniques is the so-called Holt-Winters seasonal technique, which adjusts the model parameters in response to changes in trend and seasonality within the time series data [7], [8]. These models have been widely used for their efficiency, interpretability and implementation.

More recently, ML models have significantly impacted the forecasting landscape by handling large datasets and capturing complex nonlinear relationships that traditional methods cannot. In recent years, Deep Learning (DL)-based forecasting models [9], [10] have gained popularity, motivated by the notable achievements in many fields. For instance, neural networks have been extensively studied due to their flexibility and adaptability. Simple Multi-Layer Perceptron (MLPs) were among the first to be applied to forecasting problems, demonstrating significant potential in non-linear data modeling [3], [11].

Built upon these light models, more complex architectures have progressively expanded the capabilities of neural networks in time series forecasting. Typical examples are recurrent neural network architectures such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which are designed to maintain information in memory for long periods without the risk of vanishing gradients – a common issue in traditional recurrent networks [12], [13]. On a related note, Convolutional Neural Networks (CNNs), which are fundamentally inspired by MLPs, are also extensively employed in time series forecasting. These architectures are particularly efficient at processing temporal sequences due to their strong spatial pattern recognition capabilities. The combination of CNNs with LSTMs has resulted in models that efficiently process both spatial and temporal dependencies, enhancing forecasting accuracy [14]. These models have started to outperform established benchmarks in complex forecasting tasks, motivating a significant shift towards more complex network structures. Unfortunately, as the majority of the models mentioned above are inspired by MLP architecture,

they tend to have poor scaling law [15], i.e., the number of parameters in MLPs networks do not scale linear with the number of layers, and often lack interpretability.

A recent study in reference [16], which caught the attention of the research community, introduces Kolmogorov-Arnold Networks (KANs), a novel neural network architecture designed to potentially replace traditional multilayer perceptrons. KANs represent a disruptive paradigm shift, and as a potential game changer have recently attracted the interest of the AI community worldwide. They are inspired by the Kolmogorov-Arnold representation theorem [17]–[19]. Unlike MLPs, which are inspired by the universal approximation theorem, KANs take advantage of this representation theorem to generate a different architecture. They innovate by replacing linear weights with spline-based univariate functions along the edges of the network, which are structured as learnable activation functions. This design not only enhances the accuracy and interpretability of the networks, but also enables them to achieve comparable or superior results with smaller network sizes across various tasks, such as data fitting and solving partial differential equations. While KANs show promise in improving the efficiency and interpretability of neural network architectures, the study acknowledges the necessity for further research into their robustness when applied to diverse datasets and their compatibility with other deep learning architectures. These areas are crucial for understanding the full potential and limitations of KANs.

Our paper is a prospective study that investigates the application of KANs to time series forecasting. We aim to evaluate the practicality of KANs in real-world scenarios, to the best of the authors' knowledge, not previously explored in the literature, analyzing their efficiency regarding the number of trainable parameters and discussing how the additional degrees of freedom might affect forecasting performance. Herein, we will assess the performance using real-world satellite traffic data. This exploration seeks to further validate KANs as a versatile tool in advanced neural network design for time series forecasting, although more comprehensive studies are required to optimize their use across broader applications. Finally, we note that due to the early stage of KANs, it is fair to compare them as a potential alternative to MLPs, but further investigation is needed to develop more complex solutions that can compete with advanced architectures such as LSTMs, GRUs, and CNNs, already well-established on the MLP-based architectures [20], [21].

This paper is structured as follows. Section 2 presents the problem statement, providing fundamental background on the Kolmogorov-Arnold representation theorem and describes our generalized KANs for time series forecasting. Section 3 introduces the experimental setup description. Simulation results analyzing the performance of KANs with real-world datasets are shown in Section 4. Finally, concluding remarks are provided in Section 5.

II. PROBLEM STATEMENT

We formulate the traffic forecasting problem as a time series at time t represented by y_t . Our objective is to predict the future values of the series

$$\mathbf{y}_{t_0:T} = [y_{t_0}, y_{t_0+1}, \dots, y_{t_0+T}] \quad (1)$$

based solely on its historical values

$$\mathbf{x}_{t_0-c:t_0-1} = [x_{t_0-c}, \dots, x_{t_0-2}, x_{t_0-1}] \quad (2)$$

where t_0 denotes the starting point from which future values $y_t, t = t_0, \dots, T$ are to be predicted. We differentiate the historical time range $[t_0 - c, t_0 - 1]$ and the forecast range $[t_0, T]$ as the context and prediction lengths, respectively. Our approach focuses on generating point forecasts for each time step in the prediction length, aiming to achieve accurate and reliable forecasts. Figure 1 shows an exemplary time series.

A. Kolmogorov-Arnold representation background

Contrary to MLPs, which are based on universal approximation theorem, KANs rely on the Kolmogorov-Arnold representation theorem, also known as the Kolmogorov-Arnold superposition theorem. A fundamental result in the theory of dynamical systems and ergodic theory. It was independently formulated by Andrey Kolmogorov and Vladimir Arnold in the mid-20th century.

The theorem states that any multivariate continuous function f , which depends on $\mathbf{x} = [x_1, x_2, \dots, x_n]$, on a bounded domain, can be represented as the finite composition of simpler continuous functions, involving only one variable. Formally, a real, smooth, and continuous multivariate function $f(\mathbf{x}) : [0, 1]^n \rightarrow \mathbb{R}$ can be represented by the finite superposition of univariate functions [17]:

$$f(\mathbf{x}) = \sum_{i=1}^{2n+1} \Phi_i \left(\sum_{j=1}^n \phi_{i,j}(x_j) \right), \quad (3)$$

where $\Phi_i : \mathbb{R} \rightarrow \mathbb{R}$ and $\phi_{i,j} : [0, 1] \rightarrow \mathbb{R}$ denote the so-called outer and inner functions, respectively. One might initially perceive this development as highly advantageous for ML. The task of learning a high-dimensional function simplifies to learning a polynomial number of one dimensional functions. Nevertheless, these 1-dimensional functions can exhibit non-smooth characteristics, rendering them potentially unlearnable in practical contexts. As a result of this problematic behavior, the Kolmogorov-Arnold representation theorem has been traditionally disregarded in machine learning circles, recognized as theoretically solid, but ineffective in practice. Unexpectedly, the theoretical result in [16] has recently emerged as a potential game changer, paving the way for new network architectures, inspired by the Kolmogorov-Arnold theorem.

B. Kolmogorov-Arnold network background

The authors in [16] mention that equation (3) has two layers of non-linearities, with $2n + 1$ terms in the middle layer. Thus, we only need to find the proper functions inner univariate functions $\phi_{i,j}$ and Φ_i that approximate the function. The one

dimensional inner functions $\phi_{i,j}$ can be approximated using B-splines. A spline is a smooth curve defined by a set of control points or knots. Splines are often used to interpolate or approximate data points in a smooth and continuous manner. A spline is defined by the order k ($k = 3$ is a common value), which refers to the degree of the polynomial functions used to interpolate or approximate the curve between control points. The number of intervals, denoted by G , refers to the number of segments or subintervals between adjacent control points. In spline interpolation, the data points are connected by these segments to form a smooth curve (of $G + 1$ grid points). Although splines other than B-splines could also be considered, this is the approach proposed in [16]. Equation (3) can be represented as a 2-layer (or analogous 2-depth) network, with activation functions placed at the edges (instead of at the nodes) and nodes performing a simple summation. Such two-layer network is too simplistic to effectively approximate any arbitrary function with smooth splines. For this reason, reference [16] extends the ideas discussed above by proposing a generalized architecture with wider and deeper KANs.

A KAN layer is defined by a matrix Φ [16] composed by univariate functions $\{\phi_{i,j}(\cdot)\}$ with $i = 1, \dots, N_{in}$ and $j = 1, \dots, N_{out}$, where N_{in} and N_{out} denote the number of inputs and the number of outputs, respectively, and $\phi_{i,j}$ are the trainable spline functions described above. Note according to the previous definition, the Kolmogorov-Arnold representation theorem presented in Section II-A can be expressed as a two-layer KAN. The inner functions constitute a KAN layer with $N_{in} = n$ and $N_{out} = 2n + 1$, while the external functions constitute another KAN layer with $N_{in} = 2n + 1$ and $N_{out} = 1$.

Let us define the shape of a KAN by $[n_1, \dots, n_{L+1}]$, where L denotes the number of layers of the KAN. It is worth noting the Kolmogorov-Arnold theorem is defined by a KAN of shape $[n, 2n + 1, 1]$. A generic deeper KAN can be expressed by the composition L layers:

$$\mathbf{y} = \text{KAN}(\mathbf{x}) = (\Phi_L \circ \Phi_{L-1} \circ \dots \circ \Phi_1)\mathbf{x}. \quad (4)$$

Notice that all the operations are differentiable. Consequently, KANs can be trained with backpropagation. Despite their elegant mathematical foundation, KANs are simply combinations of splines and MLPs, which effectively exploit each other's strengths while mitigating their respective weaknesses. Splines stand out for their accuracy on low-dimensional functions and allow transition between various resolutions. Nevertheless, they suffer from a major dimensionality problem due to their inability to effectively exploit compositional structures. In contrast, MLPs experience a lower dimensionality problem, due to their ability to learn features, but exhibit lower accuracy than splines in low dimensions due to their inability to optimize univariate functions effectively. KANs have by their construction 2 levels of degrees of freedom. Consequently, KANs possess the capability not only to acquire features, owing to their external resemblance to MLPs, but also to optimize these acquired features with a high degree of accuracy, facilitated by their internal resemblance to splines. To learn

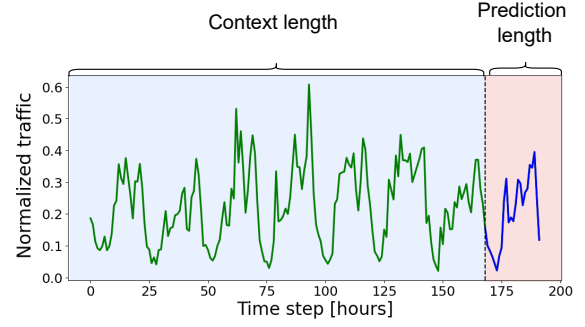


Fig. 1: Example of normalized satellite traffic series data with the conditioning and prediction lengths denoted in blue, and red, respectively.

features accurately, KANs can capture compositional structure (external degrees of freedom), but also effectively approximate univariate functions (internal degrees of freedom with the splines). It should be noted that by increasing the number of layers L or the dimension of the grid G , we are increasing the number of parameters and, consequently, the complexity of the network. This approach constitutes an alternative to traditional DL models, which are currently relying on MLP architectures and motivates our extension of this work.

C. KAN time series forecasting network

We frame our traffic forecasting problem as a supervised learning framework consisting of a training dataset with input-output $\{\mathbf{x}_{t_0-c:t_0-1}, \mathbf{y}_{t_0:T}\}$ in the condition and prediction lengths. We want to find f that approximates $\mathbf{y}_{t_0:T}$, i.e., $\mathbf{y}_{t_0:T} \approx f(\mathbf{x}_{t_0-c:t_0-1})$. For ease of notation, we describe our framework as a two-layer (2-depth) KAN $[N_i, n, N_o]$ (note that to comply with the original paper notation, the input layer is not accounted as a layer per se). The output and input layers will be comprised of N_o , and N_i nodes corresponding to the total amount of time steps in (1) and (2), while the transformation/hidden layer of n nodes. The inner functions constitute a KAN layer with $N_{in} = N_i$ and $N_{out} = n$, while the external functions constitute another KAN layer with $N_{in} = n$ and $N_{out} = N_o$. Our KAN can be expressed by the composition of 2 layers:

$$\mathbf{y} = \text{KAN}(\mathbf{x}) = (\Phi_2 \circ \Phi_1)\mathbf{x}, \quad (5)$$

where the output functions Φ_2 generates the N_o outputs values corresponding to (1) by doing the transformation from the previous layers, i.e, we predict the T time steps. The proposed network can be used to forecast future traffic data in the prediction length, based solely on the context length.

Fig. 2 shows a generic representation for any arbitrary number of layers L as presented in (4).

III. EXPERIMENTAL SETUP

The dataset has been generated within the context of the European project 5G-STARLUST. The inputs are obtained

TABLE I: Model configurations for satellite traffic forecasting

| Model | Configuration | Time horizon (h) | Spline details | Activations |
|---------------|-------------------------------|----------------------------|-----------------------------------|--------------|
| MLP (3-depth) | [168, 300, 300, 300, 24] | Context/Prediction: 168/24 | N/A | ReLU (fixed) |
| MLP (4-depth) | [168, 300, 300, 300, 300, 24] | Context/Prediction: 168/24 | N/A | ReLU (fixed) |
| KAN (3-depth) | [168, 40, 40, 24] | Context/Prediction: 168/24 | Type: B-spline, $k = 3$, $G = 5$ | learnable |
| KAN (4-depth) | [168, 40, 40, 40, 24] | Context/Prediction: 168/24 | Type: B-spline, $k = 3$, $G = 5$ | learnable |

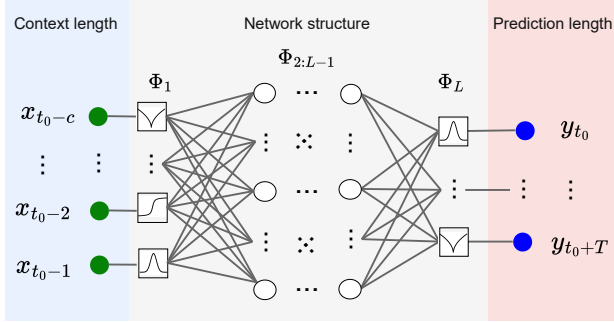


Fig. 2: Example of the flow of information in the KAN network architecture for our traffic forecasting task. Learnable activations are represented inside a square box.

from a satellite operator (SO), as a result of processing real information from a GEO satellite communication system, which provisions broadband services. The dataset is a long time series capturing aggregated traffic data. To preserve privacy, anonymous clients have been defined with more than 500 connected users, and the traffic has been normalized. The measurements are monthly long, and the time granularity is 1 hour. The traffic has been extracted per satellite beam in Megabits per second (Mbps). Although the data has been collected using a GEO satellite communication system, it is expected that user needs could be used to address LEO systems, as well. It is worth emphasizing that the data collected can be used for AI-driven predictive analysis, to forecast traffic conditions, which is essential to avoid congestion and to make efficient use of satellite resources. Endowing the network with intelligence will be beneficial to meet the different demands of satellite applications.

We aim to investigate the forecasting performance of different KAN and MLP architectures for predicting satellite traffic over a total of six beam areas. Concretely, we have a context length of 168 hours (one week) and a prediction length of 24 hours (one day). This translates to $T = 24$, $c = 168$, where $y_{t_0+T} = 192$ in (1) and (2). Our focus is on evaluating the efficacy of KAN models compared to traditional MLPs¹. We designed our experiments to compare models with similar depths but varying architectures to analyze their impact on forecasting accuracy and parameter efficiency. Table I summarizes the parameters selected for this evaluation. We

¹As KANs are in their infancy, we remark this comparison is fair instead of comparing against more complex architectures as LSTM.

have data for the six beams over one month. We use two weeks + 1 day for training and one week + 1 day for testing for all the different beams on the dataset. These test series were not seen by the network during training time. We train all the networks with 500 epochs and Adam optimizer with a learning rate of 0.001. The selected loss function minimizes the mean absolute error (MAE) of the values around the prediction length.

IV. SIMULATION RESULTS

A. Performance analysis

We analyze the forecasting performance in the prediction length for different beams over the test set. Figures 3a-c depicts the real traffic value used as input (in green) to the networks, the expected output prediction length (in blue) and the values predicted values using a KAN (in red) and MLP (in purple) of depth 4 both – see Table I for details on model configuration. In general, our results show that the predictions obtained using KANs better approximates the real traffic values than the predictions obtained using traditional MLPs.

This is particularly evident in Figure 3a. Here, KAN accurately matches rapid changes in traffic volume, which the MLP models sometimes moderately over/under-predicted, as the last part of the forecast shows. This capability suggests that KANs are better suited to adapt to sudden shifts in traffic conditions, a critical aspect of effective traffic management.

Additionally, the responsiveness of KANs is particularly noticeable in Figure 3b during fast changing traffic conditions. KAN shows a rapid adjustment to its forecast that is closely aligned with the actual traffic pattern. This is particularly noticeable in the last 6 hours of the prediction length where MLP exhibits a lag failing to capture these immediate fluctuations, which shows its worse performance to capture dynamic traffic variations. Further analysis is shown in Figure 3c, where traffic conditions are more variable and intense, demonstrated the robustness of KAN in maintaining high performance despite the complexity and higher volume. This robustness suggests that KANs can manage different scales and intensities of traffic data more effectively than MLPs, making them more reliable for deployment in varied traffic scenarios.

To further quantify the performance and advantages of using KANs for the satellite traffic forecasting task we show Table II. It shows a detailed comparison of MLPs and KANs different architectures used for evaluation over all the beams. The table displays the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the number

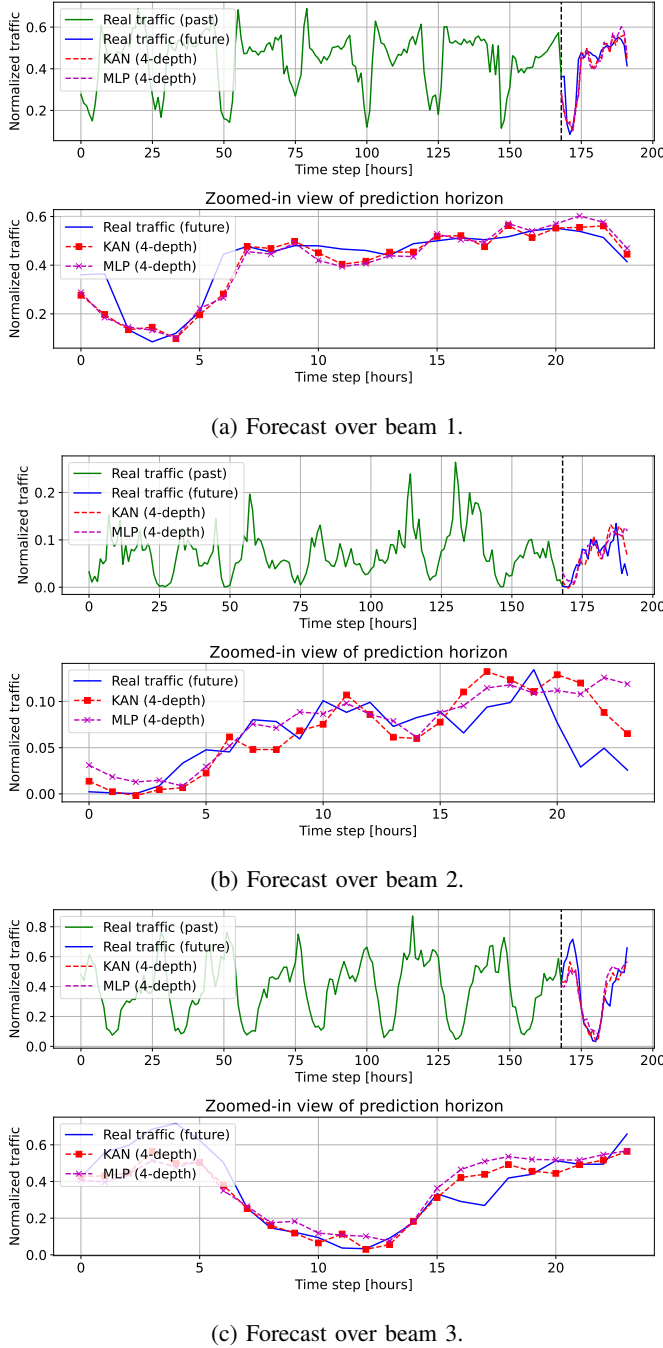


Fig. 3: Satellite traffic over three different beams with their forecasted values using a 4-depth KAN and a 4-depth MLP.

of trainable parameters for each model. Analyzing the error metrics, it becomes clear that KANs outperform MLPs, where the KAN (4-depth) is the best in performance. Its lower values in MSE and RMSE indicates its better ability to predict traffic volumes with lower deviation. Similarly, its lower values in MAE and MAPE suggests that KANs not only provides more accurate predictions but also maintains consistency across different traffic volumes, which is crucial for practical traffic

TABLE II: Results summary

| Model | MSE ($\times 10^{-3}$) | RMSE ($\times 10^{-2}$) | MAE ($\times 10^{-2}$) | MAPE | Parameters |
|---------------|--------------------------|---------------------------|--------------------------|-------------|------------|
| MLP (3-depth) | 6.34 | 7.96 | 5.41 | 0.64 | 238k |
| MLP (4-depth) | 6.12 | 7.82 | 5.55 | 1.05 | 329k |
| KAN (3-depth) | 5.99 | 7.73 | 5.51 | 0.62 | 93k |
| KAN (4-depth) | 5.08 | 7.12 | 5.06 | 0.52 | 109k |

forecasting scenarios.

Furthermore, the parameter count reveals a significant difference in model complexity. KAN models are notably more parameter-efficient, with KAN (4-depth) utilizing only 109k parameters compared to 329k parameters for MLP (4-depth) or 238k for MLP (3-depth). **This reduced complexity suggests that KANs can achieve higher or comparable forecasting accuracy with simpler and potentially lighter models.** Such efficiency is especially valuable in scenarios where computational resources are limited or where rapid model deployment is required. The results also show that with an augmentation of 16k parameters in KAN, the performance can be significantly improved, contrary to MLPs which an increment of 91k parameters does not showcase a significant improvement.

From a technical perspective, KANs leverage a theoretical foundation that provides an intrinsic advantage in modeling complex, non-linear patterns typical in traffic systems. This capability likely contributes to their flexibility and accuracy in traffic forecasting. The consistency in performance across diverse conditions also suggests that KANs have strong generalization capabilities, which is essential for models used in geographically varied locations under different traffic conditions. Moreover, besides obtaining lower error rates, our results also suggest that KANs can do so with considerably smaller number of parameters than traditional MLP networks.

B. KANs parameter-specific analysis

We provide an insightful analysis of how different configurations of nodes and grid sizes affect the performance of KANs, particularly in the context of traffic forecasting. For this analysis, we designed 3 KANs (2-depth) $[168, n, 24]$ with $n \in \{5, 10, 20\}$ and varying grids $G \in \{5, 10, 20\}$ for a $k = 3$ order B-spline. These results are shown during training time.

Figure 4 shows a clear trend where increasing the number of nodes generally results in lower loss values. This indicates that higher node counts are more effective at capturing the complex patterns in traffic data, thus improving the performance. For instance, configurations with $n = 20$ demonstrate significantly lower losses across all grid sizes compared to those with fewer nodes.

Similarly, the **grid size within the splines of KANs has a notable impact on model performance.** Larger grid sizes, when used with a significant amount of nodes ($n \in \{10, 20\}$), consistently result in better performance. However, when the amount of nodes is low ($n = 5$) the extra complexity of the grid size shows the opposite effect. When having a significant amount of nodes larger grids likely provide a more detailed basis for the spline functions, allowing the model to

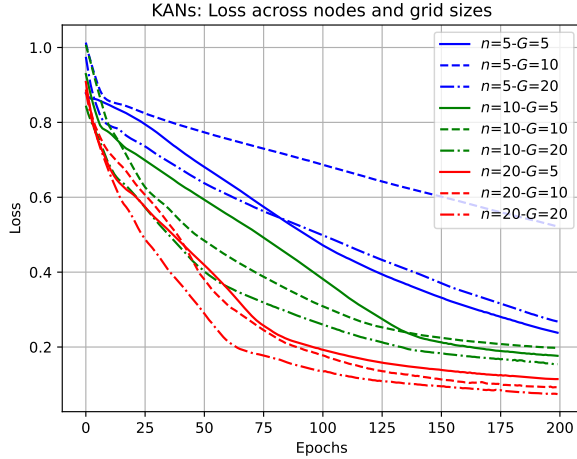


Fig. 4: Ablation comparison of KAN-specific parameters during training time.

accommodate better variations in the data, which is crucial for capturing complex temporal traffic patterns.

The best performance is observed in configurations that combine a high node count with a large grid size, such as the $n = 20$, and $G = 20$ setup. This combination likely offers the highest degree of flexibility and learning capacity, making it particularly effective for modeling the intricate dependencies found in traffic data. However, this superior performance comes at the cost of potentially higher computational demands and longer training times, as more trainable parameters are included.

These findings imply that while increasing nodes and grid sizes can significantly enhance the performance of KANs, these benefits must be weighed against the increased computational requirements. For practical applications, particularly in real-time traffic management where timely responses are critical, it is essential to strike a balance. An effective approach could involve starting with moderate settings and gradually adjusting the nodes and grid sizes based on performance assessments and computational constraints. Besides, we want to highlight that for this study continual learning was not assessed, a possibility mentioned in the original paper [16].

V. CONCLUSION

In this paper, we have performed an analysis of KANs and MLPs for satellite traffic forecasting. The results highlighted several benefits of KANs, including superior forecasting performance and greater parameter efficiency. In our analysis, we showed that KANs consistently outperformed MLPs in terms of lower error metrics and were able to achieve better results with lower computational resources. Additionally, we explored specific KAN parameters impact on performance. This study showcases the importance of optimizing node counts and grid sizes to enhance model performance. Given their effectiveness and efficiency, KANs appear to be a reasonable alternative to traditional MLPs in traffic management.

REFERENCES

- [1] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied soft computing*, vol. 90, p. 106181, 2020.
- [2] K. R. Prakarsha and G. Sharma, "Time series signal forecasting using artificial neural networks: An application on ecg signal," *Biomedical Signal Processing and Control*, vol. 76, p. 103705, 2022.
- [3] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.
- [4] X. Zhu, Y. Xiong, M. Wu, G. Nie, B. Zhang, and Z. Yang, "Weather2k: A multivariate spatio-temporal benchmark dataset for meteorological forecasting based on real-time observation data from ground weather stations," *arXiv preprint arXiv:2302.10493*, 2023.
- [5] G. E. Box and al., *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [6] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.
- [7] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Int. journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.
- [8] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management science*, vol. 6, no. 3, pp. 324–342, 1960.
- [9] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.
- [10] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.
- [11] J. F. Zhang et al., "Neural networks for time-series forecasting," *Handbook of natural computing*, vol. 1, p. 4, 2012.
- [12] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] A. Borovykh and al., "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, 2017.
- [15] G. Bachmann, S. Anagnostidis, and T. Hofmann, "Scaling mlps: A tale of inductive bias," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [16] Z. Liu and al., "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.
- [17] A. N. Kolmogorov, *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961.
- [18] J. Braun and M. Griebel, "On a constructive proof of Kolmogorov's superposition theorem," *Constructive approximation*, vol. 30, pp. 653–675, 2009.
- [19] J. Schmidt-Hieber, "The kolmogorov–arnold representation theorem revisited," *Neural networks*, vol. 137, pp. 119–126, 2021.
- [20] I. E. Livieris, E. Pintelas, and P. Pintelas, "A cnn-lstm model for gold price time-series forecasting," *Neural computing and applications*, vol. 32, pp. 17351–17360, 2020.
- [21] S. Mehtab and J. Sen, "Analysis and forecasting of financial time series using cnn and lstm-based deep learning models," in *Advances in Distributed Computing and Machine Learning: Proceedings of ICADCMML 2021*, pp. 405–423, Springer, 2022.