DOI: https://doi.org/10.14267/SEFBIS.2026.01

Horizontal Visibility Graph-Based Time Series Forecasting

FERENC FARKAS[1,2]
[1] Department of Quantitative Methods, University of Pannonia, Veszprém
[2] X1F Management and Technology Services GmbH, München
eMail: ferenc.farkas@x1f.one

ABSTRACT

Time series forecasting remains challenging when the underlying process is irregular, nonlinear, or structurally changing. This study presents a time series forecasting approach based on Horizontal Visibility Graphs (HVG), where sliding-window segments of a time series are transformed into graph structures and structural graph metrics are used as input features for forecasting models. The framework combines window-based segmentation, HVG construction, graph feature extraction, and supervised learning. The empirical analysis is conducted on a controlled subset of daily time series from the M4 dataset and compares HVG-based models with classical statistical benchmarks, namely ARIMA and Holt–Winters. The results show that HVG-derived structural metrics provide an informative feature space and can achieve competitive performance, although effectiveness is model-dependent. In a representative example, several HVG-based models outperform the statistical baselines. The findings indicate that HVG representations can serve as a complementary feature space for time series forecasting.

Keywords: time series forecasting, horizontal visibility graph, machine learning, M4 dataset

1 Introduction

Time series forecasting plays a crucial role in many scientific and economic applications, including financial market analysis, energy demand estimation, and risk management. However, many real-world time series exhibit nonlinear, irregular, or chaotic behaviour, making accurate forecasting particularly challenging. Traditional statistical forecasting models often rely on assumptions such as linearity or stationarity, which may limit their effectiveness when applied to complex dynamic systems. Large-scale forecasting benchmarks such as the M4 competition have demonstrated that forecasting performance can vary substantially across models and datasets, highlighting the importance of exploring alternative modelling approaches and feature representations [7].

In recent years, new methodological approaches have emerged that aim to capture the structural properties of time series beyond conventional statistical frameworks. One such approach is to transform time series into complex networks using visibility graph techniques, which have also been extended toward forecasting applications using link-prediction and graph-based learning mechanisms [8],[14]. In a visibility graph representation, observations from the time series become nodes, while edges are established based on visibility relationships between data points. This transformation preserves important structural characteristics of the original series while enabling the application of graph-theoretical analysis methods.

The visibility graph framework, introduced by Lacasa et al. [5], offers a way to convert temporal observations into network structures while preserving their ordering. In the Horizontal Visibility Graph (HVG) variant, edges are defined by a simplified visibility criterion, which makes the transformation computationally attractive and analytically interpretable [6]. In this setting, time series segments can be represented as graphs, from which structural metrics are extracted and used as predictive features.

This study examines whether HVG-derived structural descriptors can support forecasting on daily time series from the M4 dataset. Its contribution is empirical rather than algorithmic: it evaluates sliding-window HVG features as supervised inputs for several standard machine learning models and compares their performance with ARIMA and Holt–Winters benchmarks. The focus is on model-dependent performance and cross-series variability, not on introducing a new visibility graph construction or forecasting algorithm.

Submitted: 26/03/2026 Accepted: 20/05/2026

2 Related Work

Time series forecasting has long been a central topic in quantitative analysis and has been widely applied in economics, finance, meteorology, and engineering. Classical statistical approaches such as the Auto-Regressive Integrated Moving Average (ARIMA) model remain among the most widely used methods for modelling and predicting temporal processes. These models describe the dependence structure of a time series through autoregressive and moving average components and often require stationarity assumptions or preprocessing steps such as differencing to stabilize the statistical properties of the data [1].

In recent years, machine learning methods have increasingly been applied to time series forecasting problems. Neural network architectures such as recurrent neural networks and Long Short-Term Memory (LSTM) models have demonstrated strong capabilities in modelling nonlinear temporal dependencies and complex patterns in sequential data [3]. More broadly, machine learning approaches have been successfully used to capture nonlinear relationships in financial and economic time series, providing an alternative to traditional statistical models [12]. However, many forecasting approaches still rely on representing the time series directly as a sequence of numerical observations.

An alternative perspective has emerged from the field of complex network analysis. In this framework, time series are transformed into network structures that capture structural relationships among observations. One of the most widely used techniques for this transformation is the visibility graph method introduced by Lacasa et al. [5]. A computationally simpler variant known as the Horizontal Visibility Graph (HVG) was later proposed by Luque et al. [6]. In a visibility graph representation, each observation of the time series becomes a node in the network, and edges are created between nodes based on geometric visibility criteria. This transformation preserves the temporal ordering of observations while enabling the application of graph-theoretical tools to analyse the structural properties and underlying dynamics of the series. Comprehensive reviews of visibility graph methodologies further highlight their applicability in forecasting and classification tasks, as well as their integration with machine learning approaches [13].

Beyond descriptive analysis, visibility graph-based methods have also been extended to forecasting tasks through link-prediction, random-walk, and graph-network formulations, demonstrating that graph representations can support both structural interpretation and predictive modelling. This perspective is further supported by recent studies that apply network-based representations to real-world prediction problems, where structural properties of collaboration networks are leveraged for link prediction and decision-support purposes [4][8].

Recent studies have explored the integration of network-based representations with machine learning techniques. Graph-based approaches allow the extraction of structural descriptors that summarize the topology of the transformed network, including metrics related to connectivity, clustering, and path structure. These features can serve as informative inputs for forecasting models and may capture aspects of temporal dynamics that are not directly observable in the original time series representation. Graph neural networks and other graph-based learning frameworks have further expanded the possibilities for modelling relational structures in data [2], [11].

Overall, the growing intersection between time series analysis, complex network theory, and machine learning provides a promising direction for developing alternative forecasting methodologies. By transforming temporal data into graph representations and exploiting their structural characteristics, visibility graph-based approaches offer a complementary perspective for modelling complex and potentially nonstationary time series.

In addition to classical visibility graph approaches, recent research has increasingly focused on hybrid methodologies that combine structural representations with advanced learning paradigms. In particular, graph-based feature extraction has been integrated with ensemble learning and graph-based predictive frameworks in order to improve performance on complex and noisy time series [8]. These approaches highlight that structural transformations can serve not only as descriptive tools but also as effective feature engineering mechanisms for downstream forecasting models.

Furthermore, recent studies emphasize the importance of capturing multi-scale and nonstationary dynamics in time series. Visibility graph-based representations are particularly suitable in this context, as they preserve both local ordering and global structural characteristics of the time series, and recent methodological extensions further generalize their applicability to more complex time series structures [10]. This makes them robust to certain types of noise and nonlinear distortions, which are common in financial and economic data.

Another relevant research direction focuses on the interpretability of forecasting models. Compared to black-box deep learning approaches, graph-based feature representations combined with relatively simple machine learning models can provide more transparent insights into the relationship between structural properties and predictive outcomes. This aligns with the growing demand for explainable models in domains such as finance and risk management, where feature-based representations have been shown to support understanding of forecasting performance and model selection [9].

Visibility graph representations have already been used in forecasting and predictive modelling. This paper builds on that literature by empirically evaluating HVG-derived sliding-window descriptors across several standard machine learning models and daily M4 series. The analysis focuses on model-dependent performance and cross-series variability.

3 Methods

3.1 Dataset and Experimental Setup

The empirical evaluation is based on daily time series from the M4 forecasting dataset. The dataset is used here as a recognized public benchmark source, but the experiments do not follow the official M4 competition protocol. In particular, the official daily M4 horizon is 14 observations, whereas this study applies a longer forecasting horizon of h = 30 and an 80% - 20% train–test split. The reported results should therefore be interpreted within this experimental setup rather than as results directly comparable to the official M4 competition protocol.

The experiments use 100 daily M4 series selected by random sampling with a fixed random seed (random_state = 42) from the subset of daily series containing between 800 and 1500 observations. This restriction ensured that the selected series were long enough for the train–test split, sliding-window construction, and forecasting horizon, while keeping the sample comparable in length. The same fixed subset was used for all evaluated models.

Graph-based learning instances are generated by segmenting each series into overlapping sliding windows defined by a window size ratio (WSR). For each window, an HVG is constructed and graph-level structural descriptors are computed. In the representative configuration discussed in this paper, WSR = 0.8 is used. The D2596 series serves as an illustrative example, while the cross-series comparison reports aggregated results across the selected 100 daily M4 series.

3.2 HVG Transformation and Feature Extraction

To capture structural characteristics of time series, the observations are transformed into network representations using visibility graph techniques. The visibility graph framework, introduced by Lacasa et al. [5], provides a systematic procedure for converting time series into complex networks while preserving the temporal ordering of observations. A simplified variant known as the Horizontal Visibility Graph (HVG), proposed by Luque et al. [6], restricts visibility connections to horizontal lines and allows analytical results for several classes of time series. Due to its computational simplicity and well-defined structural properties, the HVG variant is used in this study. In the proposed approach, an HVG is constructed for each sliding-window segment, and structural graph descriptors are extracted to characterize the local dynamics of the series.

Figure 1 illustrates the transformation of a time series segment into a Horizontal Visibility Graph. In the HVG representation, each observation of the time series becomes a node, and edges are established between nodes if the corresponding observations satisfy the horizontal visibility criterion. More specifically, two observations are connected if all intermediate values between them are smaller than the minimum of the two endpoints. This transformation preserves the temporal ordering of the data while converting the time series into a network structure that can be analysed using graph-theoretical descriptors. In the left panel, the visibility relations are illustrated on a representative segment of a time series window, while the right panel displays the same structure using a standard network layout.

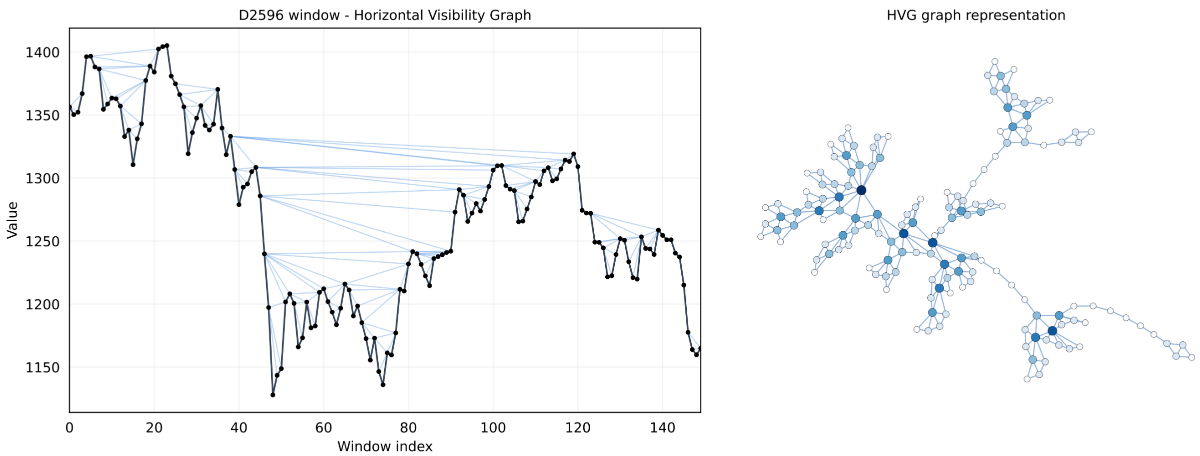

Figure 1: Example Horizontal Visibility Graph (HVG) constructed from a sliding window of the M4 daily time series D2596.

In the proposed framework, each time series is segmented into overlapping sliding windows. For every window, an HVG is constructed and a set of structural graph metrics is computed. These metrics summarize topological properties of the local graph representation and form the input feature matrix for the forecasting models.

Extracted graph features.

For each generated HVG, a set of structural graph metrics is computed in order to characterize the topology of the resulting network. The extracted feature set includes a diverse range of descriptors, such as degree-based measures, assortativity, centrality, and centralization metrics (e.g., betweenness, closeness, and PageRank), path-based measures (including average path length and diameter), efficiency-related metrics (local and global efficiency), entropy-based descriptors (degree entropy), and community-related measures (e.g., modularity). In addition, features describing network density, clustering structure, and vertex-level asymmetry are included.

Most node-level measures are aggregated at the graph level using summary statistics such as mean values or centralization indices in order to obtain fixed-length feature vectors for each sliding window. This results in a fixed-length feature vector, where each dimension corresponds to a graph-derived metric, capturing complementary structural properties of the underlying time series. All features are computed independently for each sliding window, ensuring that the feature extraction process avoids information leakage.

3.3 Forecasting Models

Two groups of forecasting models are evaluated in this study: classical statistical benchmarks and forecasting models trained on HVG-derived structural features.

Statistical baselines.

As reference models, two widely used statistical forecasting approaches are implemented: the Auto-Regressive Integrated Moving Average (ARIMA) model and the Holt–Winters exponential smoothing method. ARIMA models capture temporal dependencies through autoregressive and moving average components and often require differencing to achieve stationarity [1]. The Holt–Winters method extends exponential smoothing by incorporating level, trend, and seasonal components, making it suitable for modelling time series with evolving patterns.

Machine learning models on HVG features.

In the proposed framework, machine learning models operate on structural graph metrics extracted from the HVG representations of time series segments. These descriptors summarize topological properties of the generated graphs and serve as input features for forecasting models.

Several forecasting models are evaluated on HVG-derived structural features, including linear regression, Ridge regression, Lasso regression, ElasticNet, Support Vector Regression (SVR), Random Forest, and Extreme Gradient Boosting (XGBoost). These models are selected in order to represent a range of linear, regularized, kernel-based, and ensemble learning approaches. By training these algorithms on graph-derived structural features, the proposed methodology aims to exploit the structural information captured by the HVG transformation for forecasting purposes. All machine learning models were trained using the same set of HVG-derived features, and feature scaling was applied where required by the modelling approach.

3.4 Forecasting Pipeline Overview

To summarize the methodological workflow, the proposed forecasting framework consists of several sequential stages. First, time series are selected from the M4 dataset. Second, each time series is segmented into overlapping sliding windows in order to generate graph-based learning instances. Third, every window is transformed into a Horizontal Visibility Graph (HVG). Fourth, structural graph metrics are computed for each HVG and assembled into a feature matrix. Finally, these graph-based features serve as inputs for forecasting models whose forecasting performance is evaluated using standard error metrics. In this framework, forecasting is performed using supervised learning models trained on HVG-derived structural features rather than raw time series values.

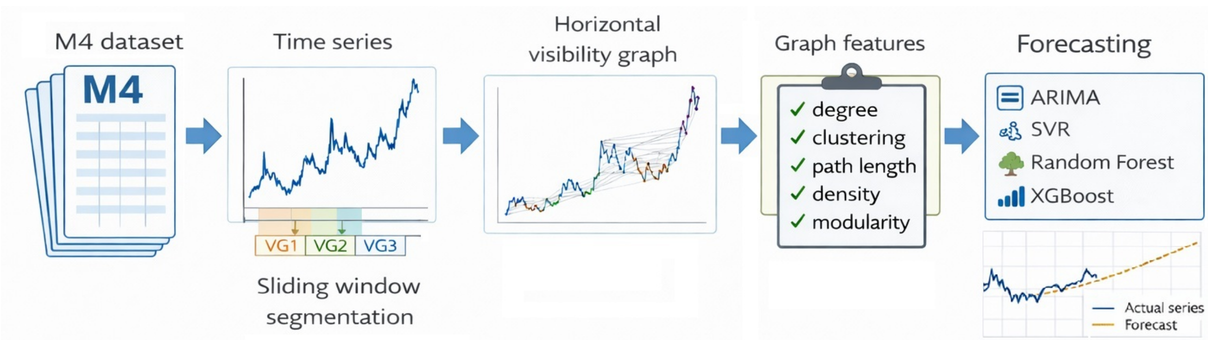

Figure 2: Overview of the HVG-based time series forecasting pipeline.

3.5 Data and Implementation Statement

The empirical analysis uses the publicly available M4 dataset. To improve transparency, the manuscript reports the main experimental settings, including the sampling rule, forecasting horizon, train–test split, window size ratio, evaluated models, and error metrics. The full implementation is part of an ongoing research project and is not released at this stage. Therefore, the framework is described as a transparent and re-implementable experimental workflow rather than as a fully reproducible software package.

4 Results

This section evaluates the forecasting performance of the proposed HVG-based framework. The results are presented in three steps. First, a representative example based on the D2596 series demonstrates the behaviour of the proposed pipeline. Second, a quantitative comparison is provided for the same configuration. Third, selected cross-series results are summarized in order to show how the relative performance of HVG-based models varies across daily M4 time series.

4.1 Forecasting Example

To illustrate the behaviour of the proposed framework, we present a representative forecasting example based on the M4 daily time series D2596.

The D2596 time series exhibits visible fluctuations and gradual changes in level that are typical for many real-world temporal processes. Such behaviour provides a suitable test case for evaluating both classical statistical forecasting models and machine learning approaches based on HVG-derived structural features.

The forecasting horizon is set to 30 observations. The time series is segmented using a window size ratio of WSR = 0.8, and the models are trained using an 80% training split.

Figure 3 shows the training fit of the evaluated models relative to the historical observations. This visualization illustrates how different forecasting approaches approximate the underlying structure of the time series during the training period. Due to the sliding-window construction and horizon, outputs are only available for later time indices.

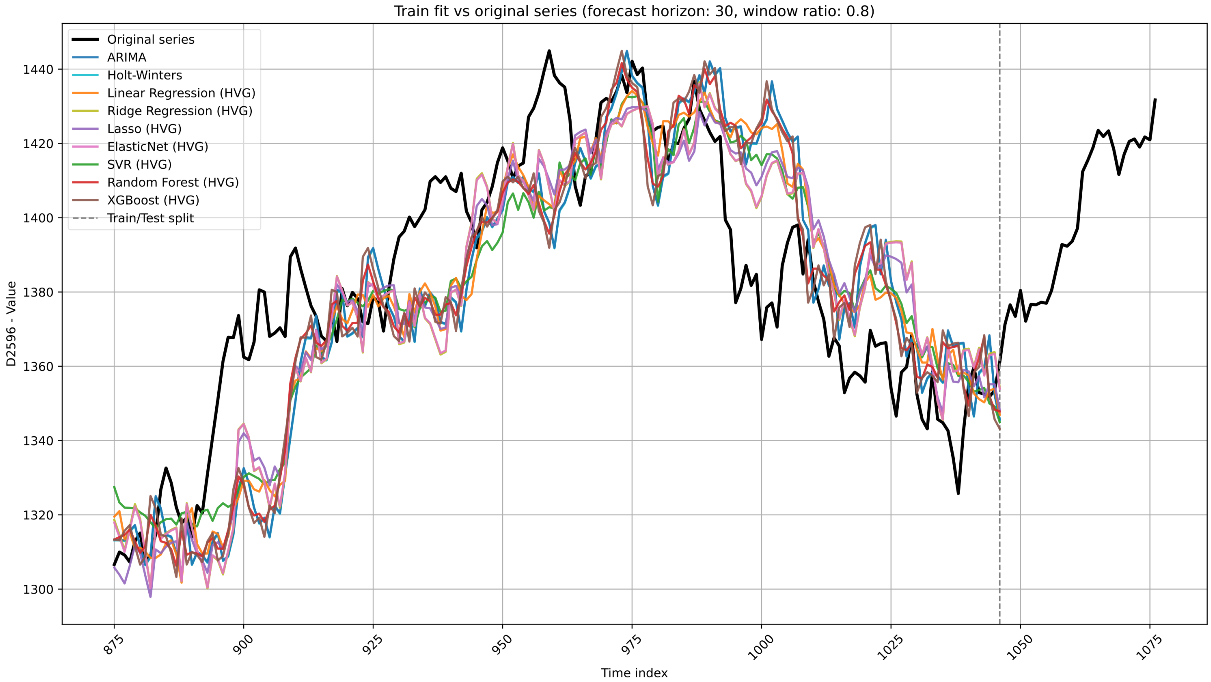

Figure 3: In-sample fit of the forecasting models for the M4 daily time series D2596 under h = 30, WSR = 0.8.

Figure 4 presents the resulting out-of-sample forecasts for the same configuration. Both classical statistical models (ARIMA and Holt–Winters) and HVG-based machine learning models are included. The forecasts illustrate that HVG-based models are able to track the overall trajectory of the series and, in several cases, improve upon the statistical benchmarks in the examined configuration.

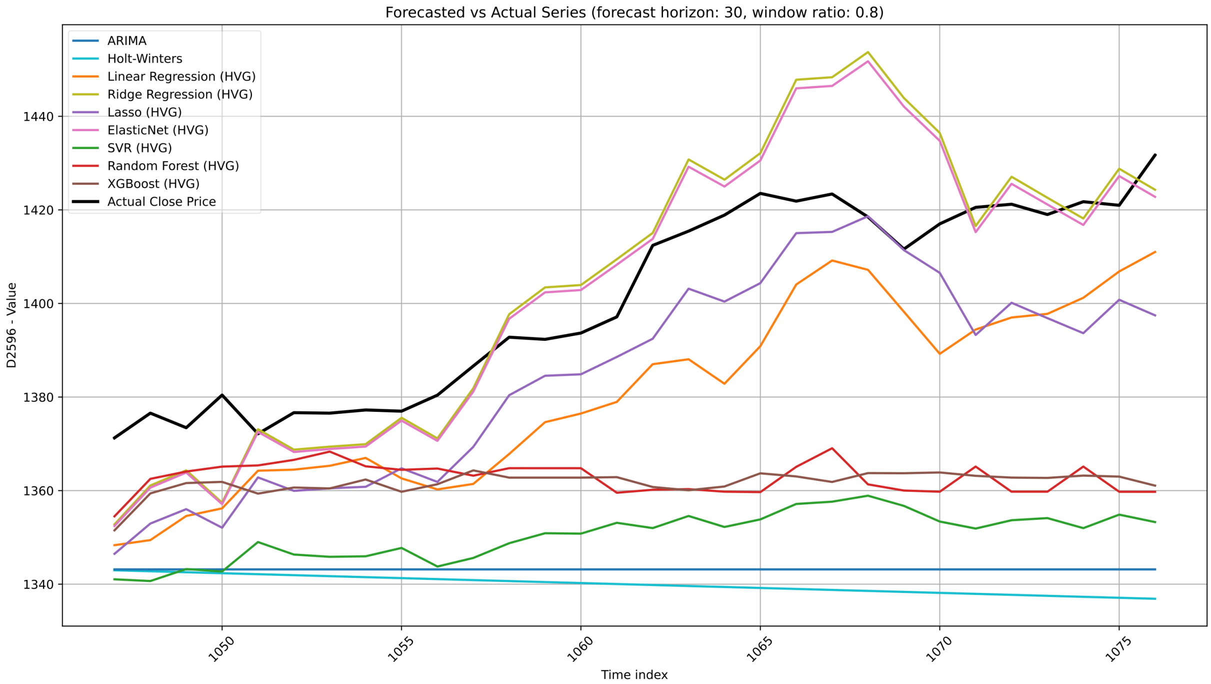

Figure 4: Out-of-sample forecast comparison for the M4 daily time series D2596 under h = 30, WSR = 0.8.

The example demonstrates the practical behaviour of the proposed HVG-based forecasting pipeline and highlights the diversity of predictive patterns produced by different modelling approaches.

4.2 Quantitative Comparison

Forecasting accuracy for the D2596 example under the same configuration is summarized in Table 1. Several standard evaluation metrics are reported, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), symmetric Mean Absolute Percentage Error (sMAPE), and Mean Absolute Scaled Error (MASE). Additionally, an Overall Weighted Average (OWA)-style relative accuracy measure is reported to summarize performance against the ARIMA benchmark within the present experimental setup. Since the forecasting horizon and train–test split differ from the official M4 competition protocol, the reported OWA values should be interpreted within this study rather than as directly comparable to results obtained under the official M4 protocol.

Table 1: Forecasting performance for the representative M4 daily time series D2596 under h = 30, WSR = 0.8.

Model	MAE	RMSE	sMAPE	MASE	OWA
Holt–Winters	60.806	64.661	4.427	14.978	1.057
ARIMA	57.578	61.044	4.186	14.183	1.000
SVR (HVG)	50.343	52.876	3.650	12.401	0.873
Random Forest (HVG)	37.986	43.632	2.739	9.357	0.657
XGBoost (HVG)	39.021	43.522	2.815	9.612	0.675
Lasso (HVG)	16.233	18.128	1.167	3.999	0.280
Ridge Regression (HVG)	11.592	14.607	0.823	2.856	0.199
ElasticNet (HVG)	**11.083**	**13.938**	**0.788**	**2.730**	**0.190**
Linear Regression (HVG)	20.178	21.278	1.450	4.970	0.348

The results show substantial variation across the evaluated models. In the examined configuration, ElasticNet trained on HVG-derived structural features achieves the best performance on all reported error metrics, closely followed by Ridge Regression. Within the experimental setup, the OWA-style value decreases from 1.000 for ARIMA to 0.190 for ElasticNet, indicating a strong improvement for this particular time series and configuration. The results also show that several other HVG-based models, including Lasso, Random Forest, and XGBoost, outperform both statistical baselines, suggesting that the HVG feature space captures useful predictive information. In contrast, standard Linear Regression, while still competitive in this example, is outperformed by its regularized counterparts.

4.3 Cross-Series Comparison

While the D2596 example provides a detailed illustration of the proposed approach, it is also important to examine performance across multiple time series. Table 2 presents aggregated results across 100 daily M4 series in order to assess the robustness of HVG-based forecasting models.

Table 2: Aggregated forecasting performance across 100 M4 daily time series under h = 30, WSR = 0.8. Median OWA, interquartile range (Q1-Q3), mean rank, and win rate are reported.

Model	Median OWA	Q1	Q3	Mean Rank	Win Rate (%)
Random Forest (HVG)	0.994	0.733	1.320	4.06	49.0
ARIMA	1.000	0.998	1.008	4.30	0.0
XGBoost (HVG)	1.021	0.765	1.455	4.35	45.0
Lasso (HVG)	1.053	0.659	2.148	4.74	43.0
Ridge Regression (HVG)	1.203	0.677	2.620	5.04	41.0
Holt–Winters	1.208	0.847	1.495	4.87	0.0
ElasticNet (HVG)	1.213	0.676	2.615	5.20	41.0
SVR (HVG)	1.404	0.689	2.450	5.13	39.0
Linear Regression (HVG)	2.872	1.414	6.671	7.31	15.0

The aggregated results provide a comprehensive view of model performance across a larger and more diverse set of daily time series. Mean rank is computed by ranking models per time series based on OWA and averaging these ranks across all series. Win rate is computed as the percentage of series where a model outperforms the best statistical baseline in terms of OWA. Among the evaluated approaches, tree-based ensemble methods exhibit the most robust performance within the HVG framework. Random Forest achieves the lowest median OWA (0.994), slightly outperforming the ARIMA benchmark, and also attains the highest win rate (49%). XGBoost shows similarly competitive behaviour, with a median OWA of 1.021 and a win rate of 45%, indicating that ensemble-based methods provide the most consistent improvements across time series.

Regularized linear models demonstrate moderate performance. Lasso achieves a relatively strong median OWA (1.053) and competitive win rate, while Ridge and ElasticNet exhibit higher variability, as reflected by wider interquartile ranges. These results suggest that while linear models can perform well in certain cases, their performance is less stable across heterogeneous time series.

A notable finding is the persistent instability of standard Linear Regression on HVG-derived features. The model exhibits a substantially higher median OWA (2.872) and a wide interquartile range, indicating poor robustness. This behaviour suggests that the HVG feature space may introduce multicollinearity or sensitivity to scaling, which negatively affects unregularized linear models, whereas regularized variants provide more stable performance, although not consistently superior across series.

The results indicate that HVG-based forecasting can outperform classical statistical methods in a substantial proportion of cases, particularly when combined with ensemble learning approaches. However, performance remains strongly model-dependent and varies across time series, highlighting the importance of model selection within the HVG-based forecasting framework.

4.4 Interpretation of Results

The results suggest that HVG-based structural descriptors may capture aspects of local time-series structure that are not directly represented by the raw observations alone. This is consistent with the relatively strong performance of ensemble methods, which can exploit interactions among graph-derived features.

The observed instability of unregularized linear models indicates that the HVG feature space may exhibit multicollinearity and varying feature relevance across series. Regularization mitigates these effects, leading to more stable but not uniformly superior performance.

HVG-derived features appear to be most effective when combined with models capable of handling nonlinearities and high-dimensional feature interactions, supporting their role as a complementary representation rather than a standalone replacement for classical approaches.

5 Conclusion

This study examined the applicability of forecasting time series using Horizontal Visibility Graph (HVG) representations. The proposed approach transforms sliding-window segments into graph structures, computes structural graph metrics, and uses these descriptors as input features for forecasting models.

The empirical analysis on daily M4 time series showed that HVG-derived features can support competitive forecasting performance. In the representative example based on the D2596 series, multiple HVG-based models, particularly regularized linear models, outperformed the classical statistical benchmarks.

The cross-series evaluation across 100 time series shows a more heterogeneous pattern: ensemble-based methods, particularly Random Forest and XGBoost, exhibit the most consistent performance across series, while unregularized linear models show instability and regularized variants provide more robust but not consistently superior results.

Overall, the results indicate that HVG-derived structural descriptors can provide an informative and complementary feature space for time series forecasting. Their usefulness, however, is model-dependent and varies across series. The findings should therefore be interpreted as evidence of conditional applicability rather than universal superiority over classical benchmarks. In practice, HVG-based models are most relevant in comparative forecasting workflows, where they are evaluated alongside statistical baselines.

The proposed version of the framework produces point forecasts only. Its computational cost may also increase for larger datasets or high-frequency time series, where HVG construction and graph-feature extraction have to be repeated many times. Future research should examine computational scalability and extend the framework beyond point forecasts, for example through prediction intervals or density forecasting. Further work may also consider additional graph descriptors, alternative graph construction methods, broader M4 subsets, hybrid forecasting approaches, and real-world economic applications where the interpretation of graph-derived features can be assessed more directly.

References

[1] Box, G. E. P., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

[2] Hamilton, W., Ying, Z. & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in neural information processing systems, 30. doi:10.48550/arXiv.1706.02216.

[3] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8):1735-1780. doi:10.1162/neco.1997.9.8.1735.

[4] Kosztyán, Zs. T., Király, F., Katona, A. I., Csizmadia, T. & Fehérvölgyi, B. (2024). Analysis and prediction of the horizon 2020 r&d&i collaboration network. Expert Systems with Applications, 255:124417. doi:10.1016/j.eswa.2024.124417.

[5] Lacasa, L., Luque, B., Ballesteros, F., Luque, J. & Nuño, J. C. (2008). From time series to complex networks: The visibility graph. Proceedings of the National Academy of Sciences, 105(13):4972-4975. doi:10.1073/pnas.0709247105.

[6] Luque, B., Lacasa, L., Ballesteros, F. & Luque, J. (2009). Horizontal visibility graphs: Exact results for random time series. Physical Review E, 80(4):046103. doi:10.1103/PhysRevE.80.046103.

[7] Makridakis, S., Spiliotis, E. & Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. International Journal of forecasting, 34(4):802-808. doi:10.1016/j.ijforecast.2018.06.001.

[8] Mao, S. & Zeng, X.-J. (2023). Simvgnets: similarity-based visibility graph networks for carbon price forecasting. Expert Systems with Applications, 230:120647. doi:10.1016/j.eswa.2023.120647.

[9] Petelin, G., Cenikj, G. & Eftimov, T. (2023). Towards understanding the importance of time-series features in automated algorithm performance prediction. Expert Systems with Applications, 213:119023. doi:10.1016/j.eswa.2022.119023.

[10] Ramezanpoor, Z., Ghazikhani, A. & Bajestani, G. S. (2024). A generalized visibility graph algorithm for analyzing biological time series having rotation in polar plane. Engineering Applications of Artificial Intelligence, 128:107557. doi:10.1016/j.engappai.2023.107557.

[11] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. (2008). The graph neural network model. IEEE Transactions on Neural Networks, 20(1):61-80. doi:10.1109/TNN.2008.2005605.

[12] Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., Tang, C. & Li, J. (2022). A survey on machine learning models for financial time series forecasting. Neurocomputing, 512:363-380. doi:10.1016/j.neucom.2022.09.003.

[13] Wen, T., Chen, H. & Cheong, K. H. (2022). Visibility graph for time series prediction and image classification: a review. Nonlinear Dynamics, 110(4):2979-2999. doi:10.1007/s11071-022-08002-4.

[14] Zhang, R., Ashuri, B. & Deng, Y. (2017). A novel method for forecasting time series based on fuzzy logic and visibility graph. Advances in Data Analysis and Classification, 11(4):759-783. doi:10.1007/s11634-017-0300-3.