mathematics

MDPI

Article

Regime-Dependent Graph Neural Networks for Enhanced Volatility Prediction in Financial Markets

Pulikandala Nithish Kumar [1], Nneka Umeorah [1,*] and Alex Alochukwu [2]

[1] School of Mathematics, Cardiff University, Cardiff CF24 4AG, UK; pulikandalan@cardiff.ac.uk
[2] Department of Mathematics, Computer Science and Physics, Albany State University, Albany, GA 31705, USA; alex.alochukwu@asurams.edu
* Correspondence: umeorahn@cardiff.ac.uk

Abstract

Accurate volatility forecasting is essential for risk management in increasingly interconnected financial markets. Traditional econometric models capture volatility clustering but struggle to model nonlinear cross-market spillovers. This study proposes a Temporal Graph Attention Network (TemporalGAT) for multi-horizon volatility forecasting, integrating LSTM-based temporal encoding with graph convolutional and attention layers to jointly model volatility persistence and inter-market dependencies. Market linkages are constructed using the Diebold–Yilmaz volatility spillover index, providing an economically interpretable representation of directional shock transmission. Using daily data from major global equity indices, the model is evaluated against econometric, machine learning, and graph-based benchmarks across multiple forecast horizons. Performance is assessed using MSE, R^2, MAFE, and MAPE, with statistical significance validated via Diebold–Mariano tests and bootstrap confidence intervals. The study further conducts a strict expanding-window robustness test, comparing fixed and dynamically re-estimated spillover graphs in a fully out-of-sample setting. Sensitivity and scenario analyses confirm robustness across hyperparameter configurations and market regimes, while results show no systematic gains from dynamic graph updating over a fixed spillover network.

Keywords: GARCH model; graph neural network; temporal GAT; volatility cluster; optimization; LSTM; volatility spillover

MSC: 91G15; 91G70; 62M10; 68T07; 05C81

Check for updates

Academic Editors: Chihwa Kao and Zhonghui Zhang

Received: 14 November 2025
Revised: 6 January 2026
Accepted: 8 January 2026
Published: 13 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Volatility forecasting is essential for effective risk management, portfolio optimization, and informed decision-making in global financial markets. Volatility reflects the degree of variation in asset prices over time and acts as a fundamental indicator of financial risk and market uncertainty. A particularly challenging characteristic of volatility is its tendency to cluster, whereby periods of high volatility are typically followed by further turbulence, and periods of calm tend to persist [1]. These volatility clusters, often triggered by macroeconomic shocks, news events, or shifts in market sentiment [2], complicate the forecasting landscape and heighten the need for robust predictive models that can adapt to evolving market regimes.

Traditional econometric models such as the generalized autoregressive conditional heteroskedasticity (GARCH) family have long been employed to capture stylized features

of financial returns, including volatility clustering and leptokurtosis. While these models provide valuable insights into time-varying volatility dynamics, they struggle to account for the nonlinear interdependencies and spillover effects that characterize increasingly interconnected global markets. As financial systems become more integrated, volatility in one market can rapidly propagate to others, demanding modelling frameworks that capture not only individual market behaviour but also the structural relationships that govern cross-market transmission.

Graph Neural Networks (GNNs) offer a powerful means of representing such interconnected systems by exploiting the graph structure underlying financial markets. Their ability to model dependencies between nodes makes them particularly suitable for volatility forecasting, where spillovers and contagion effects play a critical role. Recent studies have demonstrated the promise of GNNs for incorporating network information to enhance predictive accuracy. For example, Son et al. [3] shows that spatiotemporal GNNs combined with volatility spillover indices can improve volatility forecasts across global markets, underscoring the importance of embedding financial network structure into predictive models.

Building on these developments, this paper proposes a tailored framework for volatility forecasting: the Temporal Graph Attention Network (Temporal GAT). Our approach represents global equity markets as dynamic graphs, where nodes correspond to stock indices and edges encode time-varying interdependencies derived from either correlation networks or, more effectively, the Diebold–Yilmaz volatility spillover index. The proposed architecture adopts a temporal-first design: a Long Short-Term Memory (LSTM) first encodes the history of each index's volatility proxies, after which Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) layers capture structural spillover effects and assign adaptive attention weights to influential markets. This separation of temporal and spatial learning allows the model to capture sequential volatility dynamics alongside evolving cross-market relationships more precisely.

A key contribution of this work is the demonstration that volatility spillover networks outperform traditional correlation networks in capturing the directional transmission of market shocks. Our empirical analysis shows that spillover-based graphs yield more accurate forecasts by modelling the asymmetric and dynamic nature of financial contagion. Furthermore, the Temporal GAT is evaluated using a comprehensive set of experiments over fifteen years of data from eight major global stock indices. The model is assessed across multiple forecast horizons, from short-term to monthly volatility predictions, and benchmarked against GARCH, MLP, LSTM, and alternative GNN architectures. Extensive sensitivity and scenario analyses are conducted to evaluate robustness under different hyperparameter configurations and varying market regimes. These analyses reveal that while the model produces higher prediction errors during turbulent periods—an expected outcome given increased market unpredictability—it remains stable and maintains strong relative performance compared to competing methods.

Overall, this study advances the literature by presenting a domain-specific LSTM–GCN–GAT hybrid architecture that aligns closely with the econometric structure of volatility transmission. By integrating temporal encoding with piecewise-static spillover-based graph modelling, the Temporal GAT provides a more nuanced and accurate representation of global market interdependencies. The rest of the paper is organized as follows: Section 2 introduces the literature studies in terms of volatility clustering, machine learning approaches to volatility forecasting, GNN in financial modelling and the concepts of volatility spillovers and the GNNs. Section 3 introduces the mathematical and methodological preliminaries, Section 4 presents the full methodology, Section 5 details the empirical results,

including data visualization, sensitivity and scenario analyses, and the final section offers concluding remarks.

2. Related Works

Accurate forecasting of financial market volatility has been a central theme in financial econometrics; yet it remains a significant challenge due to the complex, non-linear, and interconnected nature of global markets. This section reviews the evolution of financial and volatility modelling, as well as the impact of the GNNs in such modelling.

2.1. Traditional Volatility Models and Early Machine Learning Approaches

Early efforts in volatility modelling were primarily dominated by econometric models. The concept of volatility clustering, where periods of high volatility are followed by periods of high volatility and vice versa, was first observed by [4] in 1963 and further developed by [5] with the introduction of the Autoregressive Conditional Heteroskedasticity (ARCH) model. Engle's work demonstrated that volatility is not constant over time but can be modelled based on past behaviours, marking a significant shift in understanding financial time series. Building on this, Bollerslev [6] extended the ARCH model by proposing the GARCH model in 1986. The GARCH model improved upon its predecessor by incorporating past volatilities in addition to past shocks, allowing for more accurate forecasting of future volatility. Traditionally, models like the GARCH model have been widely used to capture volatility clustering by analyzing time-series data. While effective to some extent, these models often fail to account for the more complex, non-linear relationships among multiple assets, or even among networks of interdependent assets, in global financial markets [7]. This limitation becomes particularly evident when markets are highly interconnected, and the volatility of one market affects others through intricate spillover effects [8].

Several variations of the GARCH model have been developed to account for specific market behaviours. For instance, the Exponential GARCH (EGARCH) model introduced by [7] addresses asymmetries in financial data, particularly the tendency for adverse market shocks to increase volatility more than positive ones. Similarly, the Threshold GARCH (TGARCH) model proposed by [9] captures the idea that markets respond differently to different types of shocks. Despite their widespread adoption, GARCH models have notable limitations. They often rely heavily on historical data, which can be problematic in rapidly evolving markets. Moreover, they assume that volatility follows a single process, overlooking the complex, interconnected relationships among global financial markets [8].

These limitations prompted the exploration of alternative approaches, including machine learning (ML) techniques, which have gained prominence due to their ability to model complex, non-linear relationships. Andersen et al. [10] demonstrated the effectiveness of realized volatility in providing more accurate volatility estimates by utilizing high-frequency intraday data. Additionally, the concept of volatility spillovers, as measured by the Volatility Spillover Index, has been instrumental in understanding how volatility in one market can influence others. Neural networks, such as LSTMs, have been successfully applied to volatility forecasting, demonstrating improved capacity to capture long-term dependencies in financial data [11,12]. Hybrid models combining machine learning with GARCH have also shown effectiveness in various markets, including energy, metals, and stock markets [13]. Other ML techniques, such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and Random Forests, have been utilized in financial forecasting. While ANNs are effective at capturing non-linear patterns, they often require extensive datasets and are prone to overfitting [14,15]. SVMs have been applied for market condition classification, and Random Forests have shown robustness, especially with large datasets.

However, a common drawback of many of these models is their tendency to treat assets independently, thereby failing to capture the inherent interconnectedness of global markets.

2.2. Graph Neural Networks in Financial Modelling

To address the challenge of interconnectedness, GNNs have emerged as powerful tools for modelling complex graph-structured systems, such as financial markets. GNNs are designed to operate on graph-structured data, providing a sophisticated way to model time-series patterns and relationships among market indices by treating them as nodes within a dynamic graph. Each edge in this graph represents a relationship or correlation between indices, capturing the dynamic interdependencies between global markets. Traditional machine learning applications handle graph-structured data by applying a preprocessing step that transforms the graph's structure into a more straightforward form, such as real-valued vectors. However, crucial information, such as the topological relationships between nodes, can be lost during preprocessing. As a result, the outcome may be unpredictably influenced by the specifics of the preprocessing algorithm [16]. Other approaches that attempt to preserve the graph-structured nature of the data prior to the processing phase have been investigated by researchers in [17–19]. This capability is crucial, as traditional machine learning methods often lose vital topological information when converting graph data into simpler forms.

In recent years, GNNs have gained attention, especially in financial modelling and research, due to their capacity to capture both structural and temporal dependencies evident in a financial system. This ability to account for both structural topology and temporal dynamics makes GNNs a powerful tool for tasks such as risk assessment, fraud detection, stock price prediction, and portfolio optimization in modern finance. Several comprehensive reviews on GNNs have been published. Bronstein et al. [20] offers an in-depth review of geometric deep learning, discussing key challenges, solutions, applications, and future directions, whereas [21] provides a detailed overview of graph convolutional networks. Other recent papers on GNN models are presented by researchers in [22–25].

The foundational work by [19] introduced a framework for applying neural networks to graph-structured data, laying the groundwork for iterative updates of node representation based on neighbour information. K. Xu et al. [26] provided a theoretical framework for analyzing the representational power of GNNs and their variants, whereas [3] recently demonstrated how GNNs, combined with a volatility spillover index, can improve the predictive accuracy of stock market volatility in different regions. GCNs aggregate information from neighbouring nodes, updating node features based on the graph's topology, making them effective for semi-supervised learning. GCNs can also incorporate temporal dynamics, which is vital for fast-paced financial markets [27]. Applications of GCNs in financial modelling include predicting stock prices using correlation graphs, as demonstrated by [28], who combined a GCN with a gated recurrent unit (GRU) to capture temporal dependencies. Other recent work on GCN implementations in finance is found in [29–33].

Further on improving GCNs, Veličković et al. [34] developed the GATs, which integrate attention mechanisms to dynamically assign weights to neighbouring nodes, enabling the model to focus on the most relevant connections. This innovation is beneficial in multivariate time series prediction [35], financial fraud detection [36], and other financial modelling frameworks in which specific market indices exert greater influence than others. GATs utilize attention mechanisms to assign varying importance to nodes within a graph, making them practical for modelling relationships between financial entities. The model enhances the traditional methods of analyzing financial data by incorporating both the GCNs and GATs [32]. This allows the model to focus on the most relevant relationships between indices, improving the accuracy of predictions related to volatility clustering.

However, traditional GATs struggle to capture temporal dependencies in data, which is crucial in the financial domain where time plays a key role.

To address this limitation, Temporal GATs have been developed to enhance GATs by incorporating temporal data. Using a temporal attention mechanism, the temporal GAT dynamically adjusts the impact of past events, allowing the model to capture not only the structure of the financial graph but also the time-dependent patterns that shape market behaviour [37,38]. This phenomenon makes temporal GATs particularly effective for financial applications, where it is crucial to account for both entity relationships and their changes over time. Unlike traditional models that consider only individual asset behaviours, the Temporal GAT model enables a deeper understanding of how assets influence one another, providing a more comprehensive analysis of market volatility. This research aims to push the boundaries of financial econometrics and machine learning by applying GNNs to better understand volatility clustering and predict it. By capturing the interconnected nature of global financial markets, this paper offers new tools for financial analysts, risk managers, and investors to improve risk assessment and make more informed decisions [39].

2.3. Volatility Spillovers and Graph-Based Models

Understanding volatility spillovers, that is, the phenomenon where volatility in one market influences volatility in another, is crucial for comprehending systemic risk and market dynamics. Diebold; Yilmaz [8,40] developed a framework to measure these spillovers using Vector Autoregressive (VAR) models. However, traditional VAR-based measures may not fully capture non-linear relationships. To overcome these limitations, recent research has increasingly turned to GNNs to model the global financial system as a dynamic graph. By representing markets as nodes and volatility relationships as edges, GNNs provide a more comprehensive understanding of volatility spillovers. A notable contribution in this area is the work of [3], who demonstrated that incorporating volatility spillover indices into a spatial-temporal GNN framework significantly improves the accuracy of volatility predictions across global markets. Their research highlighted the effectiveness of combining GNNs with spillover indices to enhance predictive accuracy for stock market volatility.

2.4. Contributions of This Research

While [3] laid the necessary groundwork by applying spatial-temporal GNNs with volatility spillovers, our research extends this knowledge through a more specialised and theoretically aligned modelling framework. In contrast to the general "spatial–temporal GNN model" examined by [3], the proposed TemporalGAT adopts a *temporal-first* architecture in which an LSTM encodes each market's volatility history before any spatial aggregation occurs. This separation enables more precise and more accurate modelling of sequential volatility dynamics. Furthermore, integrating graph attention layers enables the model to learn adaptive, time-varying importance weights for neighbouring markets, capturing heterogeneity in spillover strengths that fixed diffusion-based approaches cannot represent. Finally, the study incorporates extensive sensitivity, robustness, and scenario analyses, demonstrating that the TemporalGAT remains stable across hyperparameter configurations and maintains strong relative performance under both high- and low-volatility regimes. These distinctions collectively advance the methodological precision of temporal GNN modelling for financial applications and address key limitations in prior work.

The key contributions of this paper are as follows:

- **Development of a TemporalGAT for Volatility Forecasting:** This paper presents a novel TemporalGAT architecture tailored for volatility prediction in global stock markets. By combining LSTM-based temporal encoding with GCN and GAT layers,

the model effectively captures both sequential dependence and evolving cross-market spillovers, outperforming traditional approaches such as GARCH and MLP.

- **Volatility Spillover Index as a Superior Graph Construction Method:** The study demonstrates that constructing graphs using the Diebold–Yilmaz volatility spillover index significantly improves the modelling of directional shock transmission relative to conventional correlation-based methods, thereby enhancing forecasting accuracy and interpretability.
- **Strict Expanding Window Robustness:** Beyond the standard validation-based evaluation, the study introduces a strict expanding-window robustness check that compares fixed and dynamically re-estimated spillover graphs in a fully out-of-sample setting, showing that dynamically updating the graph structure does not yield systematic forecasting gains over a carefully constructed fixed graph, thereby strengthening the credibility and stability of the proposed modelling approach.
- **Comprehensive Sensitivity and Scenario Analysis:** The model is evaluated under a wide range of hyperparameter settings and market regimes, including both high- and low-volatility periods. These analyses highlight the robustness of the Temporal-GAT architecture and provide practical guidance for its application across diverse financial environments.

The practical significance of this research lies in its ability to capture and model complex interdependencies between multiple indices, enhance volatility predictions, and optimize risk management techniques. Traditional models like GARCH focus on each asset's volatility but fail to account for how volatility in one asset can spill over to other assets and vice versa. GNNs address this limitation by representing assets as nodes in a graph, with edges reflecting relationships such as correlations or mutual volatility impacts. This allows GNNs to capture interconnected market behaviour more effectively, yielding more accurate volatility-clustering models. Some of the practical benefits are seen in modelling complex asset relationships, cross-asset and multi-asset volatility spillovers, real-time systematic risk monitoring in terms of financial contagion, algorithmic trading and portfolio optimization for risk-adjusted returns, as well as handling correlation breakdowns in the face of market turmoil.

3. Preliminaries

This section introduces the fundamental concepts and methodologies essential for understanding the subsequent analysis of volatility clustering using GNNs. The topics covered include volatility proxy, the GARCH model, correlation, the volatility spillover index, and the architectures of GCNs and GATs. These concepts form the backbone of the proposed Temporal GAT model.

3.1. Volatility and Correlation Analytics

In the absence of intraday high-frequency data, we construct a *daily volatility proxy* based on the squared daily return. Let $P_{i,t}$ denote the closing price of asset i on day t. The daily log return is defined as

$$r_{i,t} = \ln(P_{i,t}) - \ln(P_{i,t-1}). \tag{1}$$

Following standard practice when only daily observations are available, we compute volatility using the squared daily return:

$$\widehat{\sigma}^2_{i,t} = r^2_{i,t}. \tag{2}$$

We refer to $\widehat{\sigma}_{i,t}^2$ as a *volatility proxy* rather than *realized volatility* (The term "realized volatility" is typically reserved for volatility measures constructed from high-frequency intraday returns, such as the sum of squared intraday price changes (e.g., [10]), and this is considered one limitation of this study). Despite this limitation, squared daily returns remain a widely used approximation in empirical finance when intraday data are unavailable. Furthermore, because all benchmark models in our study utilize the same volatility proxy, our comparative evaluation remains consistent and informative regarding relative predictive performance under daily-frequency constraints.

3.1.1. Generalized Autoregressive Conditional Heteroskedasticity (GARCH)

The GARCH model, introduced by [6], is a widely used statistical model for estimating volatility in financial time series data. It extends the ARCH model by incorporating both past squared returns and past variances [9], allowing for a more flexible and accurate modelling of volatility clustering [8].

Let ϵ_t be the real-valued discrete-time stochastic process, and f_t as the information set ($\sigma-$field) of all information through time t, then the standard GARCH (p, q) model is defined as

$$r_t = \mu + \epsilon_t, \quad \epsilon_t | f_{t-1} \sim N(0, h_t) \tag{3}$$

$$h_t = \omega_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j} \tag{4}$$

where (note: for $q = 0$, the process reduces to the ARCH(p) process and for $p = q = 0$, ϵ_t becomes the white noise)

$$\begin{cases} q \geq 0, \ p > 0 \\ \omega_0 > 0; \ \alpha_i \geq 0, \ \text{for } i = 1, 2, \cdots \ p \\ \beta_j \geq 0; \ \text{for } j = 1, 2, \cdots \ q \end{cases}$$

and

- r_t is the return at time t.
- μ and ϵ_t are the mean return and the error term, respectively.
- h_t is the conditional variance (volatility) at time t.
- ω, α, and β are parameters to be estimated.

The GARCH model captures volatility clustering by allowing the current variance to depend on both past squared errors (ϵ_{t-1}^2) and past variance (h_{t-1}). Since 2013, GARCH modelling has seen significant advancements, particularly with the emergence of non-linear and hybrid approaches that enhance volatility forecasting. Recent work by [41] introduces graph-based multivariate GARCH models that capture complex dependencies in time series data. Additionally, neural network-augmented GARCH models, such as Neural GARCH [42] and deep learning-enhanced realized GARCH variants [43], have improved the ability to model dynamic, non-linear volatility patterns. Other innovations include ordinal GARCH models, which allow for a flexible structure of serial dependence [44], as well as hybrid frameworks that combine GARCH with deep learning techniques [43], which have demonstrated superior performance in capturing regime shifts and long-memory effects.

3.1.2. Volatility Spillover Index

The Volatility Spillover Index measures the extent to which volatility shocks can transfer from one market to another, reflecting the interconnectedness of the global fi-

nancial market. Diebold and Yilmaz (2009, 2012) developed a framework using variance decompositions from the VAR models to quantify these spillovers [8,40]. This measure captures the transmission of volatility across financial assets or markets and provides a more dynamic and directional understanding of market interconnectedness than static correlation measures.

The calculation of the volatility spillover index typically involves a VAR model applied to a set of time series (in our case, volatility proxies of multiple market indices). The forecast error variance decomposition from the VAR model is then used to determine the contribution of shocks from market j to the forecast error variance of market i. This contribution forms the basis for the weights in our graph construction, w_{ji}, representing the spillover from market j to market i. The spillover index is calculated using the forecast error variance decompositions from the VAR model [40]:

$$\theta_{ij}^{g}(H) = \frac{\sigma_{jj}^{-1} \sum_{h=0}^{H-1} \left(e_i' A_h \sum e_j\right)^2}{\sum_{h=0}^{H-1} \left(e_i' A_h \sum A_h' e_i\right)} \tag{5}$$

where

- H is the forecast horizon.
- σ_{jj} is the standard deviation of the error term for variable j.
- e_i is the selection vector with one at the $i-th$ position and zeros elsewhere.
- A_h is the coefficient matrix at lag h.
- $\sum$ is the covariance matrix of the error terms.

The total spillover index is then given by

$$S^{g}(H) = \frac{\sum_{i,j=1;i\neq j}^{N} \tilde{\theta}_{ij}^{g}(H)}{\sum_{i,j=1} \tilde{\theta}_{ij}^{g}(H)} \times 100 = \frac{\sum_{i,j=1;i\neq j}^{N} \tilde{\theta}_{ij}^{g}(H)}{N} \times 100 \tag{6}$$

Note: To calculate the spillover index using the information available in the variance decomposition matrix, each matrix entry is normalized by

$$\tilde{\theta}_{ij}^{g}(H) = \frac{\theta_{ij}^{g}(H)}{\sum_{j=1}^{N} \theta_{ij}^{g}(H)}$$

This index provides insights into how volatility in one market influences others, which is essential for understanding systemic risk and market dynamics. The full derivation is provided in Appendix A.

3.2. Graph Theory Fundamentals

Our methodology leverages graph theory to model the interdependencies within global financial markets. A graph is formally defined as $G = (V, E)$, where V is a set of nodes (or vertices), representing the individual market indices in our context and E is a set of edges (or links), representing the relationships or connections between these market indices.

Each edge $(i, j) \in E$ indicates a relationship between node i and node j. In a weighted graph, each edge (i, j) is associated with a numerical weight w_{ij}, quantifying the strength or nature of the relationship. In a directed graph, edges have a specific direction (e.g., $i \to j$), meaning the relationship from i to j is distinct from j to i. The connectivity of a graph is represented by its adjacency matrix A, where $A_{ij} > 0$ if an edge exists from i to j, and $A_{ij} = 0$ otherwise. For weighted graphs, $A_{ij} = w_{ij}$.

3.3. Graph-Based Deep Learning Models

3.3.1. Graph Neural Networks (GNNs)

GNNs are a class of neural networks designed to operate on graph-structured data, capturing dependencies among nodes via message passing between nodes. GNNs are particularly useful for modelling relational data and have been successfully applied in various domains, including social networks, recommendation systems, and financial markets.

3.3.2. Graph Convolutional Networks (GCNs)

GCNs aggregate feature information from a node's neighbours to compute its new representation [32]. The first spatial component of the Temporal GAT architecture is a GCN layer, which captures coarse structural relationships among market indices. The GCN operates on the spillover network encoded by the Diebold–Yilmaz adjacency matrix, where edges represent the magnitude and direction of cross-market volatility transmission. The layer-wise propagation rule for a multilayer GCN is given by

$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}H^{(l)}W^{(l)}\right) \tag{7}$$

where

- $H^{(l)}$ is the feature matrix at layer l; $H^{(l)} \in \mathbb{R}^{N\times D}$.
- $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph $\mathcal{G}$ with added self-loops, and I_N is the identity matrix.
- $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.
- $W^{(l)}$ is the layer-specific trainable weight matrix.
- σ is an activation function (e.g., ReLU).

GCNs effectively capture local neighbourhood structures in graphs, making them suitable for semi-supervised learning tasks on graph-structured data. In the proposed architecture, the GCN layer takes as input the temporal embeddings generated by the LSTM and transforms them into intermediate spatial representations. This step allows the model to incorporate structural connectivity in the spillover network before applying more expressive attention-based refinements. Consequently, the GCN layer acts as a foundational spatial encoder, capturing general cross-market interactions and preparing the representations for subsequent graph attention operations.

3.3.3. Graph Attention Networks (GATs)

GATs introduce an attention mechanism to GNNs, allowing the model to assign different importance weights to different nodes in a neighbourhood. Following the GCN layers, the architecture incorporates GAT layers to learn heterogeneous spillover effects via attention-based message passing. While GCNs treat neighbouring nodes uniformly, GATs assign adaptive weights to neighbours based on their relevance, enabling the model to emphasize influential volatility transmitters. The core idea is to compute attention coefficients σ_{ij} that indicate the importance of node j's features to node i [34,45].

$$\vec{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right) \tag{8}$$

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLu}\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i||\mathbf{W}\vec{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLu}\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i||\mathbf{W}\vec{h}_k\right]\right)\right)} \tag{9}$$

where

- $\mathbf{h} = \vec{h_1}, \vec{h_2}, \cdots, \vec{h_N}; \vec{h_i} \in \mathbb{R}^F$ are the input, with N = number of nodes and F, the number of features in each node.
- $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is the weight matrix.
- $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$ is the attention mechanism's weight vector.
- $||$ denotes concatenation and $[.]^T$ is the transposition.
- $\mathcal{N}_i$ is the set of neighborhood of node i in the graph.

Through this mechanism, GAT layers learn to assign higher importance to nodes that exert stronger spillover effects, such as major global indices or structurally central markets. Multi-head attention further stabilizes the learning process and improves expressiveness by allowing the model to attend to different relational patterns simultaneously. In the Temporal GAT architecture, GAT layers refine the spatial embeddings produced by the preceding GCN layers, enabling the network to capture asymmetric, time-varying, and non-uniform volatility transmission. This makes the GAT component essential for modelling complex cross-market dynamics that uniform graph convolutions cannot capture.

3.3.4. Temporal Graph Attention Network (Temporal GAT)

The Temporal GAT combines the strengths of GCNs and GATs to model both the structural and temporal dynamics in graph-structured financial data [46]. This architecture is well-suited to tasks in which relationships among nodes evolve and in which capturing temporal behaviour is essential for modelling volatility clustering and spillovers. The model consists of three key components, which individually contribute to learning spatio-temporal dependencies across financial markets:

- **Temporal Layers:** Model the evolution of volatility proxies for each market index over time.
- **GCN Layers:** Aggregate structural information from neighbouring indices based on the spillover graph.
- **GAT Layers:** Assign adaptive attention weights to spillover linkages, emphasizing the most influential cross-market transmissions.

Temporal Layer: *Explicit LSTM-Based Temporal Modelling–TemporalGAT Module (LSTM + GCN + GAT).*

For each node i, we construct a rolling window of length w from the volatility proxy series. The input sequence at time t is given by

$$X_{i,t-w+1:t} = [x_{i,t-w+1}, \ldots, x_{i,t}], \tag{10}$$

where $x_{i,t}$ is the one-dimensional volatility feature at time t. This sequence is processed by an LSTM network with hidden size d_h:

$$(h_{i,t}, c_{i,t}) = \mathrm{LSTM}\big(X_{i,t-w+1:t}\big), \tag{11}$$

and we take the final hidden state $h_{i,t} \in \mathbb{R}^{d_h}$ as the temporal representation of node i at time t. Collecting all nodes, we form the matrix H_t, which serves as input to the graph module.

$$H_t = \begin{bmatrix} h_{1,t}^\top \\ \vdots \\ h_{N,t}^\top \end{bmatrix} \in \mathbb{R}^{N \times d_h}, \tag{12}$$

The spatial block consists of two GCN layers followed by two GAT layers, applied on the spillover (or correlation) network. Let $\tilde{A}$ denote the normalized adjacency matrix and $W^{(0)}, W^{(1)}$ the trainable weight matrices of the GCN layers. We compute

$$H_t^{(1)} = \sigma\big(\tilde{A} H_t W^{(0)}\big), \tag{13}$$

$$H_t^{(2)} = \sigma\big(\tilde{A} H_t^{(1)} W^{(1)}\big), \tag{14}$$

where $\sigma(\cdot)$ is a nonlinear activation function.

Next, we apply graph attention to the GCN features $H_t^{(2)}$. For nodes i and j, the attention coefficient is given by Equation (9). The attention-weighted node representation is given by

$$H_t^{(3)}(i) = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W H_j^{(2)}\right), \tag{15}$$

and a second GAT layer is then applied analogously to refine the node embeddings further, yielding $H_t^{(\mathrm{GAT})}$.

Finally, the output of the graph block is passed through fully connected layers to produce either a single-horizon forecast (e.g., $t+15$) or a vector of multi-horizon forecasts. For node i, the prediction is

$$\hat{y}_{i,t} = f_{\mathrm{FC}}\big(H_t^{(\mathrm{GAT})}(i)\big), \tag{16}$$

where $f_{\mathrm{FC}}(\cdot)$ denotes a small multilayer perceptron. In our main multi-horizon experiments, we set

$$\hat{y}_{i,t} \in \mathbb{R}^4, \tag{17}$$

corresponding to $(t+1, t+5, t+15, t+21)$-day ahead volatility forecasts. This architecture makes the temporal component explicit via the LSTM, while the subsequent GCN + GAT stack captures the cross-sectional spillover structure encoded in the Diebold–Yilmaz or correlation-based graphs. Thus, it can be summarized as

$$X_{i,t-w+1:t} \longrightarrow \text{LSTM} \longrightarrow \text{GCN} \longrightarrow \text{GAT} \longrightarrow \hat{y}_{i,t+h}\,.$$

3.4. Rationale for the Combined GCN–GAT Architecture and Its Distinction from Existing Temporal GNNs

The combined use of GCN and GAT layers in our TemporalGAT architecture is a deliberate design choice that leverages the complementary strengths of the two operators. GCN layers provide a stable mechanism for aggregating information over the graph's topology, making them well-suited for capturing the broad, relatively persistent structure of cross-market connectivity. By diffusing information from both direct and indirect neighbours, GCNs establish a strong baseline representation of how volatility propagates through the market network.

GAT layers, in turn, introduce an adaptive mechanism that assigns learnable attention weights to neighbouring nodes. This allows the model to focus on the most influential spillover channels at each point in time, an essential capability in financial markets, where interdependencies can shift abruptly in response to macroeconomic announcements, geopolitical events, or regime changes. The sequential use of GCN followed by GAT thus enables the model to capture long-term structural dependencies while dynamically reweighting short-term shocks, producing richer and more context-sensitive volatility forecasts.

Our architecture differs from existing temporal GNNs and "Temporal-GAT" variants in the machine-learning literature. For example, [3] builds two rolling financial networks, a Diebold–Yilmaz spillover network and a correlation network and feeds these into a diffusion convolutional recurrent neural network (DCRNN), where diffusion convolution models spatial dependence and a GRU implicitly handles temporal dynamics. Their model uses only realized volatility as an input and produces single-horizon forecasts. In contrast, our framework constructs multiple piecewise-static graph forms and explicitly separates

temporal and spatial learning: each node's volatility history is encoded by an LSTM, after which the resulting embeddings propagate through stacked GCN → GAT layers. This separation allows the model to learn directional, weighted, and time-varying interdependencies with adaptive attention, rather than relying on a single recurrent mechanism to capture both temporal and spatial structure.

Furthermore, whereas standard temporal GNNs often apply attention directly across time or operate on event-driven dynamic graphs, our approach cleanly decouples the two dimensions. Temporal persistence is modelled exclusively through a node-level LSTM applied to rolling windows of volatility proxies or GARCH volatility proxies. At the same time, spatial dependencies are captured through GCN/GAT layers operating on econometrically constructed spillover or correlation networks, rather than on learned or time-stamped interaction graphs. The GAT component is used strictly in its spatial-attention form [34], rather than as a temporal attention mechanism. Finally, the architecture is tailored to financial forecasting by supporting multi-horizon prediction and enabling formal econometric evaluation through Diebold–Mariano tests and bootstrap confidence intervals. Thus, the proposed model is not intended as a universal temporal-GNN architecture but rather as a domain-specific LSTM–GCN–GAT hybrid designed to capture volatility persistence and cross-market spillovers in an interpretable and empirically grounded manner (see Table 1).

Table 1. Comparison of Our TemporalGAT With Existing Temporal GNN Architectures.

Aspect	This Study	[3]	[47]	[37]	[34]
Primary Inputs	Volatility/GARCH volatility proxies; rolling windows; price/volume	High-frequency realized volatility (RV) only	Time-stamped interaction events	Historical stock prices encoded via Transformer; graph from price correlations	Static node features (bag-of-words, gene features)
Graph Construction	Econometric spillover/correlation networks (directed/weighted)	Net pairwise VAR spillover index + correlation graph	Continuous-time dynamic event graph	Dynamic heterogeneous graph updated daily from positive/negative price-correlation edges	Fixed citation or biological graphs (static structure)
Temporal Modelling	LSTM temporal encoder per node	DCRNN (diffusion convolution + GRU)	Continuous-time memory + temporal message passing	Transformer encoder for price history + two-stage temporal attention	None (GAT is purely spatial; no temporal component)
Number of Nodes/Assets	8 global stock indices	8 global indices (SPX, GDAXI, FCHI, FTSE, OMXSPI, N225, KS11, HSI)	Thousands	Hundreds of equities (S&P 500; CSI 300)	Up to 20 k nodes in citation networks; 60 k in PPI graphs
Evaluation Metrics	MAFE, MAPE, MSE, RMSE; DM tests, bootstrap confidence intervals	MAFE; DM test; MCS	AP, ROC-AUC	ACC, ARR, AV, MDD, ASR, CR, IR	Accuracy (transductive), Micro-F1 (inductive)
Spatial Modelling	2GCN $\rightarrow$ 2GAT	Diffusion convolution	Temporal message passing with memory	Heterogeneous Graph Attention Network	Multi-head masked self-attention over neighbors
Forecasting Task	Multi-horizon volatility forecasting ($t+1, 5, 15, 21$)	Volatility forecasting ($h = 1, 5, 10, 22$)	Link prediction; node classification	Binary price movement prediction; portfolio optimization	Node classification (Cora, Citeseer, Pubmed; PPI multilabel)
Dynamic Graph Handling	Reconstructed spillover graphs across periods; adaptive GAT	Static spillover graph	Fully dynamic continuous-time graph	Graph updated every trading day; dynamic heterogeneous	Static graphs only; no dynamic updates
Domain	Financial volatility & spillover transmission	Global realized volatility forecasting	General temporal graphs	Financial price predictions	General-purpose GNNs (citation networks, protein interactions)

4. Methodology

This section outlines the comprehensive methodology adopted to explore and analyze volatility clustering in global stock markets using a Temporal GAT approach. The research aims to advance the understanding and forecasting of volatility by leveraging the financial markets' dynamic and interconnected nature. A systematic, step-by-step approach is presented, clarifying the strategies, techniques, and tools employed throughout the study, ensuring transparency and replicability of the findings.

4.1. Problem Formulation

Given a dynamic financial system represented as a time-evolving graph sequence $\{G_t\}_{t=1}^{T}$, where each node $v_i \in V$ corresponds to a global stock market index and each edge $e_{ij,t} \in E_t$ captures the directional relationship between indices i and j at time t (e.g., through correlation or volatility spillover), the objective is to predict the volatility proxies computed from daily returns of each index at a future time step $t + h$ using historical information up to time t.

Formally, the learning task is to estimate a function:

$$\hat{y}_{i,t+h} = \mathcal{F}(G_{t-w+1:t}, X_{i,t-w+1:t}; \theta)$$

where

- $\hat{y}_{i,t+h} \in \mathbb{R}$ is the predicted volatility proxies of index i at time $t + h$.
- $X_{i,t-w+1:t} \in \mathbb{R}^{w \times d}$ is the historical sequence of node features for index i over a look-back window of size w.
- $G_{t-w+1:t}$ is the sequence of regime-dependent graph snapshots representing market structure from time $t - w + 1$ to t.
- θ denotes the model parameters to be learned.
- d is the dimensionality of node features (e.g., volatility proxies, closing price, or trading volume).

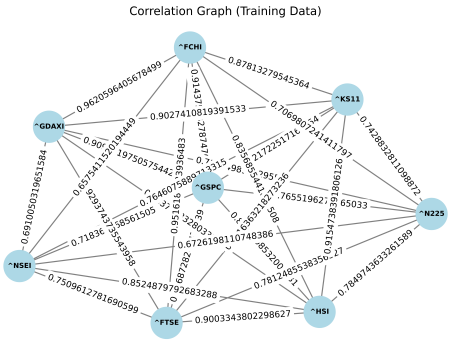
The task is modelled as a temporal graph-based regression problem. The proposed Temporal GAT is designed to capture temporal dependencies (i.e., past patterns and trends in volatility for each market index) and structural dependencies (i.e., dynamic interrelationships among market indices over time). For this study, we consider the look-back window $w \in \{5, 15, 21, 40\}$ (denoting the number of past trading days used as input), the forecast horizon $h \in \{1, 5, 15, 21\}$ (corresponding to short-term (1-day) to long-term (1-month) predictions), the node features which includes primarily volatility proxies, with comparative experiments using closing prices and trading volumes. The piecewise-static graphs G_t are constructed using either the Pearson correlation coefficient or the Volatility Spillover Index derived from a VAR model.

4.2. Graph Construction

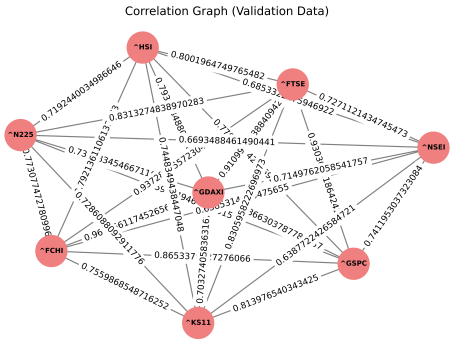
The core of our methodology lies in representing the stock market as a directed graph, where the indices are nodes and the relationships between them form directed edges. In a directed graph, each edge has a direction, indicating the flow of influence or information from one node (market index) to another. This structure is particularly suitable for capturing asymmetric relationships often observed in financial markets, in which one market can significantly impact another without necessarily experiencing a reciprocal effect [40]. We employed two primary methods for constructing these directed graphs: the Correlation Method and the Volatility Spillover Index Method.

Correlation Method: In this method, Pearson correlation coefficients between the volatility proxies of the indices during the training period were calculated. These coefficients form

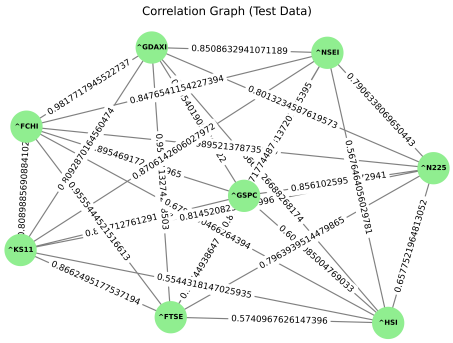
the graph's edges, yielding a symmetric adjacency matrix with self-loops on the diagonal (each entry is 1). The Net Correlation Index (NCI) for each market was calculated as the sum of its correlations with other markets [See Figure 1]. This method helps identify the strength of the correlation between different indices, providing a clear view of the interconnectedness of these markets [48,49].



(**a**) Training data



(**b**) Validation data



(**c**) Test data

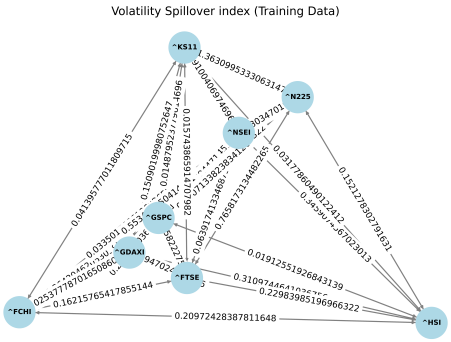
Figure 1. Visualization of graphs by the correlation method.

Volatility Spillover Index Method: Based on the framework by [8,40], this method measures the degree of volatility transmission between indices using variance decomposition from a VAR model. We employed a lag order of 4 ($p = 4$) and a 5-step ahead forecast horizon ($H = 5$) to derive these indices [See Figure 2]. The resulting spillover index matrices illustrate the directional influence one market exerts over another, capturing the dynamic nature of volatility transmission in the financial markets.

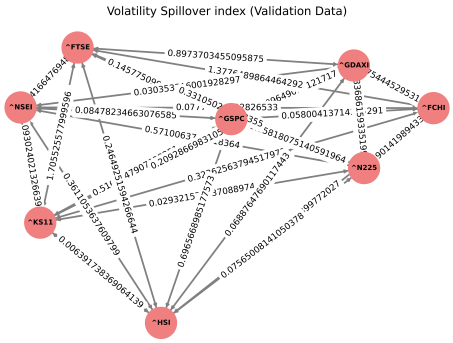
Figures 1 and 2 represent a graph where each node is a global stock market index (e.g., HSI, FTSE), and the directed edges show relationships between them. The numbers on the edges indicate the strength of the connection between two indices, which can be measured using correlation or the volatility spillover index method. For instance, HSI has a connection strength of 0.229 (spillover method) to FTSE in the training data. The comparison between training, testing, and validation data reflects how these relationships change over time, with the numbers indicating varying connection strengths.

Consider the training dataset in Figure 1, when stock indices are highly correlated, as seen between indices like GSPC vs. FCHI, HSI, KS11 (correlations of 0.91, 0.87, 0.90 respectively), KS11 vs. FTSE, HSI, GSPC, GDAXI (correlations of 0.91, 0.92, 0.90, 0.90 respectively), FTSE vs. GDAXI, HSI (correlations of 0.92, 0.90 respectively) and GDAXI vs. FCHI (correlation of 0.96), it suggests that these markets are likely to experience volatility clustering together. If one market (e.g., GSPC) enters a period of high volatility due to market turbulence or tensions in the USA, this volatility can affect the French market

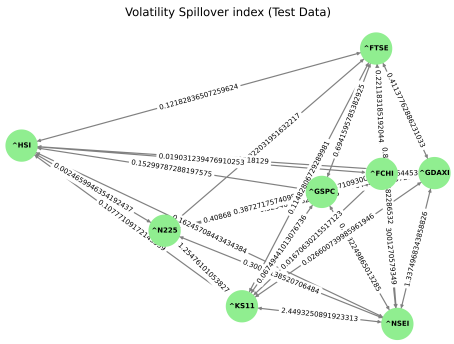
(CAC 40) because the two markets are highly correlated. In addition, moderate-to-low correlations (e.g., between NSEI vs. FCHI, and N225 and GDAXI, with correlations of 0.66, 0.67, and 0.69, respectively) indicate that while there is some shared volatility, these indices do not always cluster together. Thus, these cluster effects reflect interconnectedness because regions with strong trade links, similar industry exposures, or shared investor bases tend to exhibit co-movement in volatility. This is especially true for global markets such as the US and Europe, as well as for regional clusters such as Europe and Asia.



(**a**) Training data



(**b**) Validation data



(**c**) Test data

Figure 2. Visualization of graphs by the volatility spillover method.

4.3. Graph Construction via Volatility Spillovers

To capture cross-market dependence in volatility dynamics, we construct directed graphs based on volatility spillover effects using the Diebold–Yilmaz framework. Specifically, volatility proxies for all indices are modelled jointly using a Vector Autoregression (VAR), and generalized forecast error variance decompositions (GFEVD) are employed to quantify the proportion of forecast uncertainty transmitted from one market to another.

Let $\mathbf{RV}_t = (RV_{1,t}, \ldots, RV_{N,t})^\top$ denote the vector of volatility proxies across N equity indices. For a given data partition, a VAR model is estimated on $\mathbf{RV}_t$, and the resulting GFEVD yields a spillover matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$, where the (i, j)-th entry measures the contribution of shocks in market j to the forecast error variance of market i. This matrix is interpreted as a weighted, directed adjacency matrix, with nodes representing indices and edge weights capturing the magnitude of volatility transmission.

To ensure a strict separation of information sets, the spillover networks are constructed in a *piecewise-static, regime-dependent* manner. Separate adjacency matrices are estimated for the training, validation, and test periods, using only the volatility proxies data contained within each respective partition. Within each period, the resulting graph topology is held fixed and represents the average spillover structure of that regime. This design prevents

information leakage across evaluation phases while providing a stable and statistically robust representation of cross-market volatility linkages.

We emphasize that the resulting graphs are not updated at every forecasting origin within a period. While rolling-window or fully time-varying spillover networks are conceptually appealing, such approaches can be statistically unstable in small systems (eight stock indices) and short samples, particularly when VAR-based variance decompositions are employed. Accordingly, the adopted piecewise-static construction offers a principled trade-off between econometric reliability and temporal segmentation. Extending the framework to fully dynamic spillover networks is left for future research.

Remark 1 (On Graph Construction and Look-Ahead Bias)**.** *The spillover-based adjacency matrices are constructed separately for the training, validation, and test sets, using only information available within each respective period. As a result, no future observations beyond the boundaries of a given partition are used when defining the graph topology for that period. While the graph remains fixed within each regime, this piecewise-static design avoids look-ahead bias across evaluation phases and preserves a strictly out-of-sample forecasting setup. A fully rolling or expanding-window graph construction, although feasible, is beyond the scope of this study and left for future work.*

4.4. Node Features

The target variable of interest is the daily volatility proxies $\sigma_{i,t}$ for each index i. In the baseline specification, the volatility proxy is also used as the primary node feature. Specifically, for each node i and time t, we construct a look-back window of length w,

$$X_{i,t} = \left[\sigma_{i,t-w+1}, \sigma_{i,t-w+1}, \cdots \sigma_{i,t}\right],$$

which serves as the input feature vector for node i in the spillover graph at time t. The TemporalGAT therefore learns a mapping from past volatility proxy and contemporaneous network structure to future volatility proxy.

4.5. Model Architecture Overview

The proposed Temporal GAT integrates temporal sequence modelling with graph-based spatial learning, enabling joint extraction of time-series patterns and cross-market spillover effects. The architecture is designed to forecast volatility proxy across multiple international equity indices by leveraging both historical dynamics and contemporaneous interconnections encoded in the Diebold–Yilmaz spillover network. At a high level, the architecture processes information through four sequential modules:

1. Temporal Module (LSTM): Each market index is represented by a rolling look-back window of volatility proxies. An LSTM network transforms this sequence into a temporal embedding that captures persistence, structural breaks, and nonlinear dynamics in volatility.
2. GCN Layers: Two GCN layers first aggregate neighbourhood information based on the adjacency structure, producing a shared spatial embedding for each node. The first GCN layer transforms the node features from their initial dimension to a hidden dimension. This transformation is tested with hidden dimensions of 32, 64, and 128 in our implementation to determine the optimal size. The second GCN layer further processes these transformed features by applying a ReLU activation function to introduce nonlinearity, which is crucial for capturing complex patterns.
3. GAT Layers: The output is passed to two GAT layers, which adaptively learn the influence weights across markets through attention coefficients. This enables the model to differentiate between strong and weak spillover relationships. These layers focus on significant nodes, prioritizing crucial information within the graph. The attention

mechanism in each GAT layer is configured with multiple heads (specifically 4 or 8 heads in our tests), allowing the model to learn different aspects of the data from multiple representation subspaces simultaneously. This setup enhances the model's ability to capture diverse relational patterns among data points [50].

4. Multi-Horizon Output Head: A series of three fully connected layers maps the spatio-temporal embeddings to volatility forecasts across multiple future horizons. They incorporate a ReLU activation function, and each applies nonlinear transformations to capture higher-order interactions. The final output layer produces a scalar one-step-ahead volatility forecast for each index. These layers are crucial in synthesizing the learned graph-based features into a comprehensive form suitable for the final prediction task [19].
5. Training and Optimisation: The model is trained end-to-end using the Adam optimiser and the mean squared error (MSE) loss function. The entire architecture—temporal feature extraction, spatial propagation, and prediction—is updated jointly to minimize forecasting error. A grid search strategy is utilized to fine-tune the model's hyper-parameters, including the number of hidden dimensions, attention heads, and the learning rate. The learning rate values tested are 0.0001, 0.001, and 0.01. This optimization involves training the model across a predefined grid of parameter combinations and monitoring performance through the Mean Squared Error (MSE) on a validation set. The goal is to minimize MSE across 70 training epochs, refining the model's ability to forecast market volatility accurately.

The overall pseudocode is provided in Algorithm 1. Thus, the combination of temporal sequence learning and graph-based relational modelling allows the architecture to capture both volatility clustering in individual markets and contagion effects across the global financial network.

Figure 3 illustrates the complete processing pipeline. The model first extracts temporal features through an LSTM, then applies graph convolutions and attention mechanisms to capture spatial spillovers, and finally outputs a vector of predicted volatilities for horizons $h \in \{1, 5, 15, 21\}$.

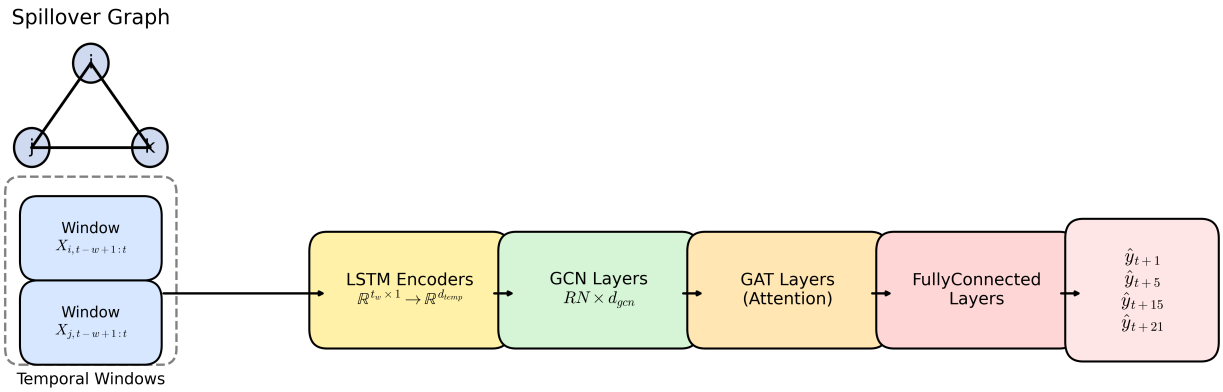


Figure 3. General design for the Temporal GAT model.

Algorithm 1 Pseudocode for the Temporal GAT Multi-Horizon Forecasting Framework

Initialize model parameters, tickers, date range, rolling window w, and forecasting horizons $\mathcal{H} = \{1, 5, 15, 21\}$.

1: **Data Collection and Preprocessing:**
- Download daily price data for each index using `yfinance`.
- Compute volatility proxies using squared returns over a rolling window.
- Split each series into training, validation, and test subsets.

2: **Construct Spillover Networks:**
- For each dataset (train/validation/test), construct a VAR model.
- Compute Diebold–Yilmaz spillover matrices using FEVD decomposition.
- Build a directed graph $G = (V, E)$ where edge weights correspond to spillover intensities.

3: **Graph-to-PyG Conversion:**
- For each node i, extract the last w volatility proxies as temporal input window $X_{i,t-w+1:t}$.
- For each forecast horizon $h \in \mathcal{H}$, define targets $y_{i,h} = RV_{i,t+h}$.
- Construct PyTorch-Geometric (version 2.9.0) `Data` objects containing:

$$\mathbf{x} = \left[X_1 \cdots X_N\right]^T, \qquad \mathbf{y} = \left[y_{1,\mathcal{H}} \cdots y_{N,\mathcal{H}}\right]^T, \qquad \texttt{edge_index}$$

4: **Model Architecture (TemporalGAT):**
- **Temporal Layer:** Apply an LSTM to each window:

$$\text{LSTM} : \mathbb{R}^{w \times 1} \to \mathbb{R}^{d_{\text{temp}}}.$$

- **GCN Layers:** Apply two stacked GCN layers to incorporate first-order graph structure.
- **GAT Layers:** Apply multi-head attention to capture weighted spillover importance.
- **Output Head:** Feed-forward layers produce a multi-horizon vector:

$$\hat{\mathbf{y}}_i = [\hat{y}_{i,t+1}, \hat{y}_{i,t+5}, \hat{y}_{i,t+15}, \hat{y}_{i,t+21}].$$

5: **Training:**
- For each epoch:
  - Forward pass: compute predictions $\hat{\mathbf{y}}$.
  - Compute multi-horizon loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2.$$

  - Backpropagate gradients and update model parameters.

6: **Hyperparameter Grid Search:**
- Define search sets for hidden dimension, attention heads, and learning rate.
- For each parameter combination:
  - Train the TemporalGAT model.
  - Evaluate validation loss and record best configuration.

7: **Evaluation:**
- Compute multi-horizon metrics for each $h \in \mathcal{H}$:

$$\text{MAFE, MSE, RMSE, MAPE, } \text{R}^2.$$

- Compute per-index metrics:

$$\text{Metrics}(i, h) = f(y_{i,h}, \hat{y}_{i,h}).$$

- Generate tables of cross-index forecast accuracy.

8: **Return:** Trained model, best hyperparameters, and multi-horizon forecast performance.

4.6. Other Models for Comparison

To evaluate the performance of the Temporal GAT model (TGATM), we compared it against six alternative models, each utilizing different methodologies:

- Baseline Model(BM): The BM is crafted to process the volatility proxies for transforming and capturing the volatility spillover index of financial markets without integrating graph-based complexities. This model is included to isolate the contribution of graph structure and temporal modelling. The model receives each index's feature vector independently and performs forecasting without any spatial or temporal interaction between nodes. The architecture consists of three fully connected hidden layers with ReLU activations and dropout regularization, followed by a final linear layer that outputs the 15-day-ahead volatility forecast. Since the MLP processes each index in isolation and lacks access to spillover relationships or historical sequences, it serves as a minimal, non-graph and non-temporal benchmark for evaluating the value added by both the spatial message-passing components and the temporal encoders used in the TemporalGAT and GARCH-TGAT models. The architecture comprises three hidden layers, each with dimensions configurable to be either 32, 64, or 128, allowing the model to adapt its complexity to the richness of the input data.
- Static GNN-GATM (SGNN-GATM): As a non-temporal benchmark, we implement an SGNN-GATM that uses only cross-sectional spillover structure and ignores all time-series dynamics. Each index is represented by a node whose feature is the most recent observed volatility value, and information is propagated solely across the static spillover graph. The model consists of a single GCN layer followed by a single GAT layer, enabling capture of first- and second-order spatial dependencies but no temporal evolution. A fully connected output layer then maps the learned node embeddings to a one-step-ahead volatility prediction. Because this model does not use historical windows, recurrent units, or temporal attention mechanisms, it serves as a minimal spatial baseline. All hyperparameters (hidden dimensions and attention heads) are kept consistent with those used in TGATM for fair comparison.
- Deeper GNN-GATM (DGNN-GATM): To ensure that the weaker performance of the simple static GNN is not merely due to under-capacity, we additionally construct a DGNN-GATM baseline. This model extends the shallow spatial architecture by stacking multiple graph convolutional layers ($GCN \rightarrow GCN \rightarrow GAT \rightarrow GAT$), allowing the network to aggregate spillover information from higher-order neighbourhoods without incorporating any temporal structure. Node features consist solely of the most recent observed volatility value, with no historical window or recurrent component. After spatial message passing, two fully connected layers refine the node embeddings before a final linear layer produces the 15-day-ahead forecast. All hyperparameters (hidden dimensions and attention heads) are kept consistent with those used in TGATM for fair comparison.
- GARCH Temporal GAT Model (GARCH-TGATM): The GARCH-TGATM incorporates the GARCH methodology for transforming raw data into a format suitable for graph-based analysis, enhancing the traditional volatility modelling approach. This model leverages the strengths of both GCNs and GATs to effectively model the dynamic relationships within financial markets. To integrate econometric volatility structure into the graph-based forecasting framework, we construct a GARCH-TGAT model in which node features are derived from GARCH(1,1) conditional volatility estimates. For each index i, a GARCH(1,1) model is fitted to daily log-returns, yielding the conditional volatility series $\sigma_t^{(i)}$. The TGAT node feature vector is formed by taking the most recent W values $(\sigma_{t-W+1}^{(i)}, \ldots, \sigma_t^{(i)})$, which encode the nonlinear persistence, clustering, and mean-reversion properties captured by the GARCH process. These GARCH-based temporal windows are fed into the TGAT architecture, which consists of an LSTM temporal encoder followed by stacked GCN and GAT layers that learn the cross-market spillover structure. Thus, unlike TGAT variants operating on raw

price-based volatility measures, the GARCH-TGAT explicitly embeds traditional econometric volatility dynamics within a temporal-spatial neural representation.

- Correlation Temporal GAT Model (C-TGATM): The Correlation-TGAT model uses a correlation-based graph structure to capture static interdependencies between global equity indices. Instead of estimating directional spillovers, the model constructs an undirected graph where edges represent statistically significant return–volatility correlations computed from a rolling window of historical data. This graph encodes the strength of co-movement between markets, allowing TGAT to propagate temporal features across highly correlated nodes. Each node is assigned a temporal sequence of volatility proxies (or GARCH-based volatility proxies), which is then processed through the Temporal-GAT architecture combining LSTM-based temporal encoding and GAT-based spatial attention. The resulting framework provides a benchmark that isolates pure correlation-driven connectivity, enabling comparison with the Spillover-TGAT model, which uses structural VAR-based spillover linkages.
- Long Short-Term Memory (LSTM): To isolate the contribution of temporal dynamics from the graph-learning component, we include a pure LSTM network as a non-graph baseline. The LSTM operates solely on the historical volatility sequence of each index, without incorporating any cross-market relational structure. For each node, the model receives a sliding window of past volatility proxies and predicts the 15-day-ahead volatility target. This baseline is intentionally structured to match the temporal depth of the TGAT model, ensuring a fair comparison focused exclusively on temporal modelling capacity. Because LSTMs are well known for their ability to capture long-range dependencies and nonlinear dynamics in time-series data, this experiment helps determine whether the performance improvements observed in TGAT stem from the graph-based message passing or simply from the temporal modelling framework.

5. Empirical Results and Discussion

This section presents the data visualization, model analysis, sensitivity studies, and robustness tests.

5.1. Data Visualization and Analysis

The data used in this study focuses on eight major global market indices: GSPC—S&P 500 (USA), GDAXI—DAX (Germany), FCHI—CAC 40 (France), FTSE—FTSE 100 (UK), NSEI—Nifty 50 (India), N225—Nikkei 225 (Japan), KS11—KOSPI (South Korea) and HSI—Hang Seng Index (Hong Kong). These indices were selected due to their significant influence on global financial markets [3]. The dataset was sourced from Yahoo Finance and spans November 2007 to June 2022. The volatility proxy (VP) data were computed using daily adjusted closing prices, following the approach suggested by Andersen et al. [51]. The dataset was divided into three subsets: a training set (1891 datasets spanning from November 2007 to August 2014), a validation set (756 datasets from September 2014 to December 2017), and a test set (1136 datasets from January 2018 to June 2022). These subsets, respectively, cover 50%, 20%, and 30% of the total data, ensuring a robust evaluation of the model's performance across different time periods [3].

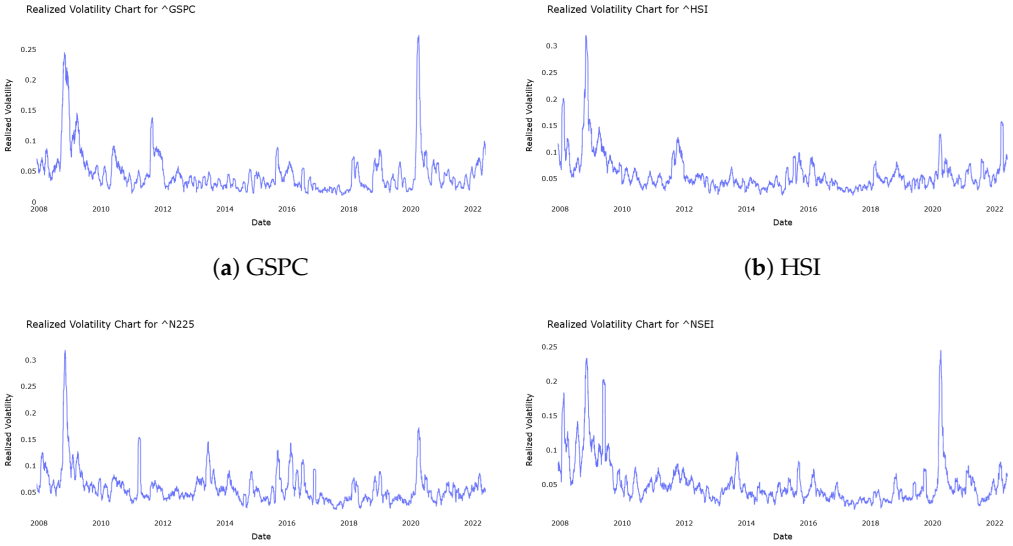
Descriptive statistics (Table 2) of the volatility proxy data were computed to gain insights into the underlying distribution and characteristics of the market indices. The mean volatility ranged from 0.046 to 0.059, reflecting the average level across the indices. The standard deviation, varying between 0.028 and 0.034, indicated the extent of dispersion in the volatility data. Skewness and kurtosis metrics highlighted the non-normality of the volatility distributions, with positive skewness values ranging from 2.28 to 3.23 and kurtosis values ranging from 10.47 to 20.04, suggesting heavy tails. The Augmented Dickey–Fuller

(ADF) test results confirmed the stationarity of the data across all indices, with p-values significantly below 0.05. These statistics ensure that the time series data is stable and suitable for GNN modelling without further transformations [52].

Table 2. Statistical properties of selected indices.

Ticker	Mean	Std. Deviation	Skewness	Kurtosis	ADF Statistic	ADF p-Value
GSPC	0.047322	0.034962	3.144922	16.101063	−5.540181	1.70867×10^{-6}
GDAXI	0.054093	0.038324	3.144921	15.566453	−5.604956	1.06869×10^{-6}
FCHI	0.057695	0.02082	2.288022	14.471417	−5.423957	3.22759×10^{-6}
FTSE	0.040868	0.023625	2.838757	13.337729	−5.237459	7.06389×10^{-6}
NSEI	0.052276	0.03306	2.763412	12.677778	−4.887057	3.69459×10^{-5}
N225	0.05592	0.031903	3.234541	20.204222	−5.484947	7.20239×10^{-7}
KS11	0.048637	0.028237	3.098161	13.453776	−4.902201	3.45269×10^{-5}
HSI	0.085911	0.033861	3.188015	18.540499	−4.435595	2.56614×10^{-4}

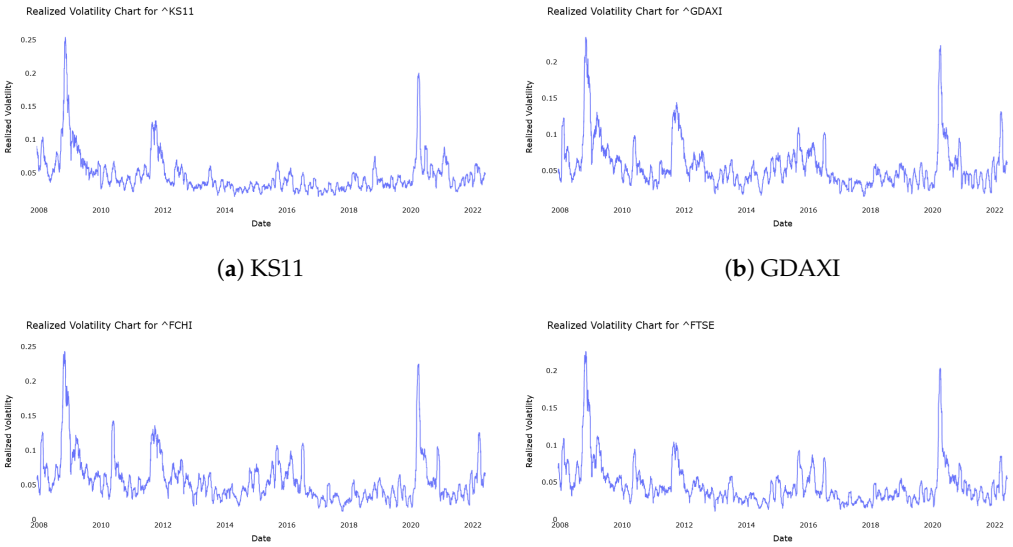
To provide a visual representation of the volatility proxy data across the eight global market indices, a time-series plot was generated that captures the daily adjusted closing prices. Figures 4 and 5 present the volatility proxy data for the eight selected indices, highlighting periods of high and low volatility. The graph illustrates the volatility proxy data over the training, validation, and test periods, enabling a better understanding of market behaviour during major global events, such as the 2008 financial crisis, Brexit, and the COVID-19 pandemic, all of which contributed to significant market volatility. The analysis of volatility proxy data reveals significant clustering behaviour across global market indices, particularly during major financial events.



(**a**) GSPC (**b**) HSI

(**c**) N225 (**d**) NSEI

Figure 4. Volatility proxy data for selected indices (Part I).



(**a**) KS11 (**b**) GDAXI

(**c**) FCHI (**d**) FTSE

Figure 5. Volatility proxy data for selected indices (Part II).

Furthermore, to evaluate and compare the models' forecasting performance, several commonly used error metrics are calculated. These metrics quantify forecast accuracy by measuring deviations between observed and predicted values. Below are some of the assessment metrics employed in this study.

Let y_t represent the actual observed value at time t, and $\hat{y}_t$ the corresponding forecasted value and n, the sample size. Then,

- Mean Absolute Forecast Error (MAFE):

$$\text{MAFE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

- Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2$$

- Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \quad (\text{where } y_t \neq 0)$$

- R-squared:

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

5.2. Comparative Analysis with Traditional Econometric Models

The empirical results in Table 3 show that the TGATM consistently outperforms the classical econometric benchmarks — GARCH(1,1), EGARCH(1,1), and HAR-RV, across both evaluation metrics. Temporal GAT achieves the lowest average MSE (0.01165), indicating superior robustness to large forecast deviations and volatility shocks. Its ability to minimize significant errors suggests that the model effectively captures nonlinear dependencies, cross-asset spillovers, and regime-switching patterns that traditional parametric models

are unable to represent. Furthermore, Temporal GAT achieves the smallest average MAFE (1.81×10^{-4}), substantially lower than that of the econometric alternatives. This highlights its exceptional precision in predicting daily volatility levels across all indices, providing forecasts that more closely align with observed market dynamics.

The econometric models perform relatively worse because they rely on rigid parametric structures that impose smooth and gradual volatility adjustments, causing them to lag during periods of rapid volatility shifts. Although EGARCH offers modest improvements by modeling asymmetry, its performance remains constrained by static functional forms. In contrast, the Temporal GAT architecture leverages graph attention mechanisms and deep temporal representations, enabling it to learn complex, nonlinear relationships in the data and adapt effectively to abrupt market changes. The combined superiority of TGAT in both MSE and MAFE underscores its advantage in capturing real-world volatility behaviour, suggesting that graph-based deep learning approaches provide a more accurate and resilient framework for financial volatility forecasting than traditional econometric methods.

Table 3. Out-of-sample error values (TGATM vs. Econometric models).

Indices	TGATM	GARCH(1,1)	EGARCH(1,1)	HAR-RV
		MSE		
GSPC	0.022141	0.111975	0.030199	0.055457
GDAXI	0.008971	0.023510	0.032299	0.027697
FCHI	0.010928	0.033062	0.037769	0.038240
FTSE	0.017812	0.026167	0.021205	0.034271
NSEI	0.017396	0.032884	0.027441	0.036480
N225	0.000373	0.013279	0.012935	0.015714
KS11	0.010277	0.016869	0.016905	0.018984
HSI	0.005280	0.034410	0.031404	0.045763
Average	**0.011647**	**0.036520**	**0.026270**	**0.034076**
		MAFE		
GSPC	4.90×10^{-4}	1.25×10^{-2}	9.12×10^{-4}	3.08×10^{-3}
GDAXI	8.05×10^{-5}	5.53×10^{-4}	1.04×10^{-3}	7.67×10^{-4}
FCHI	1.19×10^{-4}	1.09×10^{-3}	1.43×10^{-3}	1.46×10^{-3}
FTSE	3.17×10^{-4}	6.85×10^{-4}	4.50×10^{-4}	1.18×10^{-3}
NSEI	3.03×10^{-4}	1.08×10^{-3}	7.53×10^{-4}	1.33×10^{-3}
N225	1.39×10^{-7}	1.76×10^{-4}	1.67×10^{-4}	2.47×10^{-4}
KS11	1.06×10^{-4}	2.85×10^{-4}	2.86×10^{-4}	3.60×10^{-4}
HSI	2.79×10^{-5}	1.18×10^{-3}	9.86×10^{-4}	2.09×10^{-3}
Average	$\mathbf{1.81 \times 10^{-4}}$	$\mathbf{2.2 \times 10^{-3}}$	$\mathbf{7.53 \times 10^{-4}}$	$\mathbf{1.31 \times 10^{-3}}$

Note: **Bold** values in the footer denote the average MSE and MAFE across all indices.

5.3. Comparative Analysis with ML-Related Models

This section compares the out-of-sample error values for the following models: BM, LSTM, SGNN-GATM, DGNN-GATM, TGATM, GARCH-TGATM, C-TGATM. The forecasting results presented in Table 4 compare the predictive performance of several neural and econometric architectures across eight major global equity indices (window size of 15 days).

The TGATM consistently achieves the lowest or near-lowest MAFE and MSE values for most markets, particularly for indices with strong temporal dependencies such as the S&P 500 (GSPC), DAX (GDAXI), and Hang Seng Index (HSI). This demonstrates the model's ability to capture dynamic cross-market interactions effectively. In contrast, the GARCH-augmented variant of TGAT model performs notably worse, with substantially higher errors across all indices. These findings suggest that integrating GARCH-type volatility

priors may introduce noise that disrupts the temporal attention mechanism rather than complementing it. Meanwhile, models such as the DGNN-GATM and LSTM perform competitively, especially on relatively stable markets like N225 and KS11, highlighting that purely temporal or purely structural learning can be effective when market regimes are less volatile.

Table 4. Out-of-sample error values for forecast window value of 15.

Indices	TGATM	GARCH-TGATM	DGNN-GATM	BM	C-TGATM	SGNN-GATM	LSTM
				MAFE			
GSPC	0.021264	0.036186	0.023119	0.024957	0.049222	0.025125	0.022225
GDAXI	0.008093	0.033118	0.009949	0.011840	0.019283	0.011954	0.009449
FCHI	0.010050	0.033684	0.011906	0.013765	0.025039	0.013912	0.011180
FTSE	0.016934	0.034712	0.018789	0.020612	0.011514	0.020795	0.017746
NSEI	0.000518	0.034465	0.018374	0.020216	0.013853	0.020380	0.017515
N225	0.000505	0.032778	0.001351	0.003284	0.015841	0.003357	0.001188
KS11	0.009399	0.033281	0.011254	0.013131	0.004507	0.013260	0.010652
HSI	0.004403	0.032198	0.006258	0.008171	0.043946	0.008264	0.005931
				MSE			
GSPC	4.52×10^{-4}	1.31×10^{-3}	5.35×10^{-4}	6.23×10^{-4}	2.42×10^{-3}	6.31×10^{-4}	4.94×10^{-4}
GDAXI	6.55×10^{-5}	1.10×10^{-3}	9.90×10^{-5}	1.40×10^{-4}	3.72×10^{-4}	1.43×10^{-4}	8.93×10^{-5}
FCHI	1.01×10^{-4}	1.13×10^{-3}	1.42×10^{-4}	1.89×10^{-4}	6.27×10^{-4}	1.94×10^{-4}	1.25×10^{-4}
FTSE	2.87×10^{-4}	1.20×10^{-3}	3.53×10^{-4}	4.25×10^{-4}	1.33×10^{-4}	4.32×10^{-4}	3.15×10^{-4}
NSEI	2.73×10^{-4}	1.19×10^{-3}	3.38×10^{-4}	4.09×10^{-4}	1.92×10^{-4}	4.15×10^{-4}	3.07×10^{-4}
N225	2.55×10^{-7}	1.07×10^{-3}	1.83×10^{-6}	1.08×10^{-5}	2.51×10^{-4}	1.13×10^{-5}	1.41×10^{-6}
KS11	8.83×10^{-5}	1.11×10^{-3}	1.27×10^{-4}	1.72×10^{-4}	2.03×10^{-5}	1.76×10^{-4}	1.13×10^{-4}
HSI	1.94×10^{-5}	1.04×10^{-3}	3.92×10^{-5}	6.68×10^{-5}	1.93×10^{-3}	6.83×10^{-5}	3.52×10^{-5}

The C-TGAT model performs well on some indices (e.g., KS11) but, overall, exhibits larger errors than the TGATM, indicating that static correlation structures alone are insufficient to capture the nonlinear, time-varying dependencies inherent in global financial markets. Traditional baseline approaches, including the BM and SGNN models, show consistently higher error magnitudes, reinforcing the importance of temporal attention and piecewise-static graph modelling in financial forecasting. Collectively, these results highlight the superiority of the TGAT family, especially the standard TGATM, over static or correlation-based architectures. The performance differences across indices also underscore the varying degrees of temporal complexity and interconnectedness among global markets, demonstrating the necessity of models capable of capturing both sequential patterns and evolving inter-market relationships.

Furthermore, based on the average errors (See Figure 6), TGATM achieves the lowest MAFE and MSE, confirming it as the most accurate and stable model across all indices. LSTM and DGNN-GATM also show strong performance, indicating that temporal and graph-based structures can generalize well but still fall short of TGATM's efficiency. In contrast, GARCH-TGATM records the highest average errors, suggesting that incorporating GARCH-based volatility components negatively impacts forecasting accuracy.

Remark 2. *When constructing a financial forecast model, it is essential to balance the trade-off between forecast window size and the model's predictive performance. For tasks that require short-term accuracy, such as day trading, a shorter forecast window can improve performance by focusing on near-term patterns and increasing reliability. On the other hand, medium- to long-term forecasting requires a larger forecast window size to capture broader trends. In the case of the*

TGATM, reducing the window size tends to improve short-term forecasts, but an optimal window size is needed, as a smaller window can lead to model overfitting, thereby degrading performance.

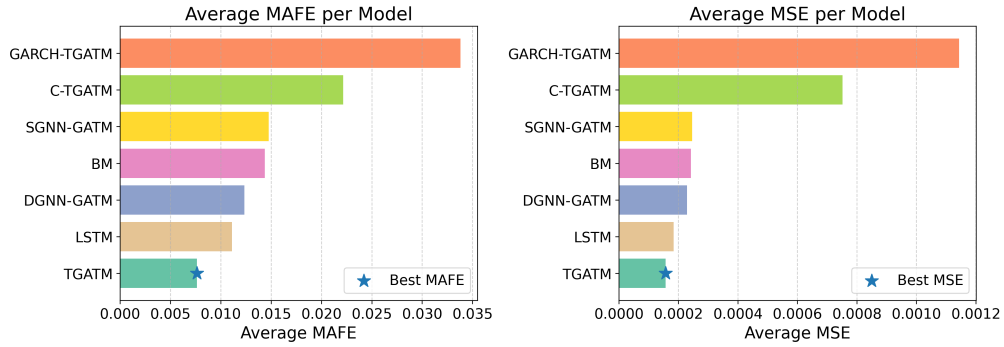


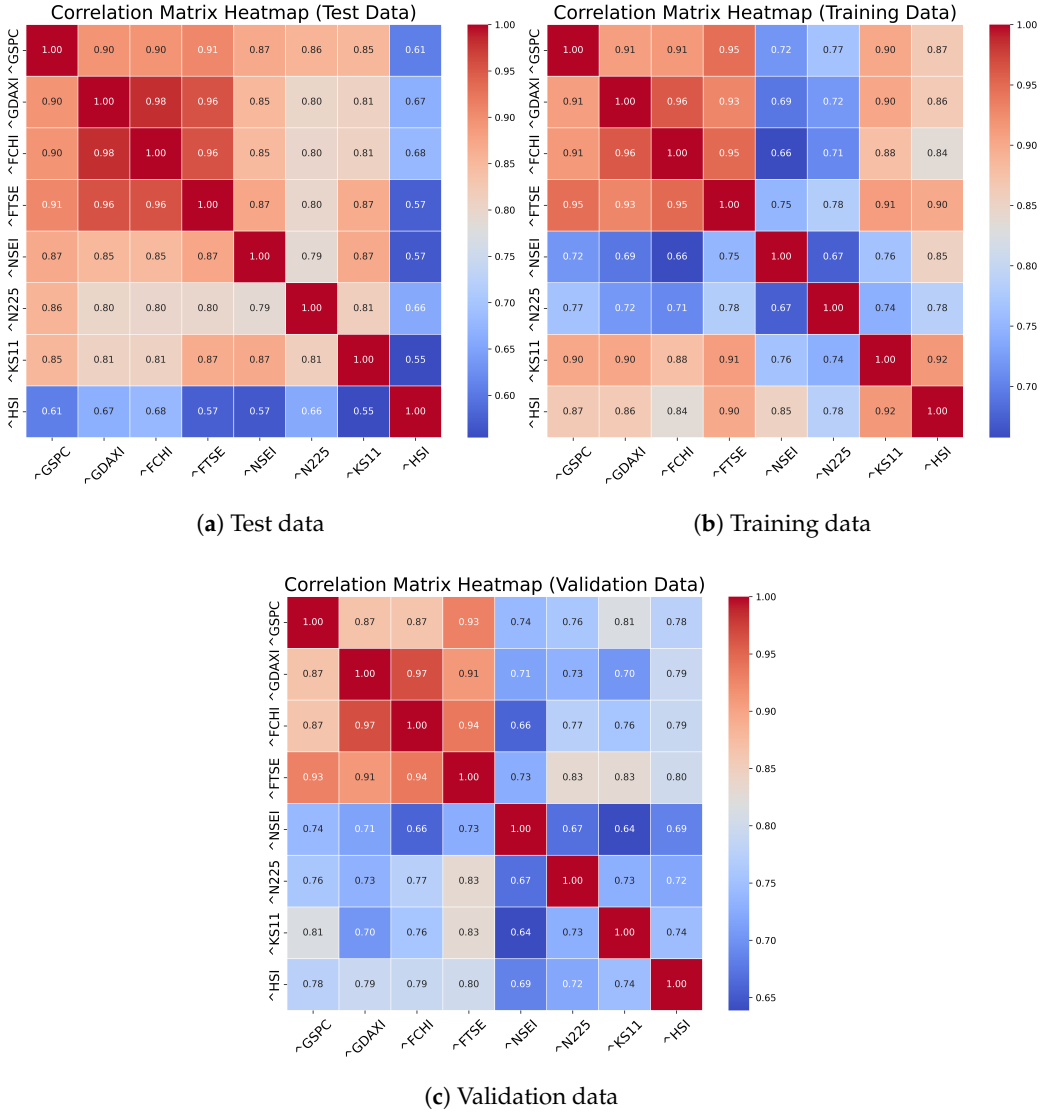
Figure 6. Model-wise comparison of average prediction errors.

5.4. Model Analysis

In this section, we analyze the distinctions between the correlation matrix and volatility spillover index heatmaps as tools for assessing relationships among market indices. Both methodologies provide valuable insights into the interconnectedness of financial markets, but they do so through different avenues. The correlation matrix provides a snapshot of linear relationships, while the volatility spillover index heatmap offers a dynamic view of how volatility transmits between indices over time. Utilizing both tools together can provide a more comprehensive understanding of market interactions and risk dynamics.

In Figure 7, we display the correlation matrix heatmaps for the training, validation and testing datasets. The figures show high interdependence across all US and European Markets datasets, with consistently strong correlations (above 0.90) between the S&P 500 and major European indices like FTSE, GDAXI, and FCHI. These relationships indicate that Western markets move closely together, likely due to similar economic factors. For the Asian markets, the HSI shows the most independence, with lower correlations in the test data, especially with the US market. Japan (N225) shows moderate correlations, while South Korea (KS11) is more aligned with global markets. India (NSEI) shows increasing integration over time, with its correlations rising in the test data. Finally, regarding changes across datasets, we observed that training data exhibits the most robust correlations, particularly between the US, European, and South Korean markets. The validation data presents slightly lower correlations but maintains the same general trends. In contrast, the test data shows more variability, especially with HSI (Hong Kong) becoming more independent and NSEI (India) increasing its correlations with global indices.

In contrast, the volatility spillover index focuses specifically on the transmission of volatility between indices. Using the Diebold–Yilmaz methodology, this approach captures how shocks to one index affect the volatility of others over time. The spillover index matrix provides a directional measure of volatility transfer, illustrating which indices are net transmitters or receivers of volatility. This is particularly useful during periods of market stress, as it identifies the channels through which volatility propagates, offering a deeper understanding of market dynamics beyond mere correlation.

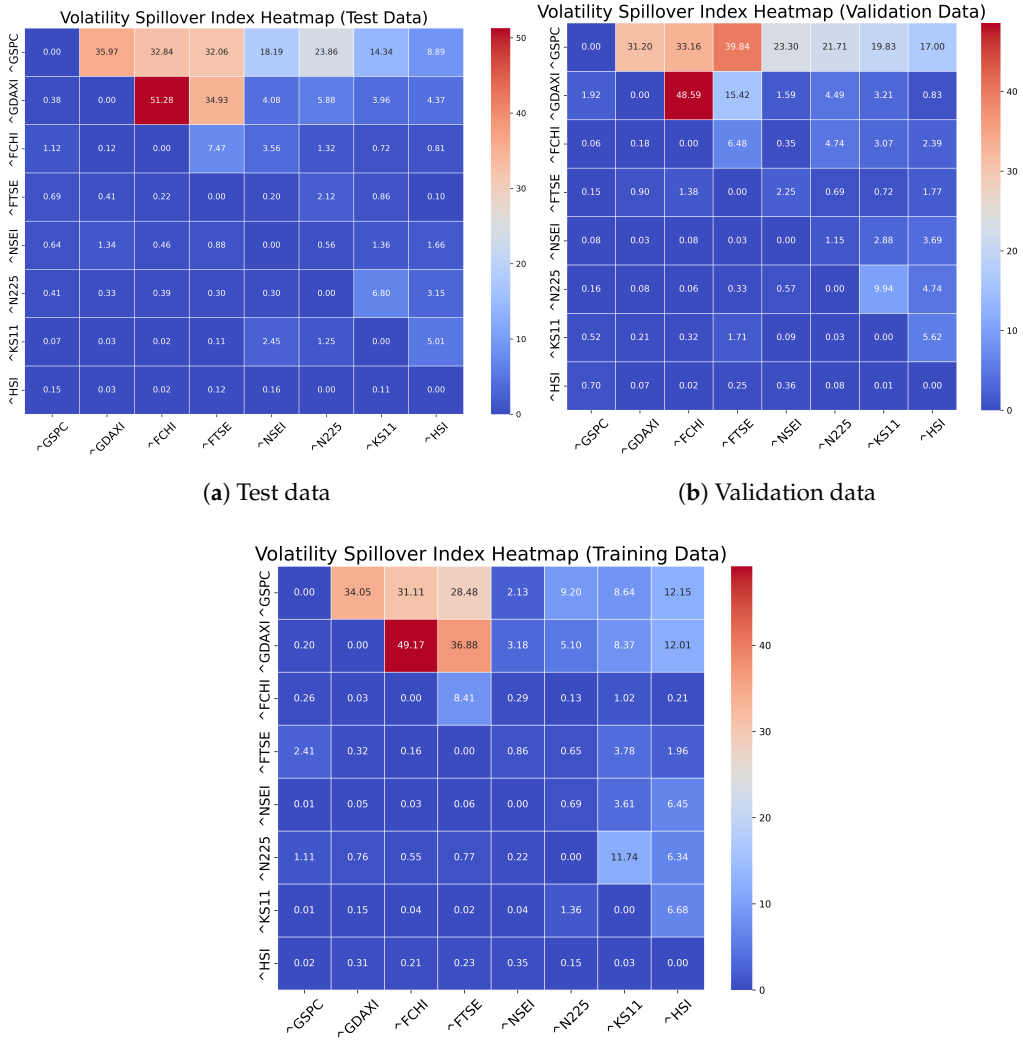


(**a**) Test data (**b**) Training data

(**c**) Validation data

Figure 7. Correlation index heatmaps of train, validation and test.

In Figure 8, we observe the strongest spillover (49.17) in the training dataset, which indicates that fluctuations in GDAXI significantly influence FCHI's volatility. This relationship remains robust in both the training and test datasets (51.28), suggesting a consistent dynamic between these indices. There is also a spillover value of 34.05 in the training dataset, suggesting a significant relationship, and these indicate that movements in the S&P 500 can affect the volatility of GDAXI. Also, the spillover value remains high at 35.97 in the test dataset, further confirming the interconnectedness between these indices. The diagram also indicates some changes in the spillover graphs, especially between FCHI and GSPC. We observed that the spillover from the training data (18.19) to the test data (14.34) decreased, suggesting that the influence of French market volatility on US markets may be weaker. This phenomenon could result from varying market conditions or economic factors that affect regions differently over time. Finally, the spillover between the HSI and the NSEI is low. These indices often exhibit spillover values close to zero, indicating that fluctuations in the other markets studied have less influence on them. For example, NSEI has several spillover values of 0.00, highlighting its independence.



(**a**) Test data (**b**) Validation data

(**c**) Training data

Figure 8. Volatility Spillover index heatmaps of train, validation and test.

Thus, the Temporal GAT model, when applied to volatility spillover indices, exhibits superior performance compared to models based solely on correlation analysis. The model identifies not just the relationships between indices but also how volatility evolves and impacts markets over time, leading to a more robust prediction and analysis framework. This makes the Temporal GAT model a valuable tool for understanding the complexities of financial markets, especially in periods of heightened volatility.

5.5. Sensitivity Analysis

The sensitivity analysis in this section focuses on evaluating the Temporal GAT model's response to varying configurations and input parameters. The aim is to assess how changes in the temporal aspects and node features impact the model's performance. We investigate three specific configurations, starting with variations in time window size to capture different levels of temporal dependencies, then evaluating the impact of different node features and concluding with analyzing the influence of hyperparameter tuning. Each of these aspects is explored in the subsections below.

5.5.1. Temporal Aspects (Time Window Size)

The time window size is a critical parameter that determines the extent of historical data used for volatility prediction. The Temporal GAT model is exposed to different temporal patterns by varying the time window size, thereby capturing short- or long-term dependencies in financial time series data. In this analysis, we analyze the trend using four time window sizes (Note: Window size of 21 represents approximately one trading month.), 5, 15, 21, and 40 days, to investigate their effect on the prediction accuracy and model stability. Table 5 summarizes the comparative performance of different time window sizes for each metric—MAFE, MSE, RMSE, and MAPE across the eight market indices. This comparison highlights the Temporal GAT model's sensitivity to temporal aspects and guides the selection of an optimal window size across different forecasting scenarios.

Table 5. MAFE, MSE, RMSE and MAPE values for Different Window Sizes.

Indices	Window Size 5	Window Size 15	Window Size 21	Window Size 40
		MAFE		
GSPC	0.012568	0.009735	0.013323	0.014385
GDAXI	0.009990	0.008592	0.008714	0.012625
FCHI	0.010539	0.007646	0.008255	0.011707
FTSE	0.010932	0.005371	0.007062	0.005915
NSEI	0.009947	0.007371	0.009622	0.010220
N225	0.011270	0.011244	0.011852	0.018505
KS11	0.011495	0.009918	0.013362	0.014923
HSI	0.008951	0.006308	0.006439	0.008495
		MSE		
GSPC	0.000192	0.000124	0.000215	0.000249
GDAXI	0.000152	0.000123	0.000126	0.000248
FCHI	0.000168	0.000108	0.000116	0.000220
FTSE	0.000154	0.000040	0.000067	0.000052
NSEI	0.000127	0.000106	0.000169	0.000184
N225	0.000244	0.000268	0.000300	0.000607
KS11	0.000158	0.000156	0.000263	0.000349
HSI	0.000123	0.000081	0.000086	0.000122
		RMSE		
GSPC	0.013854	0.011133	0.014654	0.015787
GDAXI	0.012341	0.011079	0.011232	0.015744
FCHI	0.012975	0.010415	0.010778	0.014818
FTSE	0.012402	0.006360	0.008201	0.007222
NSEI	0.011273	0.010314	0.013000	0.013582
N225	0.015627	0.016380	0.017319	0.024636
KS11	0.012561	0.012508	0.016211	0.018693
HSI	0.011093	0.008983	0.009269	0.011024

Table 5. *Cont.*

Indices	Window Size 5	Window Size 15	Window Size 21	Window Size 40
		MAPE		
GSPC	163.77	50.09	54.71	39.59
GDAXI	81.60	21.27	18.58	16.45
FCHI	83.23	20.40	20.03	16.50
FTSE	104.93	22.98	25.55	14.62
NSEI	89.19	26.02	28.67	20.52
N225	75.61	23.24	20.56	21.86
KS11	inf	40.78	44.77	33.51
HSI	59.45	15.64	13.95	12.85

From Table 5 and Figure 9, we observe that across most indices, the error metrics (MAFE, MSE, RMSE) generally increase with window size. When window sizes are smaller, such as 15, FTSE performs well across a wide range of error metrics, suggesting that it is simpler to forecast with high accuracy. Across all metrics and window sizes, N225 typically exhibits higher errors, especially at larger window sizes (e.g., window size 40), making accurate forecasting more difficult. In particular, MAPE shows a wide range of errors in GSPC, indicating that its percentage forecast error varies considerably across window sizes.

Furthermore, regarding the MAFE, we observe that the GSPC and GDAXI exhibit increasing errors as the window size increases, whereas the FTSE has the lowest MAFE values, particularly for smaller window sizes. For the MSE, the errors generally increase with larger window sizes across all indices, though some, like FTSE, maintain relatively low values throughout. For the RMSE, the errors grow with window size, with N225 consistently showing higher RMSE than other indices. Finally, the GSPC and KS11 demonstrate the highest variation in percentage error, while HSI and GDAXI tend to have more stable MAPE values across different window sizes when considering MAPE. Hence, this indicates that smaller window sizes generally lead to more accurate forecasts, whereas larger window sizes result in increased error and reduced model precision.

5.5.2. Graph Properties (Node Features)

In this section, we explore the Temporal GAT model's sensitivity to variations in node features. Within the proposed framework, each stock index is represented as a node whose features are constructed from different market-based time-series inputs, enabling the model to capture both temporal dynamics and cross-market spillover effects. Four node feature configurations are evaluated: volatility proxies, closing price, trading volume, and a combined price-and-volume specification, to assess how distinct sources of market information contribute to forecasting volatility proxies. Models using volatility proxies or closing prices as node features achieve comparatively strong predictive accuracy, reflecting the fact that both past volatility and price levels carry information relevant to future volatility dynamics.

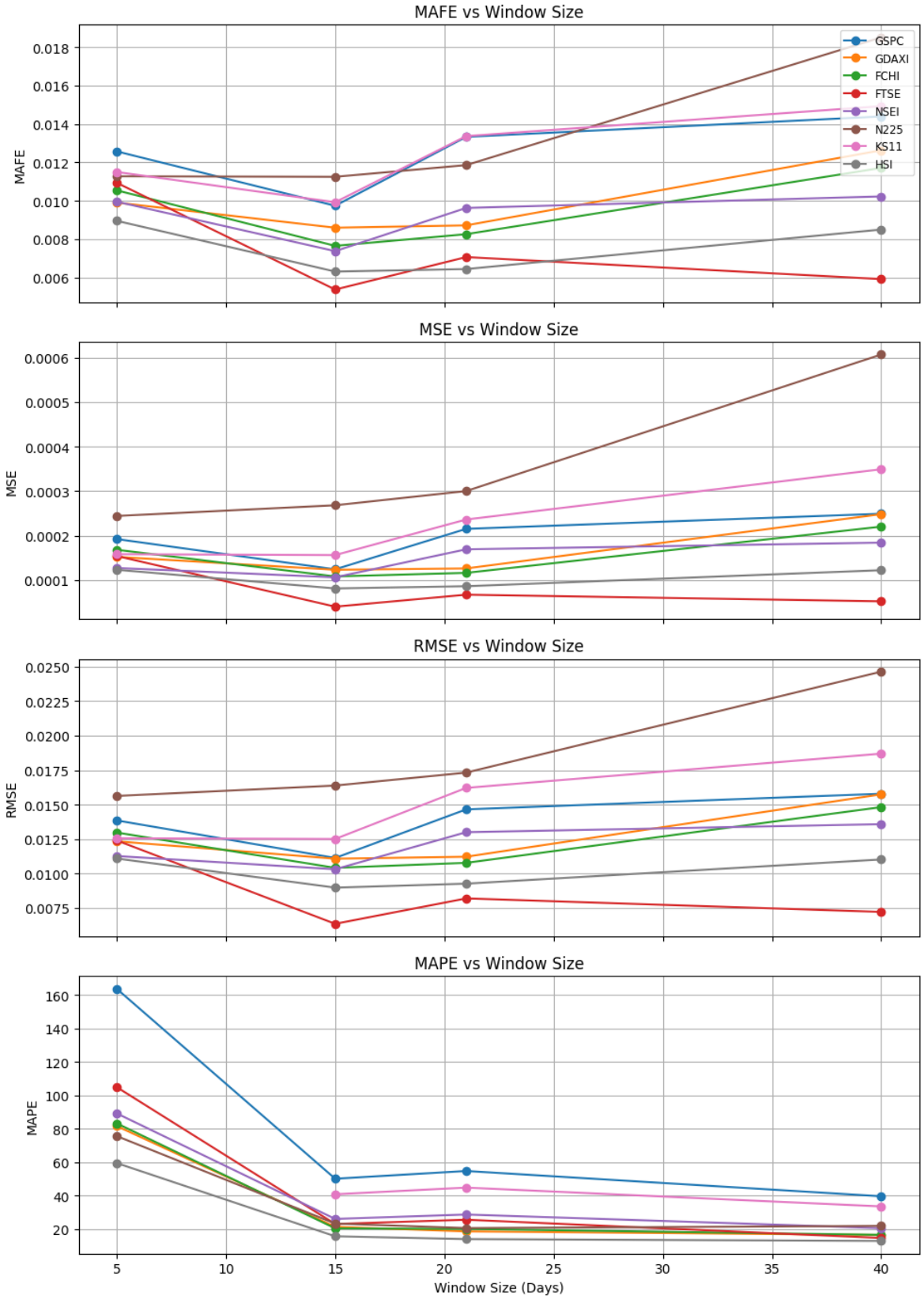


Figure 9. MAFE , MSE, RMSE and MAPE values for Different Window Sizes.

In contrast, trading volume used in isolation yields substantially poorer results, consistent with its well-documented noisiness, regime dependence, and relatively weak short-run relationship with future volatility when not conditioned on price movements [53]. Volume has unstable and limited predictive power for future return volatility once information in

prices and returns is taken into account. In most cases, its apparent explanatory power disappears entirely when proper return-based volatility dynamics are modelled [53]. Thus, feeding the GNN with volume alone deprives the model of the more direct signals about price variability and return magnitudes that are critical for short-horizon volatility forecasting, leading to much weaker performance than feature sets that include volatility proxies or closing prices.

To provide a richer representation, the combined specification assigns each node a two-dimensional feature vector that integrates the closing price and trading volume. This design allows the GNN to learn interactions between price dynamics and liquidity conditions and to evaluate whether volume adds incremental predictive value once contextualized by price behaviour, offering a more comprehensive view of how different market features jointly shape volatility outcomes.

The empirical results presented in Table 6 reinforce these findings across multiple evaluation metrics (MAFE, MSE, MAPE, and R^2) for all eight indices. Averaged across all indices, the RV feature set shows the strongest performance, achieving the lowest average errors (MAFE = 0.0117, MSE = 0.000281, MAPE = 30.68%) and the least negative R^2 (-0.028), confirming that past volatility proxy remains the most informative predictor of future volatility. The CP configuration performs moderately with higher error values (MAFE = 0.5866; MSE = 0.5235) but still substantially better than the V specification, which performs the worst by a wide margin across all metrics (MAFE = 0.8723; MSE = 0.8765; MAPE = 198.9%; $R^2 = -2.400$), indicating that volume alone contributes little meaningful predictive content. The P+V configuration markedly improves on the volume-only model and achieves error levels close to RV (MAFE = 0.0158; MSE = 0.000416), demonstrating that trading volume becomes informative primarily when contextualised by closing-price information.

Table 6. Forecasting performance across node feature specifications.

	GSPC	GDAXI	FCHI	FTSE	NSEI	N225	KSII	HSI
				MAFE				
RV	0.01038	0.01203	0.01369	0.00953	0.01147	0.01647	0.00958	0.01054
CP	0.53324	0.65825	0.54672	0.39054	0.62633	0.62317	0.56203	0.74837
V	0.70695	0.94387	1.00618	0.77059	1.85584	0.83266	0.72560	0.65147
P + V	0.01512	0.01669	0.01940	0.01352	0.01837	0.02243	0.00144	0.01445
				MSE				
RV	0.00020	0.00031	0.00033	0.00020	0.00027	0.00052	0.00019	0.00023
CP	0.34938	0.50789	0.43353	0.22730	0.47758	0.57127	0.70354	0.79847
V	0.67781	1.09315	1.29038	0.89725	4.02200	1.00947	0.96413	0.75261
P + V	0.00030	0.00042	0.00053	0.00029	0.00043	0.00073	0.00028	0.00032
				MAPE				
RV	37.04211	22.93590	34.40292	23.65555	34.19568	33.55663	33.73245	21.24218
CP	21.81313	24.96826	30.48960	78.10573	25.68947	22.99444	38.66971	43.49560
V	303.37488	158.96262	168.50649	299.64072	106.86623	161.10662	163.27484	224.37820
P + V	62.84003	44.57061	57.61976	43.13610	63.60684	51.70399	54.46960	36.23097
				R^2				
RV	0.17858	0.35234	0.38208	0.38691	−0.63729	0.32501	−0.93044	−0.28627
CP	0.17633	0.03132	0.06643	0.53770	0.08088	−0.16583	−0.29396	0.28376
V	−1.18369	−2.97524	−1.68093	−1.07109	−6.06159	−0.48633	0.03771	−0.28328
P + V	−0.02347	0.11547	0.00925	0.11637	−1.59084	0.05192	−1.90584	0.01636

Overall, the results show that the choice of node features substantially influences forecasting performance. While closing prices provide moderately informative signals for volatility prediction, trading volume alone offers a highly unstable and weak foundation for short-term forecasts, as reflected in its significant average errors and strongly negative R^2 values. In contrast, the volatility proxy and the combined price-and-volume specification deliver the highest accuracy, indicating that these feature sets best capture the dynamics relevant for the Temporal GAT model across multiple forecast horizons.

Remark 3 (Volume-Based Features)**.** *The strongly negative R^2 values for volume-based models likely stem from the pronounced non-stationarity of raw trading volume, which exhibits long-term trends and structural shifts that standardization alone cannot remove. During the data preprocessing phase of this research, the volume is z-score normalized on a per-asset basis. This preprocessing addresses scale differences but does not correct underlying temporal instabilities that hinder generalization. As a result, volume provides a limited predictive signal for future volatility in the considered setting, leading to degraded out-of-sample performance. This explains the inferior performance of volume-based models relative to return- or volatility-based specifications and motivates the focus on volatility proxies as the primary predictive features.*

5.5.3. Model Hyperparameters

In this section, we explore the sensitivity of the Temporal GAT model to variations in key hyperparameters, including hidden dimensions, number of heads, and learning rates. Adjusting these parameters is crucial in determining the model's ability to generalize and capture complex relationships within the data. To identify the optimal configuration, we conducted a comprehensive grid search over a range of hyperparameter values, as described below.

1. **Hidden Dimensions**: We experimented with three values for the hidden dimensions: 32, 64, and 128. The hidden dimension defines the size of the feature space in the hidden layers of the model. A larger hidden dimension allows the model to learn more complex patterns, but can lead to over-fitting if not appropriately regularised.
2. **Number of Heads in GAT Layer**: We tested two values for the number of heads: 4 and 8. The number of heads controls the level of attention the model can distribute across different nodes in the graph, influencing the aggregation of information from neighbouring nodes.
3. **Learning Rates**: We evaluated three learning rates, 0.0001, 0.001, and 0.01, to determine the optimal step size for updating the model's parameters. An appropriate learning rate ensures effective convergence during training while avoiding oscillations or premature stagnation.

Furthermore, the model was implemented and trained using Google Colab with GPU acceleration. The training set covered 50% of the data, with 20% used for validation and 30% for testing. Performance metrics included MAFE, MSE, R^2, and MAPE, computed across different forecasting horizons (h = 1, 5, 15, 21) corresponding to short-term (1 day), mid-term (1 week and 2 weeks), and long-term (1 month) forecasts. The model's robustness was tested under different market conditions, including the COVID-19 crisis, which introduced significant market stress. The grid search was conducted over all possible combinations of these hyperparameters, yielding 18 configurations. Each configuration was rigorously evaluated over 70 epochs using MSE as the loss function. The training and validation losses for each configuration were tracked to monitor convergence and identify the best-performing model. The results showed that the configuration with a hidden dimension of 64, 4 heads, and a learning rate of 0.001 achieved the lowest validation loss, indicating

the best generalization capability. Table 7 summarizes the performance metrics for the best configuration.

Table 7. Performance of the best model configuration.

Hyperparameter	Value
Hidden Dimensions	64
Number of Heads	4
Learning Rate	0.001
Final Training MSE	0.000085
Final Validation MSE	0.000163
Final Validation MAPE	34.71%

The training and validation loss trends for the best configuration are shown in Figure 10. The loss values decrease steadily over the epochs, converging to a low value without significant overfitting. This indicates that the selected hyperparameters provide a good balance between model complexity and training stability.

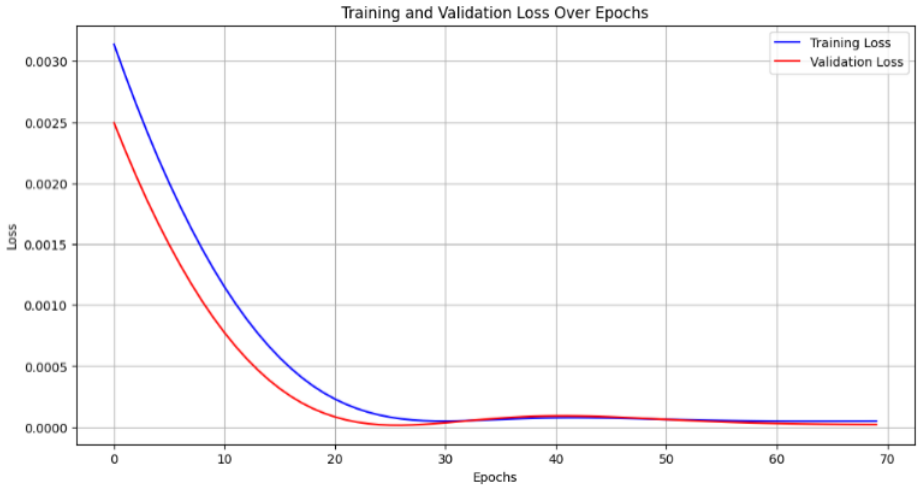


Figure 10. Loss function for optimal hyperparameters.

Overall, the hyperparameter tuning process revealed that moderate hidden dimensions and fewer heads in the GAT layer, combined with a learning rate of 0.001, yielded the best predictive performance. These settings will be used as the default configuration for subsequent model evaluations.

Remark 4 (Layer Depth in GATs)**.** *While attention heads and hidden dimensions are important hyperparameters in GATs, the number of stacked layers plays a more critical role in defining the model's effective receptive field. Each additional GAT layer allows a node to aggregate information from one hop farther in the graph. Consequently, a two-layer GAT aggregates from a node's second-order neighbourhood, capturing both direct and indirect structural dependencies. This study adopts two GAT layers, reflecting an assumption that volatility spillovers are primarily transmitted within two degrees of separation. This configuration offers a practical balance between model expressiveness and the risk of over-smoothing, which can degrade performance in deeper GNNs. Empirical trials confirmed that increasing the number of layers beyond two yielded marginal improvement but increased training instability. However, this architectural choice implicitly limits the model's capacity to capture long-range dependencies that may be relevant in global financial systems. Future*

extensions could investigate deeper GAT architectures or dynamic neighbourhood expansion to model higher-order spillover pathways more effectively.

5.6. Robustness Test

Robustness tests were conducted to evaluate the performance of the Temporal GAT model under different market conditions.

5.6.1. Scenario Analysis Using Temporal GAT

Specifically, two distinct periods were selected: a high-volatility period (1 May 2008 to 1 September 2009) and a low-volatility period (1 April 2014 to 1 March 2016). The comparison between these periods allows a detailed examination of how the model performs under markedly different levels of market uncertainty and price fluctuations. During the high-volatility period, the model was exposed to significant market turbulence, including the 2008 global financial crisis. This period is characterized by abrupt changes in asset prices and increased uncertainty. As a result, the volatility proxies were significantly higher, and the prediction errors tended to increase. Table 8 shows the error metrics for a high-volatility period for each stock index across different forecast horizons.

Table 8. Error Metrics by Horizon and Index for high-volatility period.

Horizon	Index	MAFE	MSE	RMSE	MAPE
1	GSPC	0.010560	0.000164	0.012824	8.99%
1	GDAXI	0.009123	0.000117	0.010810	8.83%
5	FCHI	0.009927	0.000153	0.012350	10.07%
10	FTSE	0.016336	0.000389	0.019730	19.42%
22	NSEI	0.012764	0.000255	0.015970	15.27%

For the **Short-term Forecasts (Horizon 1)**, the MSE and the RMSE values for the high-volatility period were generally higher compared to the low-volatility period, indicating that the model struggled to capture the rapid price changes accurately. The MAPE for most indices ranged from 7.9% to 18.8%, reflecting the difficulty of predicting extreme price movements. FTSE and NSEI exhibited the highest prediction errors, likely due to their higher exposure to the financial crisis. For the **Medium-term Forecasts (Horizon 5 and 15)**, and with increased horizons, the prediction errors compounded, as shown by the increasing MSE and RMSE values. This suggests that while the Temporal GAT model could capture short-term fluctuations, its ability to predict long-term trends in highly volatile environments diminished. This effect is most noticeable in NSEI and KS11, where the MAPE reached 18% and 17%, respectively, for horizon 15. Finally, the errors became more pronounced for the **Long-term Forecasts (Horizon 21)**, with MAPE values exceeding 16% for most indices. This outcome is expected, as long-term predictions under high-volatility conditions are inherently challenging due to the increased uncertainty in market movements.

Next, we consider the low-volatility period, spanning from 1 April 2014 to 1 March 2016, which represents a more stable market environment with less pronounced price movements. During this period, the model exhibited considerably higher prediction accuracy. Table 9 shows the error metrics for the low-volatility period for each stock index across different forecast horizons.

Table 9. Error Metrics by Horizon and Index for low-volatility period.

Horizon	Index	MAFE	MSE	RMSE	MAPE
1	GSPC	0.008006	0.000081	0.009014	28.78%
1	GDAXI	0.024644	0.000693	0.026334	37.68%
5	FCHI	0.019780	0.000554	0.023547	30.93%
10	FTSE	0.004384	0.000027	0.005168	11.86%
22	HSI	0.021983	0.000867	0.029452	30.22%

For the **Short-term Forecasts (Horizon 1)**, the MSE and RMSE values were significantly lower compared to the high-volatility period, indicating that the model could better capture the more predictable price trends. For instance, the MAPE values for most indices ranged from 11.4% to 37.6%, indicating improved predictive capability relative to the high-volatility period. This outcome suggests that the Temporal GAT model can accurately track minor price variations when the market is stable. For the **Medium-term and Long-term Forecasts (Horizon 5, 15, and 21)**, we observed that as the forecast horizon increased, the model maintained its robustness, with MAPE values stabilizing between 13% and 30% for most indices. However, there was a noticeable increase in errors for some indices, such as GDAXI and FCHI, which may be attributed to sporadic price shocks even in low-volatility environments.

Furthermore, the scenario analysis demonstrated that the Temporal GAT model responds differently across market regimes, performing better under stable conditions than during turbulent ones. During high-volatility periods, the model naturally exhibits greater prediction errors due to sudden market shocks, rapid structural changes, and heightened uncertainty that reduce the predictability of financial time series. These spikes in volatility disrupt previously learned temporal patterns and alter spillover relationships more abruptly than the model or any forecasting method can fully adapt to in real time. Nonetheless, the increase in error does not indicate model instability, and the model maintains robust relative performance. The model continues to capture evolving cross-market dynamics more effectively than traditional econometric models or static GNN benchmarks, even when the environment becomes substantially more unpredictable.

Conversely, during low-volatility periods, the model benefits from smoother temporal behaviour and more consistent spillover structures. Under these conditions, patterns in volatility proxy evolve gradually, enabling the model to leverage temporal attention and piecewise-static graph relationships more effectively. As a result, the Temporal GAT achieves lower MSE and MAPE values across all forecasting horizons. This comparison highlights the importance of market regimes in shaping forecasting accuracy and underscores that no model performs equally well across all conditions. Instead, different volatility environments require distinct modelling considerations, and the Temporal GAT's ability to remain stable while adjusting to changing structural dependencies reinforces its practical value despite the inherent challenges of forecasting during turbulent periods.

5.6.2. Model Comparison Using Diebold–Mariano Tests

To statistically evaluate the predictive accuracy of competing models, we applied the DM test [54]. The DM test assesses whether the forecast errors of two competing models differ significantly in expected loss, and it was implemented using squared-error loss functions over identical evaluation periods. A significant DM statistic (typically at the 5% level) indicates that one model statistically outperforms the other in forecasting accuracy.

The DM analysis (Table 10) provides strong and consistent evidence that the Temporal GAT architecture is the most accurate volatility forecasting model across the entire model set.

The pattern that emerges is strongly asymmetric: the TGATM consistently dominates nearly all other architectures, with large negative DM MSE statistics against GARCH–TGATM (-9.86), DGNN-GATM (-3.17), BM (-3.99), C-TGATM (-2.23), SGNN-GATM (-4.48) and LSTM (-3.06). These values indicate significantly lower forecast errors for Temporal GAT relative to all competing models, with associated p-values typically well below 0.01. This superiority underscores the importance of combining temporal sequence modelling with graph attention mechanisms, which jointly capture both cross-market spillover dynamics and nonlinear volatility evolution.

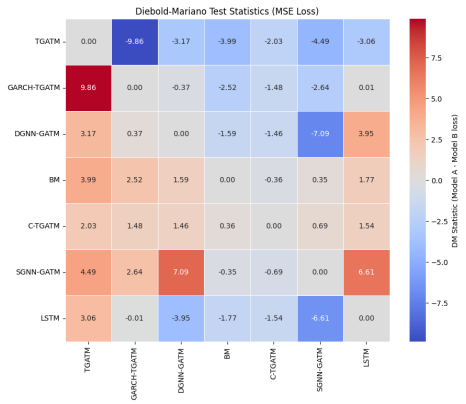
The cross-model comparison reveals broader structural insights into forecasting performance. The GARCH–TGATM, despite embedding a parametric GARCH volatility structure, performs substantially worse than the TGATM, indicating that the hybridization does not yield additional predictive benefit and may instead introduce noise. The SGNN-GATM and DGNN-GATM, lacking explicit temporal encoding, show particularly poor performance, as reflected in consistently large-magnitude DM statistics. In contrast, LSTM and BM models perform moderately well, benefiting from temporal modelling but failing to reach the accuracy of graph-aware temporal architectures. The C-TGATM sits between these extremes: although it incorporates graph structure and temporal encoding, its DM statistics against other models (particularly the TGATM) are smaller in magnitude and occasionally insignificant, suggesting more modest forecast improvements.

Table 10. Diebold–Mariano Test Results (MSE and MAFE Loss).

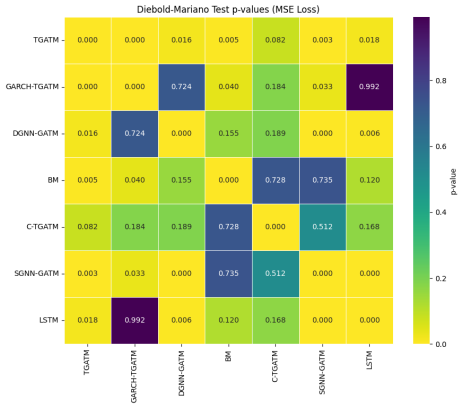
Model A	Model B	DM (MSE)	p-Value (MSE)	DM (MAFE)	p-Value (MAFE)
GARCH-TGATM	TGATM	9.863707	0.000023	7.100276	1.936×10^{-4}
GARCH-TGATM	SGNN-GATM	−2.642857	0.033285	−2.620082	3.441×10^{-2}
GARCH-TGATM	LSTM	0.010493	0.991921	0.678914	5.190×10^{-1}
DGNN-GATM	TGATM	3.171346	0.015678	3.893770	5.946×10^{-3}
DGNN-GATM	GARCH-TGATM	0.367177	0.724336	−0.319405	7.587×10^{-1}
DGNN-GATM	SGNN-GATM	−7.092131	0.000195	−inf	0.000×10^{0}
DGNN-GATM	LSTM	3.952497	0.005514	7.331685	1.584×10^{-4}
BM	TGATM	3.993997	0.005230	4.730426	2.131×10^{-3}
BM	GARCH-TGATM	2.517910	0.039932	2.517277	3.997×10^{-2}
BM	DGNN-GATM	1.593996	0.154966	1.596603	1.544×10^{-1}
BM	C-TGATM	−0.361873	0.728124	0.167757	8.715×10^{-1}
BM	SGNN-GATM	0.352707	0.734690	0.357023	7.316×10^{-1}
BM	LSTM	1.772022	0.119682	1.767575	1.205×10^{-1}
C-TGATM	TGATM	2.028853	0.082052	2.597088	3.558×10^{-2}
C-TGATM	GARCH-TGATM	1.475456	0.183596	1.136454	2.932×10^{-1}
C-TGATM	DGNN-GATM	1.455573	0.188845	1.209097	2.659×10^{-1}
C-TGATM	SGNN-GATM	0.691030	0.511804	0.122774	9.057×10^{-1}
C-TGATM	LSTM	1.537160	0.168138	1.341608	2.216×10^{-1}
SGNN-GATM	TGATM	4.487091	0.002842	6.715716	2.735×10^{-4}
LSTM	TGATM	3.059730	0.018330	3.683617	7.822×10^{-3}
LSTM	SGNN-GATM	−6.614966	0.000300	−72.518043	2.494×10^{-11}

The visualization of DM statistics and p-values (Figure 11) further reinforces these findings. Heatmaps reveal strong and consistent blocks of significant negative DM values in the Temporal GAT row and column, confirming its dominance across nearly all competitors. The MAFE-based DM tests expand this narrative: for example, the TGATM produces substantially lower absolute forecast errors than the DGNN-GATM (with DM = -3.84) and SGNN-GATM (DM = -7.09), while the SGNN-GATM exhibits extreme instability, such as an implausibly large DM (MAFE) of -72.5 when compared with LSTM. Overall, the combined MSE and MAFE evidence supports a unified conclusion: the TGAT is the only

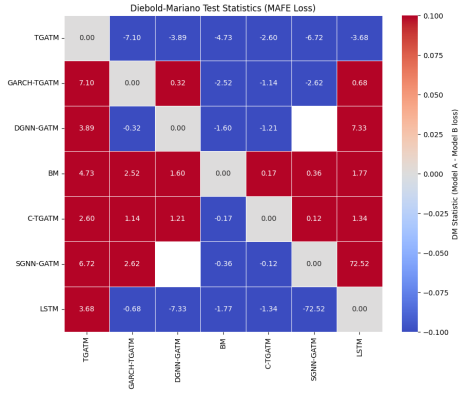
model that consistently delivers statistically superior forecasts across all major financial indices, validating it as the most reliable architecture for multi-horizon volatility prediction.



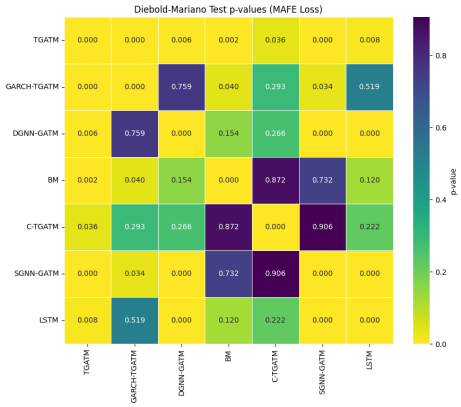
(**a**) Diebold–Mariano Test Statistics (MSE Loss)



(**b**) Diebold–Mariano Test *p*-values (MSE Loss)



(**c**) Diebold–Mariano Test Statistics (MAFE Loss)



(**d**) Diebold–Mariano Test *p*-values (MAFE Loss)

Figure 11. MSE and MAFE comparison of DM test statistics and corresponding *p*-values.

Furthermore, from the *p*-value heatmaps in Figure 11, the lighter cells represent lower *p*-values (typically below 0.05), indicating that the difference in forecasting accuracy between the two models is statistically significant and unlikely to be due to random variation; such cells show that the models do not perform equivalently. Conversely, darker cells correspond to higher *p*-values (0.05 or greater), indicating no statistically significant difference in their predictive performance, and any observed variation may reflect random noise. Applying this interpretation to the model comparisons reveals that TGATM is significantly different from all other models, which explains why TGATM is light-coloured across all comparisons. SGNN-GATM and LSTM differ significantly from most competing models, and the remaining models—GARCH-TGATM, DGNN-GATM, BM, and C-TGATM—form a cluster with no statistically significant differences among them.

5.6.3. Bootstrap Confidence Interval Analysis for Forecasting Accuracy

Bootstrap Procedure: Let $L_t(m)$ denote the forecasting loss (MAFE or MSE) of model m for index t, and let $L_t(\text{TGATM})$ denote the corresponding loss for the TGATM benchmark. For each model m, we compute the loss differential

$$d_t(m) = L_t(m) - L_t(\text{TGATM}). \tag{18}$$

The bootstrap test proceeds as follows:

- Compute the sample mean loss differential

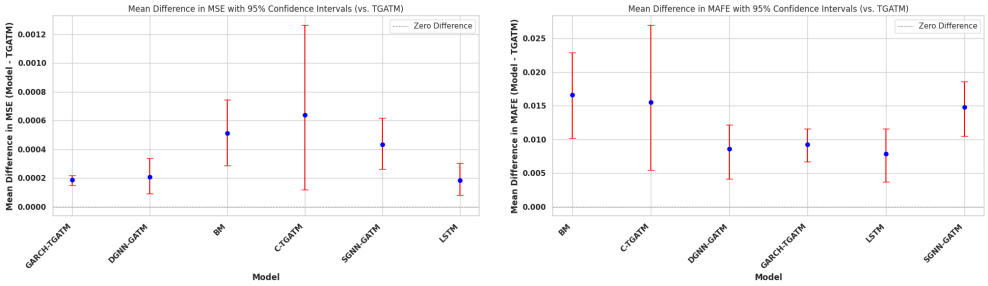
$$\bar{d}(m) = \frac{1}{T}\sum_{t=1}^{T} d_t(m). \tag{19}$$

- Generate B bootstrap samples by drawing from $\{d_1(m), d_2(m), \ldots, d_T(m)\}$ and computing the resampled mean $\bar{d}^*(m)$ for each bootstrap iteration. Note: The bootstrap procedure generates many resampled datasets (e.g., 5000). Each dataset is formed by drawing T loss differentials *with replacement* from the original set, meaning some observations may appear multiple times while others may not appear in a given resample.
- Construct a 95% confidence interval from the empirical bootstrap distribution.

Interpretation:

- If the 95% confidence interval for $\bar{d}(m)$ *excludes zero*, then TGATM performs significantly better than model m.
- If the interval *includes zero*, there is no statistically significant difference in forecasting accuracy.

To assess the statistical robustness of the performance differences between the proposed TGATM model and the benchmark forecasting architectures, we apply a nonparametric bootstrap procedure to the loss differentials. Figure 12 displays the mean differences in MAFE and MSE, respectively, between each competing model and the TGATM benchmark, together with their associated 95% bootstrap confidence intervals. Across both metrics, all confidence intervals lie strictly above zero, indicating that every competing model exhibits significantly higher forecasting errors than TGATM at the 5% level.



(**a**) 95% MSE Bootstrap CIs vs. TGATM (**b**) 95% MAFE Bootstrap CIs vs. TGATM

Figure 12. Mean differences in MAFE and MSE between each competing model and the TGATM benchmark, together with 95% bootstrap confidence intervals.

The relative widths of the intervals provide additional insight into the stability of each model's forecasting behaviour. Certain models (from MSE plot), such as DGNN-GATM, GARCH-TGATM, and LSTM, produce comparatively narrow intervals, implying more stable, though consistently inferior, forecast accuracy. Conversely, models such as C-TGATM and SGNN-GATM exhibit broader intervals, reflecting greater variability in their predictive performance. Crucially, however, even the widest intervals remain entirely above zero for both MAFE and MSE. Taken together, these results demonstrate that TGATM outperforms all benchmark models by a margin sufficiently large that sampling uncertainty does not overturn its advantage.

In our empirical application, all bootstrap confidence intervals for both MAFE and MSE lie strictly above zero, confirming that none of the competing models match the predictive accuracy of TGATM. This provides strong statistical evidence of TGATM's superior and robust forecasting performance across global financial indices.

5.6.4. Expanding-Window Forecasting with Fixed and Dynamic Spillover Graphs

To further examine the robustness of the proposed model, we adopt an expanding-window forecasting framework for multi-asset volatility proxies based on the TGATM. The experimental design considers two alternative strategies for modelling cross-asset dependence during the out-of-sample evaluation: fixing the spillover graph at the end of the validation period, and re-estimating the graph in a rolling or expanding-window manner throughout the test phase. This dual setup is specifically intended to address two key empirical challenges in graph-based forecasting, namely information leakage and structural instability.

Under the fixed-graph specification, volatility series are first aligned across all assets, and a single static spillover network is estimated once using only the training and validation samples. The resulting graph, obtained at the end of the initial validation period, is then held constant and used throughout the expanding-window forecasting exercise on the test dataset at regular forecast intervals. At each forecast origin, the training window expands while the validation window remains fixed. The TGATM is retrained from scratch using the same pre-estimated spillover graph, and multi-horizon forecasts (1, 5, 15, and 21 days ahead) are produced for all assets. By construction, no test-period observations enter the graph estimation stage, thereby preventing look-ahead bias and eliminating information leakage. Forecast accuracy is assessed using both asset-level and aggregated error metrics, which are subsequently averaged over time to summarise horizon-specific performance.

The dynamic graph specification follows the same expanding-window forecasting protocol but replaces the static dependency structure with a time-varying spillover network. In this case, the graph is re-estimated at each forecast origin within the test set using an expanding window that incorporates all information available up to that date. This rolling graph estimation allows the cross-asset volatility transmission network to evolve as new data arrive, directly addressing potential structural instability in financial markets. For each iteration, a new spillover graph is constructed, graph-based datasets are rebuilt accordingly, and the TGATM is retrained from scratch before generating multi-horizon forecasts. Performance metrics are collected across forecast origins and averaged, enabling a direct and transparent comparison with the fixed-graph benchmark.

In both setups, the volatility panel spans 3783 trading days, with the first 1891 days used for initial training and the next 756 days reserved for validation, leaving the remainder for out-of-sample testing under an expanding-window scheme. Forecasts are generated at multiple prediction origins spaced at monthly intervals of 21 trading days. The model uses a rolling feature history window of 20 days and produces forecasts at 1-, 5-, 15-, and 21-day horizons, corresponding to the maximum forecast horizon considered. The above two expanding-window graph estimations are compared with the original framework utilized in this work(We refer to it as Baseline Graph). Forecast accuracy is evaluated using per-ticker and aggregated error metrics, which are then averaged over time to summarise performance by forecast horizon. Table 11 reports the performance.

The Dynamic and Fixed Graph approaches exhibit nearly identical performance at all horizons, with very small differences in MAFE, MSE, RMSE, and MAPE, indicating that re-estimating the spillover network over time provides no systematic advantage over using a single static graph in the expanding-window test. Both graph-based test strategies show relatively large errors and strongly negative R^2 values, suggesting limited explanatory power relative to a naive benchmark. In contrast, the Baseline Graph, which reflects validation-period performance, achieves substantially lower error levels at short horizons and R^2 values closer to zero, highlighting a notable degradation in out-of-sample performance when moving from the validation setting to the more demanding expanding-window test evaluation, particularly at longer forecast horizons.

Table 11. Multi-horizon volatility forecasting performance under alternative network specifications.

Graph Type	Horizon	MAFE	MSE	RMSE	MAPE (%)	R^2
Dynamic Graph	1	0.022398	0.001277	0.024755	42.9318	−5.7202
Dynamic Graph	5	0.023415	0.001319	0.025755	45.9362	−6.2180
Dynamic Graph	15	0.024036	0.001393	0.026295	47.4470	−8.1918
Dynamic Graph	21	0.023514	0.001338	0.025778	45.3593	−5.9914
Fixed Graph	1	0.022390	0.001275	0.024749	42.9295	−5.7231
Fixed Graph	5	0.023435	0.001321	0.025773	45.9643	−6.2282
Fixed Graph	15	0.024089	0.001394	0.026349	47.5727	−8.2449
Fixed Graph	21	0.023508	0.001336	0.025772	45.3399	−5.9886
Baseline Graph	1	0.006305	0.000055	0.007442	21.4795	−0.0599
Baseline Graph	5	0.006034	0.000050	0.007104	27.8136	−0.1739
Baseline Graph	15	0.011800	0.000184	0.013566	46.3775	−3.1084
Baseline Graph	21	0.017903	0.000372	0.019276	69.4633	−6.2783

6. Conclusions

This study examined the critical task of volatility forecasting, a fundamental component of financial risk management and investment decision-making. While traditional econometric models such as GARCH effectively capture volatility clustering, they remain limited in their ability to represent the nonlinear, interconnected, and dynamically evolving structure of global financial markets. To address these limitations, this research introduced a Temporal Graph Attention Network (TemporalGAT) that models international markets as graphs, with nodes representing equity indices and edges encoding interdependencies derived from correlation and volatility spillover networks. By integrating an LSTM-based temporal encoder with GCN and GAT layers, the proposed framework jointly captures sequential volatility patterns and cross-market spillover effects, providing a richer and more flexible modelling approach than conventional methods.

Using 15 years of daily data from eight major global stock indices, the empirical evaluation demonstrates that TemporalGAT achieves strong predictive performance across multiple forecast horizons and consistently competes with or outperforms a range of benchmarks, including GARCH, MLP, LSTM, and alternative GNN architectures. Spillover-based graphs were shown to be more informative than correlation-based graphs, yielding more precise representations of directional shock transmission across markets. Forecast accuracy, assessed using MSE and MAFE, was further validated through Diebold–Mariano tests and bootstrap confidence interval analysis, confirming the statistical robustness of the comparative results. Scenario analyses indicate that, although forecast errors increase during turbulent market periods, reflecting heightened uncertainty and structural instability, the proposed model remains stable and maintains strong relative performance. Sensitivity and leave-one-out analyses further highlight the robustness of the framework and the central role of major indices such as the S&P 500 and DAX in global volatility transmission.

Overall, the findings demonstrate that Graph Neural Networks, and the TemporalGAT architecture in particular, offer a robust and interpretable framework for modelling volatility in interconnected financial markets. Several limitations also point to promising avenues for future research. First, because intraday data are not consistently available across markets, volatility is approximated using squared daily returns, which are noisier than range-based or high-frequency estimators. Future work may therefore explore the use of more efficient daily volatility measures, such as Garman–Klass or Rogers–Satchell estimators, as well as alternative loss functions such as QLIKE, which are standard in econometric volatility forecasting. In addition, while the present study adopts a piecewise-static graph structure, further extensions could incorporate fully dynamic or intraday-

informed graph construction to better capture rapid structural breaks during extreme market events. Extending the framework to other asset classes and integrating additional information sources such as macroeconomic indicators, market sentiment, or microstructure signals also represent valuable directions for advancing adaptive and robust volatility forecasting systems.

Author Contributions: Software, Formal and descriptive Analysis, writing—initial draft and revisions, P.N.K.; Conceptualization, Methodology, Acquisition, Supervision, Writing—Review & Editing, N.U.; Conceptualization, Methodology, Acquisition, Supervision, Writing—Review & Editing, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available in https://finance.yahoo.com.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of the Total Spillover Index

The normalized spillover index quantifies the proportion of the system's total forecast error variance that is attributable to spillovers, shocks originating from other variables, rather than to its own idiosyncratic innovations.

Appendix A.1. Vector Autoregression VAR(p) Model and Moving-Average Representation

Consider an N-dimensional VAR(p) model:

$$\mathbf{x}_t = \sum_{i=1}^{p} \Phi_i \mathbf{x}_{t-i} + \varepsilon_t, \qquad \varepsilon_t \sim (0, \Sigma), \tag{A1}$$

where Σ is the covariance matrix and the diagonal element is denoted by σ_{jj}.

Transforming the VAR model into an infinite moving-average representation:

$$\mathbf{x}_t = \sum_{i=0}^{\infty} A_i \varepsilon_{t-i}, \tag{A2}$$

where $A_0 = I_N$ and A_i are recursively defined by the VAR coefficients.

Appendix A.2. Forecast Error Variance

To measure the impact of one variable's shock on another, the methodology employs the Generalized Variance Decomposition (GVD). The total forecast error variance for variable i at horizon H is

$$\mathrm{Var}\big(e_i'(x_{t+H} - \mathbb{E}_t x_{t+H})\big) = \sum_{h=0}^{H-1} e_i' A_h \Sigma A_h' e_i, \tag{A3}$$

where $\sum$ is the Var matrix for the error vector ε and $\mathbb{E}_t(\cdot)$ denotes the conditional expectation.

Appendix A.3. Generalized Variance Decomposition

The generalized contribution from shocks in variable j is

$$\sigma_{jj}^{-1} \sum_{h=0}^{H-1} (e_i' A_h \Sigma e_j)^2$$

Note: We divide by σ_{jj} in order to standardize the size of shocks in variable j, since they are not orthogonal. The $H-$step generalized shock contribution from j to the forecast error variance of i is given by

$$\theta_{ij}^{(H)} = \frac{\sigma_{jj}^{-1} \sum_{h=0}^{H-1} \left(e_i' A_h \Sigma e_j\right)^2}{\sum_{h=0}^{H-1} e_i' A_h \Sigma A_h' e_i}, \qquad \text{for } i, j = 1, 2, \cdots, N. \tag{A4}$$

Since the shocks are not orthogonalized in the generalized framework, the row sums are not 1; that is, $\sum \theta_{ij}^{(H)} \neq 1$, for $j = 1, 2, \cdots, N$. Next, we perform row-wise normalization.

Appendix A.4. Normalized Spillover Index

- Row sum for variable i

$$S_i^{(H)} = \sum_{j=1}^{N} \theta_{ij}^{(H)}$$

- Normalized spillover share from j to i

$$\widetilde{\theta}_{ij}^{(H)} = \frac{\theta_{ij}^{(H)}}{S_i^H} = \frac{\theta_{ij}^{(H)}}{\sum_{k=1}^{N} \theta_{ik}^{(H)}}, \qquad \text{for } i, j = 1, 2, \cdots, N$$

Also by construction

$$\sum_{j=1}^{N} \widetilde{\theta}_{ij}^{(H)} = 1; \qquad \text{for every } i\,.$$

Appendix A.5. Total Spillover Index

The average share of variance coming from off-diagonal terms

$$\mathcal{S}^{(H)} = \frac{1}{N} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \widetilde{\theta}_{ij}^{(H)} \times 100\,, \qquad \text{where} \quad N = \sum_{i,j=1}^{N} \widetilde{\theta}_{ij}^{(H)} \tag{A5}$$

References

1. Engle, R.F. Statistical models for financial volatility. *Financ. Anal. J.* **1993**, *49*, 72–78. [CrossRef]
2. Cont, R. Empirical properties of asset returns: Stylized facts and statistical issues. *Quant. Financ.* **2001**, *1*, 223. [CrossRef]
3. Son, B.; Lee, Y.; Park, S.; Lee, J. Forecasting global stock market volatility: The impact of volatility spillover index in spatial-temporal graph-based model. *J. Forecast.* **2023**, *42*, 1539–1559. [CrossRef]
4. Mandelbrot, B. Certain speculative prices (1963). *J. Bus.* **1972**, *45*, 542–543. [CrossRef]
5. Engle, R.F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econom. J. Econom. Soc.* **1982**, *50*, 987–1007. [CrossRef]
6. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [CrossRef]
7. Nelson, D.B. Conditional heteroskedasticity in asset returns: A new approach. *Econom. J. Econom. Soc.* **1991**, *59*, 347–370. [CrossRef]
8. Diebold, F.X.; Yilmaz, K. Better to give than to receive: Predictive directional measurement of volatility spillovers. *Int. J. Forecast.* **2012**, *28*, 57–66. [CrossRef]
9. Zakoian, J.-M. Threshold heteroskedastic models. *J. Econ. Dyn. Control* **1994**, *18*, 931–955. [CrossRef]
10. Andersen, T.G.; Bollerslev, T.; Christoffersen, P.F.; Diebold, F.X. Financial risk measurement for financial risk management. In *Handbook of the Economics of Finance*; Elsevier: Amsterdam, The Netherlands, 2013; Volume 2, pp. 1127–1220. [CrossRef]
11. Kim, H.Y.; Won, C.H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst. Appl.* **2018**, *103*, 25–37. [CrossRef]
12. Liu, Y. Novel volatility forecasting using deep learning–long short term memory recurrent neural networks. *Expert Syst. Appl.* **2019**, *132*, 99–109. [CrossRef]

13. Amirshahi, B.; Lahmiri, S. Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies. *Mach. Learn. Appl.* **2023**, *12*, 100465. [CrossRef]
14. Kumar, M.; Thenmozhi, M. Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. *Int. J. Banking Account. Financ.* **2014**, *5*, 284–308. [CrossRef]
15. Kurani, A.; Doshi, P.; Vakharia, A.; Shah, M. A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Ann. Data Sci.* **2023**, *10*, 183–208. [CrossRef]
16. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Hoboken, NJ, USA, 1994.
17. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. Isdn Syst.* **1998**, *30*, 107–117. [CrossRef]
18. Frasconi, P.; Gori, M.; Sperduti, A. A general framework for adaptive processing of data structures. *IEEE Trans. Neural Netw.* **1998**, *9*, 768–786. [CrossRef]
19. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef]
20. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [CrossRef]
21. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 1–23. [CrossRef]
22. Chami, I.; Abu-El-Haija, S.; Perozzi, B.; Ré, C.; Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *J. Mach. Learn. Res.* **2022**, *23*, 1–64. Available online: https://www.jmlr.org/papers/v23/20-852.html (accessed on 7 January 2026).
23. Sawhney, R.; Agarwal, S.; Wadhwa, A.; Shah, R. Deep attentive learning for stock movement prediction from social media text and company correlations. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8415–8426. [CrossRef]
24. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef] [PubMed]
25. Zhang, Z.; Cui, P.; Zhu, W. Deep learning on graphs: A survey. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 249–270. [CrossRef]
26. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv* **2018**, arXiv:1810.00826.
27. Li, Q.; Han, Z.; Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [CrossRef]
28. Yin, X.; Yan, D.; Almudaifer, A.; Yan, S.; Zhou, Y. Forecasting stock prices using stock correlation graph: A graph convolutional network approach. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8. [CrossRef]
29. Alarab, I.; Prakoonwit, S. Graph-based lstm for anti-money laundering: Experimenting temporal graph convolutional network with bitcoin data. *Neural Process. Lett.* **2023**, *55*, 689–707. [CrossRef]
30. Chen, W.; Jiang, M.; Zhang, W.-G.; Chen, Z. A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Inf. Sci.* **2021**, *556*, 67–94. [CrossRef]
31. Chen, Y.; Wei, Z.; Huang, X. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; pp. 1655–1658. [CrossRef]
32. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
33. Ye, J.; Zhao, J.; Ye, K.; Xu, C. Multi-graph convolutional network for relationship-driven stock movement prediction. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 6702–6709. [CrossRef]
34. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
35. Wang, Y.; Aste, T. Network Filtering of Spatial-temporal GNN for Multivariate Time-series Prediction. In Proceedings of the Third ACM International Conference on AI in Finance 2022, New York, NY, USA, 2–4 November 2022; pp. 463–470. [CrossRef]
36. Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; Qi, Y. A semi-supervised graph attentive network for financial fraud detection. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 598–607. [CrossRef]
37. Xiang, S.; Cheng, D.; Shang, C.; Zhang, Y.; Liang, Y. Temporal and heterogeneous graph neural network for financial time series prediction. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management 2022, Atlanta, GA, USA, 17–21 October 2022; pp. 3584–3593. [CrossRef]
38. Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; Achan, K. Inductive representation learning on temporal graphs. *arXiv* **2020**, arXiv:2002.07962. [CrossRef]
39. Sirignano, J.; Cont, R. Universal features of price formation in financial markets: perspectives from deep learning. In *Machine Learning and AI in Finance*; Routledge: London, UK, 2021; pp. 5–15.

40. Diebold, F.X.; Yilmaz, K. Measuring financial asset return and volatility spillovers, with application to global equity markets. *Econ. J.* **2009**, *119*, 158–171. [CrossRef]
41. Hong, J.; Yan, Y.; Kuruoglu, E.E.; Chan, W.K. Multivariate time series forecasting with GARCH models on graphs. *IEEE Trans. Signal Inf. Process. Over Netw.* **2023**, *9*, 557–568. [CrossRef]
42. Yin, Z.; Barucca, P. Neural generalised autoregressive conditional heteroskedasticity. *arXiv* **2022**, arXiv:2202.11285. [CrossRef]
43. Liu, C.; Wang, C.; Tran, M.-N.; Kohn, R. Deep learning enhanced realized GARCH. *arXiv* **2023**, arXiv:2302.08002. [CrossRef]
44. Jahn, M.; Weiß, C.H. Nonlinear GARCH-type models for ordinal time series. *Stoch. Environ. Res. Risk Assess.* **2024**, *38*, 637–649. [CrossRef]
45. Bahdanau, D. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
46. De Prado, M.L. *Advances in Financial Machine Learning*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
47. Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. *arXiv* **2020**, arXiv:2006.10637. [CrossRef]
48. Chordia, T.; Roll, R.; Subrahmanyam, A. Commonality in liquidity. *J. Financ. Econ.* **2000**, *56*, 3–28. [CrossRef]
49. Karolyi, G.A.; Lee, K.-H.; Van Dijk, M.A. Understanding commonality in liquidity around the world. *J. Financ. Econ.* **2012**, *105*, 82–112. [CrossRef]
50. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 2, pp. 729–734. [CrossRef]
51. Andersen, T.G.; Bollerslev, T.; Diebold, F.X.; Ebens, H. The distribution of realized stock return volatility. *J. Financ. Econ.* **2001**, *61*, 43–76. [CrossRef]
52. Barndorff-Nielsen, O.E.; Shephard, N. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2002**, *64*, 253–280. [CrossRef]
53. Gallant, A.R.; Rossi, P.E.; Tauchen, G. Stock prices and volume. *Rev. Financ. Stud.* **1992**, *5*, 199–242. [CrossRef]
54. Diebold, F.X.; Mariano, R.S. Comparing predictive accuracy. *J. Bus. Econ. Stat.* **2002**, *20*, 134–144. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.