Received: 31 August 2025 | Revised: 24 February 2026 | Accepted: 10 April 2026 | Published online: 27 April 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–17
DOI: 10.47852/bonviewAIA62027537

BON VIEW PUBLISHING

Stock Market Movement Forecasting Using Machine Learning and Complex Network Measures

Andrii Bielinskyi[1, 2,*], Vladimir Soloviev[1, 3], Andriy Matviychuk[1] and Vitalii Bezkorovainyi[1]

[1] *Department of Artificial Intelligence, Modeling, and Statistics, Kyiv National Economic University named after Vadym Hetman, Ukraine*

[2] *Department of Economics and Digital Business, State University of Economics and Technology, Ukraine*

[3] *Department of Innovative Technologies and Methods of Teaching Natural Sciences, Southern Ukrainian National Pedagogical University named after K. D. Ushinsky, Ukraine*

Abstract: We investigate whether complexity-inspired network descriptors extracted from visibility-graph (VG) representations of price dynamics can improve short-horizon forecasting of the Standard & Poor's 500 movements. We consider two aligned forecasting tasks at a 14-day horizon: (i) Up/Down directional classification and (ii) regression of the 14-day-ahead standardized forward return. Using daily close prices from 1981 to 2025, we construct sliding window graphs (50/75/100 trading days) and compute a broad set of interpretable spectral, topological, and mesoscopic features, including spectral radius, algebraic/natural connectivity, graph index complexity, clustering/transitivity, efficiencies, assortativity, and path-length statistics. Beyond the standard ("natural") VG, we introduce a volatility-adaptive quantile VG (VAQ-VG) and two additional geometric/motif descriptors—visibility angle entropy and square-motif density—to form a multi-view feature bank. Features are lagged (1–7 days), standardized, and filtered via mutual information. Six learning algorithms (regularized linear models, random forest, k-nearest neighbors, Gaussian process, and stochastic gradient descent) are evaluated under a purged expanding cross-validation protocol with a 150-day embargo to eliminate sliding window leakage, with hyperparameters tuned by randomized search. Empirically, the directional task shows modest but consistent skill: top models reach the area under the receiver operating characteristic curve $\approx$ 0.70–0.72 with accuracy $\approx$ 0.65–0.66, remaining reliably above chance across thresholds. In contrast, point prediction of 14-day return magnitudes remains challenging, with the best R^2 below 0.20 and limited gains from more complex regressors. Overall, VG-derived features provide a compact, interpretable representation that supports leakage-robust directional forecasting, while VAQ-VG yields small, model-dependent shifts that suggest complementary structure across views rather than a decisive single best construction.

Keywords: visibility graph, complex network measures, complexity-inspired machine learning, S&P 500, stock market forecasting

1. Introduction

Financial markets exhibit complex, nonlinear behavior, making stock price forecasting a persistent challenge. According to the Efficient Market Hypothesis, price movements should be largely unpredictable [1]. Nevertheless, machine learning (ML) and deep learning (DL) approaches have advanced stock market prediction by enabling analysis of vast amounts of historical and real-time data [2, 3]. Models such as support vector machines, convolutional neural networks, and recurrent neural networks (e.g., Long Short-Term Memory (LSTM)) have been applied to predict stock prices, volatility, and trends [4]. However, traditional forecasting methods and standard ML models often struggle to handle the full complexity of financial time series [5]. This limitation has sparked interest in new features and frameworks that can more effectively represent market dynamics.

One promising avenue is the application of complexity science to financial markets. Stock markets are quintessential complex systems comprising many interacting agents, leading to emergent properties such as volatility clustering, heavy-tailed returns, and long-range correlations. Complex network models have gained traction for representing and analyzing financial systems [6, 7]. In a network representation, stocks or time points serve as nodes, and edges represent relationships such as correlation or functional links [8]. Combining network-based insights

***Corresponding author:** Andrii Bielinskyi, Department of Artificial Intelligence, Modeling, and Statistics, Kyiv National Economic University named after Vadym Hetman and Department of Economics and Digital Business, State University of Economics and Technology, Ukraine. Email: bielinskyi@kneu.dp.ua

with data-driven ML techniques is a natural next step to improve predictive performance.

The Standard & Poor's 500 (S&P 500) index is the focus of our study. Tracking 500 large publicly traded US companies, it serves as a principal indicator of overall equity market performance and is frequently used in financial modeling research. Forecasting the S&P 500 is of practical value for portfolio management and risk assessment, and it also serves as a benchmark for analyzing market efficiency and complexity.

Purpose of the presented research: Building on the visibility-graph (VG) paradigm of Lacasa et al. [9], our aim is to determine whether structure-aware, complexity-inspired descriptors provide a reliable 14-day-ahead predictive signal for the S&P 500. We transform the index into network representations and extract a rich set of spectral, topological, and geometric features and then integrate them into leakage-averse machine learning pipelines for both directional classification and return-magnitude regression. A central element of the study is a head-to-head comparison between the standard (natural) VG and a volatility-robust construction, the volatility-adaptive quantile VG (VAQ-VG). By evaluating both representations under strict time series validation, we seek to establish not only whether complex network features generalize out of sample but also where the signal resides—whether primarily in direction, in magnitude, or in both.

Novelty and contributions: First, we introduce VAQ-VG, which quantile-normalizes each rolling window to reduce level and trend effects and augments the visibility criterion with a volatility-scaled slack, yielding edge sets that are robust to heteroskedastic bursts without abandoning VG geometry. Second, we enrich the descriptor space with two new measures tailored to price-path geometry: visibility angle entropy (VAE), which summarizes the diversity of local slopes via the entropy of edge-angle distributions, and square-motif density (SMD), which captures mesoscale circuitry through the normalized density of 4-cycles, complementing triangle-based clustering. Third, we build a multi-view, multiscale feature bank that combines natural VG and VAQ-VG descriptors across windows and apply a stability-driven mutual information (MI) filter to retain features that remain informative under resampling, thereby reducing variance without discarding signal. Finally, we adopt a purged expanding cross-validation design with a temporal gap at least as large as the widest feature window, eliminating sliding window leakage and enabling fair comparison across learners.

Integrating network-derived features into ML models could enhance recognition of subtle temporal patterns, regime shifts, or anomalous behaviors in stock index data. This integration addresses a noted gap in financial modeling: while complex systems methods describe financial time series in novel ways, their predictive utility is boosted when paired with learning algorithms that can generalize from data. Our study is motivated by the expectation that combining complex network features with ML will improve stock market movement forecasting.

2. Literature Review

Network science has reframed markets as interacting systems whose structure co-evolves with macroeconomic regimes and shocks [7, 8]. Early econophysics studies revealed sectoral hierarchies and robust communities using correlation filtering and graph constructions, establishing that market co-movements are highly structured rather than random [10, 11]. Contemporary surveys in economics and econophysics underscore the role of financial networks in fragility, contagion, and policy analysis, with special attention to time variation and multilayer interdependencies (e.g., exposures, ownership, funding) [7, 12, 13].

A key empirical regularity is crisis-time densification: during periods of major stress (e.g., the 2008 financial crisis), correlation networks become more interconnected, and centrality measures spike, signaling elevated systemic risk [14, 15]. Recent multilayer studies generalize this to bank-firm or cross-market layers, where spillovers propagate along multiple channels [12, 16, 17]. These insights motivate feature designs that can summarize evolving topology rather than rely solely on contemporaneous return covariances.

Beyond inter-asset networks, a univariate time series can be directly converted into graphs using visibility criteria. The canonical VG of Lacasa et al. connects temporal samples that "see" each other under straight-line visibility [9]. Early finance applications reported heavy-tailed degree distributions, suggestive of long-range dependence and fractal structure [18–21]. Over the 2010s–2020s, multiple VG variants broadened the toolkit: Horizontal VG (HVG), limited-penetrable VG, and multivariate/-multilayer constructions [22, 23]. New constructions map series into signed or weighted VGs to track polarity and strength of local moves [24–26].

VG topology captures nonlinear traits of financial time series—long memory, intermittency, and quasi-periodic structure—which frequently manifest as small-world or scale-free patterns [20, 24, 27]. VG-based irreversibility and asymmetry indices separate persistent trending from mean-reverting turbulence [19, 28]; entropy measures on VG/HVG quantify complexity and have been linked to regime changes [26, 29]. Recent studies show that VG-based representations retain information that complements standard lagged returns and technical indicators [23, 29]. Two practical challenges remain: (i) computational burden for long, high-frequency series and (ii) occasionally opaque links between VG metrics and financial mechanisms [22, 23, 29].

Complexity science offers feature families that summarize dynamical richness beyond linear correlation. Entropy-based indicators (e.g., permutation/Jensen-Shannon, multiscale, and cross-sample entropy) quantify uncertainty and structural heterogeneity and have proven informative for financial series and their VG embeddings [19, 26, 29, 30]. Fractal viewpoints relate scaling/fractal-dimension properties and self-similarity to market efficiency, providing stable descriptors across horizons [31]. Irreversibility measures detect temporal asymmetry tied to drift/volatility bursts and have been used to flag regime shifts [19, 28]. Recurrence analysis captures geometry in reconstructed phase space and offers robust, window-wise regime diagnostics [32]. These paradigms are largely complementary to VG topology: while VG stresses geometry-of-visibility, entropy/fractal/recurrence indices summarize dispersion, scaling, and orbit structure. Combining them can therefore expand the feature space available to predictive models.

Two integration paths dominate. (i) Feature-engineering path: compute network/VG metrics in rolling windows and feed them to conventional ML classifiers/regressors. Studies report that centralities, clustering, and motif counts—alone or alongside technical factors—carry out-of-sample predictive signal or improve risk stratification [11, 13, 33]. (ii) End-to-end graph learning path: encode financial systems as graphs (e.g., by correlations, sectors, supply-chain links), then learn with graph neural networks that propagate information across edges. Recent architectures—dynamic graph attention and multi-relational Graph Neural Networks (GNNs)—jointly model temporal evolution and cross-sectional structure, consistently

outperforming non-graph baselines in trend/movement prediction and stock recommendation [34–37]. In parallel, price-graph approaches construct graphs from the trajectory itself to produce embeddings for downstream neural predictors, addressing long-range dependencies and volatility clustering [36, 37].

Within this landscape, VG-aware pipelines occupy a middle ground: they retain interpretability via explicit, geometry-guided features, yet exploit ML's ability to discover informative combinations and nonlinear interactions. Recent multivariate-VG and segmentation advances further ease integration with portfolio-level prediction by stabilizing window-wise features and focusing attention on change-points [20, 23].

Bridging descriptive network science and predictive modeling remains the central gap. Correlation networks reveal crisis-time densification and sectoral communities [10, 12, 14, 15], but their linear, undirected snapshots can miss nonlinear, path-dependent structure. VG/HVG, in turn, directly encode price-path geometry and expose irreversibility/complexity patterns [19, 26, 28, 29], yet single-view constructions may be sensitive to volatility spikes and level/trend effects. Recent finance-focused studies suggest that multiscale, multi-view graph descriptors paired with leakage-aware time series validation provide modest but consistent directional skill [18, 23, 29, 36–38].

This study proceeds as follows: we build natural VG and a volatility-adaptive, quantile-normalized VG to stabilize edges under heteroskedasticity; we aggregate spectral, topological, and geometric descriptors (including entropy-like and motif-based statistics); and we integrate them into regularized and nonparametric learners evaluated with purged, expanding cross-validation. Relative to prior work, our contribution is (i) an explicitly volatility-robust VG construction; (ii) a compact, interpretable multi-view feature bank that blends VG geometry with complexity indicators; and (iii) a leakage-averse evaluation that clarifies where and when VG-derived signals aid short-horizon forecasting.

3. Research Methodology

3.1. Network measures of complexity

A complex network is modeled as a graph $G = (V,\ E)$ with a vertex set V and an edge set E. In our setting, each vertex $v_i \in V$ encodes one observation of a univariate series (e.g., a stock price at time t_i); more generally, vertices may represent other signals (e.g., applied stress in materials). We index vertices $i = 1, \ldots, N$, where $N = |V|$. An edge $e_{ij} \in E$ connects the pair (i, j). The graph is undirected if $(i, j) \in E \Longrightarrow (j, i) \in E$; otherwise, it is directed. Unless stated, we work with simple, unweighted, loopless, undirected graphs. Connectivity is captured by the (binary) adjacency matrix $A \in \{0, 1\}^{N\times N}$ with entries

$$A_{ij} = \begin{cases} 1, & (i,j) \in E, \\ 0, & \text{otherwise}, \end{cases} \quad A_{ij} = A_{ji},\ A_{ii} = 0 \tag{1}$$

This representation supports standard network descriptors (degree, paths, motifs) used later to quantify complexity.

3.1.1. Visibility graph representation

The VG provides a canonical mapping from a scalar time series to a network. Given $\{(t_i, x_i)\}_{i=1}^{N}$ with strictly increasing times $t_1 < \ldots < t_N$, two vertices $i < j$ are linked if every intermediate datum lies below the straight-line segment joining (t_i, x_i) and (t_j, x_j). Equivalently, for all k with $t_i < t_k < t_j$,

$$x_k < x_j + (x_i - x_j)\,\frac{t_j - t_k}{t_j - t_i} \tag{2}$$

3.1.2. Volatility-Adaptive Quantile Visibility Graph (VAQ-VG)

Let $\{(t_i, x_i)\}_{i=1}^{N}$ be the series (daily closes). Fix a rolling window W. Define the rolling-quantile map

$$q_i = rank_{[i-W+1,i]}(x_i)/(W + 1) \in (0, 1) \tag{3}$$

which removes level/trend effects and yields an approximately uniform marginal per window.

Let σ_i be a local volatility (e.g., rolling mean absolute deviation or standard deviation of Δx over $[i - W + 1, i]$). We build a graph on vertices $\{1, \ldots, N\}$ with an adaptive-slack visibility rule: connect $i < j$ if for all $k \in (i, j)$:

$$q_k < q_j + (q_i - q_j)\,\frac{t_j - t_k}{t_j - t_i} + \tau_{ij}\frac{t_j - t_k}{t_j - t_i} \tag{4}$$

where the volatility-scaled slack:

$$\tau_{ij} = \kappa \cdot \left(\hat{\sigma}_i^2 + \hat{\sigma}_j^2\right)^{1/2}, \hat{\sigma}_i = \text{median-abs-dev}\,(\Delta x)_{[i-W+1,i]} \tag{5}$$

and $\kappa > 0$ controls robustness. In our case, $\kappa = 0.25$. Nevertheless, a grid search over a set of parameters κ should be used to get the most robust results. Setting $\kappa = 0$ recovers the standard VG (after quantile normalization). This construction is (i) monotone-invariant (via quantiles), (ii) robust to heteroskedastic bursts (via τ_{ij}), and (iii) faithful to the VG's geometric spirit.

3.1.3. Spectral descriptors of network structure

Spectral graph theory studies structural properties of a graph through the spectra (eigenvalues/eigenvectors) of its canonical matrices—most commonly the adjacency matrix A and the Laplacian L. For an undirected simple graph G, the Laplacian is

$$L = D - A \tag{6}$$

where D is the diagonal degree matrix with $D_{ii} = \deg(v_i)$. In addition to L, we consider the normalized Laplacian:

$$\mathscr{L} = D^{-1/2} L D^{-1/2} \tag{7}$$

whose eigenvalues are confined to the interval $[0,\ 2]$.

A central quantity derived from the Laplacian spectrum is the algebraic connectivity—$\lambda_2(L)$, the second-smallest eigenvalue of L. The multiplicity of the zero eigenvalue of L equals the number of connected components; hence, $\lambda_2(L) > 0$ if G is connected. Larger $\lambda_2(L)$ indicates stronger global connectivity/edge expansion and faster convergence of consensus/diffusion dynamics, which we interpret as greater fault tolerance and synchronization capacity of the network.

From the adjacency spectrum $\{\lambda_i(A)\}_{i=1}^{N}$, we derive several complementary measures. The graph energy is

$$E(G) = \sum_{i=1}^{N} |\lambda_i(A)| \tag{8}$$

which aggregates the magnitude of spectral modes. Note that $E(G) = 0$ is only for the edgeless graph ($A = 0$); in general, $E(G)$ increases with edge density and degree heterogeneity.

To quantify global coherence and mixing, we use the adjacency spectral gap:

$$\delta_A(G) = \lambda_1(A) - \lambda_2(A), \lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_N(A)$$

A larger δ_A typically implies weaker modular structure and faster random-walk mixing, whereas a small gap is characteristic of community structure and slow relaxation.

The spectral radius $R(G) = \max_{1 \leq i \leq N} |\lambda_i(A)|$ equals the largest adjacency eigenvalue in the undirected case and reflects hub dominance and overall connectivity scale.

Following Jun et al.'s work [39], we also integrate the natural connectivity:

$$N_c(G) = \ln\left(\frac{1}{N}\sum_{i=1}^{N} e^{\lambda_i(A)}\right) \tag{9}$$

which provides a sensitive, robust indicator of path redundancy and structural robustness: higher N_c implies richer ensembles of alternative routes and increased resilience.

Moreover, we consider spectral moments of the adjacency spectrum. For a non-negative integer k,

$$m_k(G) = \sum_{i=1}^{N} \lambda_i(A)^k \tag{10}$$

which counts the number of closed walks of length k in G. These moments capture subgraph prevalence across scales; for example, $m_3 = tr(A^3) = 6\times$ (number of triangles) in simple undirected graphs, linking m_3 to triadic closure and local clustering. Larger m_k values indicate greater structural complexity and redundancy of alternative (shortest and near-shortest) paths. In our computations, we focus on $k = 3$, which provides a compact statistic of local cyclic structure without sacrificing interpretability.

Finally, we consider the graph index complexity (*GIC*), proposed by Kim and Wilhelm's work [40], which compresses spectral information about a graph into a scale-free scalar that peaks at intermediate connectivity. First of all, we define the normalized spectral radius:

$$c = \frac{\lambda_{max}(A) - 2cos\left(\frac{\pi}{n+1}\right)}{(n-1) - 2cos\left(\frac{\pi}{n+1}\right)} \tag{11}$$

where

$$GIC = 4c(1-c) \tag{12}$$

Since $c \in [0, 1]$ for any connected n-vertex graph, $0 \leq GIC \leq 1$. *GIC* vanishes for path-like ($c = 0$) and clique-like ($c = 1$) structures and is maximized at $c = 0.5$, where $GIC = 1$. Thus, complexity is low for very sparse "ordered" topologies and for very dense, near-complete topologies, and highest at intermediate regimes.

In VGs derived from time series, larger λ_{max} typically reflects denser or more hub-dominated visibility patterns (e.g., many prominent extrema), while smaller values indicate chain-like organization. Hence, higher *GIC* values suggest richer, more intricate local–global structure, consistent with more irregular or rapidly fluctuating dynamics.

3.1.4. Topological characteristics of networks

A complex network's topology can be summarized by node- and edge-level indicators that quantify position, reachability, and control in the graph.

The most immediate notion of a node's connectedness is its (unnormalized) degree

$$d_i = \sum_{j=1}^{N} A_{ij} \tag{13}$$

which counts the number of neighbors adjacent to node i. Regarding the degree of vertices, we compute the maximum degree

$$D_{max} = \max\{d_i \mid i = 1, \ldots, N\} \tag{14}$$

We consider such degree-based features, which highlight prominent peaks in the series that are visible to many other points. A higher maximum degree indicates the presence of a time point (price) that dominates visibility over a large range, reflecting a pronounced extremum in the time series.

Let l_{ij} denote the geodesic distance between nodes i and j, defined as the length (edge count) of the shortest path connecting them, by convention $l_{ii} = 0$ and $l_{ij} = \infty$ if no path exists. Building on this notion, the closeness centrality of node i is

$$c_i = (N-1) / \sum_{j=1}^{N} l_{ij} \tag{15}$$

that is, the inverse of the mean shortest path length from i to the rest of the network. The network-level average,

$$\mathscr{C} = \frac{1}{N}\sum_{i=1}^{N} c_i \tag{16}$$

summarizes overall reachability.

To handle disconnected graphs without imputing infinite distances, Marchiori and Latora [41] proposed the harmonic centrality of node j:

$$\mathrm{Hc}_j = \sum_{\substack{i=1 \\ i \neq j}}^{N} (l_{ij})^{-1}, \text{ with } l_{ij}^{-1} = 0 \text{ if nodes } i \text{ and } j \text{ are disconnected} \tag{17}$$

which replaces an average of distances with an average of reciprocals, naturally assigning zero contribution to unreachable pairs. The global harmonic centrality GHc is the arithmetic mean over all nodes.

Betweenness centrality counts how frequently a node lies on geodesic routes between other pairs. Let $\sigma(i,j)$ be the number of shortest paths between i and j, and $\sigma(i,j \mid k)$ is the number of those paths that pass through k. Then

$$b_k = \sum_{\substack{i,j=1 \\ i,j \neq k}}^{N} \frac{\sigma(i,j \mid k)}{\sigma(i,j)} \tag{18}$$

with terms defined as zero whenever $\sigma(i,j) = 0$. In our application to the S&P 500, larger b_k highlights periods that act as "bridges" for information or shock propagation across the market.

Information centrality [42] measures the efficiency of information flow from a node to the rest of the network by aggregating

over all walks with appropriate attenuation. Let $B = \{b_{ij}\}$ be an $N \times N$ matrix defined by

$$b_{ij} = \begin{cases} 0, & \text{if nodes } i \text{ and } j \text{ are adjacent,} \\ 1, & \text{otherwise,} \end{cases} \quad \text{and } b_{ii} = 1 + d_i \tag{19}$$

and set $C = \{c_{ij}\} = B^{-1}$. The pairwise information between i and j is then

$$I_{ij} = \frac{1}{c_{ii} + c_{jj} - 2c_{ij}} \tag{20}$$

Aggregating pairwise terms via a harmonic mean yields the information centrality of node i:

$$\bar{I}_i = \left(\frac{1}{N} \sum_{j=1}^{N} \frac{1}{I_{ij}} \right)^{-1} \tag{21}$$

Using the identities $T = \sum_{j=1}^{N} c_{jj}$ and $R = \sum_{j=1}^{N} c_{ij}$, we obtain

$$\sum_{j=1}^{N} \frac{1}{I_{ij}} = Nc_{ii} + T - 2R \tag{22}$$

so that

$$I_i = \left[\frac{1}{N} (Nc_{ii} + T - 2R) \right]^{-1} = \left[c_{ii} + \frac{T - 2R}{N} \right]^{-1} \tag{23}$$

Finally, the global information centrality summarizes the network's average efficiency of information transmission. In financial networks, higher I_i indicates nodes that maintain short, redundant communication pathways to the market as a whole.

Returning to geodesic distances l_{ij}, a standard global indicator of navigability is the average shortest path length

$$L = \frac{1}{N(N-1)} \sum_{i \neq j} l_{ij} \tag{24}$$

It summarizes how many hops are typically required to connect two arbitrary vertices. In disconnected graphs, l_{ij} may be infinite; in practice, one either (i) restricts L to the largest connected component or (ii) substitutes the harmonic mean of distances (cf. harmonic closeness) to avoid infinities.

Local clustering quantifies the tendency of a node's neighbors to be mutually connected. For node i, the (triadic) local clustering coefficient is

$$C_i^3 = \frac{\sum_{k,j=1}^{N} A_{ik} A_{kj} A_{ji}}{d_i (d_i - 1)} \tag{25}$$

where the numerator counts (twice) the number of triangles incident to i. The global clustering coefficient reflects how likely neighbors form triads.

Next, Barrat and Weigt advocate the network transitivity [43]:

$$T = \frac{\sum_{i,\,k,j=1}^{N} A_{ik} A_{kj} A_{ji}}{\sum_{i,\,k,j=1}^{N} A_{ik} A_{ji}} \tag{26}$$

a global ratio that weights nodes by their participation in connected triples. In simple undirected graphs, this expression is equivalent to $T = 3\times$ (number of triangles)/(number of connected triples); hence, it emphasizes genuine triadic closure regardless of degree heterogeneity.

Following Latora and Marchiori [41], efficiency quantifies how effectively information (or any diffusive process) is exchanged over a network when signals travel along shortest paths. The global efficiency is

$$E_{glob} = \frac{1}{N(N-1)} \sum_{i \neq j} (l_{ij})^{-1} \tag{27}$$

Larger values indicate that, on average, vertices can reach each other through short routes, that is, efficient large-scale communication.

A basic summary of how densely a graph is wired is its edge density:

$$\rho = \frac{E}{E_{max}}, \; E_{max} = \frac{N(N-1)}{2} \tag{28}$$

where E is the number of edges in a simple undirected graph. Thus $0 \leq \rho \leq 1$, with $\rho = 1$ for the complete graph and $\rho = 0$ for the edgeless graph. While ρ is a convenient scalar indicator of overall connectedness, it does not capture heterogeneity (e.g., hubs vs. tails) or mesoscale organization (communities).

Degree-degree correlations reveal whether vertices tend to connect to others with similar (or dissimilar) degrees. A standard diagnostic is the average degree of nearest neighbors for vertices of degree d [44]:

$$d_{nn}(d) = \sum_{d'} d' P(d'|d) \tag{29}$$

where $P(d'|d)$ is the conditional probability that a vertex of degree d is attached to a vertex of degree d'. If degrees are uncorrelated, $P(d'|d)$ does not depend on d, and thus, $d_{nn}(d)$ is constant. An increasing $d_{nn}(d)$ indicates assortative mixing (high-degree vertices preferentially connect to other high-degree vertices), whereas a decreasing trend indicates disassortative mixing [45].

At the node level, the (weighted) average nearest-neighbor degree of vertex i is

$$d_i^w = \frac{1}{s_i} \sum_{j=1}^{N} A_{ij} w_{ij} d_j, \; s_i = \sum_{j=1}^{N} A_{ij} w_{ij} \tag{30}$$

where s_i is the node "strength," A_{ij} the adjacency entry, w_{ij} the edge weight (here $w_{ij} = 1$), and d_j the degree of neighbor j. Large d_i^w signals that i tends to be embedded among high-degree neighbors; small values indicate the opposite. Aggregating d_i^w by degree classes yields an empirical estimate of $d_{nn}(d)$.

A complementary, scalar summary is the assortativity coefficient, that is, the Pearson correlation of a nodal attribute across the two ends of an edge. Let e_{xy} be the mixing matrix for a categorical or discretized scalar attribute (e.g., degree class), with normalization $\sum_{x,y} e_{xy} = 1$, row sums $a_x = \sum_y e_{xy}$, and column sums $b_y = \sum_x e_{xy}$. The assortativity is

$$r = \frac{\sum_{xy} xy \left(e_{xy} - a_x b_y \right)}{\sigma_a \sigma_b}, \; -1 \leq r \leq 1 \tag{31}$$

where σ_a and σ_b are the standard deviations of the endpoint distributions $\{a_x\}$ and $\{b_y\}$. Values $r > 0$ indicate assortativity, $r < 0$ disassortativity, and $r = 0$ the absence of linear mixing by the chosen attribute.

Distances also characterize the network's "size" and centrality structure. The eccentricity of node i is the greatest geodesic distance from i to any other node,

$$\epsilon(i) = \max_j l_{ij} \tag{32}$$

The diameter is the maximum eccentricity, $diam = \max_i \epsilon(i) = \max_i \max_j l_{ij}$, that is, the longest among all shortest paths. The radius is the minimum eccentricity, $rad = \min_i \epsilon(i)$.

"Small-world" structure combines short typical distances with pronounced local clustering. Operationally, this means that (i) the mean geodesic distance L grows slowly (approximately $\log N$) with network size and is comparable to that of an appropriate random graph, while (ii) the clustering is substantially higher than random. Humphries and Gurney formalized this via the small-worldness index:

$$S = \frac{C/C_r}{L/L_r} \tag{33}$$

where C is the global clustering coefficient of the graph under study, L is its average shortest path length, and C_r and L_r are the corresponding quantities measured on a null model (typically an Erdős–Rényi graph with the same N and expected edge density $\rho = 2E/[N(N-1)]$, or a degree-preserving randomization). By construction, $S > 1$ indicates a small-world regime (high clustering relative to random without an accompanying inflation of path lengths), whereas $S \approx 1$ suggests random-like structure and $S < 1$ the absence of small-world organization.

The Travelling-Salesperson Tour Cost (TSP) summarizes network compactness. The tour cost is the total distance when visiting every node exactly once and returning to the start.

Lower TSP cost suggests a tightly knit graph with efficient information circulation; higher cost indicates a sparse, chain-like structure with longer bottlenecks.

We use a standard deterministic approximation sufficient for sliding window analysis.

Visibility angle entropy: VGs encode a time series as a geometric network whose edges reflect unobstructed "lines of sight" between samples. Beyond degree- or path-based summaries, the distribution of edge orientations (slopes) captures the local mix of rising vs. falling segments and their steepness. The VAE quantifies this geometric heterogeneity.

According to the proposed approach, for any edge, define its orientation:

$$\theta_{ij} = \arctan\left(\frac{x_j - x_i}{t_j - t_i}\right) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \tag{34}$$

Let $\mathscr{B} = B_1, \ldots, B_B$ be a partition of $(-\pi/2, \pi/2)$ into B equal-width bins, and let p_b be the empirical (possibly weighted) frequency of angles in bin B_b. The normalized VAE is

$$\mathrm{VAE} = -(\log B)^{-1} \sum_{b=1}^{B} p_b \log p_b \tag{35}$$

Higher VAE indicates a broad, diversified mixture of edge orientations (frequent alternation of slopes and/or co-existence of multiple characteristic steepnesses), typically associated with more irregular or volatile regimes. Lower VAE indicates concentration of orientations (e.g., predominantly gentle upward visibility), consistent with trend-dominated or smoother regimes.

Square-motif density: Triangles quantify clustering at the smallest cycle scale. In VGs of time series, 4-cycles (squares) reflect alternating configurations and provide a complementary mesoscopic motif to triangles. The SMD normalizes the prevalence of such 4-node circuits by the number of possible 4-node subsets.

Let $C_4(G)$ denote the number of 4-cycles in G. We define

$$\mathrm{SMD} = C_4(g) / \binom{|V|}{4} \in [0,\ 1] \tag{36}$$

that is, the probability that four uniformly sampled distinct vertices induce a 4-cycle.

Higher SMD indicates richer alternating local geometry and mesoscopic circuitry in the visibility structure, often arising when the series oscillates between relative highs and lows within the window. Lower SMD suggests more chain-like or tree-like visibility, consistent with strong monotone segments or highly assortative peaks/troughs that do not interleave to complete 4-cycles.

Each of these features is computed for every VG derived from the S&P 500 series. Notably, our feature set includes classical network metrics (degree, clustering, path lengths, etc.) alongside domain-specific measures like small-worldness and Graph Index Complexity (GIC) that were identified as useful in recent VG studies. Each measure serves as a candidate indicator of structural changes in the time series data, allowing us to track the system's behavior (e.g. onset of financial crash) through the lens of graph-theoretic complexity. By referencing the methodology of previous authors, we ensure our feature calculations follow established definitions and best practices in the field.

3.2. Machine learning approaches

We evaluate a diverse set of ML algorithms on the forecasting tasks, using the selected network features as inputs. Both classification and regression models are considered, mirroring the two task formulations (directional classification and numeric prediction). The following models are applied in our experiments:

1) **Logistic Regression**—a linear classification model used for predicting the probability of the index going up or down. We use an ℓ_2-regularized logistic regression for binary classification of next-day forward returns.
2) **Random Forest (RF)**—an ensemble of decision trees that can be used for both regression and classification. We use RF Regressor for predicting next-day returns (regression) and RF Classifier for direction (classification). The ensemble nature of this model helps capture nonlinear interactions between VG features.
3) ***k* − Nearest Neighbors (*k*-NN)**—a nonparametric model that predicts based on the most similar instances in the feature space. For regression, it predicts the average of the k-nearest neighbors' target values; for classification, it uses majority voting among neighbors. k-NN provides a simple benchmark to gauge if complex graph features yield localized patterns useful for forecasting.
4) **Gaussian Process (GP)**—a probabilistic nonlinear model. We employ GP Regressor (with an RBF kernel) for the regression task, which can capture complex functional mappings from network features to future returns along with uncertainty estimates. For classification, a GP Classifier (using a Laplace approximation) is used to model the probability of upward movement.
5) **Stochastic Gradient Descent (SGD)**—a simple yet scalable linear model optimized via SGD. We configure SGD Regressor for predicting returns and SGD Classifier (with logistic loss)

for classification. The SGD models serve as high-bias (simple linear) models that are less likely to overfit, providing a baseline for linear separability of the features.

6) **Ridge Regression/Classification**—a linear model with ℓ_2 regularization. For regression, ridge regression fits a linear model to predict the next-day return while controlling complexity. For classification, we use a Ridge Classifier, which is equivalent to an ℓ_2-regularized least-squares classification approach. Ridge serves to test the predictive power of a linear combination of the graph features, penalizing large coefficients to prevent overfitting.

Each model is trained and tested on the same window-based feature datasets. By including a mix of linear and nonlinear models, as well as ensemble methods, we assess the robustness of VG features for forecasting.

3.3. Time series analysis and sliding window approach

To account for the non-stationarity of financial markets over long periods, we adopt an overlapping sliding window approach to construct multiple VGs from different time segments. Instead of building one VG for the entire 45-year span (which could obscure local patterns), we partition the S&P 500 series into shorter windows and analyze each window's network features. We choose window lengths of 50, 75, and 100 trading days as the maximum span for each VG to account for short-, medium-, and long-term correlations. For the calculated complexity indicators combined into a single array, the time period under study is from December 23, 1981, to August 21, 2025.

We slide the window forward with a fixed step (creating overlapping segments) so that consecutive windows share most of their data. Each windowed segment is converted into its own VG. By scanning across the time series, we obtain a sequence of VGs and their features, creating a time series of networks. This allows us to track changes in network metrics over time and use them for rolling fashion forecasting. Using a sliding window with overlap mitigates look-ahead bias because at each prediction point, only past data within the window is used.

Figure 1 shows the S&P 500 closing prices used for all subsequent calculations based on the VG algorithm, including the 2008 stock market crash and the Black Monday crash in 1987.

As can be seen in Figure 1(b, c), the VG constructed based on S&P closing prices may be more coherent and denser during crisis and pre-crisis periods, which should also be reflected in quantitative indicators and should yield early warning information to our ML models.

For example, Figure 2 demonstrates the comparative dynamics of two network indicators—*GIC* and *ApLen* measures—calculated in a 500-day sliding window, which should more clearly reflect the dynamic complexity of the financial graph.

Reading Figure 2(a), *GIC* tends to surge around regime shifts and turbulent phases (major drawdowns, sharp reversals, or volatile consolidations) and fall during extended, directional bull legs when price drifts persistently upward and the graph becomes more linear. The prominent *GIC* peaks line up with well-known stress periods (late-80s crash, dot-com bust/aftermath, GFC 2008-2009, the COVID shock, and the 2022 bear market), while troughs accompany smoother advances (mid-1990s, 2013-2017, parts of 2019-early 2020, late-2020/2021 uptrend). In other words, *GIC* behaves like a regime/complexity thermometer: high values mark periods when the return process exhibits frequent alternation of extrema and diverse time-scale interactions; low values track tranquil, trend-dominated structure. Therefore, higher *GIC* appears when the series develops heterogeneous "hub-like" connections and a richer mix of local motifs; lower *GIC* reflects a more chain-like, uniform structure.

ApLen in Figure 2(b) behaves as a regime indicator that is complementary to measures like small-worldness or *GIC*: low ApLen flags turbulent, mean-reverting phases with dense short-cutting; high *ApLen* accompanies smoother, trend-dominated phases where the VG stretches toward a chain.

We study two forecasting targets defined at the same horizon h (trading days): (i) directional classification, where the model predicts whether the market moves Up or Down over the next h days, and (ii) magnitude regression, where the model predicts the h-day-ahead standardized forward returns.

The h-day forward returns are computed as

$$R_{t+h} = \frac{S_{t+h} - S_t}{S_t} \tag{37}$$

where S_t is the closing price of S&P 500 at day t, S_{t+h} is the closing price at the future time horizon h (in our case $h = 14$), and the standardized forward returns are calculated as

$$r_{t+h} = \frac{R_{t+h} - \mu_{t,w}}{\sigma_{t,w}} \tag{38}$$

Figure 1

S&P 500 closing prices and demonstrative VG representations. Panel (a) shows the entire studied period of S&P 500 time series; panels (b–c) illustrate VGs built from contrasting windows, with crisis/pre-crisis segments (including the 1987 and 2008 crashes) producing visibly denser, more coherent connectivity

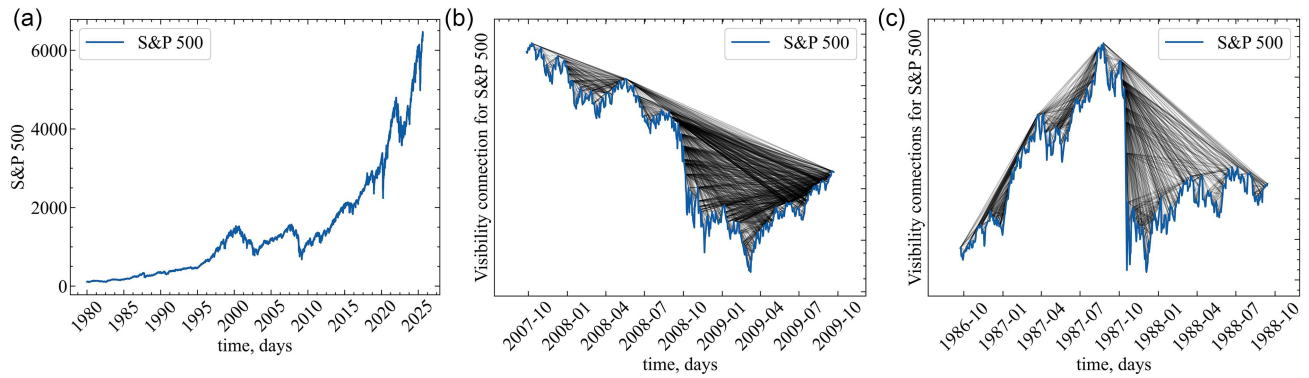
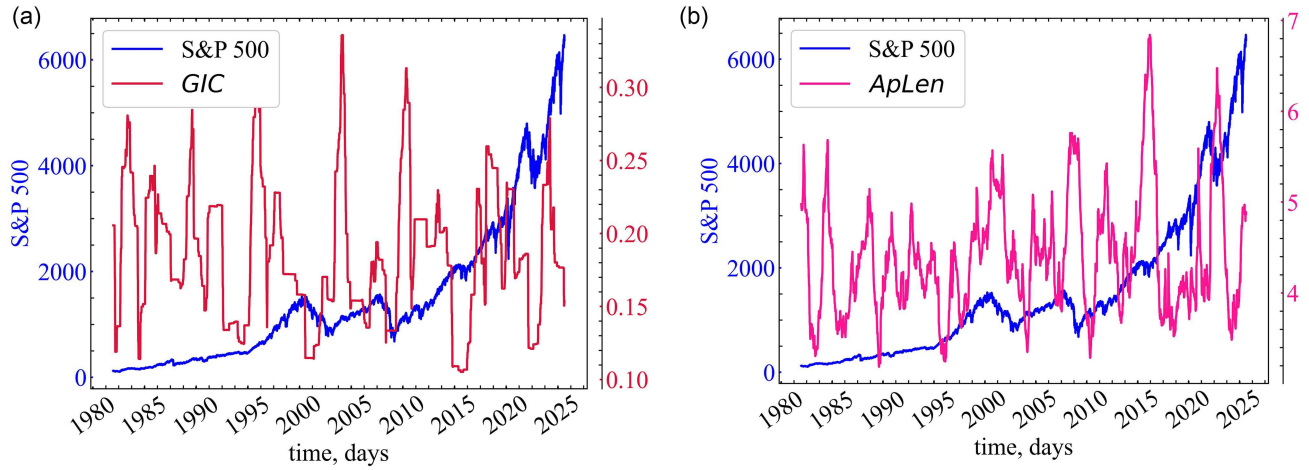


Figure 2

Network-complexity indicators computed in a 500-day sliding window. (a) Graph Index Complexity (*GIC*) versus the S&P 500; (b) Average Path Length (*ApLen*) versus the S&P 500. *GIC* peaks around turbulent regime shifts and declines during extended directional advances; *ApLen* provides a complementary regime signal, falling in choppy, mean-reverting phases and rising in smoother, trend-dominated periods



where $\mu_{t,w}$ and $\sigma_{t,w}$ are rolling mean and standard deviation of R_{t+h} calculated with the rolling window size w of 50 days. Standardization of the target variable regarding the sliding window approach makes it possible to balance the predicted classes (the "Up" and "Down" directions).

For classification, labels are derived from the sign of the forward-return target:

$$y_{t+h} = \begin{cases} +1\ (Up), & r_{t+h} > 0 \\ -1\ (Down), & r_{t+h} < 0 \end{cases} \tag{39}$$

Figure 3 represents the dynamics of 14-day forward-looking returns and the histogram of generated classes, which are going to be predicted with ML models.

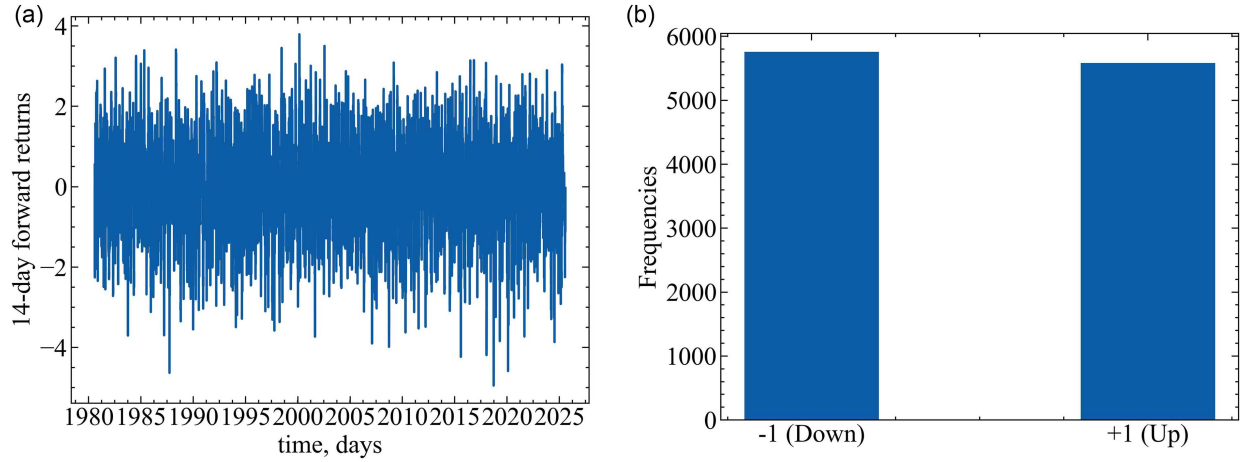
Each network feature was standardized before model input using z-score normalization (zero mean, unit variance). Feature standardization is critical because many learning algorithms are scale-dependent: distance-based methods, kernel machines, and regularized classifiers implicitly assume commensurate feature magnitudes. Standardization improves optimization conditioning, makes penalty terms comparable across coefficients, and facilitates fair feature selection.

We intentionally use price-only VG features to isolate the predictive content of VG topology. Incorporating exogenous covariates (e.g., VIX, rates, macro/sentiment), with careful timestamp and release-lag handling, is a natural extension—especially for regime conditioning and return-magnitude modeling.

For short-horizon forecasting, we explicitly encode recent serial dependence by constructing lagged features from every predictor. For a 14-day-ahead task ($h = 14$), the feature set at time t contains only information available up to t (lags $1, \ldots, 7$), while the target is y_{t+14}; this alignment prevents look-ahead bias and is maintained under walk-forward cross-validation. After generating lags, the resulting features are standardized with a rolling window of 50 days. The one-week cutoff reflects the dominant autocorrelation scale in our data; nevertheless, experiments with longer lag horizons are also possible.

Figure 3

Target construction for learning tasks. (a) Fourteen-day forward standardized returns r_{t+h}; (b) histogram of derived class labels. Forward returns R_{t+h} are standardized with a rolling window ($w = 50$) and binarized



3.4. Feature selection

Predictions are made regarding the whole set of features and selected with a feature selection approach. Given the extensive set of candidate graph features extracted (dozens per window), we apply a feature selection step to focus on the most predictive ones and reduce dimensionality. We employ an MI-based filter to rank the features by their relevance to the forecasting targets. MI measures the dependency between each feature and the target variable, capturing nonlinear relationships. We compute separate MI rankings for the regression task (using the 14-day-ahead standardized return or price change as the target) and for the classification task (using the 14-day-ahead direction label as the target class). This yields two ordered lists of features by importance.

Using these rankings, we retain only the top features that contribute the most information. In practice, we apply a percentile cutoff: the top 10% of features (those whose MI scores fall in the highest decile) are selected for model training. All other lower-ranked features are dropped. This percentile threshold is chosen to be fairly stringent, drastically reducing the feature set and thus mitigating overfitting, while still keeping the most informative indicators. By selecting features separately for regression and classification, we allow that the most useful network features for predicting the magnitude of returns may differ from those for predicting just the direction. In both cases, MI provides an information-theoretic criterion for feature relevance, ensuring that the retained VG features have a strong connection to the market outcome of interest. After this selection, we typically end up with a manageable subset of features (on the order of tens out of hundreds initially computed) for each predictive task.

3.5. Model calibration and cross-validation

3.5.1. Hyperparameters tuning and cross-validation approach

We tune the hyperparameters of each model using a rigorous time series–aware cross-validation procedure. RandomizedSearchCV is used to sample hyperparameter combinations for each model over a defined search space (for example, number of trees and maximum depth for RF, or length scale for the Gaussian process kernel). Randomized search is preferred over grid search to efficiently explore a wide range of hyperparameters without an exponential explosion of combinations.

Crucially, we employ a time series cross-validation strategy to evaluate each hyperparameter combination, rather than random k-fold CV, in order to preserve the chronological order of data and avoid look-ahead bias. We use an expanding TimeSeriesSplit with five splits, meaning the data is partitioned into five sequential train–test folds. In each fold, earlier windows are used for training and later windows for testing. To further prevent any leakage due to overlapping windows (since adjacent sliding windows share a lot of data), we impose a gap of 150 days (equivalent to one full window length) between the end of each training set and the start of the test set in the cross-validation split. In practice, if one fold uses data up to time T for training, the test fold begins at $T + 150$ days, skipping the intermediate period. This gap ensures that no portion of a sliding window used for training is even partially present in the test data for that fold, thus eliminating overlap between train and test VGs. Adopting this gap time CV technique is recommended to avoid over-optimistic performance when using sliding windows. We perform five such splits moving forward in time, so that we train on progressively later portions of the series and validate on the subsequent segment, covering the entire dataset in sequence. Model hyperparameters are selected based on the average cross-validation score across these folds.

During the randomized search, each candidate model is evaluated on each fold using the appropriate scoring metric. The time gap between training and testing ensures evaluation on truly unseen future data, emulating real forecasting conditions. The result is an optimized hyperparameter set for every algorithm, chosen to maximize predictive performance without temporal leakage.

3.5.2. Scoring metrics

Let $\{(x_t, r_{t+h})\}_{t=t_0}^{T-1}$ denote the time-ordered sample, where x_t are features (VG-based descriptors computed in a sliding window) and r_{t+h} is the h-day forward return (as we have mentioned, $h = 14$, in our case). For directional forecasting, define labels $y_{t+h} = \text{sign}(r_{t+h}) \in \{+1, -1\}$.

For a regression problem, given a parametric predictor $f_\theta(x_t) \approx r_{t+h}$, the in-sample loss is the median absolute error (MedAE):

$$\text{MedAE}(\theta) = \text{median}\left(|r_{1+h} - f_\theta(x_1)|, \ldots, |r_{t+h} - f_\theta(x_t)|\right) \tag{40}$$

For a hyperparameter vector $h \in \mathscr{H}$ and model $\theta(h)$ trained on the k-th training block, the selection criterion is

$$\hat{h}_{\text{reg}} = \arg\min_{h \in \mathscr{H}} \frac{1}{K} \sum_{k=1}^{K} \text{MedAE}\left(\theta_k(h); \mathscr{V}_k\right) \tag{41}$$

where $\mathscr{V}_k$ is the k-th validation block.

After calibration, the final model is refit on the full training range with $\hat{h}_{\text{reg}}$.

For a classification task (direction movement), the area under the receiver operating characteristic curve (ROC-AUC) is maximized.

3.5.3. Performance evaluation

Define residuals $e_i = y_i - \hat{y}_i$ and $\bar{y} = n^{-1}\sum_{i=1}^{n} y_i$. The empirical performance metrics used for evaluation are:

1) Mean absolute error: $\text{MAE} = n^{-1}\sum_{i=1}^{n} |e_i|$.
2) Mean squared error: $\text{MSE} = n^{-1}\sum_{i=1}^{n} e_i^2$.
3) Root mean squared error: $\text{RMSE} = \sqrt{MSE}$.
4) Coefficient of determination (R^2): $1 - \left[\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 / \sum_{i=1}^{n}(y_i - \bar{y})^2\right]$.

For binary classification, the following performance metrics can be defined:

1) Accuracy: $\text{ACC} = TP + TN/(TP + TN + FP + FN)$.
2) Precision (positive predicted value): $\text{Prec} = TP/(TP + FP)$.
3) Recall (sensitivity/true positive rate): $\text{Rec} = TP/(TP + FN)$.
4) F1-score (harmonic mean of precision and recall): $F1 = (2\ Prec \cdot Rec)/(Prec + Rec)$.

In multi-class classification, metrics such as precision, recall, and F1-score are first computed per class. To summarize performance with one number, we need to average those per-class scores. Here, macro and weighted averages are used.

4. Experimental Design and Evaluation

In general, we performed both classification and regression experiments using VG features derived from the S&P 500 time series. In the *classification* formulation, each observation is labeled Up/Down by the sign of the 14-day-ahead target $y_{t,14}$, and models were trained to predict direction. In the *regression* formulation, models directly forecast the 14-day-ahead standardized forward return $r_{t,14}$. All features were pre-selected by MI, a model-free criterion that retains only those variables most informative about the target.

Performance metrics were chosen according to the task. For classification, we report accuracy and ROC-AUC curve. For regression, we use MAE, MSE, RMSE, and the R^2 score. These metrics capture average forecast error and explained variance.

4.1. Classification performance results

Figure 4 illustrates ROC-AUC curves derived for six ML models trained for market direction forecasting.

Figure 4 shows that all ROC traces lie clearly above the diagonal chance line, confirming consistent (though still moderate) discriminatory power across decision thresholds. For natural VG, five methods form an almost indistinguishable top group—logistic, ridge, SGD, RF, and Gaussian process—each achieving ROC-AUC = 0.72, while k-NN remains the weakest performer (AUC = 0.67). Under VAQ-VG, the overall ranking is preserved, but the curves shift slightly downward: logistic, ridge, and Gaussian process reach AUC = 0.71, RF and SGD reach AUC = 0.70, and k-NN declines to AUC = 0.65. Hence, the VAQ-VG transformation induces only small, model-dependent changes (≈ 0.01–0.02 AUC) while maintaining the same performance ordering, indicating that the directional signal extracted from VG features is robust to volatility-adaptive quantile normalization rather than being specific to a single graph variant. The tight overlap among the leading five models further suggests that the predictive information in the selected VG descriptors is captured similarly by both linear margins and the tested nonlinear learners, whereas the distance-based k-NN remains comparatively less competitive.

Tables 1, 2, 3, 4, 5, and 6 present classification performance metrics for each studied model.

Across Tables 1–6, the thresholded classification metrics confirm the same message as the ROC curves: performance is consistently above chance but still moderate in absolute magnitude. Under natural VG, five models are tightly clustered: SGD, RF, logistic, ridge, and Gaussian process all achieve accuracy ≈ 0.659–0.663, with very similar macro/weighted summaries. In contrast, k-NN is clearly weaker, indicating that the distance-based decision rule is less aligned with the geometry induced by the selected VG features in the current setting.

A key structural pattern across all models is the recall asymmetry between classes. With natural VG, models recover Down days with high recall (~0.76–0.80) while Up recall is systematically lower (~0.52–0.56), even when Up precision remains comparatively high. This implies the decision boundary is effectively safer in calling Down moves, consistent with a modest signal-to-noise ratio.

Figure 4
ROC-AUC curves for six classifiers trained on (a) natural VG features and (b) VAQ-VG features and evaluated on the test set, indicating modest but consistent discrimination above chance. The diagonal dashed line denotes the chance level (*AUC* = 0.5)

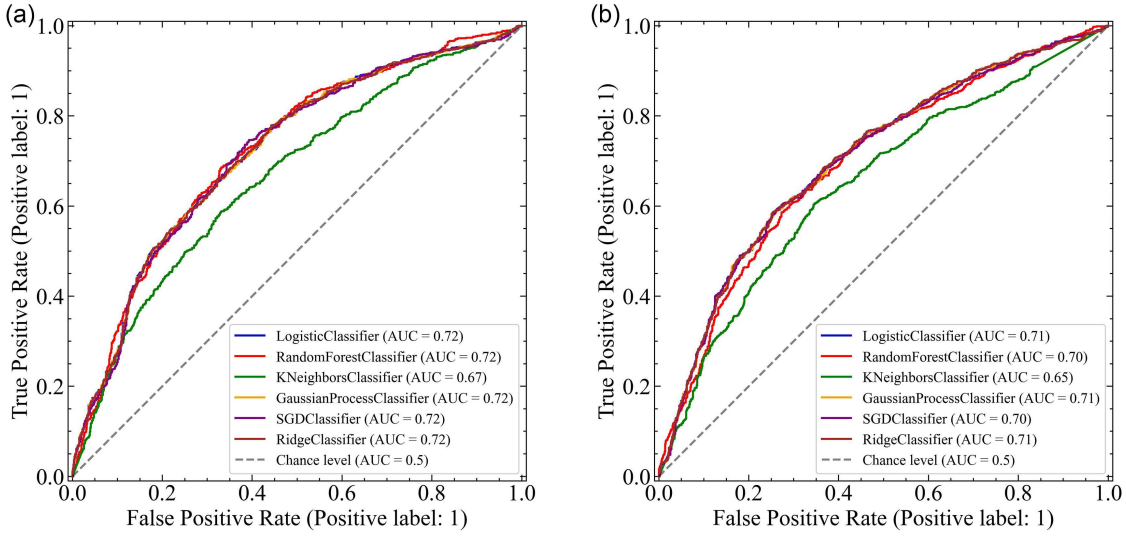


Table 1
Performance metrics for stochastic gradient descent classifier

	Natural VG			VAQ-VG		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Down	0.624	0.797	0.700	0.615	0.810	0.699
Up	0.720	0.521	0.605	0.723	0.494	0.587
Accuracy	0.659	0.659	0.659	0.652	0.652	0.652
Macro average	0.672	0.659	0.652	0.669	0.652	0.643
Weighted average	0.672	0.659	0.652	0.669	0.652	0.643

Table 2
Performance metrics for random forest classifier

	Natural VG			VAQ-VG		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Down	0.634	0.764	0.693	0.608	0.784	0.685
Up	0.705	0.560	0.624	0.697	0.495	0.579
Accuracy	0.662	0.662	0.662	0.639	0.639	0.639
Macro average	0.669	0.662	0.659	0.652	0.640	0.632
Weighted average	0.669	0.662	0.659	0.652	0.639	0.632

Table 3
Performance metrics for logistic regression classifier

	Natural VG			VAQ-VG		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Down	0.633	0.771	0.695	0.618	0.789	0.693
Up	0.709	0.555	0.622	0.710	0.515	0.597
Accuracy	0.663	0.663	0.663	0.652	0.652	0.652
Macro average	0.671	0.663	0.659	0.664	0.652	0.645
Weighted average	0.671	0.663	0.659	0.664	0.652	0.645

Table 4
Performance metrics for ridge regression classifier

	Natural VG			VAQ-VG		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Down	0.630	0.785	0.699	0.616	0.798	0.695
Up	0.716	0.541	0.616	0.715	0.505	0.592
Accuracy	0.663	0.663	0.663	0.651	0.651	0.651
Macro average	0.673	0.663	0.658	0.666	0.651	0.644
Weighted average	0.673	0.663	0.658	0.666	0.651	0.643

Table 5
Performance metrics for Gaussian process classifier

	Natural VG			VAQ-VG		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Down	0.632	0.779	0.698	0.618	0.789	0.693
Up	0.714	0.548	0.620	0.709	0.514	0.596
Accuracy	0.663	0.663	0.663	0.651	0.651	0.651
Macro average	0.673	0.664	0.659	0.664	0.651	0.645
Weighted average	0.673	0.663	0.659	0.664	0.651	0.644

Table 6
Performance metrics for *k*-nearest neighbors' classifier

	Natural VG			VAQ-VG		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Down	0.602	0.712	0.652	0.581	0.774	0.663
Up	0.649	0.530	0.583	0.663	0.443	0.531
Accuracy	0.621	0.621	0.621	0.608	0.608	0.608
Macro average	0.625	0.621	0.618	0.622	0.608	0.597
Weighted average	0.625	0.621	0.618	0.622	0.608	0.597

Introducing VAQ-VG preserves the overall ranking, but it shifts the operating-point metrics slightly downward for most methods. The leading linear/kernel baselines remain competitive, while SGD stays close. The largest drop appears for RF and for *k*-NN, suggesting that the VAQ-VG normalization affects how these models translate scores into hard labels at the default threshold. Importantly, VAQ-VG tends to maintain or even increase Down recall, but this often comes at the expense of Up recall, widening the sensitivity imbalance.

These tables clarify that differences between natural and VAQ-VG are largely operating-point effects: the ROC results indicate stable ranking quality, while per-threshold metrics reveal that small calibration changes can materially move accuracy/F1 even when AUC changes little. Decision-threshold optimization is the primary lever for addressing the persistent Up-recall deficit.

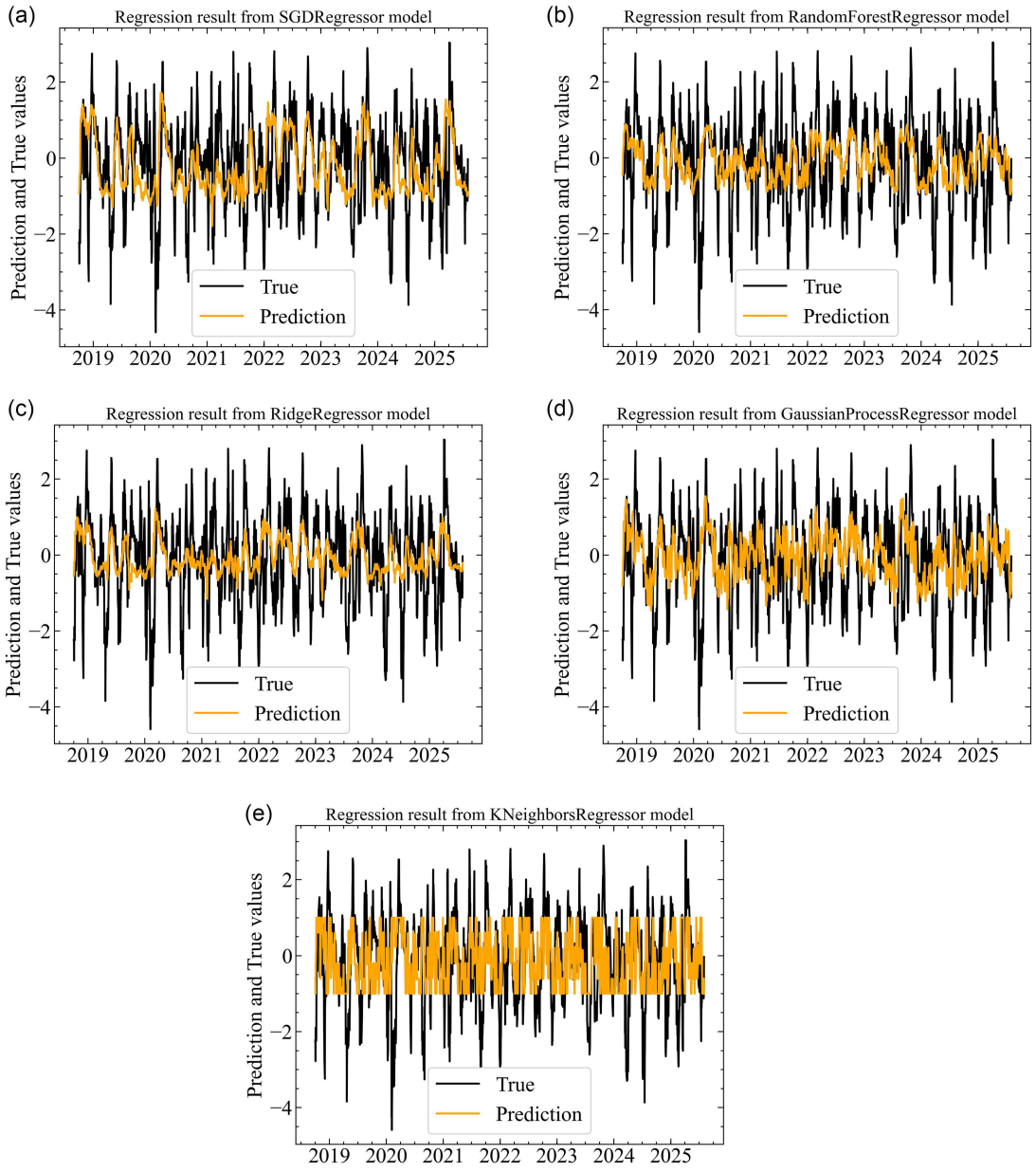
4.2. Regression performance results

Figure 5 presents forecasts of 14-day-ahead standardized forward returns obtained from five regression models, which were trained on natural VG features.

Figure 5 visualizes predicted vs. realized 14-day-ahead standardized forward returns from five regression models trained on natural VG features. Across all panels, the realized series (black) is dominated by intermittent, high-amplitude spikes, whereas the point forecasts (orange) behave like low-pass, shrinkage estimates: they remain centered close to zero, fluctuate within a markedly narrower band than the truth, and only express slow/medium-frequency undulations rather than abrupt jumps. This qualitative pattern is consistent with short-horizon equity returns being close to a noisy, heavy-tailed process: under such conditions, the

Figure 5

Regression results regarding natural VG features: predicted versus true 14-day-ahead standardized returns. Panels (a–e) correspond to stochastic gradient descent, random forest, ridge regressor, Gaussian process, and *k*-nearest neighbors' models



risk-minimizing point forecast (especially with regularization and ensemble averaging) is naturally pulled toward the conditional center, so magnitude predictability remains limited even when directional discrimination is non-trivial.

Substantively, Figure 5 reinforces that the current VG feature representation is better at encoding gradual regime/structure information than at forecasting return magnitudes: the models extract a weak, smoother component but cannot track the realized tail dynamics that dominate short-horizon regression error.

Figure 6 presents forecasts of 14-day-ahead standardized forward returns obtained from five regression models, which were trained on VAQ-VG features.

Figure 6 shows that the same regression limitation persists under VAQ-VG: point forecasts remain strongly center-biased and display substantially lower variance than the realized 14-day returns. The volatility-adaptive, rolling-quantile normalization used in VAQ-VG acts as an additional stabilizer of the input geometry, so the learned predictors continue to emphasize the persistent/low-frequency component while treating abrupt jumps as largely unpredictable. In practical terms, VAQ-VG does not convert the task into an amplitude-forecasting problem; instead, it maintains a representation that is informative mainly for relative movements and sign.

Across models, the qualitative ordering remains unchanged: SGD and ridge deliver smooth trajectories; RF shows strong shrinkage; Gaussian process remains locally adaptive but compresses extremes; *k*-NN continues to show blocky output. Figure 6 reinforces that with current feature engineering and leak-free validation, VG-based descriptors provide modest directional information, while return magnitude is better approached

Figure 6

Regression results regarding VAQ-VG features: predicted versus true 14-day-ahead standardized returns. Panels (a–e) correspond to stochastic gradient descent, random forest, ridge regressor, Gaussian process, and *k*-nearest neighbors' models

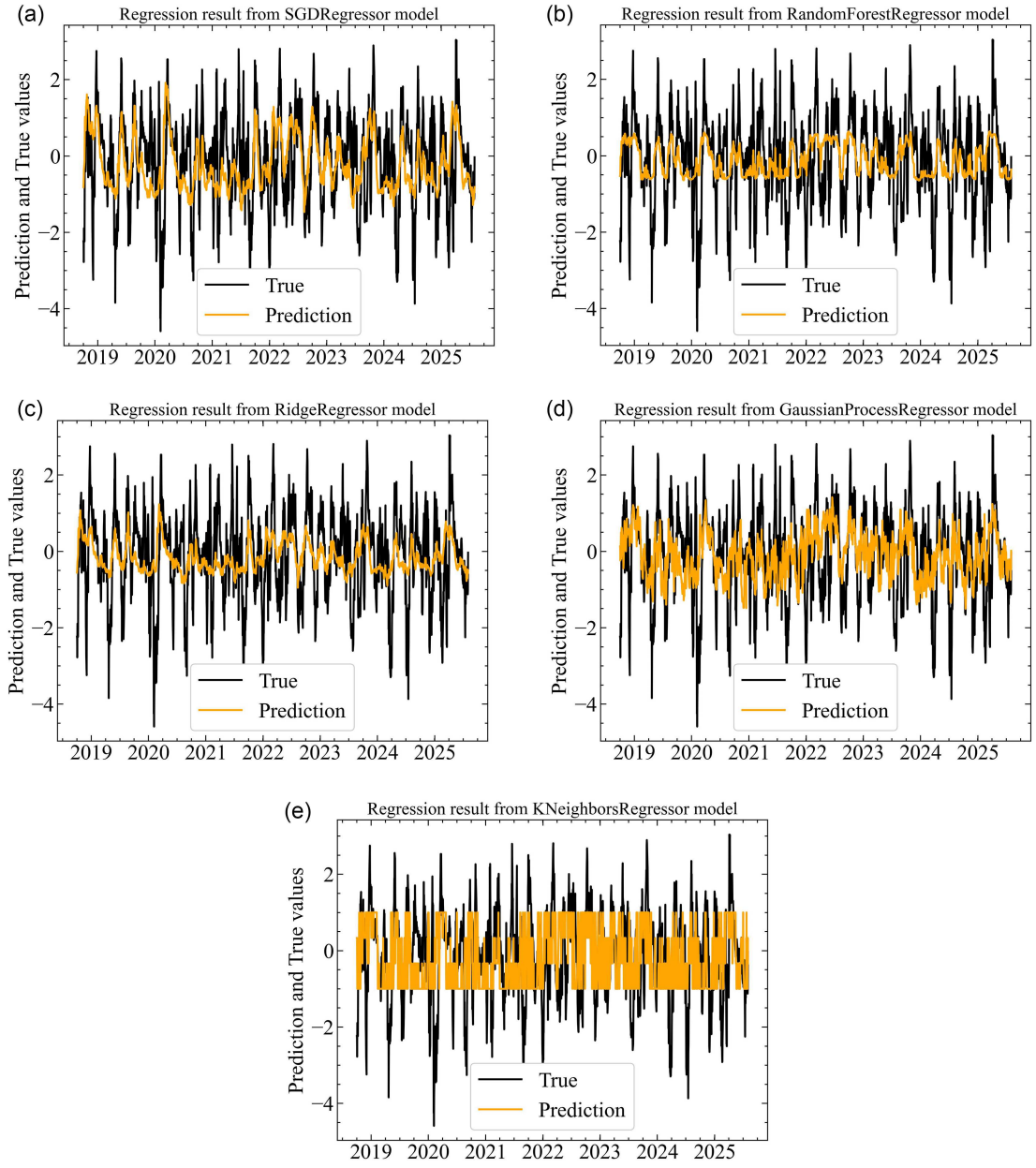


Table 7
Regression performance on 14-day-ahead standardized returns using VG features—natural VG and VAQ-VG

	Natural VG				VAQ-VG			
	R^2	MAE	MSE	RMSE	R^2	MAE	MSE	RMSE
Stochastic gradient descent	0.150	0.862	1.142	1.069	0.171	0.855	1.115	1.056
Random forest	0.160	0.833	1.129	1.062	0.187	0.828	1.093	1.046
Ridge	0.177	0.824	1.106	1.051	0.189	0.836	1.090	1.044
Gaussian process	0.009	0.912	1.332	1.154	0.048	0.896	1.280	1.131
k-nearest neighbors	–0.140	0.969	1.533	1.238	–0.152	0.994	1.548	1.244

via distributional targets or explicit regime augmentation rather than single-point forecasts.

Table 7 demonstrates the quantitative regression performance of ML algorithms fitted on natural VG and VAQ-VG features.

Table 7 quantifies the regression task (14-day-ahead standardized returns for the S&P 500) and makes two points clear. First, magnitude forecasting remains intrinsically hard in this setup: even the best specifications explain only a small fraction of variance, while error levels stay relatively high and close across the top models. This matches the visual evidence in the regression plots, where predictions remain strongly centered and under-react to tail moves.

Overall, Table 7 supports a conservative conclusion: with the current VG feature set and leak-free validation, algorithmic sophistication offers limited marginal returns beyond strong baselines (ridge/RF/SGD), while improvements are more likely to come from target/feature redesign—for example, forecasting volatility-scaled returns or distributional targets (quantiles/intervals), and adding leak-free regime/scale covariates—rather than expecting materially higher point-forecast accuracy from swapping regressors alone.

5. Limitations and Future Directions

While our results demonstrate the potential of VG features in forecasting, several limitations of this study must be acknowledged.

Future studies should broaden the feature space beyond VG metrics. Other complexity science features—such as entropy-based measures [30, 46, 47], fractal dimensions (to quantify self-similarity or long memory in price series) [31], measures from recurrence quantification analysis [32], non-extensive Tsallis statistics [48], and Lyapunov exponents (for chaos detection) [49]—could be integrated to capture facets of market behavior that network topology alone might miss. Combining VG features with these complementary complexity indicators may provide a richer characterization of the time series, potentially boosting predictive power.

We employed MI as the sole feature selection criterion. While effective for detecting nonlinear associations, MI evaluates variables individually and does not account for feature interactions or redundancy [50]. This filter-based approach may yield a suboptimal predictor subset. Future work could employ wrapper or embedded methods, dimensionality reduction (Principal Component Analysis (PCA), autoencoders), or representation learning to discover informative composite features while avoiding redundancy and overfitting.

A logical next step is to explore DL and hybrid approaches [51–53]. Recurrent models (LSTM, GRU) and transformer architectures could capture sequential and long-range dependencies. GNNs offer a way to learn directly from VG structure rather than hand-crafted features, and recent studies confirm their effectiveness for stock trend prediction. Hybrid systems integrating ML with domain-specific reasoning—such as neuro-fuzzy models [54–58]—have also demonstrated improved accuracy in financial network applications [11, 59].

This study addressed a single forecasting horizon (14-day ahead) and a single target (the S&P 500). Real-world forecasting often demands multi-horizon predictions; exploring different horizons and targets is left for future work.

In summary, by addressing these limitations—incorporating a wider set of complexity features, adopting more powerful learning architectures, refining feature selection, and broadening the prediction scope—future work can build a more comprehensive and effective framework for stock market movement forecasting.

6. Conclusions

This study examined whether complexity-inspired measures extracted from VG representations of the S&P 500 time series can aid short-horizon forecasting. We extended the baseline natural VG with VAQ-VG that (i) quantile-normalizes each window to reduce level/trend effects and (ii) introduces a volatility-scaled visibility slack to improve robustness in heteroskedastic regimes. We also added two geometric/mesoscopic indicators—the VAE and the SMD—to a broad spectral/topological feature set computed in sliding windows. Features were standardized, lagged, filtered via MI, and evaluated with walk-forward cross-validation using a large temporal purge gap to eliminate sliding window leakage.

Empirically, directional classification achieved modest but consistent skill above chance across six learning algorithms. With natural VG features, the strongest models reached ROC-AUC $\approx$ 0.70–0.72 and accuracy $\approx$ 0.65–0.66, confirming that network structure carries a usable directional signal. Introducing VAQ-VG largely preserved the ranking of methods and the overall performance level, with small, model-dependent shifts. Class-wise results were consistent across both representations: recall was higher for the Down class, while precision tended to be higher for the Up class, suggesting conservative Up predictions and a tendency to miss some Up periods—an imbalance that can be reduced via threshold calibration or cost-sensitive training.

By contrast, point regression of 14-day returns remained difficult under both feature sets. Predicted series were strongly shrunk toward zero and exhibited substantially lower variance than realized returns, and the best models achieved only limited variance explanation (best R^2 below 0.20). This behavior is consistent with a near-martingale target, heavy-tailed shocks, and the smoothing induced by regularization and ensembling. Taken together, the regression results and classification

performance indicate that the current VG feature set encodes more information about direction than magnitude.

Practically, the results suggest that even in a highly efficient market, there is a detectable (though small) signal in complex structural features. Our findings align with recent studies reporting performance boosts when incorporating network-derived information, reinforcing that market complexity can be better tamed with hybrid approaches.

The consistency of results across multiple ML models suggests that the informational content of VG features is robustly recognized regardless of the specific algorithm. This underlines that feature engineering grounded in complexity theory can matter more than algorithmic sophistication when seeking an edge in prediction.

In conclusion, our study provides proof of concept that integrating complex network features with ML adds value in forecasting stock market movements. This interdisciplinary strategy combines the interpretability and theoretical richness of network/complexity measures with the adaptability of ML algorithms, offering a meaningful step forward in understanding and predicting stock market dynamics.

Acknowledgment

The authors are grateful to the Armed Forces of Ukraine for defending our country and enabling us to conduct this research.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in GitHub at https://github.com/Butman2099/Complex-systems-book and https://github.com/Butman2099/Complex-Network-Paper-for-Artificial-Intelligence-and-Applications-AIA-journal.

Author Contribution Statement

Andrii Bielinskyi: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Vladimir Soloviev:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration. **Andriy Matviychuk:** Conceptualization, Methodology, Validation, Investigation, Writing – original draft, Writing – review & editing. **Vitalii Bezkorovainyi:** Methodology, Software, Formal analysis, Investigation, Data curation, Writing – review & editing, Visualization.

References

[1] Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417. https://doi.org/10.2307/2325486

[2] Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, *197*, 116659. https://doi.org/10.1016/j.eswa.2022.116659

[3] Patel, M., Jariwala, K., & Chattopadhyay, C. (2024). A systematic review on graph neural network-based methods for stock market forecasting. *ACM Computing Surveys*, *57*(2), 1–38. https://doi.org/10.1145/3696411

[4] Akşehir, Z. D., & Kılıç, E. (2024). Analyzing the critical steps in deep learning-based stock forecasting: A literature review. *PeerJ Computer Science*, *10*, 1–45. https://doi.org/10.7717/peerj-cs.2312

[5] Sonkavde, G., Dharrao, D. S., Bongale, A. M., Deokate, S. T., Doreswamy, D., & Bhat, S. K. (2023). Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. *International Journal of Financial Studies*, *11*(3), 94. https://doi.org/10.3390/ijfs11030094

[6] Mantegna, R. N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems*, *11*(1), 193–197. https://doi.org/10.1007/s100510050929

[7] Elliott, M., & Golub, B. (2022). Networks and economic fragility. *Annual Review of Economics*, *14*(1), 665–696. https://doi.org/10.1146/annurev-economics-051520-021647

[8] Newman, M. E. (2010). *Networks: An introduction*. UK: Oxford University Press.

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

[10] Xing, J., Li, B., & Yang, Y. (2023). Community detection and clustering characteristics analysis of the stock market. *Managerial and Decision Economics*, *44*(7), 3893–3906. https://doi.org/10.1002/mde.3929

[11] Li, B., & Zhang, X. (2024). Systemic risk and financial networks. *The Quarterly Review of Economics and Finance*, *94*, 25–36. https://doi.org/10.1016/j.qref.2023.12.012

[12] Wang, G., Zhao, H., Shu, L., & Song, L. (2025). Understanding risk spillover in multi-layer financial networks: The role of bank-firm connections. *International Review of Financial Analysis*, *105*, 104404. https://doi.org/10.1016/j.irfa.2025.104404

[13] Marti, G., Nielsen, F., Bińkowski, M., & Donnat, P. (2021). A review of two decades of correlations, hierarchies, networks and clustering in financial markets. In *Progress in Information Geometry: Theory and Applications* (pp. 245–274). https://doi.org/10.1007/978-3-030-65459-7_10

[14] Park, J., Cho, C. H., & Lee, J. W. (2022). A perspective on complex networks in the stock market. *Frontiers in Physics*, *10*, 1–4. https://doi.org/10.3389/fphy.2022.1097489

[15] Shi, H. L., & Chen, H. (2022). Network structures for asset return co-movement: Evidence from the Chinese stock market. *Frontiers in Physics*, *10*, 1–14. https://doi.org/10.3389/fphy.2022.593493

[16] Qiu, Y., Sun, X., Xiong, X., & Si, S. (2025). Systemic financial risk analysis of the US based on the complex network. *Journal of Management Science and Engineering*, *10*(3), 414–433. https://doi.org/10.1016/j.jmse.2025.05.002

[17] Cao, J., Wen, F., Stanley, H. E., & Wang, X. (2021). Multilayer financial networks and systemic importance: Evidence from

China. *International Review of Financial Analysis*, *78*, 101882. https://doi.org/10.1016/j.irfa.2021.101882

[18] Zeng, Z., & Chen, Y. (2025). Identifying and forecasting recurrently emerging stock trend structures via rising visibility graphs. *Forecasting*, *7*(2), 1–20. https://doi.org/10.3390/forecast7020026

[19] Yin, Y., Wang, X., Wang, W., Li, Q., & Shang, P. (2023). Multiscale cross-sample entropy based on visibility graph for quantifying time series irreversibility. *Communications in Nonlinear Science and Numerical Simulation*, *124*, 107308. https://doi.org/10.1016/j.cnsns.2023.107308

[20] Freitas Silva, V., Silva, M. E., Ribeiro, P., & Silva, F. (2025). Multilayer horizontal visibility graphs for multivariate time series analysis. *Data Mining and Knowledge Discovery*, *39*(3), 1–42. https://doi.org/10.1007/s10618-025-01089-4

[21] Lacasa, L., Luque, B., Luque, J., & Nuno, J. C. (2009). The visibility graph: A new method for estimating the Hurst exponent of fractional Brownian motion. *EPL (Europhysics Letters)*, *86*(3), 30001. https://doi.org/10.1209/0295-5075/86/30001

[22] Azizi, H., & Sulaimany, S. (2024). A review of visibility graph analysis. *IEEE Access*, *12*, 93517–93530. https://doi.org/10.1109/ACCESS.2024.3401485

[23] Hu, J., Chu, C., Zhu, P., & Yuan, M. (2023). Visibility graph-based segmentation of multivariate time series data and its application. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *33*(9), 093123. https://doi.org/10.1063/5.0152881

[24] Han, M., Fan, Q., & Ling, G. (2022). Multiscale online-horizontal-visibility-graph correlation analysis of financial market. *Physica A: Statistical Mechanics and its Applications*, *607*, 128195. https://doi.org/10.1016/j.physa.2022.128195

[25] Gao, M., & Ge, R. (2024). Mapping time series into signed networks via horizontal visibility graph. *Physica A: Statistical Mechanics and its Applications*, *633*, 129404. https://doi.org/10.1016/j.physa.2023.129404

[26] Zhu, J., & Wei, D. (2021). Analysis of stock market based on visibility graph and structure entropy. *Physica A: Statistical Mechanics and its Applications*, *576*, 126036. https://doi.org/10.1016/j.physa.2021.126036

[27] Yang, Y. H., Liu, Y. L., & Shao, Y. H. (2025). Visibility graph analysis of crude oil futures markets: Insights from the COVID-19 pandemic and Russia–Ukraine conflict. *Fluctuation and Noise Letters*, *24*(03), 2550021. https://doi.org/10.1142/S021947752550021X

[28] Fan, Y., Yang, Y., Wang, Z., & Gao, M. (2025). Instability of financial time series revealed by irreversibility analysis. *Entropy*, *27*(4), 402. https://doi.org/10.3390/e27040402

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

[30] Zunino, L., Olivares, F., Ribeiro, H. V., & Rosso, O. A. (2022). Permutation Jensen-Shannon distance: A versatile and fast symbolic tool for complex time-series analysis. *Physical Review E*, *105*(4), 045310. https://doi.org/10.1103/PhysRevE.105.045310

[31] Brouty, X., & Garcin, M. (2024). Fractal properties, information theory, and market efficiency. *Chaos, Solitons & Fractals*, *180*, 114543. https://doi.org/10.1016/j.chaos.2024.114543

[32] Marwan, N., & Kraemer, K. H. (2023). Trends in recurrence analysis of dynamical systems. *The European Physical Journal Special Topics*, *232*(1), 5–27. https://doi.org/10.1140/epjs/s11734-022-00739-8

[33] Zhou, Y., Xie, C., Wang, G. J., Zhu, Y., & Uddin, G. S. (2023). Analysing and forecasting co-movement between innovative and traditional financial assets based on complex network and machine learning. *Research in International Business and Finance*, *64*, 101846. https://doi.org/10.1016/j.ribaf.2022.101846

[34] Lei, Z., Zhang, C., Xu, Y., & Li, X. (2024). DR-GAT: Dynamic routing graph attention network for stock recommendation. *Information Sciences*, *654*, 119833. https://doi.org/10.1016/j.ins.2023.119833

[35] Qian, H., Zhou, H., Zhao, Q., Chen, H., Yao, H., Wang, J., . . . , & Zhou, J. (2024). MDGNN: Multi-relational dynamic graph neural network for comprehensive and dynamic stock investment prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(13), 14642–14650. https://doi.org/10.1609/aaai.v38i13.29381

[36] Pan, Y., Hu, W., Ge, X., & Lin, A. (2025). A novel forecasting method for time series based on vector visibility graphs. *Physica A: Statistical Mechanics and its Applications*, *677*, 130839. https://doi.org/10.1016/j.physa.2025.130839

[37] Chen, Z., Huang, Z., & Zhou, Y. (2023). Predicting stock trend using GNN. *Highlights in Science, Engineering and Technology*, *39*, 816–822. https://doi.org/10.54097/hset.v39i.6649

[38] Peng, B., & Gao, S. (2025). Vector visibility graph for rare event classification in complex system multivariate time series data. *Systems Science & Control Engineering*, *13*(1), 2546844. https://doi.org/10.1080/21642583.2025.2546844

[39] Jun, W. U., Barahona, M., Yue-Jin, T., & Hong-Zhong, D. (2010). Natural connectivity of complex networks. *Chinese Physics Letters*, *27*(7), 078902. https://doi.org/10.1088/0256-307x/27/7/078902

[40] Kim, J., & Wilhelm, T. (2008). What is a complex graph? *Physica A: Statistical Mechanics and its Applications*, *387*(11), 2637–2652. https://doi.org/10.1016/j.physa.2008.01.015

[41] Latora, V., & Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical Review Letters*, *87*(19), 198701. https://doi.org/10.1103/PhysRevLett.87.198701

[42] Stephenson, K., & Zelen, M. (1989). Rethinking centrality: Methods and examples. *Social Networks*, *11*(1), 1–37. https://doi.org/10.1016/0378-8733(89)90016-6

[43] Barrat, A., & Weigt, M. (2000). On the properties of small-world network models. *The European Physical Journal B-Condensed Matter and Complex Systems*, *13*(3), 547–560. https://doi.org/10.1007/s100510050067

[44] So, M. K., Mak, A. S., Chan, J. N., & Chu, A. M. (2023). Standardized local assortativity in networks and systemic risk in financial markets. *PLOS One*, *18*(10), e0292327. https://doi.org/10.1371/journal.pone.0292327

[45] Newman, M. E. (2002). Assortative mixing in networks. *Physical Review Letters*, *89*(20), 208701. https://doi.org/10.1103/PhysRevLett.89.208701

[46] Papla, D., & Siedlecki, R. (2024). Entropy as a tool for the analysis of stock market efficiency during periods of crisis. *Entropy*, *26*(12), 1079. https://doi.org/10.3390/e26121079

[47] Bielinskyi, A. O., Soloviev, V. N., Matviychuk, A. V., & Kmytiuk, T. L. (2025). Capturing bitcoin market dynamics: Assessing advanced permutation entropy metrics as early-warning indicators. *CEUR Workshop Proceedings*, *3988*, 83–96.

[48] Bielinskyi, A. O., Matviychuk, A. V., Serdyuk, O. A., Semerikov, S. O., Solovieva, V. V., & Soloviev, V. N. (2021). Correlational and non-extensive nature of carbon dioxide pricing market. In *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*, 183–199. https://doi.org/10.1007/978-3-031-14841-5_12

[49] Gezer, A. (2025). Chaos in bitcoin cryptocurrency metrics: Analysis and forecasts. *Arabian Journal for Science and Engineering*, *50*(8), 5869–5884. https://doi.org/10.1007/s13369-024-09357-z

[50] Hu, L., Gao, L., Li, Y., Zhang, P., & Gao, W. (2022). Feature-specific mutual information variation for multi-label feature selection. *Information Sciences*, *593*, 449–471. https://doi.org/10.1016/j.ins.2022.02.024

[51] Kmytiuk, T., & Majore, G. (2021). Time series forecasting of agricultural product prices using Elman and Jordan recurrent neural networks. *Neuro-Fuzzy Modeling Techniques in Economics*, *10*, 67–85. http://doi.org/10.33111/nfmte.2021.067

[52] Omole, O., & Enke, D. (2024). Deep learning for Bitcoin price direction prediction: Models and trading strategies empirically compared. *Financial Innovation*, *10*(1), 117. https://doi.org/10.1186/s40854-024-00643-1

[53] Derbentsev, V., Bezkorovainyi, V., Silchenko, M., Hrabariev, A., & Pomazun, O. (2021). Deep learning approach for short-term forecasting trend movement of stock indices. In *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology*, 607–612. https://doi.org/10.1109/PICST54195.2021.9772235

[54] Bielinskyi, A., Soloviev, V., Solovieva, V., & Velykoivanenko, H. (2022). Fuzzy time series forecasting using semantic artificial intelligence tools. *Neuro-Fuzzy Modeling Techniques in Economics*, *11*, 157–198. https://doi.org/10.33111/nfmte.2022.157

[55] Orozco-Castañeda, J. M., Alzate-Vargas, S., & Bedoya-Valencia, D. (2024). Evaluating volatility using an ANFIS model for financial time series prediction. *Risks*, *12*(10), 156. https://doi.org/10.3390/risks12100156

[56] Kozlovskyi, S., Syniehub, P., Kozlovskyi, A., & Lavrov, R. (2022). Intellectual capital management of the business community based on the neuro-fuzzy hybrid system. *Neuro-Fuzzy Modeling Techniques in Economics*, *11*, 25–47. http://doi.org/10.33111/nfmte.2022.025

[57] Kaminskyi, A., & Nehrey, M. (2023). Fuzzy clustering approach to portfolio management considering ESG criteria: Empirical evidence from the investment strategies of the EURO STOXX Index. *Neuro-Fuzzy Modeling Techniques in Economics*, *12*, 40–66. http://doi.org/10.33111/nfmte.2023.040

[58] Kabachii, V., Maslii, R., Kozlovskyi, S., & Dronchack, O. (2023). Identifying moments of decision making on trade in financial time series using fuzzy cluster analysis. *Neuro-Fuzzy Modeling Techniques in Economics*, *12*, 175–205. http://doi.org/10.33111/nfmte.2023.175

[59] Liu, Z., Wang, L., Huang, C., & Yang, B. (2024). Assessing interconnectedness and systemic importance of Chinese financial institutions. *Iscience*, *27*(8), 110474. https://doi.org/10.1016/j.isci.2024.110474

How to Cite: Bielinskyi, A., Soloviev, V., Matviychuk, A., & Bezkorovainyi, V. (2026). Stock Market Movement Forecasting Using Machine Learning and Complex Network Measures. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62027537