

## Article

# Forecasting Artificial Intelligence News Sentiment Index: Traditional vs. Image-Based Deep-Learning Models

Gianina-Maria Petraşcu <sup>1</sup>, Ioana Bîrlan <sup>1,\*</sup> , Cristina-Rodica Boboc <sup>2</sup>  and Adriana AnaMaria Davidescu <sup>2</sup> <sup>1</sup> Doctoral School of Economic Cybernetics and Statistics, Bucharest University of Economic Studies, 010374 Bucharest, Romania; gianina.petrascu@csie.ase.ro<sup>2</sup> Department of Statistics and Econometrics, Bucharest University of Economic Studies, 010374 Bucharest, Romania; cristina.boboc@csie.ase.ro (C.-R.B.); adriana.alexandru@csie.ase.ro (A.A.D.)

\* Correspondence: ioana.birlan@csie.ase.ro

## Abstract

Rapid AI adoption has intensified public and media attention toward AI-related developments, making news sentiment an increasingly important indicator of expectations and perceptions. This study constructs a global daily AI News Sentiment Index using data from the GDELT Global Knowledge Graph and examines the forecasting properties of the index using statistical, sequential deep learning and image-based forecasting methods. The period of observation of the dataset spans from January 2016 to December 2025, and the sample consists of 3635 daily observations. A sentiment index is constructed from the positive and negative sentiments and is analyzed together with its constituent series. For the empirical framework, ARIMA, ARIMA–GARCH, ETS, LightGBM, LSTM, CNN, TCN, GAN, and convolutional models based on Gramian Angular Summation Field (GASF) and Gramian Angular Difference Field (GADF) transformations are compared using accuracy evaluation on the out-of-sample test set. In this study, we find no evidence that increasing model complexity leads to better forecasting performance for the AI news sentiment series. In most cases, sequential forecasting models perform better than their image-based counterparts, while GASF and GADF transformations do not deliver consistently better forecasting performance for sentiment series of different types and different forecasting horizons. This result indicates that transforming noisy sentiment time series into an image may hide rather than preserve useful information for forecasting purposes. The study contributes to the growing literature on AI news sentiment forecasting by providing a comprehensive comparison of statistical, sequential, and image-based forecasting paradigms and offers practical insights for researchers, policymakers, and practitioners interested in monitoring AI-related expectations and sentiment dynamics.



Academic Editors: Anatoliy Swishchuk and Raymond Lee

Received: 26 May 2026

Revised: 13 July 2026

Accepted: 14 July 2026

Published: 4 August 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

**Keywords:** artificial intelligence; sentiment analysis; time series forecasting; deep learning; Gramian Angular Field; convolutional neural networks

**MSC:** 62M10; 62M45; 68T07; 68T50; 91B84

## 1. Introduction

The growing diffusion of artificial intelligence technologies has intensified public and media attention toward AI-related developments. News-based sentiment indicators provide a high-frequency measure of public perception and expectations, complementing survey-based indicators and macroeconomic data. Large-scale news databases such

as GDELT enable the construction of daily sentiment measures based on global media coverage, offering a rich source for empirical analysis.

Building on [1], this study constructs a daily Artificial Intelligence News Sentiment Index using GDELT data accessed through Google BigQuery. The dataset covers the period starting from 1 January 2016 until 31 December 2025, allowing for an extended time-series analysis of AI-related sentiment dynamics.

This paper compares traditional univariate time series models with deep-learning approaches, including both sequence-based architectures and image-based models. In particular, it applies a Gramian Angular Field transformation to convert sentiment time series into images, which are then used as inputs for CNN models. This approach allows the extraction of spatial features that encode temporal dependencies, providing an alternative framework for sentiment forecasting.

The study contributes to the literature in three ways. First, it constructs a daily AI-related news sentiment index using a large global media dataset. Second, it provides a systematic comparison between traditional forecasting models, deep-learning models, and image-based deep-learning models. Third, it evaluates whether image-based representations improve predictive performance relative to that of conventional time-series modeling approaches.

## 2. Literature Review

News-based sentiment indicators have become an important empirical tool in economics and finance, capturing forward-looking expectations embedded in media coverage. Recent studies show that sentiment extracted from policy communications, financial news, and social media contains predictive information for macroeconomic outcomes and asset prices, complementing traditional quantitative indicators.

In the context of economic policy communication, ref. [2] show that sentiment derived from FOMC statements can be forecasted using both traditional time-series models and deep-learning approaches, with neural architectures such as CNNs and GANs providing improved predictive performance.

A parallel literature examines the role of news sentiment in financial markets; ref. [3] demonstrates that global and ESG-related news sentiment significantly improves stock price forecasting when integrated into deep-learning frameworks, highlighting the asymmetric effects of sentiment during market downturns.

Similarly, ref. [4] shows that sentiment extracted from social media, combined with technical indicators in hybrid deep-learning models, improves stock-price prediction accuracy.

Beyond sentiment modeling, recent methodological advances propose transforming time series into image representations to exploit convolutional neural networks. Reference [5] converts financial time series into Gramian Angular Field images and shows that two-dimensional representations allow CNNs to capture global temporal structures and improve forecasting accuracy relative to that of one-dimensional time-series models. Reference [6] further compares vision-based deep-learning models with numerical approaches for stock prediction, highlighting that visual representations of time-series patterns provide complementary predictive information.

Sequence-based deep learning has emerged as the dominant approach for time-series forecasting. References [7,8] demonstrate that LSTM- and GRU-based architectures incorporating news sentiment improve stock-price forecasting. Bidirectional variants further enhance performance: Reference [9] shows that Bi-LSTM models achieve superior cryptocurrency forecasting when combined with sentiment features, while ref. [10] uses news sentiment analysis to achieve higher accuracy in gold price forecasting using Bi-LSTM with FinBERT sentiment analysis. Hybrid architectures amplify these gains, i.e., ref. [11]

integrates deep autoencoders with LSTM for noise removal, and ref. [12] combines CNN–LSTM–Transformer, achieving an  $R^2$  of 99.28% for energy demand forecasting.

Temporal Convolutional Networks (TCNs) provide efficient alternatives to recurrent architectures. Zhang et al. demonstrate that CEEMDAN–TCN models outperform LSTM by decomposing time series before applying temporal convolutions, enabling parallel processing while maintaining temporal ordering.

Image-based representations leverage computer vision for time-series analysis. Reference [13] introduces the foundational GAF methodology, encoding normalized time series as polar coordinates, with Gramian matrices capturing temporal correlations. Recent applications extend this approach: ref. [14] achieves predictive maintenance improvements through GAF-based image encoding for bearing fault diagnosis, while ref. [15] shows that GAF enhances transfer learning in financial forecasting. Alternative encodings such as Markov Transition Fields and Recurrence Plots capture complementary temporal characteristics, ref. [16] demonstrates robust image encoding techniques while maintaining computational efficiency, and ref. [17] shows patch-level contrastive learning with Vision Transformers, capturing fine-grained temporal features.

Generative Adversarial Networks enable probabilistic forecasting with uncertainty quantification. Reference [18] introduces Fin-GAN, incorporating economics-driven loss functions, achieving superior risk-adjusted returns vs. those of LSTM and ARIMA. TimeGAN [19] addresses data scarcity by generating synthetic sequences that preserve temporal dynamics. Topology-aware extensions prove powerful: ref. [20] achieves a 318% Sharpe ratio improvement through persistent homology integration.

Empirical comparisons consistently favor deep learning. Reference [21] shows that LSTM achieves an 84–87% error reduction vs. that of ARIMA across multiple datasets. Additionally, ref. [22] finds that hybrid attention-based models outperform individual approaches across six industry sectors. This is also shown by [23,24], which demonstrate improvements through LSTM–ARIMA ensembles. However, ref. [25] challenges transformer superiority, showing that simple linear models outperform sophisticated architectures on benchmark datasets due to temporal ordering preservation.

Media sentiment serves as high-frequency expectations measure. This is shown by [1], which showed that GDELT-derived news attention metrics improve directional accuracy by 2.4 percentage points over that of prices alone. News-based Economic Policy Uncertainty indices exemplify practical application: ref. [26] develops Nigeria’s EPU index predicting exchange rates and inflation, while ref. [27] demonstrates the causal impact of the cable news-based EPU on major asset returns. Reference [28] shows that sentiment improves volatility forecasting, with 14.99% gains on extreme variation days.

Advanced NLP enables domain-specific sentiment extraction. Shah et al. apply FinBERT to financial sentiment using LSTM forecasting, substantially reducing errors vs. those of generic analyzers. AI-specific sentiment indices remain underdeveloped despite AI’s growing importance. This is shown by [29], which documents AI coverage surging post-ChatGPT with rising emotional polarization, while ref. [30] show media attention increased tenfold with discourse centering on experts. Additionally, ref. [31] find risk-benefit tradeoffs and value expectations drive public AI acceptance, mediated by literacy and communication.

Multimodal integration combines heterogeneous representations. Reference [32] demonstrates CNN–LSTM–Transformer hybrids achieve superior performance through complementary feature extraction.

Despite progress, critical gaps persist. Sentiment forecasting and image-based modeling remain separate research strands. GAF and alternative encodings for sentiment time series lack exploration. AI-specific sentiment indices from global news coverage are

underdeveloped, and GAN approaches for sentiment forecasting require deeper investigation. The present study addresses these gaps by constructing daily AI news sentiment indices from GDELT, comparing traditional models with image-based deep learning, and evaluating whether GAF–CNN frameworks enhance sentiment prediction—integrating media analysis, expectations formation, and advanced forecasting techniques.

### 3. Materials and Methods

The daily AI-related news sentiment was constructed from the GDELT Global Knowledge Graph (GKG) 2.1 dataset, accessed through Google BigQuery. Data extraction, pre-processing, and sentiment index construction were performed in Google BigQuery using SQL queries. The sample covers 1 January 2016 to 31 December 2025, yielding 3635 daily observations, with the year 2025 reserved as the out-of-sample test period. The resulting dataset was then imported into Google Colab, where all subsequent statistical analyses, deep learning model development, forecasting, and model evaluation were implemented in Python 3.12.13.

A record-level timestamp is reconstructed by parsing the first fourteen characters (YYYYMMDDHHMMSS) of the GKG record identifier into a BigQuery `TIMESTAMP`. AI-related records are identified by a case-insensitive match for the terms “artificial intelligence”, “machine learning”, “deep learning”, and “generative AI” in the `AllNames` (entity/name) field. Sentiment is taken from the `V2Tone` field, a comma-delimited list whose leading elements are the document tone, the positive score, and the negative score: by construction, the tone equals the positive score minus the negative score, while the positive and negative scores give the percentage of positively and negatively charged words in the document (each on a 0–100 scale) [32].

The resulting dataset is then aggregated at the daily level using `DATE(ts)` and simple averaging across all matching records per day:

Avg\_Tone: daily average of AvgTone;

PositiveS: daily average of PositiveScore;

NegativeS: daily average of NegativeScore.

Type conversion is handled defensively using `SAFE_CAST` to avoid query failures from malformed records.

Daily aggregation by simple averaging follows the standard GDELT Global Knowledge Graph methodology, in which the document-level tone scores are averaged to summarize the tone of coverage over a period [32,33]. This convention is the basis of GDELT-derived news-sentiment measures [1,34] and is adopted here for comparability and reproducibility.

Following the normalized polarity approach used in sentiment-index forecasting studies [35,36], the AI News Sentiment Index is computed from the daily positive and negative components as

$$Sentiment_t = \frac{Positive_t - Negative_t}{Positive_t + Negative_t} \quad (1)$$

This scaling bounds the index between  $[-1, 1]$  and standardizes the measure across days with different overall sentiment intensity.

#### 3.1. ARMA–GARCH Models

Classical time-series modeling techniques rely on the assumption of stationarity to produce robust results. Stationarity implies that the statistical properties of the series remain constant over time, exhibiting no trend, no seasonality, and constant variance.

A trend reflects a long-term systematic increase or decrease in the level of the series and may be either deterministic or stochastic. Seasonality arises when the series is influenced by recurring, calendar-based patterns, such as daily, weekly, monthly or quarterly cycles.

We applied the Box–Jenkins methodology to model and forecast the sentiment time series. Stationarity was first assessed using a combination of graphical tools and formal statistical tests. Specifically, we examined the autocorrelation function (ACF) and the partial autocorrelation function (PACF). The ACF measures the correlation between and its lagged values, while the PACF captures the direct correlation at lag  $k$ , the net of the effects of shorter lags.

Formally, the autocorrelation at lag  $k$  is defined as

$$\rho(k) = \frac{\text{Cov}(y_t, y_{t-k})}{\sqrt{\text{Var}(y_t)\text{Var}(y_{t-k})}} \quad (2)$$

All sentiment series (Positive, Negative and SI) were formally tested for stationarity using two complementary unit-root tests: the Augmented Dickey–Fuller (ADF) test and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test. The ADF test is based on the regression

$$\Delta y_t = \alpha + \beta_t + \gamma_{t-1} + \sum_{i=1}^p \delta \Delta y_{t-i} + \varepsilon_t \quad (3)$$

where the null hypothesis  $H_0: \gamma = 0$  indicates the presence of a unit root. In contrast, the KPSS test assumes stationarity under the null hypothesis and decomposes the series as

$$y_t = \mu_t + r_t + \varepsilon_t, \quad r_t = r_{t-1} + u_t \quad (4)$$

where stationarity corresponds to  $\sigma_u^2 = 0$ .

Since no clear seasonal patterns were detected, stationarity was achieved through first differencing of the original series

$$y_t^{(1)} = y_t - y_{t-1} \quad (5)$$

The stationary series are modeled using the ARIMA process of the form

$$\varphi(L)(1-L)^d y_t = \theta(L)\varepsilon_t \quad (6)$$

where  $\varphi(L) = 1 - \varphi_1 L - \dots - \varphi_p L^p$  and  $\theta(L) = 1 + \theta_1 L + \dots + \theta_q L^q$  are lag polynomials,  $d = 1$  denotes the differencing order, and  $\varepsilon_t$  is white noise.

Model orders  $(p, q)$  are selected using a grid search strategy, with maximum lag lengths guided by the ACF and PACF, with the final specification chosen by the minimum Akaike Information Criterion (AIC) among diagnostically checked candidates on the training sample.

Due to the high-frequency nature of the data, pronounced heteroskedasticity was observed in the residuals of the mean equations (AR(I)MA models) for all time series. In order to evaluate the adequacy of the mean models and the reliability of their forecasts, we conducted diagnostic checks on the residuals, including the Ljung–Box test for residual autocorrelation, the Jarque–Bera test for normality, and Engle’s ARCH-LM test for conditional heteroskedasticity.

The ARCH-LM test is based on the auxiliary regression

$$\varepsilon_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i \varepsilon_{t-i}^2 + u_t \quad (7)$$

with the null hypothesis of homoskedasticity  $H_0: \alpha_1 = \dots = \alpha_m = 0$ .

Significant ARCH effects were detected, indicating the presence of volatility clustering and time-varying conditional variance. We therefore model the conditional variance explicitly, comparing two specifications selected by AIC on the training residuals: the symmetric GARCH(1,1) model [37] and the asymmetric GJR-GARCH(1,1) [38], each estimated under Gaussian and Student- $t$  innovations.

The standard GARCH(1,1) model is given by

$$\varepsilon_t = \sigma_t z_t, \quad z_t \sim N(0, 1) \quad (8)$$

$$\sigma_t^2 = w + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \quad (9)$$

where  $w > 0$ ,  $\alpha \geq 0$ , and  $\beta \geq 0$ . This framework allows the model to capture persistent volatility dynamics and improves both in-sample fit and forecast reliability.

To capture the leverage effect, whereby negative sentiment shocks may raise conditional variance more than do positive shocks of equal magnitude, we also estimate the asymmetric GJR-GARCH(1,1) model, as follows:

$$\sigma_t^2 = \omega + (\alpha + \gamma I_{t-1}) \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2, \quad I_{t-1} = \begin{cases} 1, & \varepsilon_{t-1} < 0 \\ 0, & \varepsilon_{t-1} \geq 0 \end{cases} \quad (10)$$

where  $I_{t-1}$  equals one when the previous shock is negative, adding the asymmetric response  $\gamma$ .

For each series, the volatility specification (GARCH vs. GJR-GARCH) and the innovation distribution (Normal vs. Student- $t$ ) are chosen by the minimum AIC on the training sample.

### 3.2. ETS Models

To complement ARIMA-type specifications, forecasts are also generated using Exponential Smoothing State Space (ETS) models. ETS models provide a flexible framework for time-series forecasting by decomposing the observed process into latent components such as level, trend, and (if present) seasonality, while allowing different functional forms for the error term. In contrast to ARIMA models, ETS models do not rely on differencing to remove non-stationarity; instead, they model evolving components directly in a state space representation.

Following the framework in [31], an ETS model is typically denoted as ETS(Error, Trend, Seasonal), where each component can be additive, multiplicative, or absent. In this study, due to the absence of clear seasonal behavior, the seasonal component is either excluded or included only when supported empirically by model selection.

Let  $y_t$  denote the observed daily sentiment series. In the additive-error case, the observation equation takes the form

$$y_t = y_{t|t-1} + \varepsilon_t \sim i.i.d.(0, \sigma^2), \quad (11)$$

where  $y_{t|t-1}$  is the one-step-ahead forecast based on the state at  $t - 1$ .

For the level-only model (ETS(A,N,N)), the dynamics are

$$y_t = l_{t-1} + \varepsilon_t, \quad l_t = l_{t-1} + \alpha \varepsilon_t, \quad (12)$$

where  $l_t$  denotes the latent level, and  $\alpha \in (0, 1)$  is the smoothing parameter.

For the additive trend model (ETS(A, A, N)), the specification becomes

$$y_t = l_{t-1} + \beta_{t-1} + \varepsilon_t, \quad (13)$$

with level and trend evolving according to the smoothing parameters  $\alpha$  and  $\beta$ .

To prevent excessive trend extrapolation, a damped trend variant (ETS(A,  $A_d$ , N)) is also considered, in which the trend component is scaled by a damping factor  $0 < \varphi < 1$ .

Seasonal extensions follow standard ETS formulations but serve primarily as robustness checks rather than baseline specifications.

ETS model selection evaluates all admissible combinations of error type, trend specification, damped trend, and weekly ( $m = 7$ ) seasonality, with the specification for each series selected by the minimum AIC on the training sample.

After selecting the optimal ETS specification, the model is fitted on the full training sample and used to generate an  $h$ -step-ahead forecast. Forecasts correspond to conditional expectations of future observations, given the information set at the end of the training period.

### 3.3. LSTM Models

The LSTM models were trained by minimizing the Mean Squared Error (MSE) loss function, defined as

$$RMSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (14)$$

where  $y_i$  is the observed sentiment value at time  $i$ ,  $\hat{y}_i$  is the model's predicted value, and  $n$  represents the number of observations in the batch.

Like all neural models in this study, the LSTM was optimized with the Adam algorithm. The hyperparameter search space and the tuning procedure shared by the neural models are described together in Section 3.10 (Table 1), after all model families have been introduced.

**Table 1.** Hyperparameter search space for the neural sequence and image models (LSTM, 1D CNN, TCN, CNN-GASF, CNN-GADF). Configurations were tuned using grid, random, and Bayesian (TPE) searches.

| Hyperparameter           | Value/Range                                                                 | Description                                                               |
|--------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Look-back window         | 14, 30, 60, 90, 120<br>(14, 30, 60 for the image models)                    | Number of past time-steps used as the input window                        |
| Units/filters            | 32, 64, 96, 128                                                             | Hidden units (LSTM) or convolutional filters (1D CNN, TCN, CNN-GASF/GADF) |
| Kernel size              | 5 (1D CNN); 3 (TCN)                                                         | Width of the convolutional filters (fixed)                                |
| Dilated residual blocks  | 4; dilations 1, 2, 4, 8 (TCN only)                                          | Depth of the temporal convolution stack                                   |
| Dropout rate             | 0.0, 0.1, 0.2, 0.3                                                          | Dropout fraction applied after each block                                 |
| Learning rate            | $10^{-2}$ , $5 \times 10^{-3}$ , $10^{-3}$ , $5 \times 10^{-4}$ , $10^{-4}$ | Step size for the Adam optimizer                                          |
| Batch size               | 32, 64, 128                                                                 | Number of samples per gradient update                                     |
| Epochs (tuning/final)    | 12/80                                                                       | Early stopping on val_loss, patience = 8                                  |
| Validation split         | 0.15                                                                        | Fraction of training data used for validation                             |
| Evaluations per strategy | 30                                                                          | Equal budget for grid/random/Bayesian search                              |
| Seed                     | 42                                                                          | Fixed seed for reproducibility (NumPy/TensorFlow)                         |

### 3.4. CNN Models

To benchmark deep-learning performance against classical time series models, forecasts are generated using a one-dimensional Convolutional Neural Network (1D CNN). CNNs are well suited for extracting local temporal patterns from sequential data through convolutional filters and pooling operations. In this setting, the model is trained to predict the next-day sentiment value from a fixed-length window of past observations.

Let  $y_t$  denote the daily sentiment series. The series is transformed into a supervised learning dataset using a sliding window of length  $L$ . For each time  $t$ , the input vector and target are defined as

$$x_t = [y_{t-L}, y_{t-L+1}, \dots, y_{t-1}]^T \quad (15)$$

The learning objective is to approximate the nonlinear mapping, as follows:

$$y_t = f_\theta(x_t) \quad (16)$$

where  $f_\theta$  is the CNN parameterized by weights  $\theta$ .

Each series was split into a training set made up of all observations before 2025. The test set contains all observations after 2025.

The CNN was trained on standardized inputs to stabilize optimization. Using the training sample mean and variance, values were scaled as

$$z_t = \frac{y_t - \mu_{train}}{\sigma_{train}} \quad (17)$$

Predictions were generated in standardized units and then transformed back to the original scale via inverse mapping.

To avoid information leakage at the start of the test period, the first test input window is constructed by concatenating the last scaled training observations with the scaled test series.

The implemented 1D CNN used two convolution-pooling blocks followed by fully connected layers. For an input sequence  $x_t$ , a 1D convolutional layer computes feature map  $m$  as

$$h_t^{(m)} = \sigma \left( b^{(m)} + \sum_{j=0}^{K-1} w_j^{(m)} x_{t-j} \right) \quad (18)$$

where  $K$  is the kernel size,  $w_j^{(m)}$  represents filter weights,  $b^{(m)}$  is a bias term, and  $\sigma(\cdot)$  is the ReLU activation. Max pooling is used to downsample  $\sigma(\cdot)$ .

The final prediction is produced by a linear output layer, as follows:

$$\hat{y}_t = w^T g_t + c \quad (19)$$

where  $g_t$  denotes the last hidden representation after flattening and dense layers.

The model training used the Adam optimizer with a learning rate  $\eta$ . For each candidate configuration, the model was trained with a 15% validation split from the training sample, preserving temporal order, and early stopping monitored the validation loss and restored the best weights. The best CNN specification was selected by minimum validation RMSE.

### 3.5. Gramian Angular Field

In order to explore an image-based representation of the sentiment series, we transformed the time windows into Gramian Angular Field (GAF) images, following the work of [11], and applied a two-dimensional convolutional neural network (2D CNN) that captures pairwise temporal correlations in a spatial format. We consider both the Gramian Angular Summation Field (GASF) and the Gramian Angular Difference Field (GADF) variants.

Each sentiment time series was rescaled in the range of  $[-1, 1]$ , as follows:

$$\tilde{x}_{-1} = \frac{(x_i - \max(X)) + (x_i - \min(X))}{(\max(X) - \min(X))} \quad (20)$$

Next, 1D time series are transformed into a GAF representation by converting Cartesian points to the polar coordinates and using GADF.

$$\phi_i = \arccos(\tilde{x}_i) \quad (21)$$

$$GADF = \begin{bmatrix} \sin(\phi_1 - \phi_1) & \dots & \sin(\phi_1 - \phi_n) \\ \vdots & \ddots & \vdots \\ \sin(\phi_n - \phi_1) & \dots & \sin(\phi_n - \phi_n) \end{bmatrix} \quad (22)$$

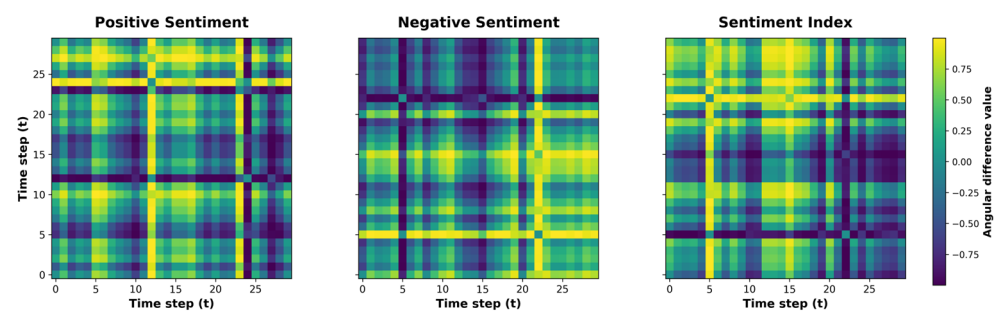
The GADF defined above emphasizes relative, directional change and discards absolute level. For completeness and comparison, we also employ the Gramian Angular

Summation Field (GASF), which encodes the cosine of the sum of the polar angles and therefore, preserves absolute level and global structure.

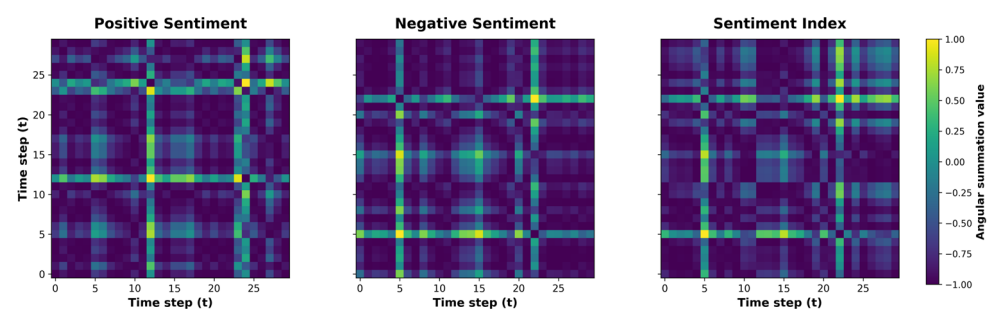
$$GASF = \begin{bmatrix} \cos(\varnothing_1 + \varnothing_1) & \dots & \cos(\varnothing_1 + \varnothing_n) \\ \vdots & \ddots & \vdots \\ \cos(\varnothing_n + \varnothing_1) & \dots & \cos(\varnothing_n + \varnothing_n) \end{bmatrix} \quad (23)$$

While the GASF is symmetrical and retains level information, the GADF is antisymmetrical, with a zero diagonal, and is more sensitive to local dynamics. The two encodings are complementary and may yield different forecasting behavior, which motivates evaluating both.

Each time series of length was transformed into a GADF (Figure 1) and a GASF (Figure 2) of size.



**Figure 1.** Example GADF encodings of Positive, Negative and Sentiment Index series (look-back window = 30).



**Figure 2.** Example GASF encodings of Positive, Negative and Sentiment Index series (look-back window = 30).

### 3.6. Temporal Convolutional Network (TCN)

As an efficient alternative to recurrent models, a Temporal Convolutional Network (TCN) [13] stacks dilated causal convolutions whose receptive field grows exponentially with depth, while strict temporal ordering is preserved. With kernel size  $K$  and an exponentially increasing dilation rate, the layer- $l$  activation is

$$h_t^{(l)} = \text{ReLU}\left(\sum_{i=0}^{K-1} W_i^{(l)} h_{t-di}^{(l-1)}\right), \quad d = 2^{l-1} \quad (24)$$

Residual connections stabilize training. The TCN used four residual blocks (dilations 1, 2, 4, 8) and a fixed kernel size of 3 and was tuned over the neural search space reported in Table 1.

### 3.7. LightGBM

As a strong non-deep baseline, we include the gradient-boosted decision-tree model LightGBM, which fits an additive ensemble of regression trees on lagged features.

$$F_M(x) = \sum_{m=1}^M v f_m(x), \quad f_m = \operatorname{argmin}_f \sum_t \ell(y_t, F_{m-1}(x_t) + f(x_t)) \quad (25)$$

where  $v$  is the shrinkage (learning rate), and  $f_m$  is the  $m$ -th tree. LightGBM was tuned over the look-back window, number of estimators, number of leaves, and learning rate, as reported in Table 2.

**Table 2.** Hyperparameter search space for the LightGBM and GAN models.

| Model    | Hyperparameter       | Value/Range                                                  | Description                                  |
|----------|----------------------|--------------------------------------------------------------|----------------------------------------------|
| LightGBM | Look-back window     | 30, 60, 90                                                   | Number of lagged features used as predictors |
|          | Number of estimators | 200, 400, 800, 1200                                          | Number of boosting trees                     |
|          | Learning rate        | 0.01, 0.03, 0.05, 0.1                                        | Shrinkage applied per boosting step          |
|          | Number of leaves     | 15, 31, 63, 127                                              | Maximum number of leaves per tree            |
|          | Subsample/colsample  | 0.8/0.8                                                      | Row/feature subsampling (fixed)              |
| GAN      | Look-back window     | 30, 60, 90                                                   | Length of the conditioning window            |
|          | Noise dimension      | 8, 16, 32                                                    | Dimension of the latent input vector         |
|          | Learning rate        | $1 \times 10^{-4}$ , $2 \times 10^{-4}$ , $5 \times 10^{-4}$ | Adam step size (generator and discriminator) |
|          | Epochs               | 120                                                          | Adversarial training epochs (fixed)          |
|          | Seed                 | 42                                                           | Fixed seed for reproducibility               |

### 3.8. Generative Adversarial Network (GAN)

A generative-adversarial baseline trains a generator  $G$  and a discriminator  $D$  in the minmax, as follows:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (26)$$

Conditioned on the look-back window, the generator produces one-step-ahead forecasts, while the discriminator distinguishes generated from observed continuations. The GAN was tuned over the look-back window, noise dimension, and learning rate (Table 2), with a bounded number of critic updates per generator step to mitigate instability on this short univariate series.

### 3.9. Hyperparameter Optimization and Model Selection

All models were tuned under a single framework so that their comparison is fair and reproducible. Hyperparameters were optimized with three complementary strategies on the same search space under an equal budget:

- (i) Grid search, which evaluates an evenly spaced, budgeted subset of the Cartesian product of the candidate values;
- (ii) Random search [39], which samples configurations from the same ranges and is more efficient in higher-dimensional spaces;
- (iii) Bayesian optimization with the Tree-Structured Parzen Estimator (TPE) [40] implemented in Optuna [41], which builds a probabilistic surrogate of the validation error.

The best configuration across the three strategies, scored by RMSE on the validation sample, was retained. All runs fixed the random seed [42], and final models were trained with a 15% validation split and early stopping.

The recurrent (LSTM), one-dimensional convolutional (1D CNN), temporal-convolutional (TCN), and image-based convolutional (CNN-GASF, CNN-GADF) architectures all map a fixed look-back window of past sentiment to the next-day value and are trained by gradient descent on the same standardized inputs; therefore, they share the common search space with that shown in Table 1. Tuning the same look-back window, capacity (units or filters), dropout, learning rate, and batch size for every architecture removes unequal tuning efforts, so any performance gap reflects the architecture itself rather than the search. The ranges bracket values that are standard in the time-series deep-learning literature, while remaining computationally feasible: look-back windows from two weeks to four months span short- and medium-memory dynamics. Widths of 32–128 units match the small-to-moderate capacity appropriate for a single daily series of roughly 3300 training points; a dropout up to 0.3 and a logarithmic learning-rate grid ( $10^{-2}$  to  $10^{-4}$ ) guard against over- and under-fitting; and early stopping caps training length. The image models additionally restrict the look-back to {14, 30, 60} because the Gramian Angular Field encodes an  $n$ -step window as an  $n \times n$  image, so larger windows inflate the input quadratically without adding information.

LightGBM and GAN models fall outside the neural search space and are tuned over architecture-appropriate grids (Table 2). LightGBM is a gradient-boosted tree ensemble using lagged features, so its capacity is governed by the number of trees, the number of leaves, and the shrinkage rate.

The GAN is governed by the latent noise dimension and a deliberately small learning rate, reflecting the well-documented instability of adversarial training on short univariate series. Its look-back grid matches that of the other models so that the conditioning window is comparable.

### 3.10. Model Performance Comparison

To evaluate the predictive accuracy of the developed models, we computed standard error metrics: the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE).

Model selection follows a strict framework: ARIMA and ETS specifications are chosen by the Akaike Information Criterion (AIC) on the training sample, whereas the deep-learning models are selected by validation RMSE.

The 2025 test set is used exclusively for the final cross-model comparison, for which we assess statistical significance with the Diebold–Mariano test.

MAE measures the average absolute difference between observed and predicted values and is computed as

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (27)$$

RMSE is a common metric for evaluating the accuracy of predictive models, measuring the squared differences between predicted and observed values. It is computed as

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right)^{1/2} \quad (28)$$

The main purpose of the Diebold–Mariano (DM) test is to determine whether there is a significant difference in the forecasting accuracy of two models. Using two sequences of forecasts, the quality of each model is evaluated through an error loss function. The null hypothesis of the test states that there is no significant difference in forecasting accuracy between the two models [42].

$$DM = \frac{\bar{d}}{\sqrt{\hat{\omega}/P}}, \quad (29)$$

where  $\bar{d}$  represents the average loss differential and  $\hat{\omega}$  denotes a consistent estimator of the asymptotic variance associated with  $\sqrt{P}$ .

#### 4. Results

Table 3 provides the descriptive statistics for the three sentiment series: Positive Sentiment, Negative Sentiment, and Sentiment Index (SI). The Positive Sentiment series has a relatively higher mean, at 3.933, than the Negative Sentiment, which has a lower mean of 2.736, indicating that Positive Sentiment is predominant in the data set. The Sentiment Index, being a composite measure, has a near-zero mean of 0.184, with low volatility, as indicated by a standard deviation of 0.092. The minimum and maximum values of each series indicate that Negative Sentiment has a wider range of values, from 1.382 to 6.976.

**Table 3.** Descriptive statistics.

| Indicator          | Positive  | Negative | Sentiment Index (SI) |
|--------------------|-----------|----------|----------------------|
| Mean               | 3.933014  | 2.736372 | 0.183856             |
| Standard Error     | 0.005736  | 0.008352 | 0.001530             |
| Median             | 3.959862  | 2.640959 | 0.186184             |
| Standard Deviation | 0.345835  | 0.503575 | 0.092222             |
| Kurtosis           | 1.807083  | 1.851469 | 0.538841             |
| Skewness           | −0.227322 | 0.751344 | −0.436154            |
| Minimum            | 1.245346  | 1.382346 | −0.265093            |
| Maximum            | 5.782512  | 6.975749 | 0.499151             |
| Count              | 3635      | 3635     | 3635                 |

Figure 3 shows the evolution of Positive Sentiment, Negative Sentiment, and the Sentiment Index over the entire sample period, with the vertical dashed line indicating the start of the 2025 out-of-sample test data. While the Positive and Negative Sentiment series display strong volatility clustering, the Sentiment Index series displays a less extreme pattern.

The strong Negative Sentiment spike in 2023 corresponds to the end of the 2022–2023 generative AI hype cycle, as well as to growing worries about the sustainability, regulatory issues, energy usage, and ethical implications of widespread AI adoption. Such strong sentiment reversals are a major problem for linear forecasting methods and highlight the need for more flexible and nonlinear techniques.

Table 4 reports the candidate ARIMA models ranked by training-sample AIC. The AIC-selected order is ARIMA(3,1,3) for Positive Sentiment, ARIMA(3,1,3) for Negative Sentiment, and ARIMA(2,1,3) for the Sentiment Index.

While a reasonable linear forecast has been achieved with ARIMA models, diagnostic tests of the residuals indicated a significant presence of ARCH effects, thus violating the constant variance condition of the normal distribution of errors. This has led to the estimation of variance equations using (G)ARCH models to better capture the time-varying volatility of the sentiment series.



**Figure 3.** Train and test data for Positive, Negative, and Sentiment Index.

**Table 4.** Top three ARIMA models ranked by training-sample AIC.

| No.                | p | d | q | AIC      | BIC      | RMSE-Test | MAE-Test |
|--------------------|---|---|---|----------|----------|-----------|----------|
| Positive Sentiment |   |   |   |          |          |           |          |
| 1                  | 3 | 1 | 3 | 421.44   | 464.12   | 0.1891    | 0.1418   |
| 2                  | 2 | 1 | 3 | 481.16   | 517.74   | 0.1876    | 0.1405   |
| 3                  | 3 | 1 | 1 | 553.29   | 583.77   | 0.1901    | 0.1414   |
| Negative Sentiment |   |   |   |          |          |           |          |
| 1                  | 3 | 1 | 3 | 2312.53  | 2355.21  | 0.2401    | 0.1717   |
| 2                  | 2 | 1 | 3 | 2339.04  | 2375.63  | 0.2419    | 0.1746   |
| 3                  | 3 | 1 | 1 | 2450.54  | 2481.03  | 0.2425    | 0.1727   |
| Sentiment Index    |   |   |   |          |          |           |          |
| 1                  | 2 | 1 | 3 | −8093.84 | −8057.25 | 0.0479    | 0.0346   |
| 2                  | 3 | 1 | 3 | −8090.22 | −8047.54 | 0.0479    | 0.0346   |
| 3                  | 3 | 1 | 2 | −7922.17 | −7885.59 | 0.0489    | 0.0352   |

ARIMA–GARCH models were estimated using the AIC-selected ARIMA mean specifications, comparing GARCH(1,1) and GJR-GARCH(1,1) under Gaussian and Student-*t* innovations (Table 5). Student-*t* innovations were selected for all series, with the asymmetric GJR-GARCH model preferred for Negative Sentiment and the Sentiment Index. However, none of the volatility models improved one-step-ahead forecasting accuracy relative to that of the corresponding ARIMA models. Although the GARCH extensions captured conditional volatility, they did not enhance point forecasts, particularly for event-driven sentiment shocks. For the Sentiment Index, ARIMA remained superior (RMSE = 0.0479

vs. 0.0543 for GJR-GARCH), indicating that modeling conditional variance provides little additional predictive value for this series.

**Table 5.** Selected ARMA–GARCH models: one-step ahead metrics.

| Series             | Model                                              | RMSE-Test | MAE-Test |
|--------------------|----------------------------------------------------|-----------|----------|
| Positive Sentiment | ARIMA(3,1,3)–GARCH(1,1),<br>Student- <i>t</i>      | 0.2057    | 0.1540   |
| Negative Sentiment | ARIMA(3,1,3)–GJR-<br>GARCH(1,1), Student- <i>t</i> | 0.2620    | 0.1888   |
| Sentiment Index    | ARIMA(2,1,3)–GJR-<br>GARCH(1,1), Student- <i>t</i> | 0.0543    | 0.0393   |

Table 6 summarizes the AIC-selected ETS specification, ETS(A,N,A)-additive error, no trend, and additive weekly seasonality, for all three series. For one-period-ahead forecasts, the simple mean models (regular ARIMA and ETS) perform better than do the ARIMA–GARCH models for all three time series. In the case of Sentiment Index, ARIMA shows a RMSE of 0.0479, and ETS gives a RMSE of 0.0487, while ARIMA–GARCH yields 0.0543. The ranking is the same for both Positive and Negative sentiment. These findings corroborate the previous findings from the GARCH analysis, according to which the addition of a conditional variance component does not increase the precision of one-period-ahead point forecasts. This is because RMSE and MAE measure the accuracy of the conditional mean rather than the conditional variance, and therefore, improved volatility modeling does not imply better point forecasts. However, even though ETS, being adaptive and having weekly seasonality, competes with ARIMA and outperforms volatility enhanced models, it does not exceed the forecasting power of ARIMA model.

**Table 6.** Selected ETS models.

| Series             | Model      | RMSE-Test | MAE-Test |
|--------------------|------------|-----------|----------|
| Positive Sentiment | ETS(A,N,A) | 0.1898    | 0.1429   |
| Negative Sentiment | ETS(A,N,A) | 0.2464    | 0.1787   |
| Sentiment Index    | ETS(A,N,A) | 0.0487    | 0.0353   |

Table 7 summarizes the hyperparameter configurations selected through grid search, random search, and Bayesian optimization. The optimal configurations varied across sentiment series and model classes, indicating that AI news sentiment exhibits heterogeneous temporal dependencies. LSTM and 1D CNN generally favored shorter look-back windows (14–30 days), whereas TCN and LightGBM often selected longer windows (60–90 days), suggesting that some architectures benefit from a broader historical context. TCN models converged toward lower learning rates and larger filter sizes, while LSTM and 1D CNN preferred simpler configurations, indicating that most predictive information is contained in local temporal dynamics. Image-based CNN models showed greater variability and typically required larger filter sets and longer look-back windows, reflecting the additional complexity introduced by GASF and GADF transformations. Despite this increased complexity, they did not consistently outperform sequential models, suggesting that image-based representations do not preserve predictive information more effectively than does direct sequence modeling for noisy sentiment series. Overall, no single hyperparameter configuration or architecture proved universally optimal, highlighting that forecasting performance depends on both the characteristics of the sentiment series and the underlying modeling approach.

**Table 7.** Selected deep-learning model parameters.

| Architecture         | Best Configuration                                                     |
|----------------------|------------------------------------------------------------------------|
| Positive Sentiment   |                                                                        |
| LSTM                 | look-back = 30, units = 128, dropout = 0.2, lr = 0.01, batch = 128     |
| 1D CNN               | look-back = 30, filters = 32, dropout = 0.0, lr = 0.01, batch = 128    |
| TCN                  | look-back = 90, filters = 128, dropout = 0.1, lr = 0.0001, batch = 64  |
| LightGBM             | look-back = 90, n_estimators = 800, num_leaves = 15, lr = 0.01         |
| GAN                  | look-back = 30, noise dim = 32, lr = 0.0002                            |
| CNN + GASF           | look-back = 60, filters = 64, dropout = 0.0, lr = 0.01, batch = 32     |
| CNN + GADF           | look-back = 30, filters = 128, dropout = 0.0, lr = 0.005, batch = 128  |
| Negative Sentiment   |                                                                        |
| LSTM                 | look-back = 14, units = 64, dropout = 0.1, lr = 0.01, batch = 32       |
| 1D CNN               | look-back = 14, filters = 64, dropout = 0.1, lr = 0.01, batch = 32     |
| TCN                  | look-back = 60, filters = 96, dropout = 0.0, lr = 0.0001, batch = 32   |
| LightGBM             | look-back = 60, n_estimators = 400, num_leaves = 15, lr = 0.01         |
| GAN                  | look-back = 30, noise dim = 16, lr = 0.0001                            |
| CNN + GASF           | look-back = 30, filters = 128, dropout = 0.0, lr = 0.01, batch = 64    |
| CNN + GADF           | look-back = 90, filters = 96, dropout = 0.2, lr = 0.01, batch = 64     |
| Sentiment Index (SI) |                                                                        |
| LSTM                 | look-back = 60, units = 32, dropout = 0.3, lr = 0.005, batch = 32      |
| 1D CNN               | look-back = 14, filters = 128, dropout = 0.0, lr = 0.001, batch = 64   |
| TCN                  | look-back = 60, filters = 32, dropout = 0.2, lr = 0.001, batch = 32    |
| LightGBM             | look-back = 60, n_estimators = 400, num_leaves = 15, lr = 0.01         |
| GAN                  | look-back = 30, noise dim = 32, lr = 0.0002                            |
| CNN + GASF           | look-back = 60, filters = 32, dropout = 0.3, lr = 0.0005, batch = 64   |
| CNN + GADF           | look-back = 60, filters = 128, dropout = 0.3, lr = 0.0001, batch = 128 |

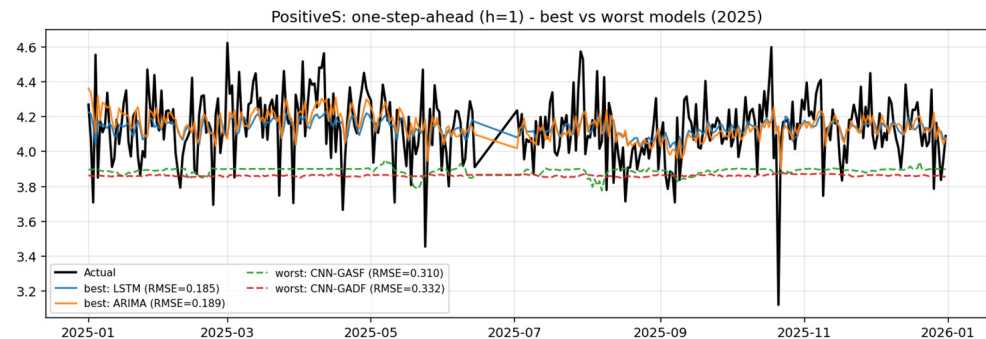
Table 8 consolidates the one-step-ahead out-of-sample results for all ten models. The well-specified classical models are highly competitive: ARIMA attains the lowest Sentiment-Index RMSE (0.0479), LightGBM the lowest Negative RMSE (0.2398), and LSTM the lowest Positive RMSE (0.1853), with negligible gaps among ARIMA, ETS, LightGBM, and the best sequence models.

**Table 8.** Performance metrics for Positive Sentiment, Negative Sentiment, and Sentiment Index (one-step-ahead).

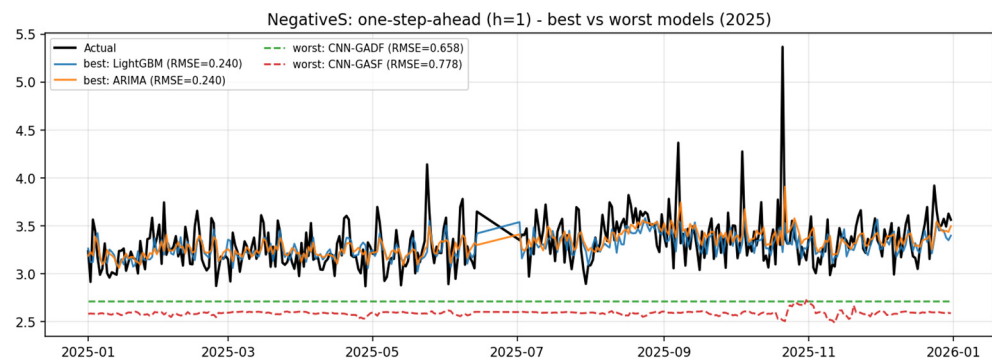
| Model       | Positive Sentiment |        | Negative Sentiment |        | Sentiment Index |        |
|-------------|--------------------|--------|--------------------|--------|-----------------|--------|
|             | RMSE               | MAE    | RMSE               | MAE    | RMSE            | MAE    |
| ARIMA       | 0.1891             | 0.1418 | 0.2401             | 0.1717 | 0.0479          | 0.0346 |
| ARIMA-GARCH | 0.2057             | 0.1540 | 0.2620             | 0.1888 | 0.0543          | 0.0393 |
| ETS         | 0.1898             | 0.1429 | 0.2464             | 0.1787 | 0.0487          | 0.0353 |
| LSTM        | 0.1853             | 0.1393 | 0.2500             | 0.1773 | 0.0528          | 0.0376 |
| 1D CNN      | 0.1941             | 0.1465 | 0.2560             | 0.1817 | 0.0530          | 0.0377 |
| TCN         | 0.1923             | 0.1432 | 0.2783             | 0.2023 | 0.0497          | 0.0354 |
| LightGBM    | 0.1921             | 0.1452 | 0.2398             | 0.1693 | 0.0486          | 0.0344 |
| GAN         | 0.1965             | 0.1492 | 0.2873             | 0.2104 | 0.0534          | 0.0388 |
| CNN-GASF    | 0.3097             | 0.2681 | 0.7778             | 0.7344 | 0.1010          | 0.0877 |
| CNN-GADF    | 0.3319             | 0.2908 | 0.6579             | 0.6073 | 0.1010          | 0.0887 |

Figures 4–6 plot the one-step-ahead out-of-sample forecasts of the best models against the realized series for each sentiment measure. The best-performing models generally track the overall level and short-term movements of the Positive Sentiment and Sentiment Index series. However, significant sentiment spikes remain difficult to predict, with even the

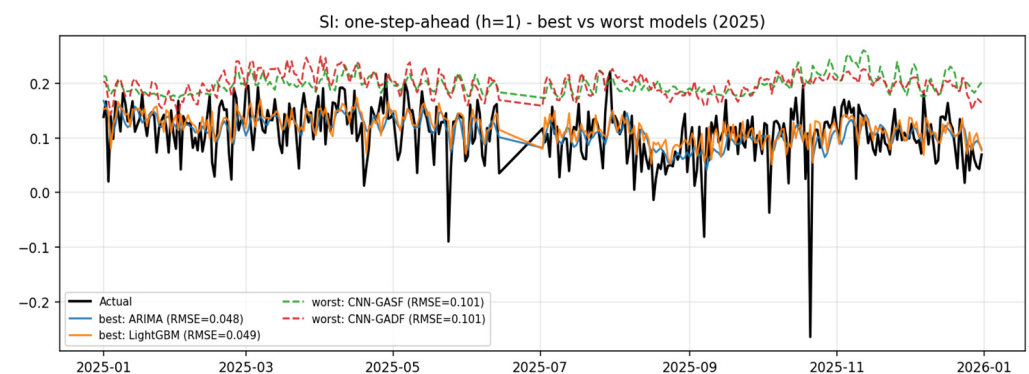
best models failing to anticipate the largest positive and negative deviations from trend. This limitation is particularly evident for the Negative Sentiment series, where forecast errors are concentrated around a small number of extreme observations. The image-based CNN-GASF and CNN-GADF models perform substantially worse, exhibiting persistent bias and oversmoothing across all three series.



**Figure 4.** One-step-ahead out-of-sample forecasts of Positive Sentiment over the 2025 test period.



**Figure 5.** One-step-ahead out-of-sample forecasts of Negative Sentiment over the 2025 test period; the discrete sentiment shocks are the main source of forecast error.



**Figure 6.** One-step-ahead out-of-sample forecasts of the Sentiment Index over the 2025 test period.

Increasing model complexity does not translate into superior accuracy. The Diebold–Mariano tests (Table 9) show that ARIMA is statistically indistinguishable from ETS, LightGBM, and TCN for the Sentiment Index, while it significantly outperforms the GAN, ARIMA-GARCH, LSTM, and 1D CNN, and outperforms both image-based encodings ( $DM > 16, p < 0.001$ ).

**Table 9.** Diebold–Mariano tests for the Sentiment Index at  $h = 1$  (squared-error loss) against the most accurate model, ARIMA.

| Comparison (vs. ARIMA) | DM Statistic | <i>p</i> -Value | Conclusion                |
|------------------------|--------------|-----------------|---------------------------|
| ETS                    | 0.746        | 0.456           | No significant difference |
| LightGBM               | 0.955        | 0.340           | No significant difference |
| TCN                    | 1.749        | 0.081           | No significant difference |
| LSTM                   | 2.999        | 0.003           | ARIMA more accurate       |
| 1D CNN                 | 3.204        | 0.002           | ARIMA more accurate       |
| ARIMA–GARCH            | 3.390        | 0.001           | ARIMA more accurate       |
| GAN                    | 3.718        | 0.000           | ARIMA more accurate       |
| CNN–GASF               | 16.453       | 0.000           | ARIMA more accurate       |
| CNN–GADF               | 16.794       | 0.000           | ARIMA more accurate       |

Both image-based models (CNN–GASF and CNN–GADF) performed far worse than every sequential model across all series and horizons. Transforming a noisy, weakly autocorrelated sentiment series into image space discards level information (GADF) or blurs local direction (GASF) while amplifying high-frequency noise, so direct sequential modeling is preferable for AI sentiment forecasting.

Table 10 compares forecasting accuracy across models, sentiment measures, and forecast horizons. Forecast errors are consistently lower for the aggregate Sentiment Index than for the Positive and Negative Sentiment series, indicating that the composite index is more stable and predictable. Across models, ARIMA, ETS, LightGBM, and LSTM generally achieve the highest forecasting accuracy, whereas the image-based CNN–GASF and CNN–GADF models consistently perform the worst. Although forecast errors increase with the prediction horizon, model rankings remain broadly unchanged, with the best-performing models showing only modest deterioration and more complex architectures (particularly ARIMA–GARCH, GAN, and TCN) exhibiting larger declines in accuracy during recursive forecasting.

**Table 10.** Multi-horizon forecasting accuracy: out-of-sample (test-set) RMSE at horizons  $h = 1, 7$ , and 30 days and over the full recursive forecast of the entire 348-day test period.

| Model       | Series          | $h = 1$ | $h = 7$ | $h = 30$ | Full (348-Step) |
|-------------|-----------------|---------|---------|----------|-----------------|
| ARIMA       | Negative        | 0.2401  | 0.2474  | 0.2543   | 0.2689          |
|             | Positive        | 0.1891  | 0.1903  | 0.1978   | 0.2639          |
|             | Sentiment Index | 0.0479  | 0.0494  | 0.0524   | 0.0590          |
| ARIMA–GARCH | Negative        | 0.2620  | 0.2783  | 0.3303   | 0.8614          |
|             | Positive        | 0.2057  | 0.2129  | 0.2338   | 0.2786          |
|             | Sentiment Index | 0.0543  | 0.0559  | 0.0681   | 0.2534          |
| ETS         | Negative        | 0.2464  | 0.2560  | 0.2643   | 0.2687          |
|             | Positive        | 0.1898  | 0.1935  | 0.2038   | 0.2408          |
|             | Sentiment Index | 0.0487  | 0.0504  | 0.0537   | 0.0569          |
| LSTM        | Negative        | 0.2500  | 0.2477  | 0.2569   | 0.2539          |
|             | Positive        | 0.1853  | 0.1853  | 0.1882   | 0.1887          |
|             | Sentiment Index | 0.0528  | 0.0613  | 0.0794   | 0.0958          |
| 1D CNN      | Negative        | 0.2560  | 0.2564  | 0.2741   | 0.2789          |
|             | Positive        | 0.1941  | 0.1952  | 0.1919   | 0.1929          |
|             | Sentiment Index | 0.0530  | 0.0632  | 0.0889   | 0.0905          |

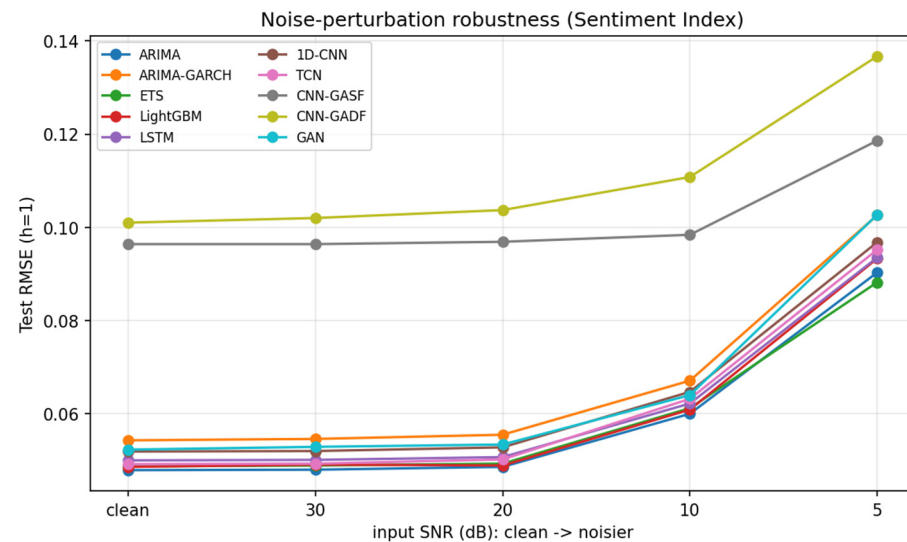
Table 10. Cont.

| Model    | Series          | h = 1  | h = 7  | h = 30 | Full (348-Step) |
|----------|-----------------|--------|--------|--------|-----------------|
| TCN      | Negative        | 0.2783 | 0.3497 | 0.4975 | 0.6231          |
|          | Positive        | 0.1923 | 0.1935 | 0.2039 | 0.3621          |
|          | Sentiment Index | 0.0497 | 0.0520 | 0.0589 | 0.0576          |
| LightGBM | Negative        | 0.2398 | 0.2465 | 0.2684 | 0.2556          |
|          | Positive        | 0.1921 | 0.1928 | 0.1955 | 0.2238          |
|          | Sentiment Index | 0.0486 | 0.0503 | 0.0558 | 0.0711          |
| GAN      | Negative        | 0.2873 | 0.2971 | 0.4948 | 3.2516          |
|          | Positive        | 0.1965 | 0.1957 | 0.1939 | 0.1953          |
|          | Sentiment Index | 0.0534 | 0.0544 | 0.0603 | 0.1071          |
| CNN-GASF | Negative        | 0.7778 | 0.8260 | 0.7328 | 0.7773          |
|          | Positive        | 0.3097 | 0.3116 | 4.0463 | 3.7197          |
|          | Sentiment Index | 0.1010 | 0.1165 | 0.1207 | 0.0927          |
| CNN-GADF | Negative        | 0.6579 | 0.6596 | 0.6706 | 0.6579          |
|          | Positive        | 0.3319 | 0.3364 | 0.3152 | 0.3282          |
|          | Sentiment Index | 0.1010 | 0.1016 | 0.0878 | 0.0965          |

Table 11 evaluates forecasting robustness under progressively increasing additive Gaussian noise. As expected, forecasting accuracy deteriorates for all models as the signal-to-noise ratio decreases. Nevertheless, the relative ranking of the models remains largely unchanged: ARIMA, ETS, LightGBM, and LSTM consistently achieve the lowest RMSE values, while the image-based CNN-GASF and CNN-GADF models remain the least accurate (Figure 7). The persistence of these rankings under noisy conditions indicates that the superiority of the best-performing models is robust and not driven by idiosyncratic characteristics of the original series, further supporting the suitability of simpler forecasting approaches for AI news sentiment.

Table 11. Sentiment-Index RMSE under additive Gaussian noise at decreasing signal-to-noise ratios (dB).

| Model       | SNR (dB) |        |        |        |        |
|-------------|----------|--------|--------|--------|--------|
|             | Clean    | 30     | 20     | 10     | 5      |
| ARIMA       | 0.0479   | 0.0480 | 0.0486 | 0.0600 | 0.0903 |
| ARIMA-GARCH | 0.0543   | 0.0546 | 0.0555 | 0.0671 | 0.1026 |
| ETS         | 0.0487   | 0.0489 | 0.0493 | 0.0612 | 0.0882 |
| LightGBM    | 0.0486   | 0.0491 | 0.0489 | 0.0609 | 0.0933 |
| LSTM        | 0.0500   | 0.0501 | 0.0507 | 0.0622 | 0.0935 |
| 1D CNN      | 0.0519   | 0.0520 | 0.0528 | 0.0647 | 0.0968 |
| TCN         | 0.0492   | 0.0493 | 0.0502 | 0.0632 | 0.0952 |
| GAN         | 0.0523   | 0.0529 | 0.0534 | 0.0640 | 0.1027 |
| CNN-GASF    | 0.0964   | 0.0964 | 0.0969 | 0.0984 | 0.1186 |
| CNN-GADF    | 0.1010   | 0.1020 | 0.1037 | 0.1108 | 0.1367 |



**Figure 7.** Sentiment-Index RMSE as a function of signal-to-noise ratio.

### Computational Cost

Table 12 compares the computational cost of the forecasting models. The statistical benchmarks are the most computationally efficient, with ETS (10 parameters) training in less than 0.25 s and ARIMA (6 parameters) in under 4 s, while both also exhibit negligible inference time. LightGBM remains similarly efficient despite its larger number of parameters. In contrast, the deep learning and image-based models require substantially greater computational resources, with model sizes ranging from 7505 parameters for LSTM to nearly two million for CNN-GADF, resulting in considerably longer training and inference times.

**Table 12.** Computational cost: trainable parameters, training time, and inference latency per forecast step.

| Model    | Parameters | Train Time (s) | Inference (ms/Step) |
|----------|------------|----------------|---------------------|
| ETS      | 10         | 0.25           | 0.46                |
| ARIMA    | 6          | 3.58           | 1.32                |
| LightGBM | 400        | 0.45           | 0.82                |
| GAN      | 8257       | 31.36          | 5.24                |
| LSTM     | 7505       | 13.73          | 16.77               |
| 1D CNN   | 102,117    | 5.97           | 9.54                |
| TCN      | 21,953     | 21.66          | 34.22               |
| CNN-GASF | 470,497    | 10.15          | 9.48                |
| CNN-GADF | 1,992,193  | 28.81          | 10.02               |

However, this additional complexity does not translate into superior forecasting performance. As shown in Tables 10 and 11, simpler models such as ARIMA, ETS, and LightGBM consistently match or outperform the more complex architectures, indicating that parsimonious models provide the best trade-off between forecasting accuracy and computational efficiency for AI-related sentiment series.

Validation RMSE for the best model trained with the three hyperparameters search techniques, namely grid search, random search and TPE, is illustrated in Table 13. The results are highly consistent across search methods, with only minor differences in validation performance for each model. For instance, the best RMSE for the LSTM model ranges between 0.0501 and 0.0511, while that of the TCN is between 0.0486 and 0.0491.

**Table 13.** Best validation RMSE for the Sentiment Index by hyperparameter-search strategy.

| Model (SI) | Grid   | Random | Bayesian (TPE) |
|------------|--------|--------|----------------|
| LSTM       | 0.0509 | 0.0501 | 0.0511         |
| 1D CNN     | 0.0514 | 0.0514 | 0.0510         |
| TCN        | 0.0486 | 0.0491 | 0.0491         |
| LightGBM   | 0.0520 | 0.0529 | 0.0520         |
| GAN        | 0.0498 | 0.0490 | 0.0491         |
| CNN–GASF   | 0.0934 | 0.0950 | 0.0903         |
| CNN–GADF   | 0.0943 | 0.0947 | 0.0951         |

The order of performance of the model classes is also maintained throughout the different search techniques. TCN, GAN, LSTM, and 1D CNN have similar validation errors, irrespective of the search technique used, while the image-based CNN–GASF and CNN–GADF models continue to exhibit the highest validation errors. Similarly, no search strategy yields a meaningful improvement in the performance of the image-based approaches.

## 5. Practical Implications

The findings carry several practical implications. Financial analysts and quantitative researchers can make an accurate and efficient prediction of daily AI sentiment through the use of parsimonious models like ARIMA and LightGBM, both of which need minimal computational power. The findings indicate that adding a considerable amount of complexity to the structure does not result in improvements in forecasting accuracy.

In the case of policymakers and researchers engaged in AI governance studies, the sentiment index can serve as an easily replicable indicator of the AI-related media sentiment that can be calculated frequently. While it is still challenging to predict the most significant sentiment shocks, strong declines in the index often occur during times of increased regulation and ethical discussion on AI issues.

The results emphasize the necessity to compare advanced deep-learning algorithms with statistically specified baselines. Under certain conditions, such as relatively short and noisy sentiment data sets, simpler models can outperform significantly more complex models in terms of forecasting accuracy and require far less computational effort.

## 6. Limitations

Several limitations should be acknowledged. First, the GDELT V2Tone scores are generated by a general-purpose sentiment analysis system rather than by a model specifically designed for AI-related content. The use of a domain-specific sentiment model could affect both the level and the predictability of the resulting sentiment index. Second, the analysis adopts a univariate framework and does not incorporate potentially informative covariates such as AI-related equity prices, trading volumes, or major regulatory events. Third, the sentiment index is constructed using simple daily averages and does not account for intraday variation, article-level heterogeneity, or alternative weighting schemes. Finally, the image-based approaches are restricted to GASF and GADF representations implemented using a standard 2D CNN architecture and fixed window lengths. Alternative image encodings, network architectures, or tuning strategies may yield different results.

## 7. Future Work

Future research could develop along several directions. First, the generic GDELT sentiment measures could be replaced with indicators derived from AI-specific language models. Second, the sentiment index could be incorporated into multivariate forecasting frameworks that include AI-related equity prices, trading volumes, and event indicators. Third,

future work could focus on modeling the abrupt sentiment shocks that characterize the Negative Sentiment series through regime-switching, intervention, or structural-break models. Fourth, additional deep-learning architectures, including Transformer- and Informer-based models, could be evaluated. Finally, future studies may explore alternative image and multimodal representations, such as Markov Transition Fields and Recurrence Plots, to assess whether different encoding approaches can improve forecasting performance for AI-related sentiment series.

## 8. Conclusions

This study examined the forecasting performance of classical time-series methods and deep-learning models for daily measures of AI-related sentiment, including Positive Sentiment, Negative Sentiment, and a Sentiment Index. The results indicate that all three series are characterized by asymmetry, volatility, and occasional extreme observations, particularly in the case of Negative Sentiment.

Among the classical approaches, ARIMA provides a strong benchmark, while the GARCH extension offers limited additional predictive value at short horizons. ETS also remains highly competitive, suggesting that level and trend dynamics are more important than conditional variance effects for these sentiment series.

Across models, greater complexity does not translate into systematically higher forecasting accuracy. The best-performing parsimonious models achieve forecasting performance comparable to and in many cases, better than, substantially more complex architectures. In particular, the image-based CNN–GASF and CNN–GADF approaches perform consistently worse across sentiment measures and forecast horizons despite their substantially higher computational cost. One possible explanation is that the high volatility and episodic shocks characterizing these sentiment series limit the ability of image-based representations to extract stable predictive patterns. These findings contribute to a better understanding of the conditions under which image-based forecasting methods may be effective.

Visual inspection of the forecasts further indicates that the best-performing models are generally able to reproduce the overall level and short-term movements of the sentiment series. However, none of the models successfully anticipate the largest sentiment shocks, with forecast errors concentrated around periods of unusually large positive or negative sentiment changes. This suggests that the principal forecasting challenge lies not in modeling the underlying sentiment dynamics, but in anticipating rare and abrupt shifts in media attention and tone.

Overall, the results indicate that AI-related sentiment can be monitored and forecast effectively using relatively simple forecasting models. Additionally, the findings highlight the importance of evaluating model complexity against demonstrable improvements in predictive performance, particularly when working with short, noisy, and event-driven sentiment series.

Beyond its methodological contribution, the study also carries broader implications for stakeholders monitoring the socioeconomic effects of AI diffusion. Recent evidence shows that AI adoption is reshaping employment structures, skill requirements, and human resource practices, with heterogeneous outcomes that depend strongly on institutional and organizational contexts rather than on the technology alone [43]. In parallel, generative AI tools such as large language models are being progressively integrated into managerial and sustainability practices, including in small and medium-sized enterprises where they support strategic decision making, stakeholder communication, and the alignment of business goals with wider societal objectives [44]. In this environment, high-frequency AI news sentiment indices provide a timely, complementary signal that can help policymakers,

business managers, and investors interpret shifts in public perception and anticipate periods of increased attention to AI-related developments. Therefore, the forecasting framework proposed in this paper offers a practical instrument for tracking the evolving narrative surrounding AI, which itself is a key driver of adoption, regulation, and long-term labor market and organizational transformation.

**Author Contributions:** Conceptualization, G.-M.P., I.B., C.-R.B. and A.A.D.; methodology, G.-M.P., I.B., A.A.D. and C.-R.B.; software, G.-M.P. and I.B.; validation, G.-M.P., I.B., C.-R.B. and A.A.D.; formal analysis, G.-M.P. and I.B.; investigation, G.-M.P., I.B. and C.-R.B.; resources, G.-M.P., C.-R.B. and A.A.D.; data curation, G.-M.P. and I.B.; writing—original draft, G.-M.P. and I.B.; writing—review and editing, G.-M.P., I.B., C.-R.B. and A.A.D.; visualization, G.-M.P.; supervision, C.-R.B. and A.A.D.; project administration, C.-R.B. and A.A.D.; funding acquisition, C.-R.B. and A.A.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was carried out within the project “MA’AT: Modelul Autonom de Asistență Textuală”, project code 330941, funded through PCIDIF 2021–2027. This work was also funded by Fundația Comunitară București an ING Hubs Romania through the ING Hubs Romania Robotics and Programming Scholarship Fund (2025).

**Data Availability Statement:** The original contributions presented in this study are included in the article. The data used in this study were derived from the GDELT Global Knowledge Graph 2.1 database, accessed through Google BigQuery. Further inquiries can be directed to the corresponding author.

**Acknowledgments:** During the preparation of this work, the authors utilized ChatGPT 4 to enhance readability and language clarity. Following the use of this tool, the authors meticulously reviewed and revised the content as necessary and assume full responsibility for the final content of the publication.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Theofilou, A.; Nastis, S.A.; Mattas, K.; Theofilou, K. AI-Driven News-Enhanced Machine Learning for Short-Term Corn Futures Price Forecasting. *Appl. Sci.* **2026**, *16*, 1337. [\[CrossRef\]](#)
2. Lee, K.-J.; Kim, J.-M. Deep Learning Approaches for Forecasting the FOMC Statement Sentiment Index. *Commun. Stat. Appl. Methods* **2025**, *32*, 631–654. [\[CrossRef\]](#)
3. Leone, F.; Marazzina, D.; Rosamilia, N. What’s News with You: Price Forecasting with Global and ESG Sentiment Scores. *Financ. Res. Open* **2025**, *2*, 100013. [\[CrossRef\]](#)
4. Das, N.; Sadhukhan, B.; Ghosh, R.; Chakrabarti, S. Developing Hybrid Deep Learning Models for Stock Price Prediction Using Enhanced Twitter Sentiment Score and Technical Indicators. *Comput. Econ.* **2024**, *64*, 3407–3446. [\[CrossRef\]](#)
5. Sarıkoç, M.; Çelik, M. Convolutional Neural Network Analysis of the Gramian Angular Field for Oil Services Exchange Traded Fund (ETF) Prediction. *Int. J. Manag. Inf. Syst. Comput. Sci.* **2025**, *9*, 108–120. [\[CrossRef\]](#)
6. Chen, Q. Comparing Vision-Instruct LLMs, Vision-Based Deep Learning, and Numeric Models for Stock Movement Prediction. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*. [\[CrossRef\]](#)
7. Dahal, K.R.; Pokhrel, N.R.; Gaire, S.; Mahatara, S.; Joshi, R.P.; Gupta, A.; Banjade, H.R.; Joshi, J. A Comparative Study on Effect of News Sentiment on Stock Price Prediction with Deep Learning Architecture. *PLoS ONE* **2023**, *18*, e0284695. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Shahi, T.B.; Shrestha, A.; Neupane, A.; Guo, W. Stock Price Forecasting with Deep Learning: A Comparative Study. *Mathematics* **2020**, *8*, 1441. [\[CrossRef\]](#)
9. Gupta, B.B.; Gaurav, A.; Piñeiro-Chousa, J.; López-Cabarcos, M.Á.; González López, I. Predicting the Variation of Decentralised Finance Cryptocurrency Prices Using Deep Learning and a BiLSTM-LSTM Based Approach. *Enterp. Inf. Syst.* **2025**, *19*, 2483456. [\[CrossRef\]](#)
10. Kirtika, G.; Pandya, R.J.; Iyer, S. Deep Learning and Natural Language Processing Integrated Gold Price Forecasting. In Proceedings of the 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 6–8 March 2025; pp. 1–6. [\[CrossRef\]](#)
11. Wang, Z.; Oates, T. Imaging Time-Series to Improve Classification and Imputation. In Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina, 25–31 July 2015; pp. 3939–3945.

12. Teleron, J.I.; Gonzales, S.L.; Fajardo, A.C. A Modified Energy Demand Forecasting Model Using Hybrid CNN-LSTM with Transformer for Univariate Time Series. *Int. J. Comput. Exp. Sci. Eng.* **2025**, *11*, 4117–4125. [\[CrossRef\]](#)
13. Zhang, X.; Wang, Y.; Chen, L. A New Time Series Forecasting Model Based on Complete Ensemble Empirical Mode Decomposition with Adaptive Noise and Temporal Convolutional Network. *Complexity* **2022**, *2022*, 9542464.
14. Kim, Y.; Yoon, S.; Shin, J.; Lee, Y. Patch-Level Contrastive Learning for Improved Time Series Classification with Gramian Angular Field. *ACM Trans. Intell. Syst. Technol.* **2025**, *17*, 1–21. [\[CrossRef\]](#)
15. Zhou, Y.; Long, X.; Sun, M.; Chen, Z. Bearing Fault Diagnosis Based on Gramian Angular Field and DenseNet. *Math. Biosci. Eng.* **2022**, *19*, 14086–14101. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Long, H.-W.; Ho, O.-I.; He, Q.-Q.; Si, Y.-W. Transfer Learning in Financial Time Series with Gramian Angular Field (Student Abstract). *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 29418–29420. [\[CrossRef\]](#)
17. Al Joumaa, H.; Al-Shrouf, L.; Jelali, M. Time Series-to-Image Encoding for Classification Using Convolutional Neural Networks: A Novel and Robust Approach. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 155. [\[CrossRef\]](#)
18. Vuletić, M.; Prenzel, F.; Cucuringu, M. Fin-GAN: Forecasting and Classifying Financial Time Series via Generative Adversarial Networks. *Quant. Financ.* **2024**, *24*, 175–199. [\[CrossRef\]](#)
19. Yoon, J.; Jarrett, D.; van der Schaar, M. Time-Series Generative Adversarial Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 32), Vancouver, BC, Canada, 8–14 December 2019; pp. 5508–5518.
20. Karbasian, M.; Ahangarzadeh, A.; Manshaei, M.H.; Zahabi, S.J. TF-GAN: Topology-Aware Generative Adversarial Network for Financial Time Series Forecasting. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF '25)*; ACM: New York, NY, USA, 2025; pp. 229–237. [\[CrossRef\]](#)
21. Siami-Namini, S.; Tavakoli, N.; Siami Namin, A. A Comparison of ARIMA and LSTM in Forecasting Time Series. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1394–1401. [\[CrossRef\]](#)
22. Tong, Z.; Li, H.; Zhang, Y. A Comparison of ARIMA, LSTM, XGBOOST and Hybrid in Stock Price Prediction. In Proceedings of the 2023 5th International Conference on Applied Machine Learning (ICAML), Dalian, China, 21–23 July 2023; pp. 18–25. [\[CrossRef\]](#)
23. Mochurad, L.; Dereviannyi, A. An Ensemble Approach Integrating LSTM and ARIMA Models for Enhanced Financial Market Predictions. *R. Soc. Open Sci.* **2024**, *11*, 240699. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
25. Salisu, A.; Salisu, S.; Salisu, S. A News-Based Economic Policy Uncertainty Index for Nigeria. *Qual. Quant.* **2024**, *58*, 4987–5002. [\[CrossRef\]](#)
26. Halousková, M.; Lyócsa Š Forecasting, U.S. Equity Market Volatility with Attention and Sentiment to the Economy. *arXiv* **2025**, arXiv:2503.19767.
27. Halder, S. FinBERT-LSTM: Deep Learning Based Stock Price Prediction Using News Sentiment Analysis. *arXiv* **2022**, arXiv:2211.07392.
28. Jain, A.; Ranganathan, S. A Longitudinal Analysis of Artificial Intelligence Coverage in Technology-Focused News Media Using Latent Dirichlet Allocation and Sentiment Analysis. *Journal. Media* **2025**, *6*, 176. [\[CrossRef\]](#)
29. Diouf, L.; Schmidt, M.; Kowalski, A. How ChatGPT Changed the Media's Narratives on AI: A Semi-Automated Narrative Analysis Through Frame Semantics. *arXiv* **2024**, arXiv:2408.06120.
30. Bermejo, S.R.; Martinez, C.A.; Gonzalez, F.L. Mapping Public Perception of Artificial Intelligence: Expectations, Risk–Benefit Tradeoffs, and Value as Determinants for Societal Acceptance. *arXiv* **2024**, arXiv:2411.19356.
31. Hyndman, R.J.; Koehler, A.B.; Ord, J.K.; Snyder, R.D. *Forecasting with Exponential Smoothing: The State Space Approach*; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2008.
32. The GDELT Project. GDELT 2.0 Global Knowledge Graph (GKG) Data Format Codebook, Version 2.1. 2015. Available online: [http://data.gdeltproject.org/documentation/GDELT-Global\\_Knowledge\\_Graph\\_Codebook-V2.1.pdf](http://data.gdeltproject.org/documentation/GDELT-Global_Knowledge_Graph_Codebook-V2.1.pdf) (accessed on 27 January 2026).
33. Leetaru, K.; Schrod, P.A. GDELT: Global Data on Events, Location, and Tone, 1979–2012. In Proceedings of the International Studies Association Annual Convention, San Francisco, CA, USA, 3–6 April 2013.
34. Shen, S.; Xia, L.; Shuai, Y.; Gao, D. Measuring news media sentiment using big data for Chinese stock markets. *Pac.-Basin Financ. J.* **2022**, *74*, 101810. [\[CrossRef\]](#)
35. Eklund, J.; Kim, J.M. Forecasting consumer price index with federal open market committee sentiment index. *J. Forecast.* **2024**, *43*, 1795–1813. [\[CrossRef\]](#)
36. Ye, Z.; Wu, Y.; Chen, H.; Pan, Y.; Jiang, Q. A stacking ensemble deep learning model for bitcoin price prediction using Twitter comments on bitcoin. *Mathematics* **2022**, *10*, 1307. [\[CrossRef\]](#)
37. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [\[CrossRef\]](#)

38. Glosten, L.R.; Jagannathan, R.; Runkle, D.E. On the relation between the expected value and the volatility of the nominal excess return on stocks. *J. Financ.* **1993**, *48*, 1779–1801. [[CrossRef](#)]
39. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
40. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems 24*; Curran Associates: Red Hook, NY, USA, 2011; pp. 2546–2554.
41. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019*; ACM: New York, NY, USA, 2019; pp. 2623–2631. [[CrossRef](#)]
42. Diebold, F.X.; Mariano, R.S. Comparing predictive accuracy. *J. Bus. Econ. Stat.* **1995**, *13*, 253–263. [[CrossRef](#)]
43. Doubková, M.; Magdin, M. The Impact of AI on the Labour Market. *Manag. Mark.* **2026**, *21*, 2. [[CrossRef](#)]
44. Fonseca, L.; Oliveira, E.; Pereira, T.; Sá, J.C. Leveraging ChatGPT for Sustainability: A Framework for SMEs to Align with UN Sustainable Development Goals and Tackle Sustainable Development Challenges. *Manag. Mark.* **2024**, *19*, 471–497. [[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.