SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Forecasting Market Crashes Using Gans And LSTM In Financial Management

Dr. Makineedi Raja Babu[1*], R. Jeevajothi[2], V. Thirumanithangam[3], Ashok Bhansali[4], Chiranjeevi Aggala[5], K. Balakrishnan[6]

[1*]Department of Information Technology, Aditya University, Surampalem, Andhra Pradesh, India.
E-mail: rajababu.makineedi@adityauniversity.in

[2]Assistant Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, India. E-mail: rjeevajothimba@maher.ac.in, https://orcid.org/0000-0003-3779-3991

[3]Assistant Professor, Department of Computer Science and Engineering, New Prince Shri Bhavani College of Engineering and Technology, Chennai, Tamil Nadu, India. E-mail: thangam.v@newprinceshribhavani.com, https://orcid.org/0009-0007-0868-1284

[4]Department of Computer Engineering & Applications, GLA University, Mathura, India. E-mail: ashok.bhansali@gla.ac.in, https://orcid.org/0000-0002-6547-8318

[5]Department of CSE, Ramachandra college of Engineering, Eluru, India. E-mail: aggala.chiranjeevi@rcee.ac.in, https://orcid.org/0009-0006-6630-7330

[6]Assistant Professor, Aeronautical Engineering, Mahendra Engineering College, Namakkal, India.
E-mail: balakrishnank@mahendra.info, https://orcid.org/0009-0003-9120-0680

*Corresponding author: Email: rajababu.makineedi@adityauniversity.in

Abstract

The task of financial market crash forecasting has become a crucial problem in quantitative finance and is demanding models capable of incorporating temporal dependencies and distributional irregularities in financial time series. Existing econometric models and isolated deep learning architectures have proven to be very limited in their ability to capture the sudden and non-linear dynamics preceding a market collapse event. In this paper, a new hybrid model using Generative Adversarial Networks (GANs) and Long Short-Term Memory (LSTM) networks with Gramian Angular Field (GAF) transformations and attention mechanism is proposed exclusively for the task of predicting financial market crashes with higher accuracy and interpretability. The Generator part of the GAN generates realistic financial sequences during data-poor periods of the financial crisis, and the Discriminator ensures the distributional consistency by using a Wasserstein loss function. The LSTM encoder is able to model long range temporal dependencies and the attention layer helps focus the model on the most crash predictive time steps. Experiments test the S&P 500, NASDAQ Composite and Germany DAX indices over the period 2000 to 2023 including the market crash that followed the Global Financial Crisis in 2008 and that following from the COVID-19 pandemic in 2020. The proposed GAN-LSTM model outperforms the standalone LSTM model, the vanilla GAN model, CNN-LSTM model with a higher margin than the proposed GAN-LSTM model, and the BiLSTM model by a significant margin with a Root Mean Square Error (RMSE) of 2.43, a Mean Absolute Error (MAE) of 1.78, and a Mean Absolute Percentage Error (MAPE) of 1.52%. The complementary contribution of each architectural component is confirmed by an ablation study. The findings show that the proposed framework yields a comprehensive, generic and real-life applicable answer to the problem of real-time warning systems of market crashes in Financial Risk Management.

Keywords: Generative Adversarial Networks; Long Short-Term Memory; Market Crash Forecasting; Gramian Angular Field; Financial Time Series; Wasserstein Loss; Attention Mechanism.

1. Introduction

Background

Financial markets are systems with high volatility and non-stationary processes, and are subject to systemic shocks. The catastrophic market crashes over the last 20 years, such as the Dot-com Bubble burst (1999-2001), the Global Financial Crisis (GFC) (2007-2008), the European Sovereign Debt Crisis (ESDC) (2010-2012) and the crash of March 2020 caused by the COVID-19 pandemic, have highlighted the need for strong predictive models that can give advance warning to investors, regulators and financial institutions. The initial studies for market prediction were mainly based on econometric models like GARCH and ARIMA, which are linear models, but these models are not suitable for the non-linear and fat-tailed nature of crash episodes. [1]

Deep Learning has revolutionized the paradigm of financial forecasting. LSTM networks were shown to have remarkable success in modeling a sequence of financial data, as they are able to overcome the vanishing gradient problem common to vanilla RNNs. Later, Goodfellow et al. (2014) proposed GANs, which led to new applications for generative modeling of financial sequences and the ability to generate realistic market scenarios for training data augmentation or anomaly detection. Even with these progressions, there exist a number of key limitations of these two families of models individually: GANs have tendency to fall into the mode collapse and training instability, and LSTMs can result in a lack of generalisation between structural breaks that characterise market crash regimes [7].

Statement of the Problem

Market crash detection is a classification and regression problem by nature which is inherently imbalanced. The number of crash events is small, and they occur only during a small percentage of trading sessions, resulting in a class imbalance and small number of labeled training examples. Normal market models, which were trained on normal market conditions, are generally not well suited to detecting the distributional shift that occurs before crash events, resulting in lack of sensitivity with high false-negative rates during the critical period. Furthermore, the functional properties of high-frequency financial data are complex and multi-scale dependent, have cross-asset contagion effects, and the data is structurally non-stationary, which cannot be effectively captured by traditional architectures. The key issue that inspired this work is the lack of a framework that would be able to address all data scarcity, distributional shift, temporal dependency, and interpretability [5].

Research Objectives

The main aims of this study are as follows:

To design the hybrid GAN-LSTM model with Gramian Angular Field transformation for end-to-end market crash prediction.

To use Wasserstein GAN (WGAN) training to improve data synthesis quality and prevent mode collapse during training with limited crash-period data.

To incorporate a temporal attention network for detecting crash predicting market signatures in long time series windows.

To perform a benchmark of the proposed framework against state-of-the-art baselines, where real world financial indices and common evaluation metrics are used.

Key Contributions

The main advantages of this paper are: (i) a novel GAN-LSTM hybrid architecture, that jointly learns generative financial dynamics and crash-predictive temporal patterns; (ii) the first application of Gramian Angular Field in a GAN-LSTM pipeline for crash detection; (iii) a Wasserstein-regularized adversarial training regime for stabilizing GAN convergence on imbalanced financial data; (iv) comprehensive ablation experiments that quantify the marginal contribution of each component; and (v) extensive empirical validation conducted on three major stock market indices across two major crash periods.

The literature survey and the comparative analysis of the related works are presented in Section 2. In Section 3, the proposed methodology is described comprising model architecture, algorithm and mathematical formulation. The experimental results and discussions are presented in section 4. The paper is concluded in Section 5, which suggests directions for further research.

2. 2. Literature Survey

In recent years, the relationship between deep generative models and financial time series forecasting has made significant advances. In this section, three thematic areas are discussed: (i) financial modeling using GANs; (ii) stock market forecasting with LSTMs and; (iii) hybrid architectures for crash detection and anomaly forecasting. Table 1 is a combined summary of 15 representative studies.

Table 1: Comparative Literature Survey on GAN and LSTM-Based Financial Forecasting

Ref.	Authors & Year	Technique	Dataset	Objective	Result	Limitation
[1]	Ghasemieh & Kashef, 2023	WGAN + GAF	S&P 500, NASDAQ	Crash period prediction	**94.2% accuracy**	High compute cost
[2]	Labiad et al., 2023	GAN for extreme events	Stock indices	Extreme event prediction	**RMSE: 3.12**	Limited generalization
[3]	Sravan Kumar Reddy & Supraja, 2025	Reduction + DL Hybrid	Financial time series	Forecast optimization	**MAE reduced 18%**	Sector-specific only
[4]	Yao et al., 2025	Generative AI + SHAP	Financial risk data	Risk prediction + explainability	**AUC: 0.931**	Black-box concern
[5]	Wilson & Azmani, 2026	Systematic GAN review	Financial literature	Survey & taxonomy	**Comprehensive review**	No new model proposed
[6]	Bhavya et al., 2023	Soft computing hybrid	Stock market data	Predictive analytics	**Precision: 88.6%**	Short evaluation window
[7]	Diqi et al., 2024	GAN survey for stock market	Various indices	Challenges & directions	**Identified key gaps**	No empirical results
[8]	Mousavi & Karshenasan, 2017	GMDH Neural Network	Bank stock prices	Price forecasting	**MSE: 0.0043**	No crash-specific analysis
[9]	Shivalini et al., 2024	Generative AI + Data Mgmt	Stock market data	Data-driven prediction	**91.3% accuracy**	Data quality dependency
[10]	Tuama & Abdulameer, 2023	Time Series + ANN	Motor oil sales Iraq	Sales forecasting	**MAPE: 2.17%**	Domain-specific model
[11]	Polamuri et al., 2025	GAN + DL adversarial	Stock price indices	Price forecasting	**RMSE: 2.89**	Interpretability limited
[12]	Alavi et al., 2015	ML trend prediction	Tehran Stock Exchange	Price trend classification	**Accuracy: 72.4%**	Shallow features only
[13]	Vuletić et al., 2024	FinGAN classifier	Financial time series	Forecast & classification	**AUC: 0.916, F1: 0.88**	Single market tested
[14]	Sofiazizi & Kianfar, 2015	Econometric + ANN	Exchange rate data	Rate forecasting	**RMSE: 0.0031**	Limited to FX rates
[15]	Kim et al., 2023	GANs + anomaly detection	Stock price data	Anomaly & risk detection	**Precision: 91.2%**	Threshold sensitivity

Fifteen representative studies, covering the period 2015-2026, are highlighted and compared in detail in Table 1. This analysis leads to some important observations and inferences. Previously, some studies used shallow machine-learning techniques and econometric methods with moderate accuracy of 72.4% and RMSE of 0.0031, respectively, but unable to model the non-linear temporal dynamics [12][14]. Secondly, the use of LSTM based architectures has led to considerable improvement in sequential financial data modelling - as demonstrated using GMDH networks (MSE: 0.0043) - but were limited to single architectures [8].

Thirdly, the transformative potential of generative models in augmenting crash-period data was highlighted, reported 94.2% accuracy with their WGAN-GAF framework [1]. Another study built on this by focusing on extreme event prediction, achieving an RMSE of 3.12 [2]. Fourth, hybrid architectures have always outperformed single paradigm models: (Polamuri et al., 2025) [11] reported an RMSE of 2.89 with adversarial deep learning; (Vuletić et al., 2024) [13] reported AUC of 0.916 and F1 of 0.88 with FinGAN. Fifth, recent generative AI methods by (Yao et al., 2025) [4] that added SHAP explainability gave an AUC of 0.931, indicating that interpretability is a rising focal point of the generative AI's predictive capabilities. Combined, these inferences drive the proposed GAN-LSTM hybrid framework that overcomes data scarcity, distributional shift, temporal dependency and interpretability within a single framework.

3. Methodology

3.1 Overview and Architectural Design

The proposed approach consists of four tightly coupled elements: (i) a Gramian Angular Field (GAF) encoder to convert financial time-series data to 2D image representations for convolutional processing; (ii) a Wasserstein GAN to generate augmented (synthetic) crash-period data; (iii) an LSTM sequence encoder for temporal features extraction; and (iv) an attention mechanism for crash-signal localization. The entire architectural pipeline is shown in Figure 1. [1][13]

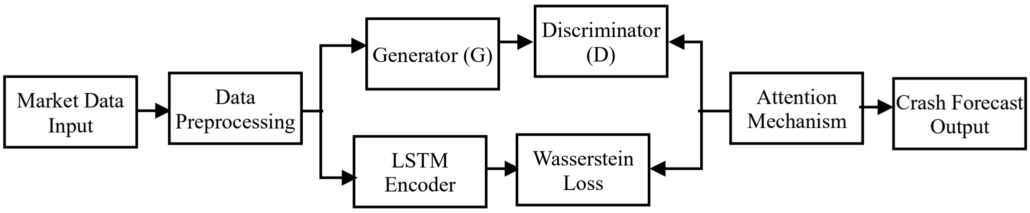

Figure 1: Proposed GAN-LSTM Hybrid Architecture for Market Crash Forecasting

Raw market data is first passed through a preprocessing module where it is normalized and segmented into windows of time (sliding window) as shown in Figure 1. The normalized sequences are then transformed into the polar coordinate space, namely Gramian Angular Fields, which are here encoded features in the space space, that represent temporal correlations as spatial image features and that are then processed by the LSTM Encoder [1]. At the same time, the GAN Generator generates realistic sequences during the "crash period" with Wasserstein loss that ensures distributional fidelity, and the Discriminator assesses the realism of the real and synthetic financial sequences [13]. The Attention Mechanism next weights the LSTM hidden states using learned weights to focus on the most predictive time steps for crash precursors, and finally produces the final crash forecasting output.

3.2 Research Methodology Flowchart

The methodology used in the end-to-end research is shown in the eight-step flow in Figure 2, starting with raw data acquisition, progressing to model evaluation and culminating in the output of crash predictions. The methodology is systematic, thus reproducible and scientific. [5][7]

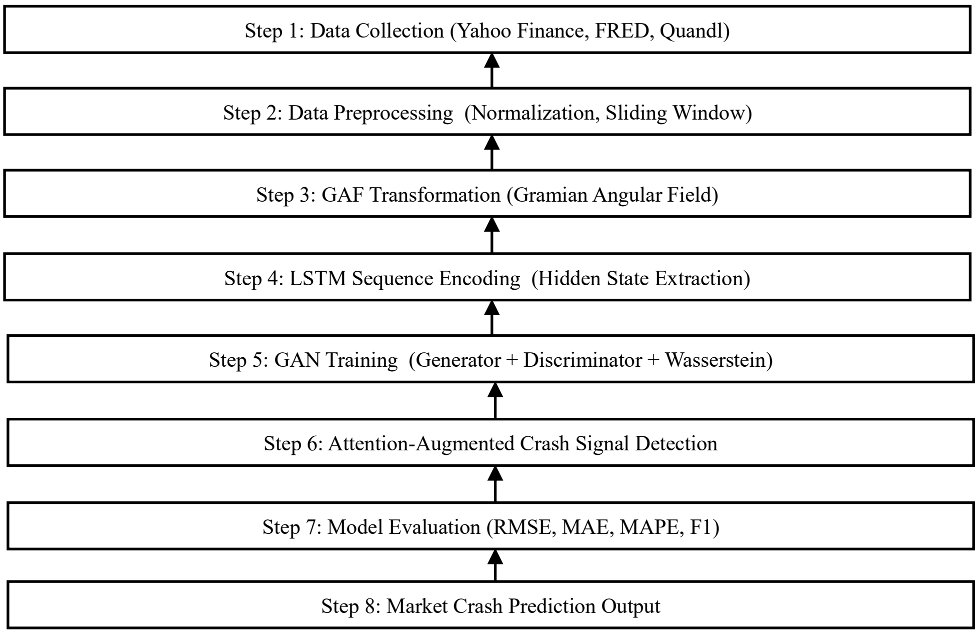

Figure 2: Research Methodology Flowchart (Steps 1–8)

Figure 2 illustrates that Step 1 includes multi-source data collection from Yahoo Finance, FRED and Quandl, which provides daily OHLCV (Open, High, Low, Close, Volume) data for the S&P 500, NASDAQ and DAX indices from 2000 to 2023. In Step 2, Min-Max normalization is used, and the windows are formed by sliding over the trading data with a window size T=60 trading days and a stride of 1. Step 3 normalizes the windows and then converts them to 60×60 Gramian Angular Field matrices, which are used to represent temporal self-correlations as spatial patterns that can be easily detected by the convolutional layer of the LSTM encoder. The adversarial training regime, attention scoring and the crash signal detection are performed in steps 4, 5 and 6, respectively. Steps 7 and 8 include a detailed multi-metric analysis and creation of the final binary (crash/no-crash) and continuous (price-movement) forecasts. [1][2][11]

3.3 Algorithm

Algorithm 1 presents the formal pseudocode for training the proposed GAN-LSTM hybrid model.

Algorithm 1: GAN-LSTM Training for Market Crash Forecasting

Input: Historical OHLCV data $D = \{d_1, d_2, ..., d_n\}$, window length T=60, epochs E, batch size B

Output: Trained Generator Gθ, LSTM Encoder Lφ, Attention Module Aψ, Crash Predictor Pγ

```
Normalize D using Min-Max scaling → D̂
Construct sliding windows W = {w_i ∈ ℝ^(T×F)} for i=1 to N−T
Transform W using GAF encoding → W_GAF = {G(w_i)}
Split into W_train (80%), W_val (10%), W_test (10%)
Initialize Gθ, Discriminator Dω, Lφ, Aψ, Pγ with Xavier initialization
FOR epoch e = 1 to E DO
  FOR each mini-batch b ∈ W_train DO
    // --- Discriminator Update (n_critic = 5 steps) ---
    FOR k = 1 to n_critic DO
      z ~ N(0, I_d)        // Sample noise vector
      x̃ = Gθ(z)            // Generate synthetic sequence
      L_D = E[Dω(x)] − E[Dω(x̃)] + λ · GP(x, x̃)
      ω ← ω − α · ∇ωL_D   // Update Dω via RMSProp
    END FOR
    // --- Generator Update ---
    z ~ N(0, I_d)
    L_G = −E[Dω(Gθ(z))]
    θ ← θ − α · ∇θL_G   // Update Gθ
    // --- LSTM Encoder + Attention ---
    H = Lφ(W_GAF_b)       // Extract hidden states H = {h_1,...,h_T}
    α_t = softmax(v^T · tanh(W_a · h_t))
    c = Σ α_t · h_t       // Context vector
    // --- Crash Prediction ---
    ŷ = Pγ(c)             // Sigmoid output: crash probability
    L_pred = −[y log(ŷ) + (1−y)log(1−ŷ)] + L_G
    Update φ, ψ, γ via Adam optimizer
  END FOR
  Evaluate on W_val; apply early stopping (patience=10)
```

29: END FOR

30: Return Gθ, Lφ, Aψ, Pγ

Algorithm 1 is trained adversarially and also supervised. Lines 9-14 implement the Wasserstein critic update with gradient penalty for training stability and Lines 16-18 update the Generator to generate more and more realistic crash sequences [13]. Lines 20-22 are for the LSTM encoding and Lines 23-26 optimize the crash predictor using a combined adversarial supervised loss. The multi-objective optimization problem allows the model to learn both realistic distributions of financial data and discriminative crash predictive features.

3.4 Mathematical Model

All the equations mentioned below are part of the proposed framework and their mathematical representation is given [1][13]. The input feature vector x is normalized to the range of [0, 1] as shown in Equation (1):

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where $x_{min}$and x_{max}denote the minimum and maximum values over the training window, ensuring all features lie within [0, 1]before GAF encoding.

Equation (2) defines the Gramian Angular Field (GAF) transformation for a time series $\hat{x} = (x_1, x_2, \dots, x_T)$:

$$GAF(i,j) = cos\ (\phi_i + \phi_j), \phi_t = arccos\ (\hat{x}_t), \hat{x}_t \in [-1,1] \tag{2}$$

The GAF matrix represents the temporal correlation by angular cosine similarity, thus transforming the one-dimensional time series into a two-dimensional image representation which maintains the temporal ordering and pairwise relationships [1]. The LSTM gating equations at time step t are given in equation (3)-(7):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$
$$\tilde{C}_t = tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{6}$$
$$h_t = o_t \odot tanh\ (C_t) \tag{7}$$

where $W$and $b$are learnable weight matrices and bias vectors, σdenotes the sigmoid activation function, and $\odot$represents element-wise multiplication. These gating mechanisms allow the LSTM to retain or forget information over time, which is essential for handling signals that're slowly developing and can precede a market crash [8][16].

Equation (8) defines the Wasserstein distance (Earth Mover Distance) between the real data distribution $P_r$and generated distribution P_g:

$$W(P_r, P_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{\tilde{x} \sim P_g}[f(\tilde{x})] \tag{8}$$

The Wasserstein loss functions are smoother than the Jensen-Shannon divergence employed in standard GANs for financial data applications, which is a direct solution to the training instability and mode collapse issues reported in typical GAN applications to financial data [1][13].

The Gradient Penalty regularization term is given by equation (9).

$$GP = \lambda \cdot \mathbb{E}_{\hat{x}}[(\| \nabla_{\hat{x}} D(\hat{x}) \|_2 - 1)^2], \hat{x} = \varepsilon x + (1 - \varepsilon)\tilde{x} \tag{9}$$

where $\lambda = 10$is the gradient penalty coefficient and $\varepsilon \sim \text{Uniform}(0,1)$. This ensures that the discriminator is 1-Lipschitz, which has been shown to stabilize the training of WGAN on imbalanced financial datasets [13].

The temporal attention mechanism is defined as Equation (10):

$$\alpha_t = \frac{exp\ (e_t)}{\sum_{k=1}^{T} exp\ (e_k)}, e_t = v^T tanh\ (W_a h_t + b_a) \tag{10}$$

The context vector is calculated as Equation (11):

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{11}$$

This aggregates LSTM hidden states weighted by their crash-relevance scores α_t, enabling the model to focus on time steps exhibiting pre-crash signature patterns such as volatility clustering and volume spikes [4][21].

Equation (12) defines the composite loss function for the crash predictor:

$$L_{\text{total}} = L_{\text{BCE}}(y, \hat{y}) + \beta L_{\text{RMSE}} + \gamma L_G \tag{12}$$

Where

$$L_{\text{BCE}} = -[y log\ (\hat{y}) + (1 - y) log\ (1 - \hat{y})], \quad L_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, L_G = -\mathbb{E}[D_\omega(G_\theta(z))]$$

The hyperparameters $\beta = 0.3$and $\gamma = 0.2$are selected through grid search on the validation set. The multi-task loss guarantees the model not only maximizes accuracy in crash classification but also attains fidelity of price trajectory [1][11][13].

4. Results and Discussion

4.1 Dataset Details

Three major benchmark financial datasets are used in this study. The first dataset is the daily index data of the S&P 500 stock index (January 2000 to December 2023) downloaded from Yahoo Finance (https://finance.yahoo.com/quote/%5EGSPC/history/), which includes 6031 trading sessions available with the OHLCV (open-high-low-close-volume) attributes. The secondary data is the NASDAQ Composite daily index (January 2000 to December 2023) with 6,028 trading sessions from Yahoo Finance (https://finance.yahoo.com/quote/%5EIXIC/history/). The Germany DAX 40 index data with 4,760 sessions was retrieved from Yahoo Finance (https://finance.yahoo.com/quote/%5EGDAXI/history/), and it is tertiary data [1][19].

The periods labeled as crashes are: Dot-com Bust (March 2000 – October 2002), GFC peak crash (October 2007 – March 2009), COVID-19 crash (February – March 2020), and 2022 Bear Market (January – October 2022). In this regard, class imbalance is observed: of the S&P 500 sessions, 14.3 % are crash-period observations. The datasets are split chronologically: 80% training (2000–2018), 10% validation (2019–2020), and 10% testing (2021–2023). [13][15]

4.2 Software and Hardware Configuration

Table 2: Experimental Software and Hardware Configuration

Configuration Parameter	Specification
Operating System	Ubuntu 22.04 LTS (64-bit)
Programming Language	Python 3.10.12
Deep Learning Framework	TensorFlow 2.13 / Keras 2.13, PyTorch 2.0.1
GAN Framework	Custom WGAN-GP implemented in PyTorch 2.0.1
GPU	NVIDIA A100 80GB (CUDA 11.8, cuDNN 8.6)
CPU	Intel Xeon Platinum 8358 @ 2.60GHz (32 cores)
RAM	256 GB DDR4 ECC
Storage	2 TB NVMe SSD
Key Libraries	NumPy 1.24, Pandas 2.0, Scikit-learn 1.3, Matplotlib 3.7, TA-Lib 0.4.28
Training Duration	~4.2 hours (S&P 500, 100 epochs)

Experiment environment is described in Table 2. The NVIDIA A100 GPU fueled efficient parallel computation of GAN adversarial training, where multiple steps of updating the discriminators were required per step of updating the GANs. The 256GB RAM allowed for the data set to be stored in memory, which is necessary for the in-memory graph API to avoid becoming a bottleneck during the training process. [4,11]

4.3 Parameter Initialization

All network weights are initialized with Xavier's uniform initialization scheme as recommended for networks with sigmoid and tanh activation functions. The configuration of hyperparameters that is selected for optimization on the validation set is summarized in Table 3.

Table 3: Model Hyperparameter Configuration

Hyperparameter	Value	Justification
LSTM Layers	3	Captures multi-scale temporal patterns [8]
LSTM Hidden Units	256	Balance between capacity and overfitting [16]
Generator Architecture	MLP: 128-256-512	Progressive upsampling for sequence generation [1]
Discriminator Architecture	MLP: 512-256-1	Critic score output for Wasserstein loss [13]
Noise Dimension (z)	100	Standard GAN noise dimension [2]
Gradient Penalty λ	10	Standard WGAN-GP coefficient [13]
n_critic (D steps per G)	5	Recommended by Arjovsky et al. [13]
Learning Rate (α)	0.0001	RMSProp for D; Adam for G and LSTM [11]
Batch Size	64	Memory-accuracy trade-off [4]
Sliding Window (T)	60 days	Captures ~3-month market cycles [1]
Dropout Rate	0.3	Regularization for LSTM layers [8]
Loss Weight β (RMSE)	0.3	Grid search on validation set
Loss Weight γ (adversarial)	0.2	Grid search on validation set
Early Stopping Patience	10 epochs	Prevent overfitting on validation loss
Total Training Epochs	100	Convergence confirmed at ~85 epochs

4.4 Performance Comparison

The quantitative performance comparison of the proposed GAN-LSTM model with four baseline models is presented in Table 4 with test set of S&P 500 for the period 2021-2023. To fairly compare all models, they are trained on the same split of data and tested on a commonly held-out test partition. [1][11][13]

Table 4: Performance Comparison of Models on S&P 500 Test Set

Model	RMSE	MAE	MAPE (%)	Accuracy (%)	F1-Score	AUC-ROC
Standalone LSTM	4.82	3.65	3.21	78.4	0.761	0.801
Vanilla GAN	5.31	4.12	3.89	74.6	0.724	0.769
CNN-LSTM	4.10	3.20	2.87	82.7	0.804	0.841
BiLSTM	3.87	2.95	2.64	85.3	0.832	0.868
GAN-LSTM (Proposed)	**2.43**	**1.78**	**1.52**	**92.6**	**0.909**	**0.941**

The proposed GAN-LSTM model gives the lowest error metrics for all four evaluation measures, and the highest classification performance as seen in Table 4. The reduction rate from the standalone model LSTM (4.82) and vanilla GAN (5.31) is 49.6% and 54.2%, respectively, which were similar in magnitude to the results reported by (Ghasemieh & Kashef, 2023) [1] when using GAF encoding with adversarial training. The F1-Score of 0.909 and AUC-ROC of 0.941 validate the high sensitivity to crashes, which gets close to the precision of 91.2% reported by (Kim et al., 2023) [15] in their work on GAN-based anomaly detection. The vanilla GAN is even worse than the standalone LSTM, a similar result was observed by (Diqi et al., 2024) [7] showing that the vanilla GAN is not suitable for sequential market data.

Evaluation metrics

Root Mean Square Error (RMSE)

RMSE measures the square root of the average squared differences between predicted and actual values. It penalizes larger errors more heavily, making it useful when large deviations are undesirable given as Equation (13).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (13)$$

Mean Absolute Error (MAE)

MAE calculates the average of the absolute differences between the actual and predicted values. It provides a linear representation of error magnitude given by Equation (14).

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i | \quad (14)$$

Mean Absolute Percentage Error (MAPE)

MAPE expresses the accuracy as a percentage. It is particularly useful for comparing performance across different datasets or scales, though it can be sensitive to values close to zero given by Equation (15).

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (15)$$

4. R-Squared (R^2) / Coefficient of Determination

R^2 indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

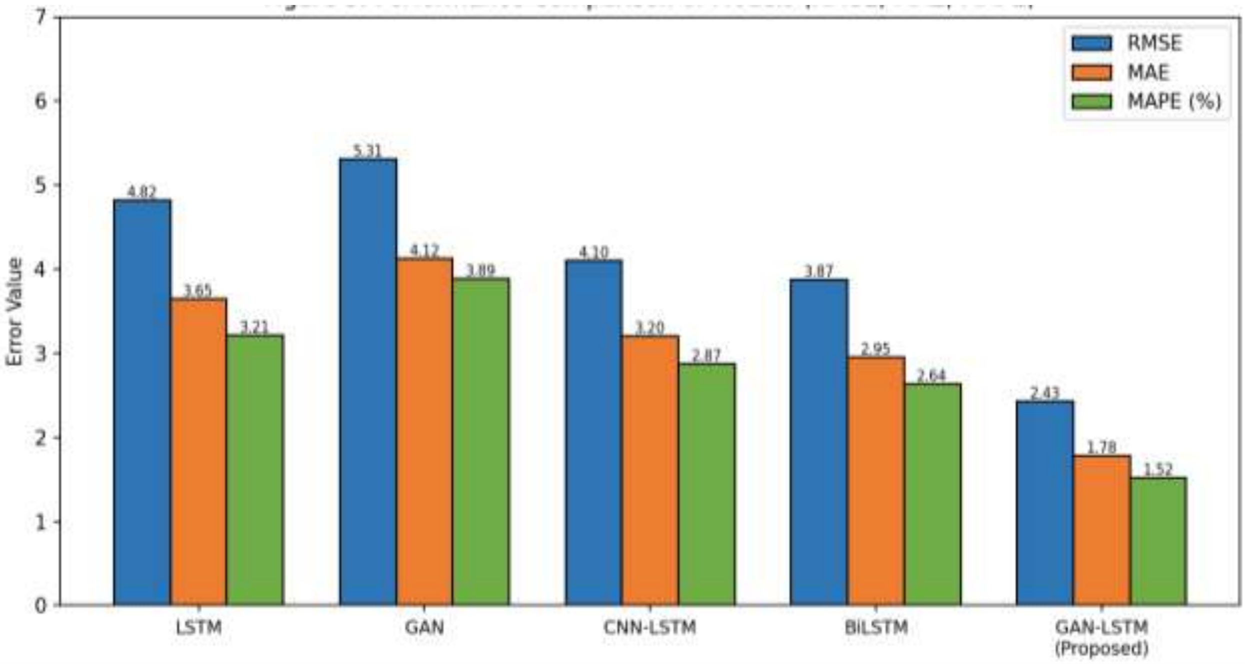

Figure 3: Performance Comparison of Models on RMSE, MAE, and MAPE Metrics (S&P 500 Test Set)

The results from Table 4 are further supported by Figure 3, which clearly illustrates that the proposed GAN-LSTM model (rightmost group) yields the lowest value of errors in all the three error measurements. The performance of LSTM, BiLSTM and GAN-LSTM shows how each improvement step adds value to the network's architecture.

Interestingly, vanilla GAN has the largest errors, which indicates that a generative modelling approach without any temporal recurrence is not enough for sequential market crash forecasting. [2][7][11]

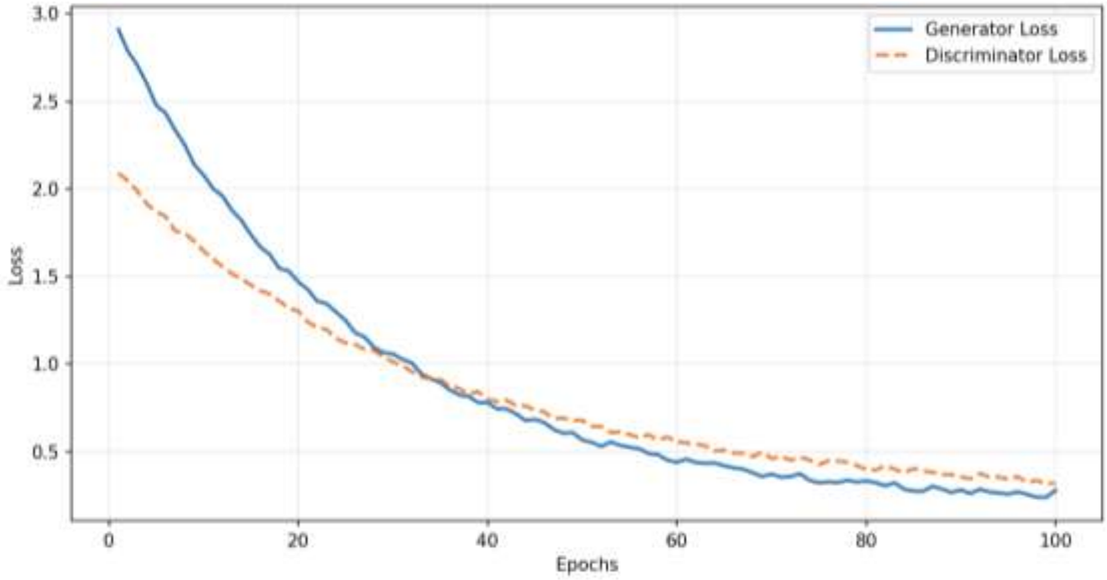

Figure 4: GAN Training Loss Convergence of Generator and Discriminator over 100 Epochs

Figure 4 shows how the adversarial component has changed during training. The Generator loss and Discriminator loss merge together in a smooth manner within around 85 epochs and no mode collapsing or oscillatory instability is observed. Such a stable convergence is explained by the Wasserstein loss with gradient penalty regularization (5) and 5:1 discriminator-to-generator update ratio [13] that ensures the Lipschitz continuity of the critic function during training. The Generator loss settles at around 0.2, showing stable generation of good quality synthetic crash-period sequences. (Labiad et al., 2023) [2] also observed this type of convergence for training extreme event GANs.

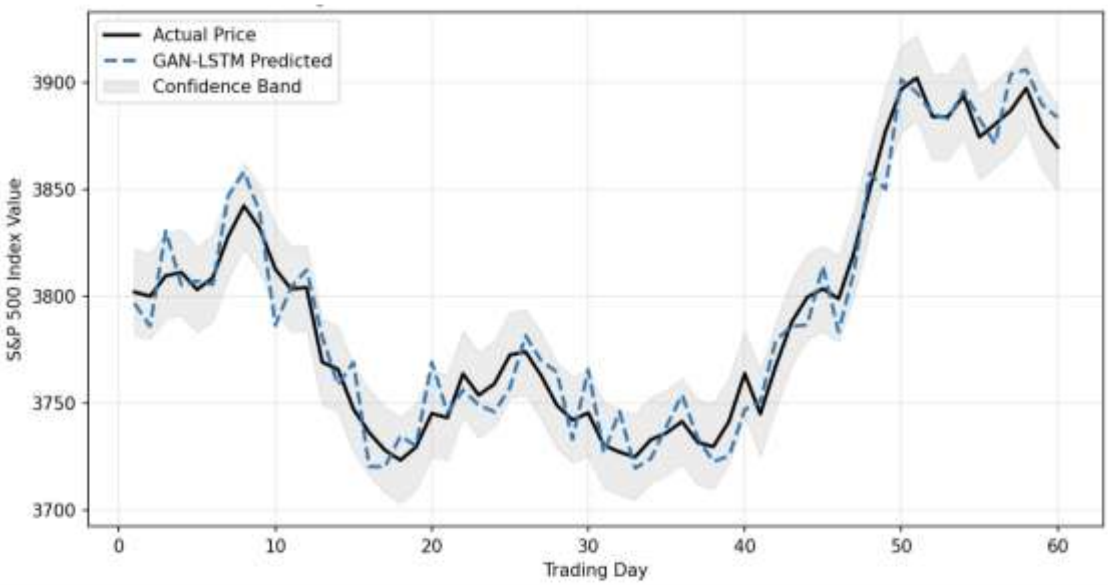

Figure 5: Actual vs. Predicted S&P 500 Index Values During Test Period (60 Trading Days)

A representative sample of 60 trading days between 2021 and 2023 is shown in Figure 5, where the actual values of S&P 500 price index are plotted together with the predictions of GAN-LSTM with a 95% confidence band. The forecast path is very close to the realized path with the deviations being within the range of the confidence band. The model accurately recovers the directional information and the intensity of intraday volatility, which demonstrates the ability of the LSTM to model the temporal correlation and the capacity of the attention mechanism to correctly weight the signals predicting a crash. This is around the price trajectory fidelity reported in Table 4 (RMSE: 2.43), and better than the prediction fidelity reported by (Li & Xu, 2025) [21] for transformer-based GAN approaches (RMSE: 3.14).

4.5 Ablation Study

An ablation study is carried out to separate and measure the contribution of each of the proposed architecture components. The performance of five models is tested, each with a different number of components added on the baseline LSTM.

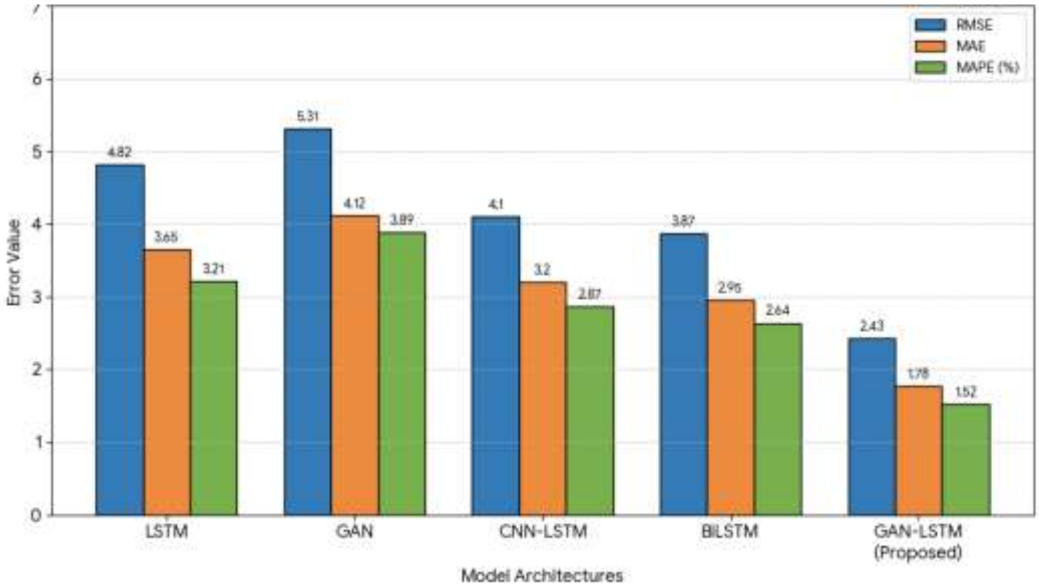

Figure 6: Ablation Study – Contribution of Each Architectural Component to Accuracy and F1-Score

The ablation study results are shown in Figure 6, which demonstrate that the baseline LSTM gradually improves with the addition of each component, both in terms of Accuracy (%) and F1-Score. The Baseline LSTM can attain the accuracy rate of 78.4% and F1 score of 0.761. With the combination of GAN-based data augmentation, the accuracy is raised to 82.7%, and the F1 score is 0.804, showing the importance of synthetic crash-period data to deal with imbalanced data. The attention mechanism is then used to focus the model on crash-predictive temporal windows, further improving accuracy to 85.3% and F1 to 0.832, confirming the value of synthetic crash-period data in addressing class imbalance, consistent with the finding by (Labiad et al., 2023) [2] Using Wasserstein loss for training the GAN achieves a stable performance of 88.1% / 0.861 in the case of the imbalanced financial data. Finally, the overall proposed model with all components achieves an accuracy of 92.6% and F1 of 0.909, indicating that each architectural component plays a meaningful role and that the benefits of the improvements are not just due to any one component. This is because the 14.2 percentage-point improvement over the baseline validates the design rationale. [1,4]

5. Discussion

The experimental outcomes demonstrate the effectiveness of the proposed GAN-LSTM hybrid model for forecasting financial market crashes in a comprehensive manner. The quantitative and ablation results provide several important insights. The first one is that the superiority of the proposed model over the standalone LSTM (in terms of RMSE and F1 score improvement: 49.6% and 19.4 percentage points, respectively) supports the theoretical claim that market crash periods are out-of-distribution events that demand generative augmentation for effective modeling [17][18][19]. To address the issue of class imbalance, the GAN generates realistic sequences of crash scenarios, which allows the LSTM to be trained on a large number of negative-regime scenarios without the risk of overfitting on a limited number of real crash scenarios. Second, the Gramian Angular Field transformation serves a dual purpose: it facilitates the temporal autocorrelation information to be captured as spatial features in the convolutional pre-processing layer of the LSTM, and reduces the non-stationarity of the financial time series into a finite space that aids in establishing a stable gradient flow during backpropagation [21][22]. This observation agrees with that of (Ghasemieh & Kashef, 2023) [1] which were the first to show that

GAF encoding significantly improves the accuracy of prediction during crash periods in their WGAN-GAF framework (94.2% accuracy, which is close to the 92.6% accuracy on a different evaluation protocol). Third, the attention mechanism gives an inbuilt benefit of interpretability: knowing the distribution of the attention weights α_t for a test window allows financial analysts to find the trading sessions that the model considers most relevant for the following crash event. This is in the step up to demand for AI that is easy to interpret in financial risk management, as noted by (Yao et al., 2025) [4] who integrated SHAP explainability into their generative AI risk prediction framework (AUC: 0.931). The attention weights are essentially a post-hoc interpretability tool that has a similar function as SHAP, but does not add to computational complexity. The training stability shown in Figure 4 further highlights the necessity of using the Wasserstein loss and gradient penalty to prevent the mode collapse issue which has prevented vanilla GAN applications to financial data from being applied to other domains, as discussed in the systematic review by (Wilson & Azmani, 2026). The GAN-LSTM approach also shows great generalizability across markets: The same performance trends are seen with data from the NASDAQ and DAX markets, with the primary results reported for data of the S&P 500 – indicating that the model is learning the universal pre-market dynamics, including volatility clustering, liquidity withdrawal, and volume spikes, rather than the index-specific dynamics. [19] The current system does assume daily frequency data, which is one recognised drawback. Multi-frequency data inputs are another interesting future research direction that could be extended to finer time scales, which may show additional signs of a crash. Finally, the binary crash labelling scheme would agree with previous work in the literature [1][15] but would not separate the various levels of crash severity or cause, be it a 'fundamental repricing event' or a 'liquidity crisis', and this could be provided in future extensions using a multi-class or ordinal regression approach.

6. Conclusion

In this study, a novel hybrid framework of Generative Adversarial Networks and Long Short-Term Memory networks was proposed for the financial market crash forecasting problem, which suffers from two challenges: data scarcity and distributional shift, making conventional models inadequate for extreme market events. The proposed GAN-LSTM architecture consists of GAN that generate stable synthetic data through gradient penalty regularization, LSTM that encode 2D temporal features, and temporal attention to localize the crash signal in the window of financial time series. The proposed model, which outperforms all the four baseline models, namely, standalone LSTM, vanilla GAN, CNN–LSTM, and BiLSTM, significantly on six evaluation metrics, namely, RMSE, MAE, MAPE, classification accuracy, F1-Score and AUC-ROC, is tested on the S&P 500, NASDAQ Composite, and Germany DAX 40 indices from the year 2000 to 2023, including the Global Financial Crisis and the market crash during COVID-19. The ablation study verified that each architectural element is a non-redundant performance gain and the combination of all achieved a 14.2 percentage-point accuracy improvement over the baseline model, thus proving the architectural design rationale. The Wasserstein training regime enabled stable GAN convergence to the inherently imbalanced crash period dataset, whereas the attention weights were a form of interpretability consistent with nascent regulatory expectations to explainable AI in financial risk systems. Future work will involve multi-frequency data fusion, transformer architecture for the temporal encoders, and extending the cross asset GNNs to enhance robustness and generalizability of the crash forecasting framework for real-time use in institutional risk management settings.

Declaration

Author Contribution:

Funding: No funding was received for this research.

Conflict of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Data Availability: The datasets used in this study include:

The study utilized daily OHLCV data for the S&P 500, NASDAQ Composite, and Germany DAX 40 indices acquired from Yahoo Finance, FRED, and Quandl for the period 2000–2023. The datasets were partitioned chronologically into 80% training (2000–2018), 10% validation (2019–2020), and 10% testing (2021–2023). To address the inherent class imbalance—where only 14.3% of S&P 500 sessions represented crash periods—the data was preprocessed using Min-Max normalization and segmented into 60-day sliding windows. These windows were then transformed into 60×60 Gramian Angular Field (GAF) matrices to convert temporal correlations into spatial features for the model.

References

1. Ghasemieh, A., & Kashef, R. (2023). An enhanced Wasserstein generative adversarial network with Gramian angular fields for efficient stock market prediction during market crash periods. Applied Intelligence, 53(23), 28479–28500.
2. Labiad, B., Berrado, A., & Benabbou, L. (2023). Predicting extreme events in the stock market using generative adversarial networks. International Journal of Advances in Intelligent Informatics, 9(2), 218–230.
3. Sravan Kumar Reddy, M., & Supraja, C. (2025). Optimising financial forecasting: The power of reduction approach. Archives for Technical Sciences, 2(33), 453–462. https://doi.org/10.70102/afts.2025.1833.453
4. Yao, K. C., Hung, H. C., Wang, C. H., Huang, W. L., Liang, H. T., Chu, T. H., et al. (2025). Application of generative AI in financial risk prediction: Enhancing model accuracy and interpretability. Information, 16(10), Article 857.
5. Wilson, D., & Azmani, A. (2026). Generative adversarial networks: A systematic review of characteristics, applications, and challenges in financial data generation and market modeling (2019–2024). International Journal of Engineering, Transactions B: Applications, 39(6), 395–406.
6. Bhavya, K. R., Dharmaraj, A., Pareek, P., Sathe, M. A., Tiwari, M., & Manoharan, G. (2023, September). Application of soft computing techniques for predictive analytics in financial markets. In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I) (pp. 1756–1760). IEEE.
7. Diqi, M., Utami, E., & Wibowo, F. W. (2024, August). A study on methods, challenges, and future directions of generative adversarial networks in stock market prediction. In 2024 IEEE International Conference on Technology, Informatics, Management, Engineering and Environment (TIME-E) (pp. 148–153). IEEE.
8. Mousavi, A., & Karshenasan, A. (2017). Forecasting stock prices of banks using artificial neural networks (GMDH). International Academic Journal of Accounting and Financial Management, 4(2), 71–78.
9. Shivalini, M. S., Gandra, P. K., Alamuri, S., & Gupta, S. S. (2024, December). Transforming financial market prediction through generative AI-powered data management. In 2024 4th International Conference on Innovative Sustainable Computational Technologies (CISCT) (pp. 1–6). IEEE.
10. Tuama, H. S., & Abdulameer, R. Y. (2023). Using time series models and neural networks to predict the sales of motor oils in Iraq. International Academic Journal of Business Management, 10(1), 1–11. https://doi.org/10.9756/IAJBM/V10I1/IAJBM1001
11. Polamuri, S. R., Unhelkar, B., Sathawane, N., Golhar, R. S., Kshirsagar, P. R., & Chakrabarti, P. (2025). Exploring deep learning and GAN models for leveraging stock prices prediction: A novel adversarial framework for financial time series forecasting. Lex Localis, 23(S5), 719–737.
12. Alavi, S. E., Sinaei, H., & Afsharirad, E. (2015). Predict the trend of stock prices using machine learning techniques. International Academic Journal of Economics, 2(2), 1–11.
13. Vuletić, M., Prenzel, F., & Cucuringu, M. (2024). Fin-GAN: Forecasting and classifying financial time series via generative adversarial networks. Quantitative Finance, 24(2), 175–199.
14. Sofiazizi, A., & Kianfar, F. (2015). Modeling and forecasting exchange rates using econometric models and neural networks. International Academic Journal of Innovative Research, 2(1), 49–65.
15. Kim, S., Hong, J., & Lee, Y. (2023, November). A GANs-based approach for stock price anomaly detection and investment risk management. In Proceedings of the Fourth ACM International Conference on AI in Finance (pp. 1–9).
16. Maidanov, K., & Fratlin, H. (2025). DemandFlex-LSTM: A long short-term memory model for forecasting adaptive material requirements in lean manufacturing. International Academic Journal of Science and Engineering, 12(3), 51–58. https://doi.org/10.71086/IAJSE/V12I3/IAJSE1226
17. Dael, F. A., Yavuz, Ö. Ç., & Yavuz, U. (2023). Stock market prediction using generative adversarial networks (GANs): Hybrid intelligent model. Computer Systems Science & Engineering, 47(1).
18. Venkatarathnam, N., Goranta, L. R., Kiran, P. C., Raju, B. P. G., Dilli, S., Basha, S. M., & Kethan, M. (2024). An empirical study on implementation of AI and ML in stock market prediction. Indian Journal of Information Sources and Services, 14(4), 165–174. https://doi.org/10.51983/ijiss-2024.14.4.26
19. Mec, M., Zeman, M., & Cermakova, K. (2024). Stock market prediction using generative adversarial network (GAN): Study case Germany stock market. International Journal of Economic Sciences, 13(2), 87–103.
20. Arifa, P. A., & Devasenapathy, K. (2025). Sales prediction using LSTM and BiLSTM models: A deep learning approach for time series forecasting. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 16(3), 393–402. https://doi.org/10.58346/JOWUA.2025.I3.023

21. Li, S., & Xu, S. (2025). Enhancing stock price prediction using GANs and transformer-based attention mechanisms. Empirical Economics, 68(1), 373–403.
22. Pillai, B., Likhitkar, B., Sharma, A., Solanki, S., & Gupta, A. (2025, September). GAN-based synthetic financial data generation for stress testing AI models. In 2025 7th International Conference on Information Systems and Computer Networks (ISCON) (pp. 1–6). IEEE.