EAI Endorsed Transactions
on AI and Robotics

Research Article EAI.EU

AC-Seq2Net: A Math-Guided Hybrid Deep Learning Framework for Explainable Geometric Pattern Recognition in Financial Time Series

Md Fazlur Rahman[1], Md. Abul Kalam Azad[2], Esrat Jahan[3], Najmus Saadat[3], Md Faysal Ahmed[1], Abul Kadar Muhammad Masum[4,*]

[1]School of Business, International American University, USA
[2]Dept. of Business and Technology Management, Islamic University of Technology, Gazipur, Bangladesh
[3]Computational Intelligence Lab, Southeast University, Dhaka, Bangladesh
[4]Department of Computer Science and Engineering, Southeast University, Dhaka, Bangladesh

Abstract

Financial market data is usually rough and volatile. This noise makes it difficult to detect clear price structures from raw intraday records. Traders still rely on geometric patterns such as symmetrical triangles to describe consolidation and potential breakouts. Deep learning models are able to acknowledge market internal conditions with high scores though they tend to acquire concealed relations. They do not capture the triangle geometry in an explicit form. This introduces a gap of representation and lowering interpretability. In order to overcome this problem, we proposed AC-Seq2Net. It is a math-guided hybrid symmetrical triangle recognition model in intraday price data. The pipeline segments the price stream into rolling windows and applies local min-max normalization to keep the scale consistent across different price levels. Ground truth labels are produced through geometric boundary extraction using ordinary least squares trendlines, upon which slope convergence and volatility contraction are derived as explicit engineered features. Unlike purely black-box sequence models, AC-Seq2Net integrates these geometric descriptors alongside hybrid temporal representation learning. A two-stage 1D-CNN extracts local structural patterns, a Bidirectional LSTM captures temporal dependencies, and an Attention mechanism highlights critical convergence boundaries. This design encodes symmetrical triangle geometry directly rather than approximating it through latent feature learning alone, bridging deep learning performance with trader-interpretable geometric reasoning validated through SHAP-based explainability. The triangle probability is produced using a sigmoid layer. Experiments on longitudinal S&P 500 and NASDAQ 100 data report 97.70% accuracy and 0.9947 AUC, confirming that geometric signals rather than random fluctuations drive model decisions.

Received on 11 February 2026; accepted on 14 May 2026; published on 08 June 2026

Keywords: Financial Time Series, Geometric Pattern Recognition, Symmetrical Triangle, Hybrid Deep Learning, Attention Mechanism; Explainable AI (XAI)

doi:10.4108/airo.11907

1. Introduction

Financial markets produce OHLC streams that are noisy, non-stationary and regime switching, which makes reading the charts manually subjective and difficult to reproduce on a large scale. To make this subjectivity smaller, candlestick or OHLC behavior are increasingly encoded into learnable structures in the recent studies and integrated into predictive pipelines. For example, the detection of patterns in the form of candlestick mining in conjunction with sequence similarity demonstrates that pattern level structure can serve as an interpretable signal for trend forecasting beyond the raw variation of the prices [1]. In practical trading pipelines candlestick recognition is also applied as a reliability filter for machine learning

*Corresponding author. Email: akmmasum@seu.edu.bd

recommendations, that is, explicit pattern checks can be applied to reduce fragile trading signals [2]. These trends validate the drive for patterns recognition that is automated and in the same scent of technical chart semantics for scalability.

A large portion of the existing literature continues to learn from visual candlestick representations as image-based encodings can fit well into CNN and vision pipelines. Candlestick-chart learning has been applied to the prediction of the prices movement, proving that visual encodings do provide discriminative cues [3]. Some of the works go further and frame buy or sell identification as an object detection problem on candlestick charts: there is evidence that detection architectures can pick up some actionable cues from the plotted charts [4]. More recent pipelines extend candlestick recognition by generative representations, in order to boost classification under a low amount of labeled data, which reinforces the appeal of chart images for deep learning [5]. However, there remains a crucial limitation: image-based models are often sensitive to rendering, scaling, and window choice, and the signal learned may fluctuate when the same geometry is plotted under different conditions. To prevent sensitivity against the appearance, other directions work directly on sequences or with structured mining. Rule-based candlestick recognition allows for transparent recognition and is simple to implement in highly volatile markets, however rules can be brittle across regimes and fail to learn temporal dependencies [6]. Symbolic and compression-style mining can efficiently find recurrent motifs in a data set, yet the found motifs are not necessarily related to precise geometric chart definitions like converging trendlines and volatility squeeze [7]. At the same time, the survey work on candlestick feature extraction emphasizes that there is a lack of unified geometry-grounded evaluation as well as consistent benchmark design in the field, which becomes a difficulty in comparing the features at the level of patterns [8]. As a result, robust pattern specific learning especially for classical formations are underdeveloped.

A second major stream combines candlestick information with sequential learning or decision-making systems. Multi-type fusion with deep reinforcement learning reveals that the fusion of candlestick charts and indicators can help trading performance, however, the fusion is often featuring level and does not provide geometric evidence for why a pattern is valid [9]. Ensemble deep RL systems that "learn candlesticks" are able to beat numerical baselines in trading crypto, but they are still trading-objective driven and do not impose formation geometry [10]. Hybrid encoders identifying visual candlesticks with the technical indicators show a better representation learning but usually model dynamics, not an explicit check of geometric constraints of the target pattern class [11]. Related chart decomposition pipelines also learn multi-scale candlestick cues for directional prediction, but do not enforce the structural validity conditions of a true symmetrical triangle [12]. Recent pair trading systems based on candlestick embeddings further validate the usefulness of visual features in end-to-end strategies but still fail to isolate triangle-specific geometric reasoning [13].

These observations reveal three gaps that are particularly important for symmetrical triangle recognition. First, many candlestick pipelines are focused on optimizing a prediction or trading reward, but fail to enforce any triangle validity constraints such as converging support and progressive range contraction. Second, image-based learning is still sensitive to scaling and chart rendering, which can distort the same underlying geometry from one platform to another. Third, even if models are able to achieve good performance, the models tend to be poorly interpretable, so it is unclear if decisions are based on meaningful geometric information, or based on spurious correlations. Recent efforts in interpretable 1D-CNN based forecasting highlight the importance of explanation tools to foster trust and also suggest that trust is built when interpretability is high, but if the features are not structurally aligned with the definition of the domain [14].

Motivated by these gaps, we design AC-Seq2Net, an end-to-end symmetrical triangle-based recognition system that avoids geometry conservation, but learns temporal dynamics. The novelty of AC-Seq2Net does not arise from introducing a fundamentally new standalone deep learning block; rather, it lies in the task-specific integration of four complementary elements: (i) rolling-window preprocessing with local normalization to reduce scale sensitivity, (ii) mathematically engineered geometric descriptors aligned with the formal geometric definition of symmetrical triangles, (iii) hybrid temporal representation learning using 1D-CNN, BiLSTM, and Attention, and (iv) SHAP-based explainability to verify that model decisions are driven by meaningful geometric evidence rather than spurious market fluctuations. In this way, AC-Seq2Net moves beyond conventional black-box financial models by combining explicit pattern validity constraints with interpretable sequence learning.

2. Literature Review

Financial time series are of course non-stationary, are noisy, and are regime dependent, so the use of subjective chart reading tends not to scale and lacks consistent reading. Large-scale reviews indicate that modern financial modeling is moving toward pipelines in ML and DL that are data-driven, but also caution about the potential effects of weak experimental protocols, leakage and incomplete reporting which can magnify the results and result in decreasing reproducibility [15]. In the same direction, it is revealed from practical surveys that many deep finance studies often have problems in translating the model accuracy into decision support that can be trusted because transparency and deployment of realism are not often addressed [16]. These results provide motivation for pattern-recognition pipelines that are not only accurate, but also robust, protocol-safe and explainable.

A huge flow of work renders chart patterns learnable through converting OHLC or candlestick series into image-like forms to allow CNNs to directly classify patterns [17]. But it is based heavily on plotting decisions, even with the same underlying price geometry, re-scale axis, change of style or rendering can push the learned decision boundary [18]. This is especially troublesome on the case of symmetrical triangles, where validity of pattern follows converging boundaries and contraction of volatility, rather than being determined by the manner in which the chart is drawn. This means that image-based methods are sensitive to the normalization decisions and window scale reducing cross market transferability. In addition to images, other researchers combine pattern cues with end-to-end trading pipelines designed with engineered indicators and multi-window characteristics, and present quantifiable decision benefits when evaluated systematically [19]. However, most systems take patterns as big families and do not impose the rigid rules of geometry that a symmetrical triangle has. Other patterns of the literature directly identify patterns with 1D sequences and show that without plotting convolution can learn useful shape features, making it more efficient and less prone to rendering bias [20]. Nevertheless, benchmark scarcity and poor labeling criteria is still a bottleneck, which restricts triangle specific generalization. Further feature-oriented work demonstrates that feature-guided by rule are capable of a more uniform representation of chart formations, yet are easily broken under regime transitions unless the rules are supported by powerful temporal learning [21]. Formalization research carried out earlier also reinforces this point by transforming chart patterns into quantifiable constraints, though these also reveal the challenge of being robust and scalable with fixed specifications alone [22]. More recently, graph-based representations model recurring pattern structures and are interpretable at the topology level, though tend to focus on global structure and ignore local geometric constraints, such as converging trendlines and squeeze behaviour [23]. Furthermore, less technical reviews of the concept of the graphic signal recognition point out that the transition between the geometry of technical charts and sequence learning is currently a blank area [24]. Financial forecasting research has also expanded beyond traditional sequential modeling toward graph-based and foundation-model-based paradigms. Recent studies on graph neural networks have shown that spatial, temporal, and inter-asset dependencies can improve financial prediction by capturing relational structure more effectively than independent sequence models [25], [26]. At the same time, emerging research on financial large language models indicates a broader shift toward context-aware and semantically enriched frameworks for financial analytics and decision support [27]. Collectively, these developments further motivate the design of hybrid, explainable, and practically deployable architectures for financial pattern recognition.

In parallel, the further development of deep sequential modeling continues with attention-based recurrent modeling for signal focus on noisy financial sequences [28]. Hybrid designs combining convolution and bidirectional sequence learning are also effective due to the separation of the local morphology extraction and temporal dependency modeling [29]. Transformer-based models further enhance the long-range dependency learning but are typically optimized for movement prediction instead of enforcing pattern validity and are often still difficult to interpret in geometrical terms [30]. As a result, many high-performing predictors do not ensure that their learned signals correspond to a well-defined symmetrical triangle especially in case of changing window length, changing volatility regime or scaling.

Explainability is now rapidly becoming considered as a fundamental requirement for financial ML, as decision evidence is required by stakeholders for auditability, risk control and trust [31]. Systematic reviews reveal that XAI has moved to the heart of finance processes, but current practices are scattered and often have no consistent evaluation protocols for explanations [32]. Complementary work on model agnostic explainability summarizes the popularity of XAI tools, while also noting that faithful explanations are dependent on feature design and problem framing, not only the explainer itself [33]. In parallel, recent studies have increasingly explored explainable sequential architectures in time-dependent prediction tasks. For example, transformer-based frameworks integrating temporal attention and interpretability mechanisms have demonstrated strong capabilities for modeling complex longitudinal dependencies while improving the transparency of model decisions in healthcare forecasting applications [34].

Similarly, applied AI systems for infrastructure predictive maintenance highlight the growing importance of reliable data-driven prediction frameworks for supporting operational decision-making in real-world environments [35]. Moreover, hybrid deep learning architectures combined with metaheuristic optimization strategies have shown improved forecasting robustness and parameter adaptability in complex nonlinear prediction tasks [36]. These developments collectively

Table 1. Comparative Summary of Recent Symmetrical Pattern Recognition Studies

Author (Year)	Dataset	Method	Key Insight	Limitation	Efficiency
Chen & Tsai et al. [17]	Simulated OHLC (GBM) + image labels	Image-CNN	Images make patterns learnable.	Plot scale sensitive.	High-dim image; expensive.
Jin & Kwon et al. [18]	KOSPI chart images	CNN	Chart design shifts accuracy.	Plot-dependent; no detection.	Very high-dim; high cost.
Lin et al. [19]	Stock time series	Pattern+ indicator ML	Patterns + indicators aid trading.	Geometry not enforced.	Compact vectors; low burden.
Qiu et al. [28]	Major index series	Attn-LSTM	Attention filters noise.	No geometry guarantee.	Seq-based; moderate cost.
Lu et al. [29]	Shanghai index (1k days)	CNN-BiLSTM +Attn	Local+temporal fusion helps.	Pattern implicit.	Hybrid deep; high overhead.
Wang et al. [30]	Multi-index benchmarks	Transformer	Long-range deps captured.	Low interpretability; no geometry.	Large representation; high demand.

motivate the design of hybrid and explainable architectures for reliable financial pattern recognition. This suggests that explainability is best when the model is based on meaningful features that are aligned with the domain. Table 1 summarizes the reviewed literature and highlights their main limitations.

Against this background, AC-Seq2Net addresses three specific gaps of symmetrical triangle recognition in concrete. First, it decreases the representation sensitivity in the preprocessing by rolling-window, and by local normalization to achieve the stability in all price scales and volatility regimes. Second, it explicitly encodes the validity of triangulation, the use of geometric descriptors that correspond with the definition thus converging trending line behavior and volatility squeeze so the model is not based on implicit learning of patterns only. Third, it combines these engineered geometric cues with a hybrid deep architecture that learns temporal-geometry representations for classification, and provides transparency through SHAP-based validation to establish that the signal shown in terms of detected triangle could be explained in terms of meaningful structure. In this way, AC-Seq2Net escapes all the limitations on image-dependence, on purely directional forecast and on ungrounded pattern label, which are available in the literature. Also, AC-Seq2Net providing a geometry preserving, sequence aware and explainable pipeline for robust symmetrical triangle recognition.

3. Methodology

This section introduces the architectural design of the proposed AC-Seq2Net which is systematically illustrated in Figure 1. The framework is a synthesis of advanced geometric feature engineering and a hybrid neural architecture consisting of dual-stage CNNs for spatial extraction, Bi-LSTMs for temporal learning and Adaptive Attention to enable automated recognition of symmetrical triangle patterns in financial time series.

3.1. Dataset Description

The foundation of the empirical study is based on a longitudinal sample taken from the 5-Year Data for S&P 500 & NASDAQ 100 dataset [37]. The data set covers the period from August 2019 to August 2024. It is composed of some 262,000 observations taken at 5-minute intervals. The main features are the timestamp, open price, high price, low price, close price, and trading volume. When the raw data are statistically explored, it becomes clear what the inherent complexity of financial time series is. As illustrated in Figure 2, the market has significant volatility clusters and stochastic noise. Price movements are frequent and non-stationary along the observed timeline. This high entropy environment makes the identification of meaningful patterns not so easy. Random fluctuations in the data tend to confound the geometric structure in the raw data.

3.2. Data Preprocessing

The process of converting raw intraday price sequences into a structured dataset is carried out in a structured pipeline which involves segmentation, normalization and geometric annotation. Initially, the continuous price stream is divided by a sliding window technique with a constant size of $n = 30$ intervals. Since financial data has different price levels, Local Min-Max Normalization is applied within each window to achieve the scale invariance. This is a scaling of the data to a range of $[0, 1]$ using the local extrema, as defined in Equation (1):

$$x_{norm} = \frac{x - \min(x_{win})}{\max(x_{win}) - \min(x_{win})} \tag{1}$$

After normalization, the ground truth is created using an automated Geometric Boundary Extraction system. Ordinary Least Squares (OLS) regression is

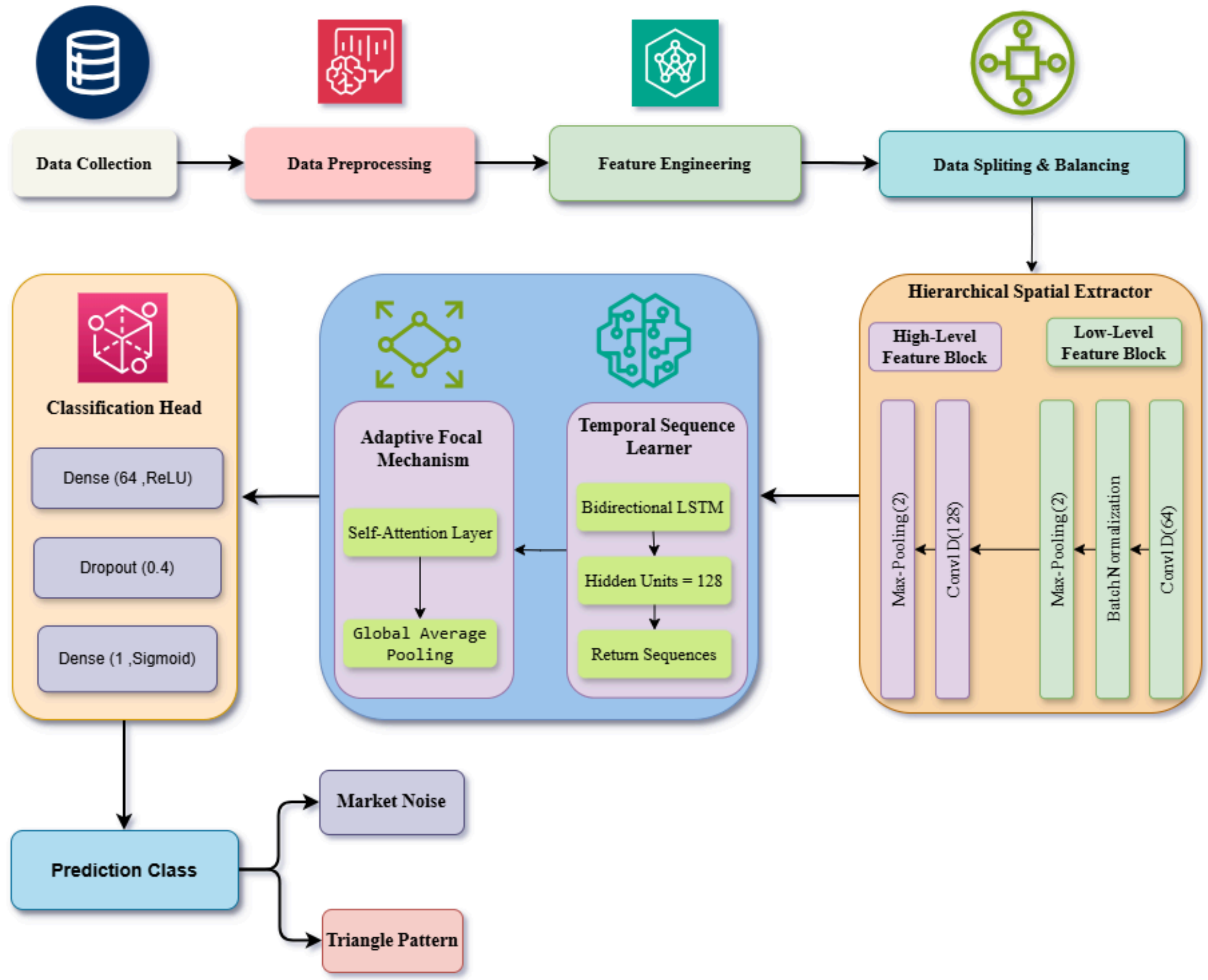

Figure 1. Systematic illustration of the proposed AC-Seq2Net framework

used to compute the first-degree polynomials for resistance (m_h) and support (m_l) lines, as given in Equation 2:

$$m = \frac{N\sum(xy) - (\sum x)(\sum y)}{N\sum x^2 - (\sum x)^2} \tag{2}$$

For each rolling window, the time axis is defined as $x =$

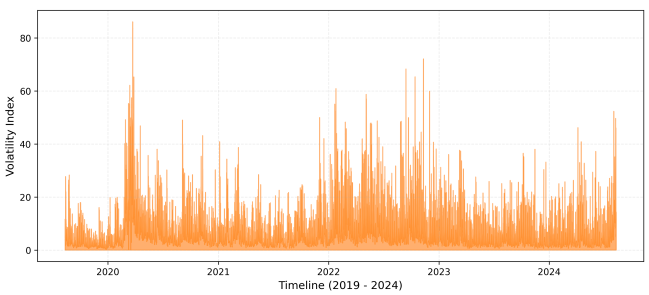

Figure 2. Illustration of volatility clustering and stochastic noise within the financial index

$\{0, 1 \ldots, n-1\}$. The upper and lower geometric boundaries are estimated separately by fitting first-order polynomials to the windowed high-price sequence and low-price sequence, respectively. In implementation, this was performed using degree-1 polynomial fitting, which is equivalent to ordinary least squares linear regression for a straight line. Thus, the slope m_h is obtained from the fitted line over the high-price series and the slope m_l the fitted line over the low-price series.

Figure 3 illustrates the geometric criterion under which a sequence is classified as a Symmetrical Triangle. Specifically, a sequence is considered valid only when the fitted upper and lower trend slopes exhibit significant convergence, defined by $m_h < -0.03$ and $m_l > 0.03$. Let $\hat{H}(x) = m_h x + c_h$ and $\hat{L}(x) = m_l x + c_l$ denote the estimated resistance and support boundaries, respectively. Convergence implies that the boundary gap $g(x) = \hat{H}(x) - \hat{L}(x)$ contracts over time. For linear boundaries, this condition is equivalent to $g'(x) = m_h - m_l < 0$. Accordingly, we enforce a minimum convergence rate by imposing the following constraint, as defined in Equation (3):

$$m_h \leq -\tau, \ m_l \geq \tau, \ \tau = 0.03 \Rightarrow m_h - m_l \leq -2\tau = -0.06 \tag{3}$$

In the implemented labeling procedure, a window is assigned to the triangle class only when the fitted upper boundary satisfies $m_h < -0.03$ and the fitted lower boundary satisfies $m_l > 0.03$. The threshold $\tau = 0.03$ was selected to enforce a minimum convergence strength and to prevent weak or visually ambiguous formations from being labeled as valid triangles.

Sequences that do not satisfy the triangle criterion are not all treated identically. Windows are assigned to the noise class when both slopes are nearly zero ($|m_h| < 0.01, |m_l| < 0.01$) or when both fitted boundaries move in the same direction. Windows that fall between the triangle and noise criteria are treated as ambiguous and excluded from supervised training. To mitigate class imbalance, random under-sampling is applied to the majority class prior to partitioning the dataset into training 70%, validation 15%, and testing 15% subsets.

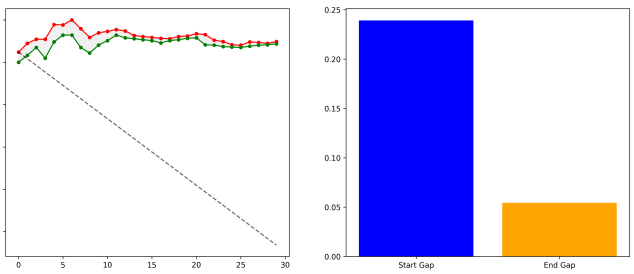

Figure 3. Geometric boundary extraction

3.3. Feature Engineering and Selection

In order to improve the discriminative ability of AC-Seq2Net model, a multivariate feature set was built to represent the temporal dynamics of market patterns as well as the geometric morphology of market patterns. Instead of only utilising univariate closing prices, the input vector at time step t was extended to incorporate normalized High, Low and Close sequences.

In addition, in order to explicitly quantify the "price squeeze" feature of symmetrical triangles, a Volatility Contraction (V_t) feature was engineered. This is mathematically defined a s t he a bsolute difference between the daily high and low prices, as defined in Equation (4):

$$V_t = |P_t^{\text{high}} - P_t^{\text{low}}| \tag{4}$$

Consequently, the final input vector for the model is represented as $X_t = \{P_t^{\text{norm}}, m_h, m_l, V_t\}$, combining both dynamic price data and static geometric attributes. Correlation Analysis determines that the selected features are providing unique and non-redundant information. In Figure 4, the heatmap shows that there is a clear demarcation between the price data and the geometric slopes. This result proves that the geometric features add new information instead of the sole repetition of the price trends, which is necessary to enable correct classification.

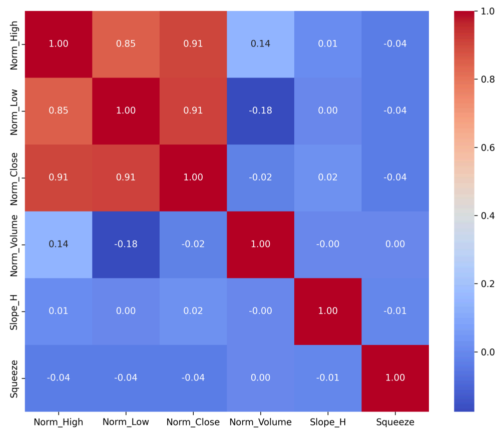

Figure 4. Correlation heatmap illustrating the statistical independence

3.4. Model Architecture: The Hybrid AC-Seq2Net

The proposed Hybrid AC-Seq2Net architecture is presented to address the shortcomings of capturing noise-robust local frameworks, as well as, the longer-range temporal dynamics. Despite the fact that various traditional models have been implemented in pattern-recognition problems. Traditional models typically display significant d rawbacks w hen learning discriminative forms in noisy market environments and complex sequential dynamics. In order to counter these issues and enhance the accuracy of Symmetrical Triangle detection, our proposed architecture will be developed as a composite of hierarchical spatial feature extraction with bidirectional sequential modeling.

The architectural pipeline is made up of 4 mathematically formalized stages:

A) Hierarchical Spatial Feature Extraction (Dual-Stage CNN)
A dual-layer Hierarchical 1D-CNN is applied at the first stage of processing to obtain geometrical structures at various scales as shown in Figure 5. The convolutional building block 1 is applied to identify low-level boundaries and short-term movements of the input sequence. The dimension reduction is then done by Max-Pooling that also aids in reducing small random fluctuations. S ubsequently a s econd convolutional block consisting of 128 filters o btains h igher-level and more complex shapes by combining the corresponding low-level features. This step is significant a s the

Symmetrical Triangle pattern is created slowly and the main features of the pattern rely on the shape as a whole and not on a specific point. By doing this hierarchical treatment the high frequency stochastic noise is filtered and the important morphological features are retained and stable feature maps are given to the next stage. The mathematical operation for feature extraction at any given time step is formally defined as in Equation (5):

$$C_t^k = \mathrm{ReLU}\left(\sum_{i=0}^{L-1} w_i^k \cdot x_{t+i} + b_k\right) \tag{5}$$

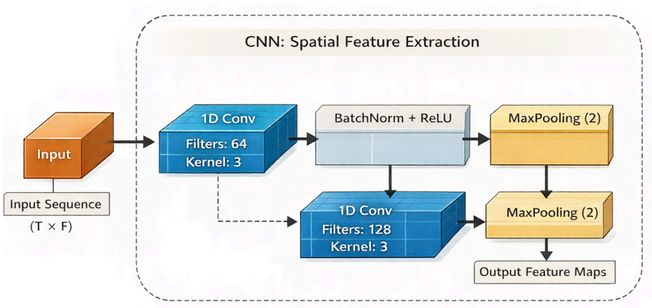

Figure 5. Dual Stage 1D CNN Block for Spatial Feature Extraction

B) Bidirectional Sequential Modeling
The feature maps with fine s patial r esolution are propagated to a Bidirectional LSTM network. Unlike the standard LSTMs that only process the data in one way, i.e., forward, the Bi-LSTM, on the other hand, will process the sequence in both forward and backward directions. This enables the model to model the "price squeeze" dynamics by simultaneously analyzing past trends and the future convergence points. The last hidden state h_t is a concatenation of both directional states, as defined in Equation 6:

$$h_t, c_t = \mathrm{LSTM}(C_t, h_{t-1}, c_{t-1}) \tag{6}$$

C) Adaptive Temporal Attention Mechanism
A Query-Value Attention Mechanism is used on the Bi-LSTM outputs in order to determine the most critical time-steps (e.g., breakout zones). This layer uses the softmax function to assign dynamic weights of importance at any given time step to each time step, as defined in Equation (7):

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \tag{7}$$

Figure 6 presents the temporal attention weight map over the pooled sequence representation. The attention distribution is clearly non-uniform, indicating that the model does not treat all temporal regions equally. Strong concentration is observed around the middle pooled steps, particularly near Steps 2–4, suggesting that the model assigns greater importance to the most informative phase of the pattern formation. This behavior supports the interpretation that the attention mechanism helps the network focus on structurally meaningful temporal regions rather than distributing importance uniformly across the sequence.

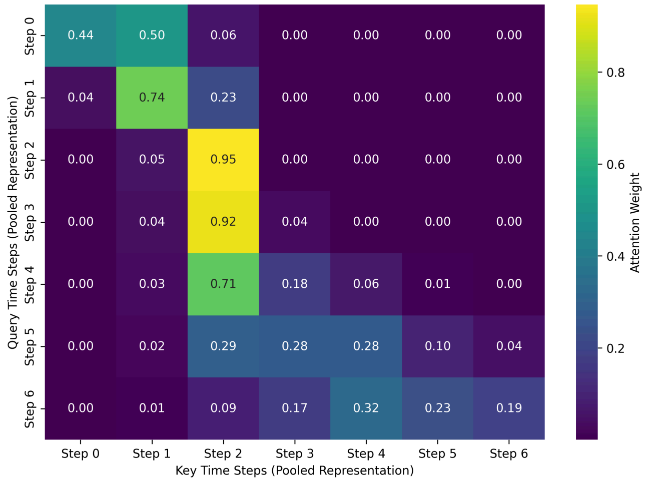

Figure 6. Temporal attention weight map over pooled sequence steps

D) Probabilistic Prediction and Decision Logic
Lastly, the attention weighted context vector is fed into the Global Average Pooling layer and an all-connected Dense Layer (64 units) that has Dropout (0.4) as a regularization measure. The Sigmoid activation function is used to obtain the final classification probability, as defined in Equation (8):

$$\hat{y} = \frac{1}{1 + e^{-(W_{\text{out}} \cdot v_{\text{context}} + b_{\text{out}})}} \tag{8}$$

When the resultant probability, denoted as P, is greater than the decision threshold, set to 0.5 then the pattern is confirmed as a valid instance failing which the pattern is dismissed as noise.

4. Result Analysis

4.1. Experimental Setup and Configuration

The proposed Hybrid AC-Seq2Net model was coded in Python on the Kaggle cloud platform with an NVIDIA Tesla P100 GPU used to accelerate the computation. The network was implemented with TensorFlow and Keras framework. Table 2 summarizes the final hyper-parameter configuration used in the reported experiments. These values were selected through validation-based model tuning, with emphasis on preserving geometric information, limiting overfitting, and ensuring stable convergence. The key hyperparameters were chosen to match the scale and noise characteristics of the problem. A window size of 30 was adopted because it preserved sufficient context for converging-boundary formation without introducing excessive unrelated variation. The two-layer CNN with 64 and 128 filters was designed to progressively extract low-level and higher-level geometric features, while a bidirectional LSTM with 128 hidden units was selected to model temporal evolution in both forward and backward directions. A dropout rate of 0.4 was applied after the dense layer to regularize the hybrid architecture under noisy intraday conditions. The learning rate of 0.0005 with Adam provided stable optimization, and a batch size of 64 offered a practical trade-off between gradient stability and computational efficiency. Early Stopping with a patience of 12 epochs and ReduceLROnPlateau with a patience of 4 epochs were used to improve generalization and prevent unnecessary overtraining.

The dataset was initially sorted in strict chronological order to preserve the temporal integrity of the price series. Rolling windows were then extracted sequentially, with each window's geometric label derived solely from its own 30-bar sequence. To address class imbalance, an equal number of positive and negative samples were drawn prior to splitting. The final 70-15-15 partitioning was performed using stratified random splitting to maintain class distribution across subsets. Since each window's label is computed entirely from within-window information, look-ahead bias at the window level is absent. However, we acknowledge that adjacent overlapping windows may share bars across subsets, and a strict chronological split with purging represents a direction for future work.

Table 2. Hyperparameter Configuration and Optimization

Hyperparameter	Optimal Value
CNN Architecture	2 Layers (64, 128 Filters)
Kernel Size	3
LSTM Type	Bidirectional
LSTM Units	128
Learning Rate	0.0005
Batch Size	64
Dropout Rate	0.4
Optimizer	Adam

4.2. Qualitative Results and Confidence-Based Visualization

This Figure 7 represents a qualitative view of the proposed AC-Seq2Net in terms of displaying several samples of the sliding-window and the confidence scores of the model. The subplots correspond to the various windows on which the red and green curves are taken to record the highs and lows of prices during the period. The confidence levels are also quite high in the presented cases that demonstrate that the model yields consistent decisions even in the case of a noisy signal. These windows depict several difficult market actions like abrupt pullbacks and sharp spikes as well as inconsistent oscillations. Such patterns are usually problematic in the standard models due to non-smooth shape and the local variations that may conceal the shape. Nevertheless, AC-Seq2Net is very confident, due to its learning of the geometry of windows on multiple levels. The CNN block attenuates high-frequency noise and retrieves local shape cues whereas the Bi-LSTM retrieves the temporal dynamics of the structure. The attention layer serves to enhance the reliability as well, focusing on the most informative parts and down-weighting the unstable ones. Consequently, the model is able to sustain confident forecasts within windows which capture challenging market conditions such as volatility bursts and non-stationary movement.

4.3. Classification Outcomes for Triangle Pattern Detection Using AC-Seq2Net

The confusion matrix of the Noise vs. Triangle classification is presented in Figure 8. These diagonal elements are dominant and indicate that there were 420 correctly identified Noise samples and 431 correctly identified Triangle samples which confirms good class separation. There is only 20 cases of misclassification meaning there is very little ambiguity in the model between true triangular structures and noisy price movements. False triangle alarms are the most frequent type of error where 16 Noise windows are falsely predicted to be Triangle. This indicates that even with some loud parts, partial contraction like behavior will still be observed that will resemble a triangle. Conversely, the percentage of missed triangles is not high, with 4 Triangle samples being unanimously predicted as Noise, and thus demonstrates that the model is sensitive to true pattern windows. By and large, the accuracy of the classifier is 97.70%, which presents that the classifier is reliable in terms of detection and pattern recognition during different market regimes. This low false-negative rate is especially significant in trading applications, since it minimizes the setting of false triangles in the presence of significant increases in noise, while allowing the algorithm to be highly robust to noise.

4.4. ROC Analysis of AC-Seq2Net for Triangle Versus Noise Classification

A comparative ROC analysis of all considered baselines and the suggested AC-Seq2Net is shown in Figure 9. ROC curve draws a line between the True Positive rate (TPR) and the False Positive rate (FPR) and a better classifier has a higher True Positive rate (TPR) and lower False Positive rate (FPR). We have achieved one of the best results on our AC-Seq2Net with an AUC of 0.991 meaning that we are more accurate in detecting

Figure 7. Qualitative visualization of AC-Seq2Net decisions across sliding windows.

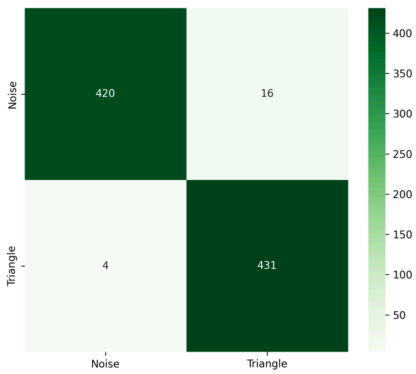

Figure 8. Confusion matrix of AC-Seq2Net model

Symmetrical Triangle windows using a smaller number of false alarms. It is interesting to note that the curve picks up quickly when even the FPR is small which shows significant performance in the area where false positives are expensive. Even though there are a number of baselines including Simple CNN Simple RNN and AdaBoost that also demonstrate a competitive performance. Proposed model is very effective since it balances local structure and long-range temporal context in a single pipeline.

The high AUC values on stock market data may be attributed by the fact chart patterns like Symmetrical Triangles still have consistent morphological clues within each sliding window despite the market being noisy. Pattern windows in this dataset are characterized by the occurrence of contraction in highs and lows as well as gradual decrease in the price range. These cues are better preserved in our AC-Seq2Net since the CNN block trains noise resistant local shape features the Bi-LSTM block trains sequential relationships throughout the whole window and the attention mechanism concentrates the decision on the most informative time steps and down weights irrelevant variation. Consequently, the model yields a more distinct pattern and noise classes resulting in a stronger ROC curve and high AUC.

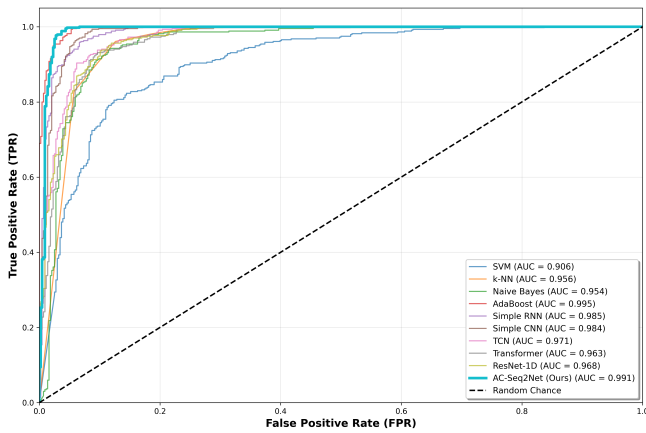

Figure 9. ROC Curve of AC-Seq2Net on the Test Set for Triangle and Noise Classes

4.5. Comparative Performance of Baseline Models and AC-Seq2Net

Table 3 provides an exhaustive comparison between AC-Seq2Net and nine representative baselines in terms of accuracy, precision, recall and F1-score. The results indicate that AC-Seq2Net achieves the best overall and the most balanced performance. Specifically, it has the highest accuracy of 97.24 percent and the highest precision of 96.38 percent, which means it has good discrimination ability with few false alarms. AC-Seq2Net also reaches a recall of 98.16 %, the highest value reached and matched only by k-NN. In the case of k-NN, the same recall is achieved with considerably low precision 83.89%, this indicate that it captures most positive samples but makes more false-positive predictions. In contrast, AC-Seq2Net is more precise and retains close to 100% sensitivity. This balanced behavior achieves the highest F1-score rate at 97.26 percent and helps to confirm the proposed method's quality to optimize detection quality and reliability.

Table 3. Performance Comparison with Baseline Models

Model Name	Accuracy	Precision	Recall	F1-Score
AdaBoost	96.67 %	96.13	97.24	96.68
1D CNN	95.29 %	92.45	98.62	95.43
Vanilla RNN	93.68 %	89.91	98.39	93.96
TCN	91.96 %	91.95	91.95	91.95
ResNet-1D	91.27 %	90.15	92.64	91.38
Time-Series Transformer	90.12 %	85.97	95.86	90.65
k-NN	89.66 %	83.89	98.16	90.46
SVM	72.67 %	92.27	49.42	64.37
AC-Seq2Net	**97.24 %**	**96.38**	**98.16**	**97.26**

An ablation study was conducted to quantify the contribution of the CNN block, BiLSTM block, and attention mechanism by removing each component individually while keeping the rest of the pipeline unchanged. The results of the corresponding classification are reported in Table 4. The overall performance of the full AC-Seq2Net is the best with an accuracy of 97.24, precision of 0.9638, recall of 0.9816 and F1-score of 97.26. Removal of the BiLSTM component leads to the largest performance decline, accuracy declines to 91.62 per cent, and F1-score declines to 91.77, suggesting that temporal dependency modeling plays a significant role in the capture of the changing convergence structure of symmetrical triangles. The removal of the CNN also leads to a decrease in performance, albeit less so, which confirms that hierarchical spatial feature extraction adds valuable local geometric information. The no-attention variant is better than the single-component variants but still worse than the full model, which indicates that the attention mechanism provides an additional refinement by emphasizing the most informative time steps. On the whole, the findings of the ablation show that the CNN, BiLSTM, and attention modules have a positive impact on the final prediction quality, and their combination in the hybrid architecture provides the most accurate and balanced performance.

Table 4. Ablation Study of AC-Seq2Net Components

Model Variant	Accuracy	Precision	Recall	F1-Score
AC-Seq2Net without CNN	0.9679	0.9492	0.9885	0.9685
AC-Seq2Net without BiLSTM	0.9162	0.9004	0.9356	0.9177
AC-Seq2Net without Attention	0.9621	0.9548	0.9701	0.9624
AC-Seq2Net (Full Hybrid)	**0.9724**	**0.9638**	**0.9816**	**0.9726**

Table 5 evaluates the effect of the rolling window length on symmetrical triangle recognition and compares the result with AC Seq2Net. Increasing the window from 15 to 45 results in improved recall from 96.94% to 98.00% but results in reduced precision from 94.07% to 93.87%. AC Seq2Net shows the best trade-off with 97.24% accuracy.

Table 5. Performance Variation Under Different Sliding Window Lengths

Window size	Accuracy	Precision	Recall	F1-score
15	95.41 %	94.07	96.94	95.48
45	95.80 %	93.87	98.00	95.89
AC-Seq2Net (30)	**97.24 %**	**96.38**	**98.16**	**97.26**

4.6. Performance comparison with previous study

The results of the above studies are summarized in Table 6, demonstrating that our proposed AC-Seq2Net reaches the highest overall performance despite the high noisiness and messiness of the input data.

Table 6. Performance comparison with previous studies

Author	Data	Model	Acc.	Prec.
Pagliaro et al. [38]	S&P 500	ExtraTrees + TI	86.08	86.14
Chen & Tsai et al. [17]	OHLC	CNN	90.70	–
Chen et al. [24]	Yahoo Finance	VAR (rolling)	96.99	–
Rida et al. [39]	Apple images	CNN–LSTM	86.00	0.92
AC-Seq2Net	**S&P 500**	**1D-CNN+BiLSTM+Attn**	**97.70**	**96.42**

Previous methods are found to have positive outcomes but most are based on the simpler feature configurations or less demanding representations. Pagliaro et al. applies an Extra Trees machine with technical indicators and reports the accuracy of 86.08% which suggests that handcrafted signals are useful but can fail to represent the complete geometric structure of chart patterns [38]. Chen and Tsai use real OHLC and test a CNN and optimize results to 90.7% but CNN based models are primarily looking at local trends and may not capture long range temporal correlations

[17]. A VAR based rolling model with GFNN at Yahoo finance gives a result of 96.99% but that is tested on another source and has no direct connection to high frequencies window noise [24]. Rida et al uses CNN LSTM on images of Apple indicators and claims a success rate of 86 percent [39]. Conversely, AC-Seq2Net produces an accuracy of 97.70% and a precision of 96.42% in the S&P500 stocks data where regime shifts and non-stationarity occur rapidly. The improvement is instigated by the architecture design. The dual stage 1D CNN eliminates the high frequency noise but also maintains the important shape information that the Bi LSTM learns the time-dependent transition of the window and the attention layer highlights the most informative parts.

4.7. Class Separation in Latent Space Using AC-Seq2Net

AC-Seq2Net obviously learns a discriminative latent space which separates the Triangle patterns from the Noise with noticeable clustering in the t-SNE projection. Blue points (Noise = 0) form a relatively compact dominant cluster on the right side of the plot while red points (Triangle = 1) appear as several concentrated sub-clusters. In Figure 10, a small overlap region around the center indicates a small number of ambiguous samples where triangle cues are weak.

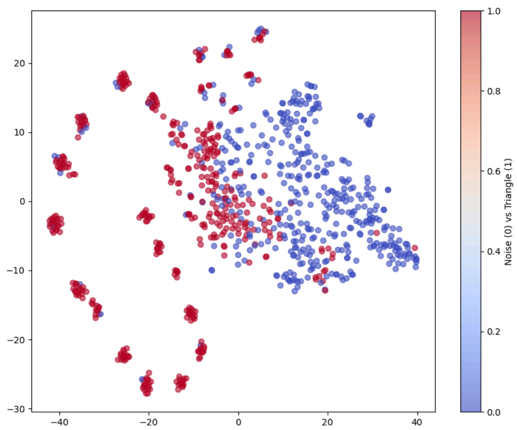

Figure 10. Latent Feature Clusters Visualization

4.8. SHAP Summary Analysis of AC-Seq2Net for Symmetrical Triangle Recognition

Figure 11 presents a SHAP summary plot which describes the contribution of the individual input features to the output of AC-Seq2Net in symmetrical triangle recognition. The features are ranked in terms of global significance, and a point corresponds to one sample. The horizontal axis indicates the SHAP value with positive values indicating augmented confidence in the model and negative values indicating diminished confidence in the model.

According to the plot, the use of AC-Seq2Net is the most dependent on normalized price descriptors and geometric validity cues. Norm Low and Slope H have the most effect indicating that the model focuses on stable price scaling and trendline activity in accordance with a symmetrical triangle. The Squeeze feature also shows a well-defined contribution range, which allows the contraction of volatility to be considered as an important confirmation signal. Moreover, Norm Volume can also be used as complementary evidence. In general, SHAP distribution supports the idea that AC-Seq2Net uses interpretable signals and pattern-relevant signals to make its predictions.

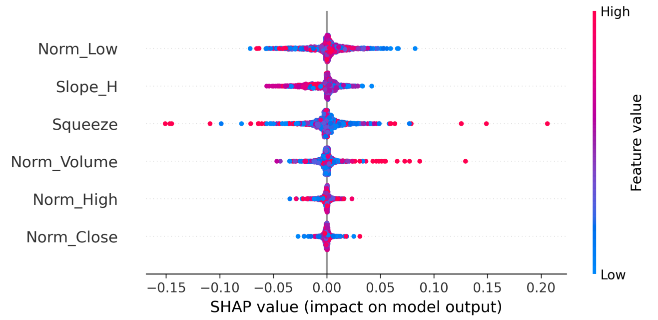

Figure 11. SHAP Summary for AC-Seq2Net Predictions

5. Discussion

A critical issue in the recognition of symmetrical triangles by automated methods is the selection of temporal context because fixed rolling windows can cut formation clues short or add confounding fluctuations. Window based detection hence involved an inherent precision recall trade off. Short windows may exclude early boundary construction, and increase the missed detection, while long windows expand the contextual coverage, and admit more non pattern movements which may resemble partial triangles.

AC Seq2Net reduces this sensitivity by balancing between geometry consistent descriptors and sequence modeling. The geometric features store the convergence and contraction of volatility in a manner compatible with the formal definition of a symmetrical triangle, whereas the hybrid CNN BiLSTM backbone learns how these cues change over time. The attention mechanism further reinforces the discrimination by giving an additional weight to informative formation phases like boundary interactions and squeeze segments. This design allows for a more balanced precision recall profile compared to pipelines only working on normalized price sequences. The SHAP analysis also supports this interpretation by showing that boundary related behavior and trendline geometry cues are the strongest in the decision process.

From a point of view of deployment, this pipeline is suitable for streaming use due to the need for inference which just needs a most recent rolling window and the lightweight feature computation. This makes the method applicable for near real time monitoring and alerting. The contribution of AC Seq2Net extends beyond the predictive performance by proposing explainable pattern-based diagnostics. Because the decision logic can be audited using attribution on geometry linked inputs, the model is able to provide structured evidence of the compression of volatility and boundary behaviour which may often precede abrupt regime transitions.

6. Conclusion and Future work

This study introduces AC-Seq2Net, a geometry-aware and explainable framework for symmetrical triangle recognition that combines rolling-window preprocessing, mathematically aligned feature engineering, hybrid sequence learning, and SHAP-based interpretability. Although the constituent deep learning blocks, namely 1D-CNN, BiLSTM, and Attention, are individually established, their integration with explicit geometric priors enables robust and interpretable recognition of symmetrical triangle patterns in noisy multiyear financial time series. Empirically, the 97.70% accuracy and 0.9947 AUC of AC Seq2Net are compared. It also achieves 99.08% recall and 96.42% precision which indicates that it is sensitive to true triangles while controlling false alarms.

Across different comparative evaluations, AC Seq2Net performs well against representative baselines such as Random Forest, Simple CNN, and Simple LSTM. In addition, SHAP based global explanations confirm that boundary related behavior and trendline geometry cues are dominant drivers of the model decisions strengthening the interpretability of the pipeline. Future work will generalize the same math guided design on other consolidation patterns and multi class recognition under other market regimes. We will also investigate cross market transferability and robustness under distribution shift and there will be a focus on streaming evaluation and automated drift detection. Finally, stability checks of attribution across assets and time will be added to enhance interpretability that the learned geometric evidence is stable under evolving market microstructures.

References

[1] Liang M, Wu S, Wang X, Chen Q. A stock time series forecasting approach incorporating candlestick patterns and sequence similarity. Expert Systems with Applications. 2022 11;205:117595.
[2] Cagliero L, Fior J, Garza P. Shortlisting machine learning-based stock trading recommendations using candlestick pattern recognition. Expert Systems with Applications. 2023 4;216:119493.
[3] Hung CC, Chen YJ. DPP: Deep predictor for price movement from candlestick charts. PLOS ONE. 2021 6;16:e0252404.
[4] Birogul S, Temur G, Kose U. YOLO Object Recognition Algorithm and "Buy-Sell Decision" Model Over 2D Candlestick Charts. IEEE Access. 2020;8:91894-915.
[5] Natarajan J, Wang W, Jiang Y, Zhang Z, Ye H, Kuang L. Generative-CNN for Pattern Recognition in Finance. In: Proceedings of the 5th ACM International Conference on AI in Finance. ACM; 2024. p. 142-9.
[6] Uzun I, Lobachev M, Kharchenko V, Schöler T, Lobachev I. Candlestick Pattern Recognition in Cryptocurrency Price Time-Series Data Using Rule-Based Data Analysis Methods. Computation. 2024 6;12:132.
[7] Nikolaou K. CPC-SAX: Data mining of financial chart patterns with symbolic aggregate approXimation and instance-based multilabel classification. The Journal of Finance and Data Science. 2024 12;10:100132.
[8] Mersal ER, Karaoğlan KM, Kutucu H. Enhancing market trend prediction using convolutional neural networks on Japanese candlestick patterns. PeerJ Computer Science. 2025 2;11:e2719.
[9] Liu P, Zhang Y, Bao F, Yao X, Zhang C. Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading. Applied Intelligence. 2023 1;53:1683-706.
[10] Jing L, Kang Y. Automated cryptocurrency trading approach using ensemble deep reinforcement learning: Learn to understand candlesticks. Expert Systems with Applications. 2024 3;237:121373.
[11] Patel M, Goyal S, Jariwala K, Chattopadhyay C. D2CTNet: An Approach to Detect Dynamics of Candlestick charts and Technical indicators for Stock Market. In: Proceedings of the 8th International Conference on Data Science and Management of Data (12th ACM IKDD CODS and 30th COMAD). ACM; 2024. p. 192-200.
[12] Hung CC, Chen YJ, Guo SJ, Hsu FC. Predicting the price movement from candlestick charts: a CNN-based approach. International Journal of Ad Hoc and Ubiquitous Computing. 2020;34:111.
[13] Kim N, Lee J, Kang Y. ISEPT: Image-Based Selection and Execution Framework for Pair Trading. In: Proceedings of the 6th ACM International Conference on AI in Finance. ACM; 2025. p. 413-21.
[14] Ranjan P, Itani R, Faccia A. An Interpretable 1D-CNN Framework for Stock Price Forecasting: A Comparative Study with LSTM and ARIMA. FinTech. 2025 11;4:63.
[15] Sezer OB, Gudelek MU, Ozbayoglu AM. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing. 2020 5;90:106181.
[16] Olorunnimbe K, Viktor H. Deep learning in the stock market—a systematic survey of practice, backtesting, and applications. Artificial Intelligence Review. 2023 3;56:2057-109.

[17] Chen JH, Tsai YC. Encoding candlesticks as images for pattern classification using convolutional neural networks. Financial Innovation. 2020 12;6:26.

[18] Jin G, Kwon O. Impact of chart image characteristics on stock price prediction with a convolutional neural network. PLOS ONE. 2021 6;16:e0253121.

[19] Lin Y, Liu S, Yang H, Wu H, Jiang B. Improving stock trading decisions based on pattern recognition using machine learning technology. PLOS ONE. 2021 8;16:e0255558.

[20] Liu LC, Si YW. 1D convolutional neural networks for chart pattern classification in financial time series. The Journal of Supercomputing. 2022;78(12):14191-214.

[21] Zheng Y, Si YW, Wong R. Feature extraction for chart pattern classification in financial time series. Knowledge and Information Systems. 2021 7;63:1807-48.

[22] Wan Y, Si YW. A formal approach to chart patterns classification in financial time series. Information Sciences. 2017 10;411:151-75.

[23] Zeng Z, Chen Y. Identifying and Forecasting Recurrently Emerging Stock Trend Structures via Rising Visibility Graphs. Forecasting. 2025 6;7:26.

[24] Chen J, Wen Y, Nanehkaran YA, Suzauddola MD, Chen W, Zhang D. Machine learning techniques for stock price prediction and graphic signal recognition. Engineering Applications of Artificial Intelligence. 2023 5;121:106038.

[25] Foroutan P, Lahmiri S. Deep learning-based spatial-temporal graph neural networks for price movement classification in crude oil and precious metal markets. Machine Learning with Applications. 2024 6;16:100552.

[26] Uygun Y, Sefer E. Financial asset price prediction with graph neural network-based temporal deep learning models. Neural Computing and Applications. 2025 10;37:25445-71.

[27] Lee J, Stevens N, Han SC. Large Language Models in Finance (FinLLMs). Neural Computing and Applications. 2025 10;37:24853-67.

[28] Qiu J, Wang B, Zhou C. Forecasting stock prices with long-short term memory neural network based on attention mechanism. PLOS ONE. 2020 1;15:e0227222.

[29] Lu W, Li J, Wang J, Qin L. A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications. 2021 5;33:4741-53.

[30] Wang C, Chen Y, Zhang S, Zhang Q. Stock market index prediction using deep Transformer model. Expert Systems with Applications. 2022 12;208:118128.

[31] Akhter T, Saona P, Abraham R, Azad MAK. Navigating the Storm: How the EU's Non-financial Reporting Directive and Board Structure Curb ESG Controversies. Journal of Interdisciplinary Research in Business, Education and Society. 2023;1(2):1-18.

[32] Černevičienė J, Kabašinskas A. Explainable artificial intelligence (XAI) in finance: a systematic literature review. Artificial Intelligence Review. 2024 7;57:216.

[33] Khan FS, Mazhar SS, Mazhar K, AlSaleh DA, Mazhar A. Model-agnostic explainable artificial intelligence methods in finance: a systematic review, recent developments, limitations, challenges and future directions. Artificial Intelligence Review. 2025 5;58:232.

[34] Zangana HM, Sulaiman MA. Explainable Transformer Models for Early Prediction of Chronic Diseases Using Longitudinal Electronic Health Records (EHRs). EAI Endorsed Transactions on AI and Robotics. 2026 2;5.

[35] Masud SB, Sozib HM, Mishu KP, Bellal RB, Ahmed MT, Jony AM, et al. AI-Driven Predictive Maintenance in Infrastructure and Facilities Management. EAI Endorsed Transactions on AI and Robotics. 2025 12;5.

[36] Jinbo H, Wang Z, Huang W, Zhou F, Wang A, Wu X, et al. An Efficient Hybrid Model With Harris Hawks Optimization Algorithm for Predicting Oat Water. EAI Endorsed Transactions on AI and Robotics. 2026 3;5.

[37] Elkhirani SE. 5-Year Data for S&P 500 & NASDAQ 100: Historical Data (2019–2024) 5 min Time Frame. Kaggle Dataset. 2024.

[38] Pagliaro A. Forecasting Significant Stock Market Price Changes Using Machine Learning: Extra Trees Classifier Leads. Electronics. 2023 11;12:4551.

[39] Rida SM, Hamza E, Taher Z. From Technical Indicators to Trading Decisions: A Deep Learning Model Combining CNN and LSTM. International Journal of Advanced Computer Science and Applications. 2024;15.