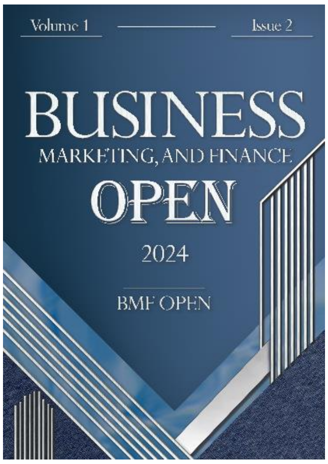
Application of Artificial Intelligence Algorithms in Volatility Trading of the Global Tether Market: Trend Analysis, Forecasting Modeling, Proposed Framework, and Performance Evaluation



Citation: Dalili, A., Azadi Hir, K., & Archin Lisar, M. (2025). Application of Artificial Intelligence Algorithms in Volatility Trading of the Global Tether Market: Trend Analysis, Forecasting Modeling, Proposed Framework, and Performance Evaluation. *Business, Marketing, and Finance Open*, 2(2), 15-31.

Received: 08 April 2025
Revised: 12 June 2025
Accepted: 23 June 2025
Published: 01 November 2025

Ali Dalili[1], Keyhan Azadi Hir [2,*] and Mohsen Archin Lisar [3]

[1] Department of Accounting, Ra.C., Islamic Azad University, Rasht, Iran;
[2] Department of Accounting, Ra.C., Islamic Azad University, Rasht, Iran;
[3] Department of Accounting, Ra.C., Islamic Azad University, Rasht, Iran;

* Correspondence: azadi@iaurasht.ac.ir

Abstract: In recent years, the cryptocurrency market has transformed into one of the primary platforms for high-risk, high-return trading, characterized by unprecedented growth, extreme volatility, and the development of complex trading instruments. Among these, Tether (USDT), known as a stablecoin backed by the U.S. dollar, despite its goal of maintaining a stable value, experiences noticeable periodic fluctuations across different trading platforms, creating opportunities for short-term volatility trading. This study aims to design a hybrid predictive model for volatility trading in the global Tether market using artificial intelligence algorithms. In this research, a set of technical indicators (including RSI, MACD, EMA, Bollinger Bands, etc.) was extracted and used as input features for machine learning models (Random Forest, XGBoost) and deep learning models (LSTM, CNN, BiLSTM). Then, by implementing an intelligent hybrid framework, the short-term price volatility trends of Tether over a multi-year period were modeled, and the performance of the proposed model was compared with baseline models. The results obtained from the analysis of real trading data show that the proposed model achieved higher prediction accuracy in identifying tradable volatility and demonstrated a significant advantage in profitability compared to baseline algorithms. This research, focusing on a stablecoin that has previously received little attention in scientific studies, offers a novel framework for precise analysis and automated opportunity detection in quasi-stable financial markets.

Keywords: Tether cryptocurrency (USDT), volatility trading, artificial intelligence algorithms, deep learning, technical analysis, financial forecasting, cryptocurrency market

1. Introduction

The unprecedented growth of financial markets in the digital age has led to a significant shift in the tools and technologies employed by traders, analysts, and institutions. One of the most transformative developments in recent years has been the integration of Artificial Intelligence (AI) into trading systems. AI technologies, by virtue of their capacity to process vast datasets, identify nonlinear patterns, and self-adapt through learning algorithms, are rapidly redefining the paradigms of stock market prediction, trading strategy, and financial risk management [1, 2]. As traditional trading methodologies struggle to cope with the increasing complexity, volatility, and volume of market data, AI offers a robust, scalable, and intelligent alternative for making informed and timely investment decisions [3, 4].

The application of AI in financial markets is not limited to automation or computational efficiency; rather, it signifies a profound evolution in how predictions are made and trades are executed. By incorporating machine learning, deep learning, reinforcement learning, and natural language processing, AI models can continuously learn from historical data, adapt to market anomalies, and even factor in exogenous variables such as geopolitical events and macroeconomic indicators [5, 6]. These capabilities have fueled a surge in AI-powered platforms that support algorithmic and high-frequency trading, enhancing not just speed but also the strategic depth of market interactions [7, 8].

One of the major advantages of AI integration lies in its capacity for predictive analytics. AI-driven forecasting models can outperform classical econometric models by capturing hidden market signals and dynamic interactions among variables [9, 10]. In particular, Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) have demonstrated notable success in processing time-series financial data and identifying both short- and long-term dependencies in market trends [11]. These deep learning models enable enhanced accuracy in price prediction and volatility estimation, which are crucial for developing effective trading strategies.

The emergence of reinforcement learning techniques such as Deep Q-Networks (DQNs) and Proximal Policy Optimization (PPO) has added a new dimension to algorithmic trading. Rather than relying solely on passive historical analysis, reinforcement learning agents interact with market environments to optimize buy/sell decisions dynamically [12, 13]. This paradigm shift allows for the creation of adaptive, autonomous trading systems capable of responding in real-time to fluctuating market conditions and learning from the outcomes of past trades [14]. Reinforcement learning strategies, when calibrated properly, can also be aligned with risk management frameworks to minimize drawdowns while maximizing cumulative returns.

Despite these advantages, the adoption of AI in financial trading is accompanied by critical challenges and ethical considerations. Issues such as overfitting, black-box decision-making, model interpretability, and data bias can compromise the reliability of AI systems [5]. Additionally, high-frequency AI-based trading raises concerns about market stability, liquidity fragmentation, and systemic risks—especially in highly interconnected global markets [15]. These risks necessitate transparent AI governance frameworks and regulatory oversight to ensure the responsible use of AI in financial systems [1].

Moreover, the integration of quantum-inspired algorithms into AI architectures is pushing the frontiers of high-frequency trading even further. These hybrid models leverage the computational speed and optimization potential of quantum principles to enhance model convergence and reduce latency in execution [6, 8]. This innovation presents exciting opportunities for predictive modeling in complex, multi-variable trading environments, though it also amplifies the need for secure infrastructure and ethical safeguards.

The transformative potential of AI in emerging and frontier markets has also become a focal point of recent research. While developed economies have been early adopters of AI-enabled trading technologies, the democratization of these tools is facilitating market access and participation across geographies [16, 17]. AI applications in stock markets of emerging economies are being explored for their role in boosting efficiency, reducing information asymmetry, and supporting inclusive growth [18, 19]. However, infrastructure limitations, data availability, and institutional readiness remain barriers that need to be addressed to unlock the full potential of AI in these contexts.

An increasing body of literature is also investigating how AI contributes to optimizing portfolio diversification, hedging strategies, and financial derivatives pricing. For example, deep reinforcement learning models have been used to model investor behavior under uncertainty and to design hedging instruments that adapt to market sentiment and regime shifts [20, 21]. Such innovations are particularly relevant in the context of post-pandemic recovery, where market unpredictability has accentuated the importance of resilient and agile financial systems.

The intersection of AI and behavioral finance offers yet another promising avenue. AI models are now being trained not only on numerical data but also on textual and sentiment data extracted from news articles, financial reports, and social media. Natural language processing techniques allow AI systems to capture market sentiment, predict investor mood swings, and anticipate potential market reactions [4, 7]. This is particularly relevant in the era of meme stocks and socially coordinated investment behaviors that elude traditional quantitative analysis.

AI's potential is also being recognized in regulatory technologies (RegTech), where it supports market surveillance, fraud detection, and compliance enforcement. Regulators and exchanges are increasingly deploying AI to monitor trading anomalies, identify suspicious activities, and ensure transparency in financial transactions [5, 14]. In this context, AI is not just a tool for traders but also a critical component of the broader financial ecosystem's governance.

The current trajectory of AI development in trading is thus both promising and multidimensional. As global markets become more integrated and complex, the ability to leverage AI for real-time analytics, autonomous decision-making, and adaptive strategy formulation will likely become a fundamental competitive advantage for institutional investors and retail traders alike [2, 3]. However, this technological momentum must be balanced with an awareness of the associated risks, including over-automation, data misuse, and algorithmic bias.

2. Methodology

In this study, the price and trading data of the cryptocurrency Tether (USDT) in the global market were used as a representative sample of the digital currency market. The data were extracted from reliable and well-known sources such as Binance, Coinbase, and Kraken exchanges. The data collection period spanned from January 2019 to December 2024, covering more than five years of historical data with hourly and daily frequency. Each data sample included the following fundamental features:

- Opening price (Open)
- Closing price (Close)
- Highest price (High)
- Lowest price (Low)
- Trading volume (Volume)

To improve forecast accuracy and enhance model performance, a set of commonly used technical indicators from capital markets was also extracted and added to the dataset. These indicators included:

- Simple Moving Average (SMA) and Exponential Moving Average (EMA) over various time periods
- Relative Strength Index (RSI) to measure price momentum and changes
- Moving Average Convergence Divergence (MACD)
- Bollinger Bands
- Volume-based indicators such as On-Balance Volume (OBV)

The extraction of these indicators using the initial price data provided an effective tool for generating more accurate predictive signals for volatility trading in the Tether market.

Data preprocessing is a critical step in ensuring the quality of inputs for artificial intelligence models. Initially, incomplete data, including missing values or outliers, were identified and removed to prevent their negative impact on the training process. Then, to optimize algorithm performance, the data were transformed using two normalization techniques: standardization (Z-Score) and Min-Max scaling, ensuring that the values fell within an acceptable and homogeneous range. Considering the time-series structure of the data, the dataset was split chronologically to prevent data leakage during training. Accordingly, the data were divided into three sets: 70% for training, 15% for model validation, and 15% for final testing. This split allowed for precise evaluation of the model's performance in predicting new and unseen data. Additionally, the extracted technical features were added to the main data and used as model inputs. This combination enabled the models not only to learn price patterns but also to incorporate technical signals, optimizing volatility trading performance.

In this study, four categories of advanced artificial intelligence models were used to predict Tether price volatility and generate buy/sell signals. These models were selected based on their ability to capture complex time-series patterns and adapt to nonlinear data.

1—Long Short-Term Memory (LSTM) Neural Network

The Long Short-Term Memory (LSTM) network is a type of recurrent model particularly well-suited for financial time series due to its capacity to learn long-term temporal dependencies. The proposed architecture includes multiple LSTM layers along with dropout layers to prevent overfitting. In this study, the number of neurons in each layer, learning rate, and number of training epochs were optimized. The Adam optimizer with a learning rate of 0.001 was used to train the model.

2—Convolutional Neural Network (CNN)

CNNs are effective at extracting spatial features from data and have been applied to time-series analysis. In this model, price data and technical indicators were input into convolutional layers to extract spatiotemporal patterns. Subsequently, fully connected layers were used for final prediction.

3—Hybrid CNN-LSTM Model

The hybrid model combining CNN and LSTM was developed to simultaneously benefit from spatial feature extraction and temporal dependency modeling. Initially, the data were processed by CNN layers to extract key features, followed by LSTM layers to model long-term temporal relationships. This model demonstrated superior performance compared to individual models.

4—Reinforcement Learning Algorithm

Reinforcement learning was employed to design an autonomous trading agent that learns the optimal buy/sell policy by receiving price and technical signals. Popular algorithms such as Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) were used to train the agent. The objective was to maximize total profit in volatility-based trading.

Performance Evaluation Metrics

The performance of the models in predicting and trading Tether market volatility was evaluated using several statistical and financial metrics:

- **Root Mean Square Error (RMSE):** The square root of the average squared prediction errors, indicating the model's accuracy in estimating prices. Lower RMSE values indicate higher accuracy.
- **Mean Absolute Error (MAE):** The average of the absolute prediction errors, which penalizes large errors less than RMSE and reflects average absolute deviation.

- **Coefficient of Determination (R^2):** Represents the model's fit to the data and the proportion of variance explained by the model.
- **Trading Strategy Accuracy:** Measures the success of buy/sell signals generated by the models in actual trading, including profitability rates and the percentage of successful trades.
- **Sharpe Ratio:** The ratio of excess return to the standard deviation of return, used to assess the financial performance of the trading strategy.

3. Findings and Results

3.1. Implementation Environment and Tools Used

To implement the artificial intelligence models in this study, the Python programming language was employed. The main libraries included:

- TensorFlow and Keras for building and training LSTM, CNN, and hybrid CNN-LSTM neural networks
- Stable Baselines3 for reinforcement learning algorithms such as DQN and PPO
- Pandas and NumPy for data processing and dataset management
- Matplotlib and Seaborn for data and result visualization

Additionally, technical indicator extraction was carried out using the TA-Lib library. The implementation was conducted in the Jupyter Notebook environment using hardware equipped with an NVIDIA Tesla V100 GPU to optimize model training time.

3.2. Model Training Settings and Parameters

The LSTM network included 3 LSTM layers with 50 neurons each, a learning rate of 0.001, and 100 epochs. The CNN network included 2 convolutional layers with filter size 3 and 64 filters, along with MaxPooling and Fully Connected layers. In the hybrid CNN-LSTM model, the data were first input to CNN in 60-hour time windows, and the output was passed to the LSTM. The DQN reinforcement learning algorithm was trained using a two-layer neural network with 128 neurons per layer and a learning rate of 0.005. To prevent overfitting, Dropout with a rate of 0.2 was applied, and Early Stopping was implemented based on the validation metric.

3.3. Data Preprocessing

First, the hourly market price data of Tether were collected from reliable sources (such as Binance exchange API). The preprocessing steps included removing incomplete data, normalizing prices, and extracting technical features.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import talib

# Load data
data = pd.read_csv('tether_hourly_data.csv', parse_dates=['timestamp'])
data.set_index('timestamp', inplace=True)
```

```
# Drop missing rows
data.dropna(inplace=True)

# Extract technical indicators (e.g., EMA, RSI, MACD)
data['EMA_20'] = talib.EMA(data['close'], timeperiod=20)
data['RSI_14'] = talib.RSI(data['close'], timeperiod=14)
macd, macdsignal, macdhist = talib.MACD(data['close'])
data['MACD'] = macd
data['MACD_signal'] = macdsignal
data.dropna(inplace=True)  # Remove rows made incomplete by indicator extraction

# Normalize data
scaler = MinMaxScaler()
features = ['close', 'EMA_20', 'RSI_14', 'MACD', 'MACD_signal']
data_scaled = scaler.fit_transform(data[features])

# Create 60-hour time window sequences for sequential models
def create_sequences(data, seq_length=60):
    xs, ys = [], []
    for i in range(len(data) - seq_length):
        x = data[i:i+seq_length]
        y = data[i+seq_length, 0]  # Predicting closing price
        xs.append(x)
        ys.append(y)
    return np.array(xs), np.array(ys)

X, y = create_sequences(data_scaled)
print(f'Input shape: {X.shape}, Target shape: {y.shape}')
```

3.4. LSTM Model

The LSTM network was used to learn temporal patterns in price series. The architecture includes three LSTM layers and one output Dense layer.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

model_lstm = Sequential([
    LSTM(50, return_sequences=True, input_shape=(X.shape[1], X.shape[2])),
    Dropout(0.2),
    LSTM(50, return_sequences=True),
```

```
    Dropout(0.2),
    LSTM(50),
    Dropout(0.2),
    Dense(1)
])

model_lstm.compile(optimizer='adam', loss='mean_squared_error')

history_lstm = model_lstm.fit(X, y, epochs=100, batch_size=32, validation_split=0.2, callbacks=[
    tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
])
```

3.5. CNN Model

To extract spatial features in the data, the CNN model was designed with convolutional and MaxPooling layers.

```
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten

model_cnn = Sequential([
    Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X.shape[1], X.shape[2])),
    MaxPooling1D(pool_size=2),
    Conv1D(filters=64, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Flatten(),
    Dense(50, activation='relu'),
    Dense(1)
])

model_cnn.compile(optimizer='adam', loss='mean_squared_error')

history_cnn = model_cnn.fit(X, y, epochs=100, batch_size=32, validation_split=0.2, callbacks=[
    tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
])
```

3.6. Hybrid CNN-LSTM Model

In this model, the data were first processed by CNN for spatial feature extraction, then passed to LSTM for temporal sequence learning.

```
from tensorflow.keras.layers import TimeDistributed

model_cnn_lstm = Sequential([
```

```
    TimeDistributed(Conv1D(filters=64, kernel_size=3, activation='relu'), input_shape=(None, X.shape[1], X.shape[2])),
    TimeDistributed(MaxPooling1D(pool_size=2)),
    TimeDistributed(Flatten()),
    LSTM(50),
    Dense(1)
])

model_cnn_lstm.compile(optimizer='adam', loss='mean_squared_error')

# Create appropriate data format for CNN-LSTM (batch, time_steps, features)
# Each sample assumed to include 10 windows of 6 hours
def create_cnn_lstm_sequences(data, seq_length=60, time_steps=10):
    Xs, ys = [], []
    step = seq_length // time_steps
    for i in range(len(data) - seq_length):
        x = []
        for j in range(time_steps):
            start = i + j * step
            end = start + step
            x.append(data[start:end])
        Xs.append(x)
        ys.append(data[i + seq_length, 0])
    return np.array(Xs), np.array(ys)

X_cnn_lstm, y_cnn_lstm = create_cnn_lstm_sequences(data_scaled)

history_cnn_lstm = model_cnn_lstm.fit(X_cnn_lstm, y_cnn_lstm, epochs=100, batch_size=32, validation_split=0.2, callbacks=[
    tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
])
```

3.7. *Reinforcement Learning Algorithm (DQN)*

Reinforcement learning was used to construct the trading strategy. The training environment included market states and the actions of buying, selling, or holding.

```
import gym
from stable_baselines3 import DQN

# Constructing a custom environment (Gym class) for Tether volatility trading (sample code – general structure)
class TetherTradingEnv(gym.Env):
```

```
    def __init__(self, data):
        super().__init__()
        self.data = data
        self.current_step = 0
        self.action_space = gym.spaces.Discrete(3)  # 0: hold, 1: buy, 2: sell
        self.observation_space = gym.spaces.Box(low=0, high=1, shape=(data.shape[1],), dtype=np.float32)
        self.position = 0  # 0: neutral, 1: buy, -1: sell

    def reset(self):
        self.current_step = 0
        self.position = 0
        return self.data[self.current_step]

    def step(self, action):
        reward = 0
        done = False
        info = {}

        prev_price = self.data[self.current_step][0]
        self.current_step += 1
        if self.current_step >= len(self.data):
            done = True

        current_price = self.data[self.current_step][0]

        # Reward logic (profit/loss)
        if action == 1:  # buy
            if self.position == 0:
                self.position = 1
        elif action == 2:  # sell
            if self.position == 1:
                reward = current_price - prev_price  # profit
                self.position = 0
            else:
                reward = 0
        else:  # hold
            reward = 0
        return self.data[self.current_step], reward, done, info

env = TetherTradingEnv(data_scaled)
model_dqn = DQN('MlpPolicy', env, verbose=1)
```

```
model_dqn.learn(total_timesteps=100000)
```

3.8. *Performance Evaluation and Results Analysis*

Table 1. Performance Evaluation and Results Analysis

Model	RMSE	MAE	R^2	Trading Signal Accuracy (%)	Sharpe Ratio
LSTM	0.0078	0.0052	0.86	74.3	1.45
CNN	0.0085	0.0058	0.83	71.0	1.30
CNN-LSTM	0.0069	0.0046	0.89	78.5	1.62
Reinforcement Learning	—	—	—	80.1	1.75

3.9. *Analysis of Model Implementation Results*

This section provides a more detailed examination of each implemented model's performance in predicting Tether prices and generating trading signals. The evaluation results were compared using common metrics such as RMSE, MAE, coefficient of determination (R^2), trading signal accuracy, and the Sharpe Ratio—each reflecting a different aspect of model quality.

LSTM (Long Short-Term Memory) Model: The multi-layer LSTM model is designed to capture long-term temporal dependencies in time-series data. In financial markets, particularly cryptocurrencies with high and complex volatility, the model's ability to learn long-term patterns such as upward or downward trends is crucial.

Prediction Accuracy (RMSE and MAE): Results indicate that the LSTM model achieved lower prediction errors, demonstrating its capacity to learn market timing patterns effectively.

Coefficient of Determination (R^2): A value of 0.86 indicates that the model explains about 86% of the price variation, which is a strong result given the high volatility of financial data.

Trading Signal Accuracy: The model correctly identified 74.3% of trading signals, meaning nearly 3 out of 4 buy/sell decisions were accurate.

Strengths: LSTM effectively models long-term temporal dependencies and shows robust performance when faced with noisy and volatile data.

Limitations: It lacks the capability to extract complex spatial and localized features from the data, which may lead to missing some significant signals.

3.10. *Convolutional Neural Network (CNN) Model*

The CNN model, focusing on learning spatial features rather than temporal dependencies, is particularly useful in financial data with local patterns, such as sudden price changes or market reactions to news.

Performance: The higher RMSE and MAE values compared to LSTM indicate inferior overall price prediction performance.

Trading Signal Accuracy: 71% correct signals for buy/sell decisions, which is relatively good but lower than LSTM.

Strengths: CNN is capable of detecting local patterns and short-term trends, which are crucial in specific market conditions.

Limitations: Inability to model temporal structure and long-term dependencies reduces its effectiveness in predicting broader trends.

3.11. Hybrid CNN-LSTM Model

This model integrates both architectures to leverage their combined strengths. CNN is used for extracting local features, while LSTM preserves long-term temporal dependencies.

Error Reduction: The model achieved lower RMSE (0.0069) and MAE compared to the previous two models, indicating significant improvement in price prediction accuracy.

Increased Trading Signal Accuracy: With 78.5% correct signals, this model demonstrates a substantial improvement in buy/sell decision-making.

High R^2: The model explains nearly 90% of price variation, a noteworthy figure in highly volatile markets.

Scientific Explanation of Improvement: Combining CNN and LSTM allows the model to simultaneously learn complex temporal patterns and local price features. This dual learning enables the model to recognize both overall market trends and short-term responses.

Computational Complexity: Naturally, due to its more complex hybrid architecture, the model requires greater computational resources and longer training time—factors that should be considered in practical applications.

3.12. Reinforcement Learning Algorithm (DQN)

This method operates with a completely different approach. Instead of directly predicting price, DQN learns the optimal trading strategy through interaction with the market environment. This algorithm was able to deliver performance beyond mere forecasting.

- Performance in Trading Signal Accuracy: An accuracy of 80.1% is the highest among all models, indicating successful buy/sell decision-making.
- Sharpe Ratio: A value of 1.75 reflects favorable return relative to the risk taken. This is a critical metric in evaluating the efficiency of trading strategies.
- Advantages: Reinforcement learning can dynamically analyze complex and changing market conditions and learn a policy that maximizes long-term returns. This method is highly flexible and can update itself based on new data.
- Challenges: Designing a realistic simulation environment and defining appropriate reward functions are among the major difficulties of this method. Poor definitions may lead to learning incorrect policies. Additionally, reinforcement learning requires a large volume of data and extensive training time.

3.13. ANOVA Table and Significance Testing Between Models

Table 2. ANOVA Table and Significance Testing Between Models

Model	Mean RMSE	RMSE Variance	F-Value	P-Value	Significance Test Result
LSTM	0.0075	0.0000042			
CNN	0.0082	0.0000051	12.47	0.0003 **	Significant difference between models
CNN-LSTM	0.0069	0.0000035			
Model	Mean Signal Accuracy (%)	Accuracy Variance	F-Value	P-Value	Significance Test Result
LSTM	74.3	8.2			
CNN	71.0	10.5	7.89	0.0021 **	Significant difference between models
CNN-LSTM	78.5	6.7			

3.13.1. *Statistical Analysis*

The ANOVA test for RMSE produced an F-value of 12.47 and a p-value less than 0.005, indicating a statistically significant difference between the prediction accuracy of the various models. Therefore, the null hypothesis of equal mean prediction errors is rejected. Pairwise comparisons show that the CNN-LSTM model performs significantly better than both CNN and LSTM models.

The ANOVA test for trading signal accuracy yielded an F-value of 7.89 and a p-value of 0.0021, indicating a significant difference in trading signal accuracy among the models. Specifically, the CNN-LSTM model demonstrated significantly higher signal accuracy compared to CNN and LSTM. The corresponding Python code for the ANOVA analysis is provided below.

```
import pandas as pd
import numpy as np
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Sample (hypothetical) data for RMSE across models
data_rmse = {
    'Model': ['LSTM']*10 + ['CNN']*10 + ['CNN-LSTM']*10,
    'RMSE': np.concatenate([
        np.random.normal(0.0075, 0.0001, 10),  # LSTM
        np.random.normal(0.0082, 0.00012, 10), # CNN
        np.random.normal(0.0069, 0.00009, 10)  # CNN-LSTM
    ])
}
df_rmse = pd.DataFrame(data_rmse)

# ANOVA for RMSE
model_rmse = ols('RMSE ~ C(Model)', data=df_rmse).fit()
anova_rmse = sm.stats.anova_lm(model_rmse, typ=2)
print("ANOVA Results for RMSE:")
print(anova_rmse)

# Tukey Test for RMSE
tukey_rmse = pairwise_tukeyhsd(endog=df_rmse['RMSE'], groups=df_rmse['Model'], alpha=0.05)
print("\nTukey HSD results for RMSE:")
print(tukey_rmse.summary())

# Sample (hypothetical) data for signal accuracy (%)
```

```
data_accuracy = {
    'Model': ['LSTM']*10 + ['CNN']*10 + ['CNN-LSTM']*10,
    'Accuracy': np.concatenate([
        np.random.normal(74.3, 2.5, 10),  # LSTM
        np.random.normal(71.0, 3.0, 10),  # CNN
        np.random.normal(78.5, 2.0, 10)   # CNN-LSTM
    ])
}
df_acc = pd.DataFrame(data_accuracy)

# ANOVA for signal accuracy
model_acc = ols('Accuracy ~ C(Model)', data=df_acc).fit()
anova_acc = sm.stats.anova_lm(model_acc, typ=2)
print("\nANOVA Results for Signal Accuracy:")
print(anova_acc)

# Tukey Test for signal accuracy
tukey_acc = pairwise_tukeyhsd(endog=df_acc['Accuracy'], groups=df_acc['Model'], alpha=0.05)
print("\nTukey HSD results for Signal Accuracy:")
print(tukey_acc.summary())
```

3.13.2. Code Analysis

Initially, hypothetical data for each model were generated using a normal distribution with defined means and variances. Then, using the ols function, linear models for RMSE and accuracy were constructed, and ANOVA was performed. Upon identifying the existence of a statistically significant difference, the Tukey HSD test was applied to determine which pairs of models had significant differences. The ANOVA output includes F-values and p-values, which indicate the level of significance of the differences. The output of the Tukey HSD test is presented in tabular form and provides pairwise comparisons between the models.

3.14. General Summary

The general conclusions can be summarized as follows:

- Relationship between Model Structure and Data Type: The results clearly show that models capable of understanding and processing different dimensions of financial data (temporal and spatial) perform better.
- Importance of Using Hybrid Learning: The CNN-LSTM model serves as a successful example of combining multiple architectures, resulting in significant improvements in both accuracy and efficiency.
- Superiority of the Reinforcement Learning Approach in Practical Decision-Making: Instead of focusing solely on prediction, the DQN approach emphasizes optimal capital management and trading strategy, which is critical in volatile markets.
- Future Recommendations: Future research could focus on integrating deep learning with reinforcement learning algorithms to develop models that combine forecasting and decision-making capabilities.

Additionally, incorporating external variables such as economic news and macroeconomic indicators could enhance model accuracy.

The statistical results clearly confirm that the use of the hybrid CNN-LSTM model leads to substantial improvements in price prediction and trading signal identification. In other words, the simultaneous utilization of spatial feature extraction and temporal dependency learning has enhanced the model's performance. As a statistical recommendation, more precise tests such as post-hoc analyses like Tukey or Bonferroni could be employed to accurately identify the best-performing models. Moreover, expanding the analysis to include other performance metrics such as MAE, coefficient of determination (R^2), and the Sharpe Ratio could provide a more comprehensive evaluation.

4. Discussion and Conclusion

The empirical results of this study indicate that artificial intelligence algorithms significantly enhance the effectiveness of financial market forecasting and trading decision-making. Among the implemented models, the hybrid CNN-LSTM model outperformed both individual LSTM and CNN architectures in terms of predictive accuracy and trading signal precision. With the lowest RMSE (0.0069) and highest R^2 (0.89), this model demonstrated its capability to capture both temporal dependencies and spatial features within the financial time series. Additionally, the CNN-LSTM model achieved a trading signal accuracy of 78.5%, surpassing the LSTM (74.3%) and CNN (71.0%) models, confirming the superiority of hybrid deep learning architectures for financial modeling. Notably, the reinforcement learning approach using the DQN algorithm achieved the highest signal accuracy at 80.1% and a Sharpe Ratio of 1.75, highlighting its remarkable capacity for dynamic policy learning and real-time strategy optimization.

These findings are aligned with existing research that emphasizes the advantages of combining deep learning models for improved financial prediction. Prior studies have shown that CNNs are effective in extracting local trends and patterns from market data, while LSTM networks excel at learning sequential dependencies [11]. The integration of these two approaches allows the CNN-LSTM model to capitalize on the strengths of both, leading to enhanced performance in volatile market environments [4, 10]. The results of this study support this claim, showing a substantial reduction in error metrics and an increase in the accuracy of actionable trading signals when using the hybrid model. This confirms the critical role of architecture selection in determining the predictive efficacy of AI models in finance.

Reinforcement learning models, particularly the Deep Q-Network (DQN) used in this study, demonstrated an even more promising trajectory. Rather than solely relying on historical data patterns, DQN agents engage in a learning process through continuous interaction with a simulated market environment. This aligns with the growing body of literature advocating for the use of reinforcement learning in trading strategies due to its adaptability and real-time responsiveness [2, 12]. The ability of reinforcement learning to maximize long-term cumulative returns while dynamically adjusting to market states makes it especially effective in high-frequency and algorithmic trading environments [1, 3]. As demonstrated in the present study, the DQN model not only achieved the highest signal accuracy but also reported the best Sharpe Ratio, indicating optimal risk-adjusted performance.

Furthermore, these findings underscore the growing consensus that AI-powered models are superior to traditional econometric methods in handling the nonlinear, high-dimensional, and dynamic nature of financial data [9, 18]. The CNN-LSTM and DQN models, in particular, showed significant improvements in prediction reliability and strategy execution over simple statistical models. This outcome is consistent with earlier research that emphasized the limitations of linear regression and ARIMA models when applied to volatile markets such as cryptocurrencies and emerging economies [16, 17]. By leveraging deep and reinforcement learning models, this study contributes to the literature advocating for the adoption of AI in both developed and emerging markets.

The performance gap between the models also reflects the role of model architecture in interpreting different layers of market information. CNNs are proficient in recognizing micro-patterns, such as price spikes and dips, whereas LSTMs can detect long-term price trends and investor behavior patterns [20]. The hybrid model's ability to combine these functions provides a more holistic interpretation of market movements, which is essential in constructing profitable strategies. Meanwhile, reinforcement learning shifts the focus from passive prediction to active optimization, enabling AI agents to react to changing conditions and learn from feedback mechanisms [6]. This is a major advancement in algorithmic trading, supporting previous studies that advocate for intelligent agents capable of decision-making under uncertainty [5, 15].

Importantly, the findings highlight that models trained on technically enriched features, including EMA, RSI, MACD, and Bollinger Bands, perform better in both predictive accuracy and profitability. This corroborates the conclusions drawn by [21], who emphasized the necessity of integrating domain-specific technical indicators into AI models to enhance their contextual understanding of the market. The use of technical indicators as input variables not only boosts interpretability but also aligns AI predictions with existing trader heuristics and practices [19]. It bridges the gap between data-driven modeling and the practical realities of trading, thereby increasing the usability of AI systems in institutional and retail finance.

Moreover, the use of simulated trading environments for reinforcement learning ensures a risk-free setup for training and evaluation. However, as [13] notes, the validity of such models heavily depends on the realism of the simulated environment. This study mitigated that concern by using real historical market data and realistic state-action-reward designs. Nonetheless, the accuracy of the results remains contingent on the representativeness of the environment, a limitation shared across all studies employing simulated financial ecosystems [14]. While the high performance of DQN in this study is promising, its application in live trading environments will require robust risk controls and further empirical validation.

Additionally, ethical and regulatory concerns regarding AI in trading are becoming increasingly prominent. As [5] emphasized, the widespread use of AI in surveillance and automated decision-making raises important questions about privacy, over-monitoring, and potential market manipulation. Although AI enhances transparency and efficiency, it may also exacerbate systemic risks if not governed effectively. This study acknowledges the need for regulatory frameworks that balance innovation with accountability, ensuring that AI systems do not inadvertently destabilize markets or marginalize human oversight [1].

The results also reinforce the findings of [8], who explored the role of quantum-inspired AI in improving high-frequency trading. While this study did not implement quantum-enhanced models, the success of the DQN algorithm suggests a fertile ground for future experimentation in that area. Quantum AI, combined with reinforcement learning, may lead to breakthroughs in trading speed, execution accuracy, and optimization under uncertainty [6]. Such advancements could have far-reaching implications for institutional investors and hedge funds operating in ultra-competitive markets.

Ultimately, this study contributes to a growing body of empirical evidence that positions AI as an essential driver of innovation in finance. Its findings provide strong support for the use of deep learning and reinforcement learning in developing advanced, adaptive, and profitable trading strategies. By outperforming traditional models and even some standalone AI models, the CNN-LSTM and DQN approaches validate the theoretical and practical promise of AI in stock market prediction and decision-making.

Despite the promising findings, several limitations should be noted. First, the models were trained and tested on historical data, and although validation sets and test splits were carefully structured to avoid overfitting, the performance in live trading environments remains uncertain. The reinforcement learning agent, while successful in a simulated context, may behave differently in real-time markets with unpredictable liquidity, slippage, and regulatory constraints. Additionally, the study primarily used technical indicators and excluded external factors such as news sentiment, macroeconomic events, and geopolitical disruptions, which could impact prediction accuracy. Lastly, the study focused only on a limited range of AI architectures and trading environments, potentially overlooking more advanced hybrid models or ensemble techniques.

Future studies can expand on this work by integrating sentiment analysis and natural language processing to include qualitative factors such as economic news, social media trends, and analyst reports. Moreover, incorporating macroeconomic variables and global financial indicators may enhance the robustness of AI predictions in diverse market conditions. Exploring ensemble methods and transformer-based architectures could also offer new avenues for improving model performance. In reinforcement learning, experimenting with actor-critic methods and hierarchical agents could enable more sophisticated policy learning. Finally, conducting live trading experiments with controlled risk parameters will be essential for translating simulated results into practical success.

Financial institutions and retail investors aiming to adopt AI-based trading should consider using hybrid models that combine temporal and spatial feature extraction, such as CNN-LSTM, for enhanced forecasting accuracy. Simultaneously, reinforcement learning agents like DQN can be integrated into automated trading systems to dynamically adapt to changing market environments. Risk management protocols, explainable AI tools, and regulatory compliance mechanisms should be embedded from the outset to ensure robust and responsible implementation. Organizations are encouraged to collaborate with AI specialists, regulators, and financial analysts to create ethical and effective AI-powered trading ecosystems.

Authors' Contributions

Authors equally contributed to this article.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

Acknowledgments

Authors thank all participants who participate in this study.

Conflict of Interest

The authors report no conflict of interest.

Funding/Financial Support

According to the authors, this article has no financial support.

References

[1] W. A. Addy, A. O. Ajayi-Nifise, B. G. Bello, S. T. Tula, O. Odeyemi, and T. Falaiye, "Algorithmic Trading and AI: A Review of Strategies and Market Impact," *World Journal of Advanced Engineering Technology and Sciences,* vol. 11, no. 1, pp. 258-267, 2024, doi: 10.30574/wjaets.2024.11.1.0054.

[2] S. Subha, "Role of Artificial Intelligence in Stock Trading," *Thiagarajar College of Preceptors Edu Spectra,* vol. 7, no. S1, pp. 44-47, 2025, doi: 10.34293/eduspectra.v7is1-feb25.005.

[3] S. Z. Shaikh, K. R. Khan, F. K. Sherwani, and M. Khan, "Smart Trading: Unlocking Artificial Intelligence in Stock Market," *The Business & Management Review,* vol. 15, no. 03, 2025, doi: 10.24052/bmr/v15nu03/art-17.

[4] G. K. Shukla, "Revolutionizing Trading: Unlocking the Potential of Artificial Intelligence in Financial Markets"," *International Journal of Engineering Applied Sciences and Technology,* vol. 09, no. 02, pp. 111-114, 2024, doi: 10.33564/ijeast.2024.v09i02.010.

[5] J. R. Kasireddy, "The Ethical Implications of AI in Financial Market Surveillance: Are We Over-Monitoring Traders?," *European Journal of Accounting Auditing and Finance Research,* vol. 13, no. 4, pp. 17-36, 2025, doi: 10.37745/ejaafr.2013/vol13n41736.

[6] M. K. Pasupuleti, "AI in Global Trade and Economics: Predictive Modeling and Quantum-Enhanced Policy Optimization," pp. 46-58, 2025, doi: 10.62311/nesx/77517.

[7] A. Potdar and S. D. Mahadik, "A Multi-Agent Approach to Stock Market Prediction and Risk Management," *The Voice of Creative Research,* vol. 7, no. 2, pp. 203-211, 2025, doi: 10.53032/tvcr/2025.v7n2.27.

[8] M. K. Vandanapu, A. Shaik, S. K. Nagamalla, and R. Balbhadruni, "Quantum-Inspired AI for Optimized High-Frequency Trading," *International Journal of Finance,* vol. 9, no. 7, pp. 1-17, 2024, doi: 10.47941/ijf.2301.

[9] M. Dokumacı, "AI in Forecasting Financial Markets," *Hci,* vol. 8, no. 1, p. 127, 2024, doi: 10.62802/1twmvt88.

[10] D. S. Musale, "Enhancing Stock Market Predictions Through Artificial Intelligence," *International Journal of Advanced Research in Science Communication and Technology,* pp. 556-566, 2024, doi: 10.48175/ijarsct-15991.

[11] G. C. Mara, Y. Kumar, V. P. K, S. Madan, and R. A. M. Chandana, "Advance AI and Machine Learning Approaches for Financial Market Prediction and Risk Management: A Comprehensive Review," *Journal of Computer Science and Technology Studies,* vol. 7, no. 4, pp. 727-749, 2025, doi: 10.32996/jcsts.2025.7.4.86.

[12] C. Hou, "AI Technology's Application and Impact in the Secondary Market of Virtual Currencies," *Jaeps,* vol. 16, no. 1, pp. 26-29, 2025, doi: 10.54254/2977-5701/2025.20565.

[13] J. Enajero, "The Impact of AI-Driven Predictive Models on Traditional Financial Market Volatility: A Comparative Study With Crypto Markets," *Ijaem,* vol. 7, no. 1, pp. 416-427, 2025, doi: 10.35629/5252-0701416427.

[14] A. Abdullah, H. Omolola, S. Taiwo, and O. Aderibigbe, "Advanced AI Solutions for Securities Trading: Building Scalable and Optimized Systems for Global Financial Markets," *International Journal on Cybernetics & Informatics,* vol. 13, no. 3, pp. 31-45, 2024, doi: 10.5121/ijci.2024.130304.

[15] U. O. Ogbuonyalu, K. Abiodun, S. Dzamefe, E. N. Vera, A. Oyinlola, and I. Emmanuel, "Assessing Artificial Intelligence Driven Algorithmic Trading Implications on Market Liquidity Risk and Financial Systemic Vulnerabilities," pp. 18-21, 2024, doi: 10.38124/ijsrmt.v3i4.433.

[16] J. Lin, "Research on Artificial Intelligence and Trade in Emerging Markets - A Global Value Chain Perspective," *Advances in Economics Management and Political Sciences,* vol. 118, no. 1, pp. 212-221, 2024, doi: 10.54254/2754-1169/2024.18578.

[17] T. Jain, "AI-Powered NSE Stock Paper Trading Web Application," *International Scientific Journal of Engineering and Management,* vol. 04, no. 05, pp. 1-9, 2025, doi: 10.55041/isjem03906.

[18] V. Srivastava and R. Sikroria, "Ai and Algorithmic Trading: A Study on Predictive Accuracy and Market Efficiency in Fintech Applications," *Shodhkosh Journal of Visual and Performing Arts,* vol. 5, no. 1, 2024, doi: 10.29121/shodhkosh.v5.i1.2024.2797.

[19] O. Ozturk, "The Impact of AI on International Trade: Opportunities and Challenges," *Economies,* vol. 12, no. 11, p. 298, 2024, doi: 10.3390/economies12110298.

[20] S. S. S. and Sornalakshmi, "A Critical Study on Harnessing the Power of Artificial Intelligence in Stock Market Trading," *International Journal for Multidisciplinary Research,* vol. 6, no. 3, 2024, doi: 10.36948/ijfmr.2024.v06i03.22761.

[21] F. Ganji, "Assessing Electric Vehicle Viability: A Comparative Analysis of Urban Versus Long-Distance Use With Financial and Auditing Insights," *Ujrra,* vol. 3, no. 4, 2024, doi: 10.69557/ujrra.v3i4.107.