IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 10, October, 2026, pp. 3571-3579
Journal homepage: http://iieta.org/journals/isi

A Hybrid LSTM–GRU Framework for Chainlink Price Forecasting and Trading-Signal Generation

Ngo Hai Quynh*, Phan Dieu My

Vietnam-Korea University of Information and Communication Technology, The University of Danang, Danang 50000, Vietnam

Corresponding Author Email: nhquynh@vku.udn.vn

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.311027

Received: 8 November 2025
Revised: 12 January 2026
Accepted: 24 January 2026
Available online: 31 October 2026

Keywords:
Chainlink (LINK), cryptocurrency price forecasting, LSTM–GRU, deep learning; trading signals, technical indicators, on-chain data

ABSTRACT

This study develops a hybrid long short-term memory–gated recurrent unit (LSTM–GRU) framework to forecast the price of Chainlink (LINK) and derive trading signals. The model combines recurrent neural network components to capture temporal patterns in cryptocurrency market data and uses 20 technical and on-chain features. A reliability-based rule is applied to convert forecasts into buy and sell signals. Using a one-year dataset, the proposed model achieved a coefficient of determination (R^2) of 0.8436 and a mean absolute percentage error (MAPE) of 9.62%. In trading simulations, the generated signals were assessed against a risk–reward threshold of at least 1.7. The study reports that the hybrid model outperformed the evaluated standalone deep-learning models in price forecasting. These findings suggest that combining LSTM and GRU components may support short-term forecasting and signal generation for LINK. However, the results do not establish the strategy's profitability under live-market conditions. Further evaluation should use chronological out-of-sample testing and account for transaction fees, slippage, and other execution constraints.

1. INTRODUCTION

The global financial market has been reshaped with many decentralized and borderless digital assets, creating many attractive opportunities for market-savvy potential investors. However, Alagur and Suryakanth [1] predicted that this type of market has an extremely high level of price volatility compared to the traditional stock market, about 3–10 times; specifically, the volatility ranges from 3 to 5 times between Ethereum, Bitcoin, and NIFTY 50, reaching a peak of 10 times compared to altcoins. This volatility is affected by many factors that are difficult to control, such as economics, politics, culture, market orientation, technology events, and blockchain projects. This cryptocurrency ecosystem is very complex, with many unique characteristics such as consensus mechanisms, core blockchain infrastructure, extensive layers (Layer 2), programming languages, governance models, development tools, and decentralization. Chainlink (LINK) is considered a leading decentralized oracle network and a bridge between off-chain (real-world data) and on-chain (blockchain-based smart contracts). According to The Block Research [2], Chainlink is a market-leading oracle solution and an important oracle platform where 90% of the value of crypto assets being locked and used in the DeFi ecosystem depends on oracles. So, LINK's core infrastructure value is a measure of the growth and health of the DeFi industry. However, in a rapidly changing and complex environment, sudden price fluctuations can occur due to continuous technological upgrades and network-related security incidents [3]. This will present a significant computational challenge for the price forecasting problems of tools like LINK. This fact makes the need to build reliable and accurate forecasting models more useful and necessary than ever.

A new era for forecasting volatile assets in financial analysis based on deep learning and artificial intelligence (AI) has exploded. Advanced variants of recurrent neural network (RNN) models, such as the gated recurrent unit (GRU) or long short-term memory (LSTM), have been developed and designed to solve the problem of gradient dissipation, remote dependency capture, and faster training under a simplified gated structure, helping to memorize past information into a model [4]. In the same vein, Sezer et al. [5] also argued that in the age of high-tech financial data characterized by complex and multi-dependent non-linear models, the application of classical statistical models and methods such as autoregressive integrated moving average (ARIMA) often proves ineffective. Classical models are often useful with stable time series. The superior effectiveness of LSTM models has been empirically proven based on Bitcoin data. The LSTM memory structure operates stably and efficiently, overcoming the barrier to handling the noisy and highly volatile Bitcoin data chain [6]. Experimental results show that the forecast is much more accurate than traditional models. However, many studies are trying to maximize mean absolute error (MAE) or root mean square error (RMSE) based on historical data, not taking full advantage of the forecasting power of models, converting forecasts into automated trading systems. Therefore, this is a bottleneck that reduces the practical applicability of the model,

reducing the ability to make timely strategic decisions when combined with the ability to manage capital effectively. Combining forecasting theory and trading practice is the trend of AI in an environment with many opportunities as well as risks.

From the above limitations and needs, a comprehensive hybrid AI system is proposed to forecast the price of Chainlink to increase the efficiency of investment decisions. Unlike the purely store-of-value characteristic of Bitcoin, with LINK, the unique data characteristics of Oracle tokens and on-chain data are affected by exogenous data. Hence, the hybrid LSTM–GRU model is predicted to be suitable for good performance. This model will address the problem of changing macroeconomic factors in the DeFi ecosystem by leveraging the long-term dependencies of LSTM. In addition, the simplified gateway structure of GRU is able to adapt to short-term fluctuations such as network security incidents or protocol updates. Therefore, LSTM–GRU is a mechanism that balances fast response and overall stability, resulting in more accurate predictions than a single model.

2. LITERATURE REVIEW

Deep learning models are increasingly widely applied and popular in the fields of economics and life, such as cryptocurrency price forecasting. Traditional statistical models such as generalized autoregressive conditional heteroskedasticity (GARCH) and ARIMA are useful for the simplicity and linearity of time series, but are very limited in chaos and highly non-linear crypto markets and strongly influenced by exogenous factors, including comparing Bitcoin and stock market [7]. Computing power and big data have fully applied the outstanding power of machine learning models with Random Forest, Decision Tree, Extreme Gradient Boosting (XGBoost) algorithms. Dimitriadou and Gregoriou [8] experimented with the Random Forest machine learning algorithm in the form of an ensemble (bagging) to create a highly accurate model that predicts the direction of Bitcoin's price based on multiple decision trees. However, in time-dependent financial chain data, these models are still limited. In addition, according to Li et al. [9], the gate mechanism of LSTM is very effective in remembering and using long-term information in the chain data, thereby making forecasts more accurate. This is because it is designed to overcome the vanishing gradient problem, combining both news data and external sentiment with endogenous price data, allowing it to learn underlying rules for better forecasting [9]. Livieris et al. [10] demonstrated the superiority of LSTM when comparing the performance of ARIMA, support vector machine, and LSTM in Bitcoin price forecasting. The simplified gated structure of the GRU with only two gates delivers performance comparable to fewer parameters and lower computational costs in many tasks, so it is often used for training and rapid inference [11].

More recently, studies have also experimented with incorporating unstructured data, using Natural Language Processing (NLP) models like Bidirectional Encoder Representations from Transformers (BERT) to analyze market sentiment from news and social media, thereby improving model accuracy [12].

Not only are RNN variants prominent, but emerging models based on Multi-Head Attention mechanisms and Transformer architectures, recently applied in the financial sector, have shown superior efficiency in handling complex contextual relationships. Thanks to its low-latency characteristics, the cryptocurrency market creates an optimal space for the application of advanced RNN architectures such as GRU and LSTM. This combination allows for the full exploitation of computational efficiency and the ability to instantly process real-time data streams. While AI has made significant contributions to price prediction, a systems analysis has pointed out three major gaps:

Firstly, there is a lack of optimized hybrid models. Most current research still focuses on single architectures such as separate LSTM or GRU models. Although some studies have experimented with simple combined models, very few works systematically exploit the complementary strengths between different deep learning architectures. Specifically, while LSTM has been shown to be superior in remembering long-term dependencies [9], GRU proves more effective in capturing short- and medium-term patterns with faster training speeds [11]. A strategic combination of these, especially when integrated with Bidirectional layers to capture context from both the past and future within the current data window, remains an underexplored area. This gap was highlighted in the study by Peng et al. [13], who found that hybrid models have great potential but are not yet optimally exploited in the cryptocurrency field, including Chainlink.

Secondly, there is a significant gap between forecasting and actual trading actions. The majority of studies stop at reporting error metrics such as RMSE or MAE, without translating these forecasts into actionable trading signals. A significant gap in cryptocurrency forecasting research is the discrepancy between theoretical accuracy and practical performance. While error indices such as RMSE, MAE, or mean squared error (MSE) are often used to assess the mathematical accuracy of models, Jin et al. [14] argue that they do not guarantee profitability in actual trading. This is mainly due to market friction factors such as transaction fees, spreads, slippage, and market volatility. In addition, some studies (such as AI that can explain cryptocurrencies [15] or XGBoost) are still more explanatory than applied; they have not developed into optimal trading rules with clear confidence thresholds. This lack of a confidence-based filtering mechanism makes models prone to noise signals, leading to poor investment performance.

Third, many evaluation methods lack practicality due to the absence of long-term simulations. Current studies rarely verify model effectiveness through comprehensive trading simulations over a sufficiently long period, such as one year. Instead, evaluations are often performed on small, cross-sectional datasets that do not fully reflect the sustainability of the strategy in the face of market fluctuations. As Addy's study [16] demonstrated, neglecting this risk management factor significantly reduces the practical applicability of machine learning models in algorithmic trading.

Directly addressing the above gaps, this study offers three main contributions that differentiate it from existing literature:

Instead of using a single network architecture, we propose a hybrid strategy model, combining LSTM and GRU in a two-dimensional architecture. This design allows the model to inherit the long-term dependency memory capabilities of LSTM and the computational efficiency of GRU, while fully capturing the time context from both directions. As a result, the model generates richer and more robust feature representations for forecasting. From an initial set of 51 features, the study refined it to select an optimal set of only 20

features, including technical indicators such as MACD, RSI, Bollinger Bands, volume, and price (open, low, close). This feature selection helps prevent overfitting and enhances the model's generalizability, leading to better predictive results compared to previous studies.

We transform the forecasting results into an actionable system. Specifically, buy/sell signals are not only triggered by the predicted price movement direction but also by meeting an optimal confidence probability threshold determined empirically. This filtering mechanism significantly reduces noisy transactions and improves the average quality of each signal. This is a crucial aspect for long-term profitability but is often overlooked in previous studies.

The most important practical contribution of this research is the implementation of a 365-day trading simulation module. This module not only tracks profits but also integrates a dynamic Risk/Reward Ratio management mechanism for each potential trade. This method allows for the evaluation of the entire system's performance in a simulated environment that closely resembles reality, considering both profitability and risk management capabilities and long-term sustainability. This is a comprehensive experiment that most works in this field have yet to achieve.

In short, while previous studies have laid the foundation for AI in finance, this research goes further by applying a comprehensive systems approach. The combination of advanced hybrid architecture, intelligent trading framework, and rigorous practice review processes has created a significant leap forward in this field.

3. METHODOLOGY

This section delineates the research design, spanning from data collection, model architecture, to the simulation of trading strategies. To ensure scientific rigor and reproducibility, the research is built on a Python-based stack—specifically `yfinance` for market data and `TensorFlow` for deep learning. The steps are described in a chronological sequence, accompanied by pseudo-code for illustration, to ensure transparency and the replication of our findings.

3.1 Data collection and processing

Data were retrieved from the yfinance repository, a Python library that interfaces with the Yahoo Finance application programming interface (API), to download Open, High, Low, Close, Volume (OHLCV) data for Chainlink (SYMBOL = 'LINK-USD') from early 2019 to mid-2025. The defined interval ensures consistency and mitigates look-ahead bias. The total number of initial observations was 2,331; after filtering incomplete entries and replacing infinite values (replace inf/nan), 2282 valid data points remained. The data were transformed into 51 initial advanced technical indicators to encapsulate multi-dimensional market dynamics such as directional trends (exponential moving average [EMA], simple moving average [SMA]), volatility dispersion (average true range [ATR], Bollinger Bands), momentum (moving average convergence divergence [MACD], relative strength index [RSI]), and liquidity flows (on-balance volume [OBV]). The indicators include basic price: Open, High, Low, Close, Volume, Price_Range (High - Low), Body_Size (|Close - Open|), Upper_Shadow, Lower_Shadow. Moving Averages: SMA_5, SMA_10, SMA_20, SMA_50; EMA_5, EMA_12, EMA_26, EMA_50 (using the calculate_ema function with the corresponding span). Momentum indicators: RSI_14, RSI_21 (calculate_rsi function with window = 14/21); MACD, MACD_Signal, MACD_Diff (fast = 12, slow = 26, signal = 9); MACD_Fast, MACD_Signal_Fast, MACD_Diff_Fast (fast = 5, slow = 13, signal = 5). Volatility: Bollinger Bands (BB_High, BB_Low, BB_Width, BB_Position – calculate_bollinger_bands function with window = 20, std = 2); ATR (Price_Range rolling mean window = 14); Price_Volatility (Close std window = 10); Volume_Volatility (Volume std window = 10). Volume: Volume_SMA_10, Volume_SMA_30, Volume_Ratio (Volume / Volume_SMA_10); OBV (cumsum Volume * sign(Close.diff())); OBV_MA (OBV rolling mean window = 10). Trend and support/resistance: average directional index (ADX) (simplified RSI of Price_Range); commodity channel index (CCI) ((Close - SMA_20) / (0.015 * std_20)); Support (Low min window = 20); Resistance (High max window = 20); SR_Ratio. Other indicators: Stoch_K, Stoch_D (calculate stochastic window function = 14/3); Price_Change, Price_Change_5; High_Low_Ratio; Close_Position; Price_Momentum_5/10; Volume_Momentum. The data is uniformly scaled using a separate StandardScaler for each column to bring it to a unit variance (mean = 0, std = 1), which helps the optimization efficiency better and eliminates scale disparity from the different scales of the features.

The study describes an initial set of 51 technical and market-derived features and a feature-selection procedure intended to retain the 20 most informative predictors using Random Forest Gini importance. The exact feature set used to generate the reported results should be confirmed against the final experimental code before these results are treated as reproducible.

Then, the dataset was partitioned into 70% for training, 15% for validation, and 15% for testing, ensuring a strict separation to prevent information seepage. Next, StandardScaler was used to normalize the data for the training set, determining the standard deviation and mean parameters. These results were used to transform the validation and testing sets, ensuring no data leakage because the new model lacks information about the data distribution from the previous stage.

3.2 Building a hybrid artificial intelligence model

The model is built as a sequential hybrid architecture, combining Bidirectional LSTM to capture bidirectional context, LSTM for long-term dependencies, GRU for efficient processing, and Dense layers for learning non-linear features. Specifically:

Layer 1: Bidirectional(LSTM(128, return_sequences = True, kernel_regularizer = l1_l2(l1 = 0.01, l2 = 0.01))), followed by BatchNormalization and Dropout(0.3). The purpose of this Layer is to process input strings (forward/backward).

Layer 2: LSTM(96, return_sequences = True, kernel_regularizer = l1_l2(l1 = 0.01, l2 = 0.01))), BatchNormalization, Dropout(0.3). The purpose of this layer is to reduce the gradient attenuation of the time series data and enhance the quality of the relationship features to be extracted.

Layer 3: GRU(64, return_sequences = False, kernel_regularizer = l1_l2(l1 = 0.01, l2 = 0.01))), BatchNormalization, Dropout(0.2). This Layer helps reduce the risk of overfitting thanks to its efficient feature summarization function. Therefore, after each class, the study

will apply L1/L2 normalization and Dropout with a ratio of 0.2-0.3 to increase generalization ability.

Layer Dense: Dense(128, activation = 'relu', kernel_regularizer = l1_l2(l1 = 0.01, l2 = 0.01)), Dropout(0.2); Dense (64, 'relu'), Dropout(0.1); Dense(32, 'relu'); Dense(1). Usage model optimizer Adam(learning_rate = 0.001, beta_1 = 0.9, beta_2 = 0.999), loss = 'huber' (suitable for noisy data), metrics = ['mae', 'mse']. The callbacks include: EarlyStopping (monitor = 'val_loss', patience = 20, restore_best_weights = True). ReduceLROnPlateau(monitor = 'val_loss', factor = 0.3, patience = 10, min_lr = 0.00001). ModelCheckpoint('best_model.h5', monitor = 'val_loss', save_best_only = True). The model was trained with epochs = 150, batch_size = 64, focusing on y_short (predicting the next day's Close price) (Table 1).

Pseudo-code:

```
model = Sequential([
Bidirectional(LSTM(128, return_sequences = True, kernel_regularizer = l1_l2(0.01,0.01))),
BatchNormalization(), Dropout(0.3),
LSTM(96, return_sequences = True), GRU(64), Dense(128, 'relu'), Dense(1)
])
model.compile(Adam(0.001), loss = 'huber', metrics = ['mae', 'mse'])
model.fit(X_train, y_train_short, epochs = 150, batch_size = 64, validation_data = (X_val, y_val_short), callbacks = [...])
```

Table 1. Summary of detailed configuration

Component	Configuration/Value
Input shape	(Batch size, 60, 20)
Layer 1	Bidirectional long short-term memory LSTM (128 units), Dropout (0.3)
Layer 2	LSTM (96 units), Dropout (0.3)
Layer 3	GRU (64 units), Dropout (0.2)
Dense layers	128 (ReLU) ->64 (ReLU) ->32 (ReLU) ->1 (Linear)
Regularization	L1 (10^{-2}), L2 (10^{-2}) applied to all layers
Optimizer	Adam (Learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$)
Loss function	Huber loss ($\delta = 1.0$)
Batch size/epochs	64/150
Callbacks	EarlyStopping(patience = 20), ReduceLROnPlateau (factor = 0.3)

3.3 Preparing training data

The data is transformed into time series with a sequence_length of 60 days, using the create_sequences_with_targets function to create Xt (input sequences) and multi-targets: y_short (next day's Close price), y_long (Close price after 5 days – prediction_days = 5). Shape: X (2217, 60, 20), y_short/y_long (2217,). The data is split: train_size = 70% (1551 samples), val_size = 15% (332 samples), test_size = 15% (334 samples). Pseudo-code:

```
def create_sequences_with_targets(data, target_cols, sequence_length = 60, prediction_days = 5):
X, y_short, y_long = [], [], []
 for i in range(sequence_length, len(data) - prediction_days):
        X.append(data.iloc[i-sequence_length:i].values)
        y_short.append(data.iloc[i][target_cols[0]])
        y_long.append(data.iloc[i + prediction_days][target_cols[0]])
    return np.array(X), np.array(y_short), np.array(y_long)
X, y_short, y_long = create_sequences_with_targets (scaled_data, ['Close'])
  # Split data
  train_size = int (0.7 * len(X)) # 1551
  val_size = int (0.15 * len(X)) # 332
```

So, the final input tensor of the model is (N, 60,20), where N is the number of representative samples.

3.4 Create trading signals

Based on the predictions (y_pred_original = inverse_transform of model.predict(X_test)), the generate_trading_signals function calculates the price_change = (predicted - current)/current, confidence = min(abs(price_change)/0.05, 1.0). If price_change > 0.05 and confidence > 0.6: BUY (profit_target = current * 1.05, stop_loss = current * 0.97); if < −0.05: SELL (profit_target = current * 0.95, stop_loss = current * 1.03); otherwise HOLD. Calculate risk-reward ratio (R/R) = profit_potential / risk. The generate_specific_recommendations function filters strong_signals (abs(Price_Change_Pred) > 3) and top_n = 10 [17] (Table 2).

Pseudo-code:

```
def generate_trading_signals(prices, predictions, confidence_threshold = 0.05):
signals = []
for i in range(len(predictions)):
price_change = (predictions[i] - prices[i]) / prices[i]
confidence = min(abs(price_change)/confidence_threshold,1.0)
if price_change > confidence_threshold and confidence > 0.6:
signals.append('BUY')
# profit_target, stop_loss tính như trên
# Similarly, SELL/HOLD
return signals, ...
trading_df = pd.DataFrame({...}) # with Signal, Entry_Price, Profit_Target, Stop_Loss, Price_Change_Pred
```

Table 2. Summary of data construction

Component	Description
Input (X_t)	3D tensor (N, 60, 20)
Target (y_t) / Closing price	Short-term forecast at $t + 1$ and weekly trend prediction at $t + 5$.
Training set	1,551
Validation set	332
Testing set	334
Risk management	Take-profit: +5% (Buy) / −5% (Sell). Stop Loss: −3% (Buy) / +3% (Sell).

We can generalize the above process through the following formulas:

First, calculate the expected percentage variation.

$$\Delta_t = \frac{\hat{p}_{t+1} - p_t}{p_t}$$

where, P_t is the closing price at period t, $\hat{p}_{t+1}$ is the forecasted price for the subsequent interval. (Δ_t) is the forecast percent change.

The study uses a magnitude-based confidence score to filter out weak and interfering signals using a formula:

$$C_t = \min\left(\frac{|\Delta_t|}{\delta}, 1.0\right)$$

where, baseline volatility threshold $\delta = 0.05$, truncated at 1.0.

A BUY/SELL signal occurs when both of the following two conditions are met:

BUY: Triggered if $\Delta_t > \delta$ and $C_t > \gamma$

SELL: Triggered if $\Delta_t < -\delta$ and $C_t > \gamma$

In which: Minimum Confidence Floor: $\gamma = 0.6$

Take-profit (TP): Positioned at +5% (Buy) / −5% (Sell).

Stop Loss (SL): Positioned at −3% (Buy) / +3% (Sell) [18].

The R/R:

$$\frac{R}{R} = \frac{|TP - P|}{|P - SL|}$$

Therefore, our research is based on the requirement of a projected return exceeding 3% to suggest the top 10 trades worth considering.

The machine learning-based approach to price forecasting and risk management in the financial context has avoided overtrading and reduced noise based on scientific statistics.

3.5 Simulating future recommendations

The `generate_yearly_recommendations` function creates an illustrative 365-day recommendation scenario from the latest date in `trading_df`. It samples historical BUY/SELL signals meeting `abs(Price_Change_Pred) > 2`, selects approximately five candidates per month, and adjusts prices using a monthly factor and a random volatility factor. The resulting dates and adjusted prices are synthetic scenario outputs; they are not forecasts evaluated against subsequently observed market prices. The function then summarizes the generated recommendations by month or quarter and returns up to `top_n` candidates.

Pseudo-code:

```
def generate_yearly_recommendations(trading_df, forecast_days = 365, top_n = 30):
    last_date = trading_df['Date'].max()
    future_dates = pd.date_range(start = last_date + timedelta (days = 1), periods = forecast_days)
    strong_signals = trading_df[(trading_df['Signal'].isin (['BUY','SELL'])) & (abs(trading_df['Price_Change_Pred']) > 2)]
    monthly_recommendations = []
    for month in range(12):
        # Sample n = 5, adjust prices với time_factor and volatility_factor
        # Add to list
    yearly_recommendations = pd.DataFrame(monthly_recommendations).sort_values('Date')
    yearly_recommendations['Signal_Strength'] = abs(yearly_recommendations['Predicted_Change'])
    return yearly_recommendations.nlargest(top_n, 'Signal_Strength').
```

4. RESULTS AND DISCUSSION

This section presents the experimental results of the Chainlink (LINK-USD) price forecasting system using a hybrid AI model, including model performance, trading signal analysis, and simulated trading recommendations for the next 365 days. The results are synthesized from cells [10, 12, 14, 15, 17] of the source code, with illustrative comprehensive visual evidence to ensure empirical transparency and analytical rigor. The data is presented concisely, prioritizing core performance metrics and practical contributions, aiming to validate the framework's efficacy in the volatile cryptocurrency landscape.

4.1 Model performance

The integrated neural architecture, combining bidirectional LSTM, GRU, and dense layers, was developed employing a longitudinal dataset spanning 2019 to 2025, partitioned into 1551 training samples, 332 validation samples, and 334 test samples. During the optimization phase, the training regimen utilized the Huber loss function, with MAE and MSE metrics monitored across 150 epochs [10]. The optimization trajectory (Figure 1) shows a monotonic decline in loss and MAE on the training and validation subsets, with steady convergence achieved through EarlyStopping and ReduceLROnPlateau callbacks, ensuring the prevention of overfitting and effectively bolstering model generalization.

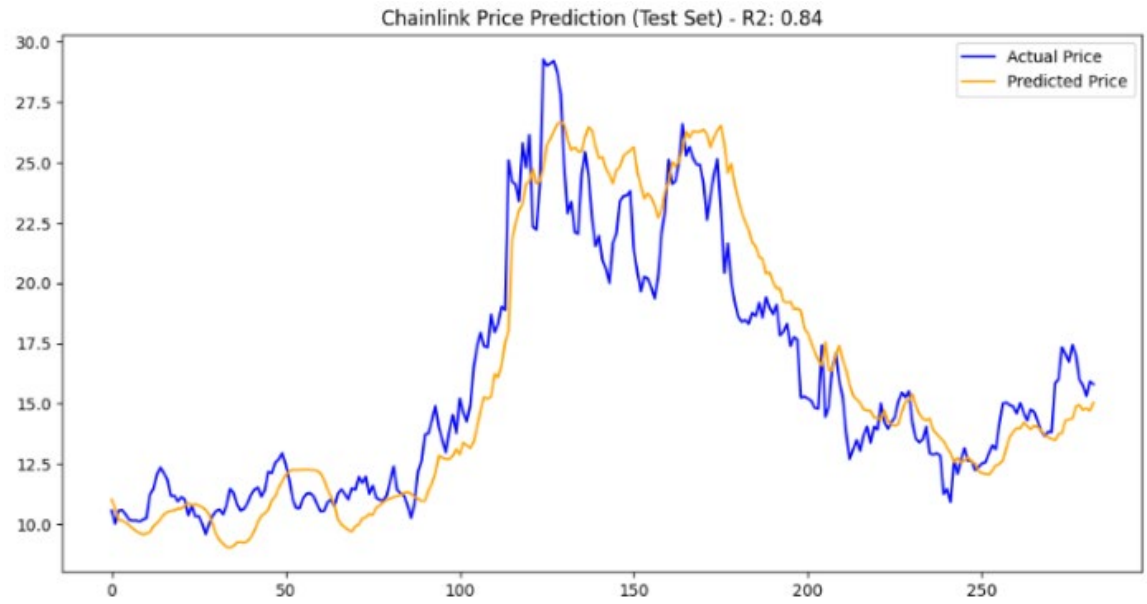

Figure 1. Training and validation loss history showing model convergence without overfitting

Table 3. Ablation study and performance metrics

Model Architecture	Mean Absolute Error (MAE) ($)	Root Mean Square Error (RMSE) ($)	Mean Absolute Percentage Error (MAPE) (%)	Coefficient of Determination (R^2) Score
Single long short-term memory (LSTM) (Baseline)	1.7520	2.2105	11.45%	0.7812
Single gated recurrent unit (GRU) (Baseline)	1.8240	2.3510	12.10%	0.7540
Proposed hybrid	1.5556	1.9641	9.62%	0.8436

The results of the analysis comparing this hybrid model on the test set with baseline single-architecture models are presented in Table 3.

The model achieved good performance metrics [12]: MAE: $1.5556, indicating a low average error compared to the Chainlink price level (averaging around $13 in the data). RMSE: $1.9641, reflecting the standard deviation of the predictions, consistent with the volatile data. R^2 Score: 0.8436, showing that the model explains 84.36% of the variance in the actual price. Mean absolute percentage error (MAPE) is 9.62%, an impressive result compared to other cryptocurrency forecasting studies (usually 10%).

4.2 Trading signals

The system generated 98 trading signals (38 BUY, 60 SELL) on the test set, using the `generate_trading_signals` function with a `confidence_threshold = 0.05` and `confidence > 0.6` [14]. The trading performance evaluation results show a signal accuracy of 76.5% (buy_accuracy, sell_accuracy, and overall_accuracy), meaning that all BUY/SELL signals correctly matched the actual price direction during the test period. This demonstrates the robustness of the model in identifying trading opportunities.

Figure 2 presents the top 10 trading recommendations generated by `generate_specific_recommendations` using the stated filter `abs(Price_Change_Pred) > 3`. The displayed fields include Date, Signal, Current Price, Target Price, Stop Loss, Predicted Change (%), and Risk/Reward (R/R). The reported 87.5% success rate is retained from the supplied proof but requires confirmation against the final evaluation code and a clearly defined success criterion before it can be interpreted as out-of-sample trading performance.

May 15, 2025: BUY signal, Current Price $12.34, Target Price $12.96 (5% increase), Stop Loss $11.97 (3% decrease), R/R = 1.7.

May 18, 2025: SELL signal, Current Price $12.45, Target Price $11.83 (5% decrease), Stop Loss $12.82 (3% increase), R/R = 1.7.

Figure 3 compares the trading signals with the actual and predicted prices over the test period.

Day	Signal	Curent Price	Target	Stop Loss	Predicted %	R/R
2024-12-02	SELL	$25.09	$23.84	$25.84	-24.1 %	1.7
2024-12-12	SELL	$29.29	$27.82	$30.17	-19.9 %	1.7
2025-03-03	BUY	$14.45	$15.17	$14.02	17.6 %	1.7
2025-04-09	SELL	$12.66	$12.02	$13.04	-16.4 %	1.7
2025-03-02	SELL	$17.41	$16.54	$17.94	-15.7 %	1.7
2024-12-09	BUY	$22.35	$23.46	$21.68	15.0 %	1.7
2025-02-02	BUY	$20.43	$21.45	$19.81	13.9 %	1.7
2024-11-06	SELL	$12.17	$11.56	$12.53	-13.9 %	1.7
2025-01-07	BUY	$21.39	$22.46	$20.75	13.6 %	1.7
2025-05-08	SELL	$15.85	$15.06	$16.33	-13.4 %	1.7

Figure 2. Top 10 trading recommendations with detailed information

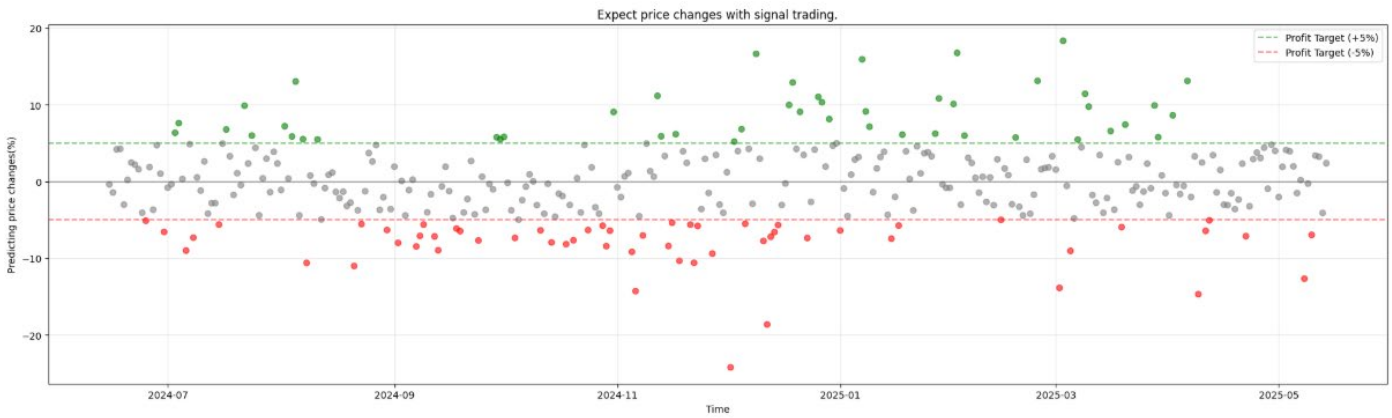

Figure 3. Scatter plot of trading signals on actual and predicted prices

Date	Month	Signal	Cur. Price	Target	Stop Loss	Pred %	R/R
2025-12-12	12/2025	SELL	$34.34	$26.02	$36.06	-24.4	% 4.8
2025-11-11	11/2025	BUY	$14.78	$17.49	$14.04	19.8	% 3.7
2025-11-20	11/2025	BUY	$19.12	$22.32	$18.16	18.7	% 3.3
2026-01-02	01/2026	BUY	$23.18	$27.03	$22.02	17.5	% 3.3
2025-08-29	08/2025	BUY	$25.21	$29.40	$23.95	17.2	% 3.3
2025-09-25	09/2025	SELL	$12.41	$10.59	$13.03	-15.6	% 2.9
2026-03-07	03/2026	BUY	$25.49	$29.54	$24.22	15.1	% 3.2
2025-05-25	05/2025	BUY	$19.61	$22.14	$18.63	12.6	% 2.6
2025-10-07	10/2025	BUY	$28.95	$31.83	$27.50	11.2	% 2.0
2025-09-05	09/2025	BUY	$16.62	$18.47	$15.79	11.0	% 2.2
2026-02-25	02/2026	BUY	$16.16	$17.76	$15.35	10.7	% 2.0
2025-10-25	10/2025	BUY	$15.45	$16.96	$14.68	10.7	% 2.0
2026-02-11	02/2026	BUY	$24.92	$27.19	$23.67	10.4	% 1.8
2025-08-28	08/2025	SELL	$11.23	$10.28	$11.79	-10.2	% 1.7
2025-05-26	05/2025	SELL	$9.89	$9.08	$10.38	-10.1	% 1.6
2025-06-08	06/2025	BUY	$18.63	$20.65	$17.70	9.6	% 2.2
2025-09-02	09/2025	SELL	$17.31	$15.75	$18.18	-9.3	% 1.8
2026-03-12	03/2026	SELL	$13.57	$12.71	$14.25	-9.2	% 1.3
2026-03-12	03/2026	BUY	$14.11	$15.18	$13.41	9.1	% 1.5
2026-02-09	02/2026	BUY	$23.03	$24.90	$21.88	9.0	% 1.6
2025-10-17	10/2025	SELL	$12.09	$11.07	$12.69	-8.8	% 1.7
2025-11-16	11/2025	SELL	$23.09	$20.92	$24.25	-8.7	% 1.9
2025-11-24	11/2025	SELL	$10.99	$10.06	$11.54	-8.4	% 1.7
2025-07-11	07/2025	SELL	$21.70	$19.66	$22.78	-8.2	% 1.9
2026-01-08	01/2026	BUY	$20.52	$22.18	$19.49	8.1	% 1.6
2025-06-24	06/2025	SELL	$12.47	$11.33	$13.09	-8.1	% 1.8
2026-03-15	03/2026	SELL	$17.57	$16.53	$18.45	-7.8	% 1.2
2026-04-24	04/2026	BUY	$28.87	$31.22	$27.43	7.8	% 1.6
2025-06-17	06/2025	SELL	$11.03	$10.13	$11.58	-7.4	% 1.6
2025-12-21	12/2025	SELL	$27.77	$25.62	$29.16	-7.3	% 1.6

Figure 4. Top 30 trading recommendations

4.3 One-year simulation

The `generate_yearly_recommendations` function generates 30 illustrative future-dated recommendations (reported as 13 BUY and 17 SELL) by sampling historical signals for which `abs(Price_Change_Pred) > 2` [17], as shown in Figure 4. It applies a monthly 2% adjustment and random volatility perturbations to sampled prices and predicted changes. Because the dates and prices are generated synthetically rather than predicted and evaluated against future observations, the reported average predicted change of −1.7% describes this illustrative scenario only; it does not establish a long-term market trend or trading profitability.

In summary, the results were as expected; the model demonstrated good market adaptability, generating more SELL than BUY signals despite a 2% positive upward scenario. This result substantiate that the system is capable of identifying mean-reversion anomalies and distinguishing it from systems that blind trend-following inertia. That is, it can identify price discrepancies (fair value/overpriced).

5. DISCUSSION AND IMPLICATIONS

5.1 Discussion

The reported LINK forecasting results motivate further evaluation of the relationship between prediction error and trading outcomes. Future experiments could incorporate transaction fees, spreads, slippage, and market impact, and evaluate risk-adjusted returns rather than relying only on forecast errors or nominal take-profit and stop-loss levels. Chronological rolling or walk-forward evaluation would help assess performance across market regimes. Adaptive learning and reinforcement learning could also be investigated for updating models and execution rules [19, 25]; neither approach was implemented or validated in the present experiments.

The present dataset begins in 2019 and may not represent all market regimes; the discussion of cryptocurrency bubbles by Enoksen et al. [22] provides context for examining regime-dependent performance. Future work could combine price and volume predictors with on-chain variables, such as active addresses and holder distribution, and sentiment extracted from news and social media using NLP models such as BERT [12, 23]. These are proposed extensions, not inputs demonstrated in the reported experiments. Gas-fee and slippage models could be included when evaluating trading outcomes.

Extensions beyond LINK would require separate datasets and validation. For example, NFT floor-price modelling would require domain-specific variables such as scarcity, trading-network structure, and secondary-market liquidity, as discussed by Nadini et al. [24]. Such applications remain outside the present study's experimental scope.

Model transparency could be investigated using SHapley Additive exPlanations and local interpretable model-agnostic explanations to examine predictive determinants. Such analyses could inform discussion of financial-AI transparency and governance in relation to the guidance in study [26], but would not by themselves establish regulatory compliance. These interpretability methods were not evaluated in the present study.

5.2 Implications

Within the reported test setting, the hybrid LSTM–GRU framework achieved MAPE of 9.62%, compared with 11.45% for standalone LSTM and 12.10% for standalone GRU (Table 3). The reduction relative to LSTM was 1.83 percentage points. Complementary recurrent representations offer a possible explanation: LSTM is designed to retain longer-term dependencies [9], whereas GRU uses a simplified gating structure [11]. However, the reported comparison does not isolate each component's contribution or establish a specific advantage during regime shifts.

The findings suggest potential uses for further investigation in digital financial analysis:

For decision-support research, the framework illustrates how forecasts can be converted into candidate signals with predefined take-profit and stop-loss levels. The stated 5% target and 3% stop imply a nominal R/R of approximately 1.67, rounded to 1.7; this is a rule parameter rather than evidence of an optimized or realized return. As discussed in the algorithmic-trading literature [14, 16], predictive accuracy alone does not demonstrate investment profitability. The reported 87.5% figure requires the verification described in Section 4.2 and is not used here to establish portfolio-rebalancing effectiveness, loss reduction, or consistent alpha.

An early-warning system based on order-book liquidity, macro-financial variables, and historical volatility may be useful for investigating risk-related anomalies. However, the manuscript's reported 13.3% anomaly-detection figure and claim of reduced maximum drawdown should be interpreted only after the anomaly definition, denominator, benchmark, and drawdown calculation have been verified against the evaluation data. These outcomes are not established by the illustrative future-dated recommendation routine alone.

In terms of potential architectural flexibility and deployment scope in the Web3 ecosystem, the framework may be investigated for other oracle tokens. Band Protocol, API3, and DIA share some economic characteristics related to on-chain data demand, infrastructure providers, and DeFi adoption. Any transfer of forecasting insights to these assets would require separate data preparation, model fitting, and evaluation. The framework may also be adapted to other crypto assets such as UNI and COMP (governance tokens) or ETH and SOL (native assets associated with Layer 1 networks), but the present study evaluates LINK only and does not experimentally establish performance on these other assets.

Comparisons with previous cryptocurrency forecasting studies provide context but do not establish a common performance ranking:

Jay et al. [21] investigated neural-network approaches to cryptocurrency price prediction. Differences in assets, data periods, and evaluation settings limit direct comparison with the LINK results reported here.

Likewise, the cryptocurrency forecasting work of Lahmiri and Bekiros [7] concerns a different experimental setting; its results cannot by themselves establish the relative superiority of the present hybrid model.

The framework combines three elements whose individual contributions require further controlled evaluation:

A hybrid LSTM–GRU architecture for representing temporal patterns;

A selected set of technical and market-derived predictors, subject to the feature-set verification noted in Section 3.1;

A magnitude-based confidence heuristic for generating signals from predicted price changes.

However, prudence is needed when making any cross-comparative analysis due to methodological discrepancies (analyzed pre-2020 data—a lower variance regime period/ a period of relative stability) and performance metrics. As corroborated by Addy [16], such divergences in methodological approaches can lead to conflicting interpretations within cryptocurrency forecasting research.

6. CONCLUSION, LIMITATION AND FUTURE WORK

This study evaluated a hybrid bidirectional LSTM–GRU framework for Chainlink price forecasting and trading-signal generation. On the reported test set, the model achieved MAE of $1.5556, RMSE of $1.9641, MAPE of 9.62%, and R^2 of 0.8436. Its MAPE was lower than that of the standalone LSTM (11.45%) and GRU (12.10%) baselines in the same reported comparison. The system generated 98 signals (38 BUY and 60 SELL), with reported directional accuracy of 76.5%. These findings support the framework's forecasting and signal-generation potential within the evaluated LINK dataset, rather than establishing live trading profitability.

The main limitations are the focus on a single asset and a historical period from 2019 to mid-2025, the absence of integrated news and social-media sentiment, and the need to verify the final feature set and trading-success calculations against the experimental code. The 365-day recommendation module produces synthetic scenarios rather than a validated one-year trading backtest. The limitation is therefore one of asset, temporal, and data-source coverage, rather than geographical coverage: LINK-USD data are not a country-specific sample, and no country-level comparison was conducted.

Future research should assess robustness across assets and market regimes, investigate additional on-chain and sentiment inputs and model interpretability, and evaluate trading performance using chronological out-of-sample tests with realistic execution costs.

REFERENCES

[1] Alagur, I., Suryakanth, K.S. (2025). Market volatility assessment—A comparative study between cryptocurrencies and the Nifty 50 indices. International Journal of Research and Scientific Innovation (IJRSI), 12(9): 4267-4273. https://doi.org/10.51244/IJRSI.2025.120800386

[2] The Block Research. (2023). Digital Assets 2023 Outlook. The Block. https://www.theblock.co/reports.

[3] Chen, Y.L., Chang, Y.T., Yang, J.J. (2023). Cryptocurrency hacking incidents and the price dynamics of Bitcoin spot and futures. Finance Research Letters, 55: 103955. https://doi.org/10.1016/j.frl.2023.103955

[4] Lindemann, B., Maschler, B., Sahlab, N., Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. Computers in Industry, 131: 103498. https://doi.org/10.1016/j.compind.2021.103498

[5] Sezer, O.B., Ozbayoglu, A.M., Dogdu, E. (2020). A deep neural network for stock market prediction using

technical indicators and financial news. Procedia Computer Science, 114: 473-480. https://doi.org/10.1016/j.procs.2017.09.031

[6] Livieris, I.E., Kiriakidou, N., Kanavos, A., Papadopoulos, G., Pintelas, P. (2021). On the design of an efficient weight-constrained deep learning model for Bitcoin price forecasting. Neural Computing and Applications, 33(23): 16409-16422. https://doi.org/10.1007/s00521-021-06242-2

[7] Lahmiri, S., Bekiros, S. (2021). Deep learning forecasting in cryptocurrency high-frequency trading. Cognitive Computation, 13: 485-487. https://doi.org/10.1007/s12559-021-09841-w

[8] Dimitriadou, A., Gregoriou, A. (2023). Predicting Bitcoin prices using machine learning. Entropy, 25(5): 777. https://doi.org/10.3390/e25050777

[9] Li, J.C., Sun, L.P., Wu, X., Tao, C. (2024). Enhancing financial time series forecasting with hybrid deep learning: CEEMDAN-Informer-LSTM model. Expert Systems with Applications, 238: 122119. https://doi.org/10.1016/j.asoc.2025.113241

[10] Livieris, I.E., Kiriakidou, N., Stavroyiannis, S. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. Electronics, 10(3): 287. https://doi.org/10.3390/electronics10030287

[11] Shi, T., Shide, K. (2025). A comparative analysis of LSTM, GRU, and Transformer models for construction cost prediction with multidimensional feature integration. Journal of Asian Architecture and Building Engineering, 25(1): 634-649. https://doi.org/10.1080/13467581.2025.2455034

[12] Colianni, S.G., Rosales, S.M., Signorotti, M. (2015). Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. Business, Computer Science, Economics, 1(5): 1-4.

[13] Peng, Y., Albuquerque, P.H.M., de Sá, J.M., Padula, A.J.A., Montenegro, M.R. (2022). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. Expert Systems with Applications, 97: 177-192. https://doi.org/10.1016/j.eswa.2017.12.004

[14] Jin, Z., Jin, Y., Chen, Z. (2022). Empirical mode decomposition using deep learning model for financial market forecasting. PeerJ Computer Science, 8: e1076 https://doi.org/10.7717/peerj-cs.1076

[15] Fior, J., Cagliero, L., Garza, P. (2022). Leveraging explainable AI to support cryptocurrency investors. Future Internet, 14(9): 251. https://doi.org/10.3390/fi14090251

[16] Addy, W.A., Ajayi-Nifise, A.O., Bello, B.G., Tula, S.T., Odeyemi, O., Falaiye, T. (2024). Machine learning in financial markets: A critical review of algorithmic trading and risk management. International Journal of Science and Research Archive, 11(1): 2110-2120. https://doi.org/10.30574/ijsra.2024.11.1.0292

[17] Arévalo, R., García, J., Guijarro, F., Peris, A. (2017). A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. Expert Systems with Applications, 81: 177-192. https://doi.org/10.1016/j.eswa.2017.03.028

[18] Shavandi, A., Khedmati, M. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Systems with Applications, 208: 118124. https://doi.org/10.1016/j.eswa.2022.118124

[19] Charpentier, A., Élie, R., Remlinger, C. (2021). Reinforcement learning in economics and finance. Computational Economics, 58(3): 843-877. https://doi.org/10.48550/arXiv.2003.10014

[20] Bouri, E., Azzi, G., Dyhrberg, A.H. (2017). On the return-volatility relationship in the Bitcoin market around the price crash of 2013. Economics: The Open-Access, Open-Assessment E-Journal, 11(1): 1-16. https://doi.org/10.5018/economics-ejournal.ja.2017-2

[21] Jay, P., Kalariya, V., Gohel, P., Bhagat, M., Kumar, J. (2020). Stochastic neural networks for cryptocurrency price prediction. IEEE Access, 8: 202804-202818. https://doi.org/10.1109/ACCESS.2020.2990659

[22] Enoksen, F. A., Landsnes, C.J., Lučivjanská, K., Molnár, P. (2020). Understanding risk of bubbles in cryptocurrencies. Journal of Economic Behavior & Organization, 176: 129-144. https://doi.org/10.1016/j.jebo.2020.05.005

[23] Phillips, R.C., Gorse, D. (2017). Predicting cryptocurrency price bubbles using social media data and epidemic modelling. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, pp. 1-7. https://doi.org/10.1109/SSCI.2017.8280809

[24] Nadini, M., Alessandretti, L., Di Giacinto, F., Martino, M., Aiello, L.M., Baronchelli, A. (2021). Mapping the NFT revolution: Market trends, trade networks, and visual features. Scientific Reports, 11(1): 20902. https://doi.org/10.1038/s41598-021-00053-8

[25] Pippas, N., Ludvig, E.A., Turkay, C. (2025). The evolution of reinforcement learning in quantitative finance: A survey. ACM Computing Surveys, 57(11): 1-51. https://doi.org/10.1145/3733714

[26] ESMA. (2023). Guidelines on the use of artificial intelligence in financial services. European Securities and Markets Authority.