Evaluation of Machine Learning and Deep Learning Investment Strategies

Mai Duc Toan[1], Assem Atallah[2], Dachawar Shrinivas Kamalkishor[3]
[1]*maiductoan130298@gmail.com*, [2]*assembassel@gmail.com*,
[3]*shrid12@yahoo.com*

Abstract

This paper evaluates the accuracy of various Machine Learning methods in predicting stock market and applies them to different stock indices for performance comparison. The results show that while Machine Learning and Deep Learning models achieve high classification accuracy, this does not necessarily guarantee superior trading performance, especially when transaction costs are considered. LSTM, GRU and Trasnformer yield promising results and outperform traditional Machine Learning models due to their ability to capture temporal dependencies in time series data. Model performance is influenced by market efficiency, with stronger results observed in Emerging markets. Finally, hybrid models do not consistently enhance performance, suggesting that their effectiveness depends on market conditions and data complexity.

Keywords: *Stock market prediction, Machine Learning, Deep Learning, Hybrid Models, Trading Strategy, Backtesting*

1. Introduction

Financial markets play an important role in the economy and development of countries. As the center of the financial sector, the stock market—characterized by its volatility and complexity—has attracted considerable attention from investors and scholars in many fields who are developing trading and optimization strategies. Stock market forecasting is attractive because it brings investors profits.

With the advancement of machine learning (ML) and Deep Learning (DL) and the explosion of available data, stock price forecasting has significantly improved. ML and DL models have demonstrated strong potential in detecting complex price patterns and improving prediction accuracy (Gu, Kelly, & Xiu, 2020). However, the application of these models in practice with transaction costs has not been widely implemented. Transaction costs affect trading as they reduce net profits (Tricker et al., 2017).

Beyond transaction costs, another critical gap lies in the lack of comprehensive model comparisons across diverse financial markets. The vast majority of research focuses mainly on only one or compares a small different structure of models, making it difficult for investors to choose the most suitable approach. Moreover, the effectiveness of ML and DL models may vary depending on market characteristics, such as volatility and liquidity, which differ significantly across developed, emerging, and frontier markets.

This paper addresses these gaps by systematically evaluating the performance of various ML and DL investment strategies across developed, emerging, and frontier markets. By leveraging advanced cross-validation techniques and backtesting frameworks, we assess model effectiveness under realistic trading conditions.

The remainder of this paper is organized as follows. Part 2 conducts a literature review of current ML and DL models used in stock price forecasting and trading strategies. Section 3 details the methodology, including data collection, model selection, and evaluation metrics. Section 4 presents the empirical results, followed by discussions in Section 5 and concluding remarks in Section 6.

2. Literature review

William et al.'s 1992 research reveals that investors can predict stock price changes using technical analysis. This raises a heavy debate on the Efficient Market Hypothesis (EMH) (Fama, 1965), which states that investors can not earn excess returns. Since then, more researchers have attempted to conduct stock market prediction research.

At the early stage, researchers mainly focus on the use of technical indicators to predict market prices. Technical analysis is a trading methodology that involves analyzing statistical trends in market activity, such as price movements and trading volumes, to make investment decisions. Technicians believe that market prices reflect all available information and that patterns and trends in price data tend to repeat over time due to consistent investor behavior. Some widely used indicators are: Moving Average (Brock et al, 1992; Hung & Yang, 2013), Moving Average Convergence Divergence (Hung, 2016; Nor, 2014; Rosillo et al, 2013), Relative Strength Index (Nor, 2014; Rosillo et al, 2013). However, the paper by Bustos & Pomares-Quimbaya, 2020 mentioned that Park and Irwin's survey (Park & Irwin, 2007) shows that some technical analysis-based strategies have limited results.

Bustos & Pomares-Quimbaya's review shows that technical indicators and social networks are the most predictive data. Their work also points out that there are missing comparisons between the different stages of development between markets. According to (Subasi et al,2021), the accuracy of the used method can be measured by obtaining the Precision, Recall, F-score and Accuracy for both the normal and the leaked dataset. For that reason, we will conduct the test with technical indicators as it is widely known and easy to compute, and the data would cover a wide range of different financial markets.

Besides that, another branch of research centered on how traditional statistical models can exploit and forecast stock market return, such as Autoregressive Model (AR) (Xiang & Fu, 2006), Autoregressive Integrated Moving Average (ARIMA) (Ariyo et al, 2014) and Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) (Arowolo, 2013). However, Deakin (1972) pointed out that financial data often violate the assumptions of statistical methods. Officer (1972) and Peiró (1994) also point out that stock return does not follow a stable normal distribution. Therefore, recent research has focused on advanced machine learning methods and deep learning models to eliminate the limitations of these assumptions.

Following the advancement of machine learning models and computing power such as Graphics Processing Units (GPUs), machine learning has shown its potential ability to predict stock market prices. This has attracted scholars and researchers, creating a trend and increasing number of papers in recent years in applying machine learning to stock price forecasting (Bustos & Pomares-Quimbaya, 2020). Traditional models such as Support Vector Machine (SVM) (Huang et al, 2005), Decision Tree (DT) and Random Forest (RF) (Imandoust & Mohammad, 2014), K-Nearest Neighbor (k-NN) (Alkhatib et al, 2013), Naive Bayes (NB) (Shihavuddin et al, 2010) and enhanced techniques like Ensemble method (Toochaei & Moeini, 2023; Tsai et al, 2011), Bagging and Boosting method (Wang et al, 2009; Zheng, 2006; Nti et al, 2009; Ribeiro & Coelho, 2020), further improved prediction accuracy and robustness.

In parallel, deep-learning approaches have gained significant traction. Recent deep learning and artificial intelligence approaches also gain sound results. Research varies among models like Artificial Neural Network (ANN) (Kara et al, 2011), Long Short Term Memory (LSTM) (Chen et al, 2015; Fister et al, 2019; Selvin et al, 2017), Recurrent Neural Network (RNN) (Zhu, 2020; Selvin et al, 2017), Convolutional Neural Network (CNN) (Selvin et al, 2017), Deep Reinforcement Learning (DRL) (Tabaro et al, 2020), Transformer (Muhammad et al, 2023; Mao et al, 2024) and hybrid model (Song & Choi, 2023). A summary of ML and DL is shown in Table 1.

Table 1. Overview of ML and DL models

Author	Model	Dataset	Factor
Xiang & Fu	AR MLP	S&P 500 Index	Historical data
Ariyo et al	ARIMA	Nokia stock, Zenith Bank	Historical data
Arowolo	GARCH	Zenith Bank	Historical data
Huang et al	SVM LDA QDA EBNN	NIKKEI 225 index	Macroeconomic S& P 500 USD/JPY Historical data
Imandoust & Mohammad	DT RF Naive Bayes classifier	TSE Index	Technical indicators Oil, gold prices USD/IRR
Alkhatib et al	k-NN	Five listed companies on ASE	Historical data
Shihavuddin et al	Naive Bayes classifier	FTSE 100 Index	Review report on stock prices
Toochaei & Moeini	Ensemble	TSE Index	57 indicators
Tsai et al	MLP CART LR Bagging	Taiwan Economic Journal (TEJ)	Financial ratios Economic indicators
Wang et al	DT Bagging-DT Boosting-DT	SSEC Index SZSE Index	accounting variables
Zheng	Boosting Bagging	Eight stocks and indices	Log-return prices
Nti et al	Ensemble Bagging Boosting	GSE Index NYSE Index BSE Index	Technical indicators
Kara et al	SVM ANN	ISE National 100 Index	Technical indicators
Chen et al	LSTM	China stock market	Historical data
Fister et al	LSTM	Bayerische Motoren Werke AG stock	Technical indicators Date data
Selvin et al	RNN,	1721 NSE-listed	Historical data

	LSTM CNN	companies	
Zhu	RNN	Apple's stock	Historical data
Tabaro et al	Deep RL	Tesla stock	Technical indicators, financial statements, Loughran–McDonald Sentiment Word Lists
Muhammad et al	Transformers	Eight listed companies on DSE	Historical data
Song & Choi	CNN-LSTM GRU-CNN Ensemble	DAX DOW S&P500	Historical data

3. Data

Figure 1 shows the flowchart of the stock price forecasting process based on ML and DL models. It includes steps such as data collection and processing, calculation of technical indicators and model training. The output data is used for accuracy testing and backtesting to assess investment perfromance.

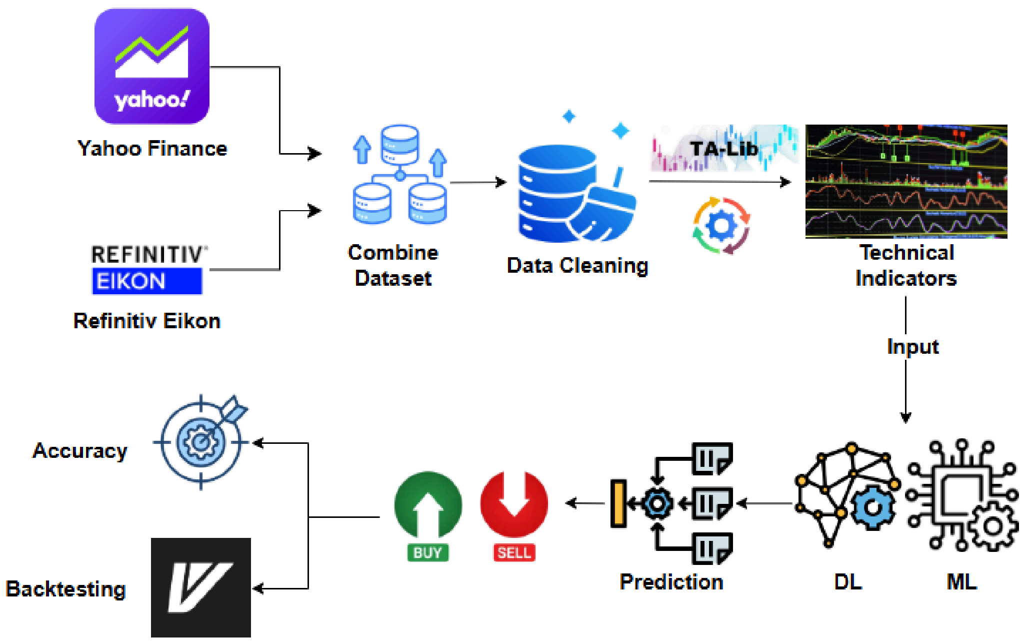


Figure 1. Process Flowchart

3.1. Data Collection

The data used in the study was collected from Yahoo Finance and Refinitiv Eikon, which are reputable data sources used by many researchers. This data includes information on Open, High, Low, Close, and Volume data of the two major indices of each region from each classified market, which are Frontier, Emerging and Developed Market. We classify markets by [MSCI market classification](). The chosen developed markets (US, UK, Japan, Germany) are among the largest in terms of market cap and GDP and are the most interested and invested by investors. They represent a good benchmark for model evaluation. Emerging markets such as China, India, and Brazil are known for rapid growth but also higher volatility, testing the adaptability of the models. Finally, there is not much data about the frontier market so we gather available market data (missing volume or discontinuous data availability). The specific countries and indices used in this study are described in Table 2.

Data details:

- Time range: From 01-01-2000 to 01-01-2025.
- Data type: open, high, low, close prices, and volume daily.

Table 2. Market Indices

Market	Country	Region	Index
Developed	United States	Americas	S&P 500
	Canada		TSX
	United Kingdom	EMEA	FTSE 100
	Germany		DAX
	Japan	APAC	NIKKEI 225
	Hong Kong		HSI
Emerging	Brazil	Americas	BOVESPA
	Mexico		MXX
	Saudi Aribia	EMEA	TASI
	Qatar		QE GENERAL
	China	APAC	SSEC
	Indonesia		JKSE
Frontier	Romania	EMEA	BET
	Vietnam	APAC	VNINDEX

Data is processed to ensure integrity and consistency, including steps such as removing missing data, detecting and handling exceptions, and removing duplicate or invalid records. This data will serve as a foundation for building and evaluating financial forecasting models while ensuring transparency and reproducibility of the research.

We then use the TA-lib and the Technical Analysis Library to calculate 102 widely used technical indicators (Lo, 2000). These indicators are divided into groups such as volatility, momentum, volume, overlap studies, price transformation, and statistics. The indicators are summarized in Table 3. More details of each technical indicator can be found in the documentation of both libraries. In addition to the indicators, we also calculate 3 Fourier Transform with 3 different components (3, 6, and 9) to decompose price data into simpler components, such as sine and cosine waves, to analyze underlying patterns or cycles.

Table 3. Technical Indicator

Python Library	Indicator Groups	Technical Indicator
TA-Lib	Overlap Studies	BBANDS, DEMA, EMA, HT_TRENDLINE, KAMA, MA, MAVP, MIDPOINT, MIDPRICE, SAR, SAREXT, T3, TEMA, TRIMA, WMA
	Momentum Indicators	ADX, ADXR, APO, AROON, AROONOSC, BOP, CCI, CMO, DX, MACD, MACDEXT, MACDFIX, MFI,

		MINUS_DI, MINUS_DM, MOM, PLUS_DI, PLUS_DM, PPO, ROC, ROCP, ROCR, RSI, STOCH, STOCHF, STOCHRSI, TRIX, ULTOSC, WLLR
	Volume Indicators	AD, ADOSC, OBV
	Cycle Indicators	HT_DCPERIOD, HT_DCPHASE, HT_PHASOR, HT_SINE, HT_TRENDMODE
	Price Transform	AVGPRICE, MEDPRICE, TYPPRICE, WCLPRICE
	Volatility Indicators	ATR, NATR, TRANGE
	Statistic Functions	LINEARREG, LINEARREG_ANGLE, LINEARREG_INTERCEPT, LINEARREG_SLOPE, STDDEV, TSF,VAR
Technical Analysis Library	Trend	dpo, Ichimoku, kst, mass_index, VortexIndicator
	Momentum	awesome_oscillator, pvo, pvo_hist, tsi
	Volatility	donchian_channel, keltner_channel, ulcer_index
	Volume	ease_of_movement, force_index, negative_volume_index, volume_price_trend, volume_weighted_average_price

3.2. Target Variable

The target variable is a binary trading signal derived from daily returns. If the return for the next 20 days exceeds a predefined positive threshold, a 'buy' signal is assigned, while a 'sell' signal is assigned if the return falls below a negative threshold. This approach helps translate raw price movements into actionable trading decisions, allowing the models to learn patterns associated with profitable trades. Also, with the predicted signals, we can backtest the strategy generated by each model, which clarifies the trading performance of ML and DL models.

3.3. Descriptive Statistic

We first look at the closing price of each market index to see then overall trend over time. We can see that from Figure 2, Figure 3 and Figure 4, except for Qatar and UK, all market exhibit a clear up trend. Also, each market include three phases: rising, falling and sideways. In addition, historical prices also include the 2008 financial crisis, which helps the model learn and remember more patterns, thereby adapting to many market conditions.

From Table 4, we can see that DAX, TSX, S&P 500, JKSE, BET and VNINDEX have the numbers of up day a lot more than the number of down day, this explain for their rally. Also, QE General and TASI Index does not have volume data before 2008, that is why the data for those market start from 2008.

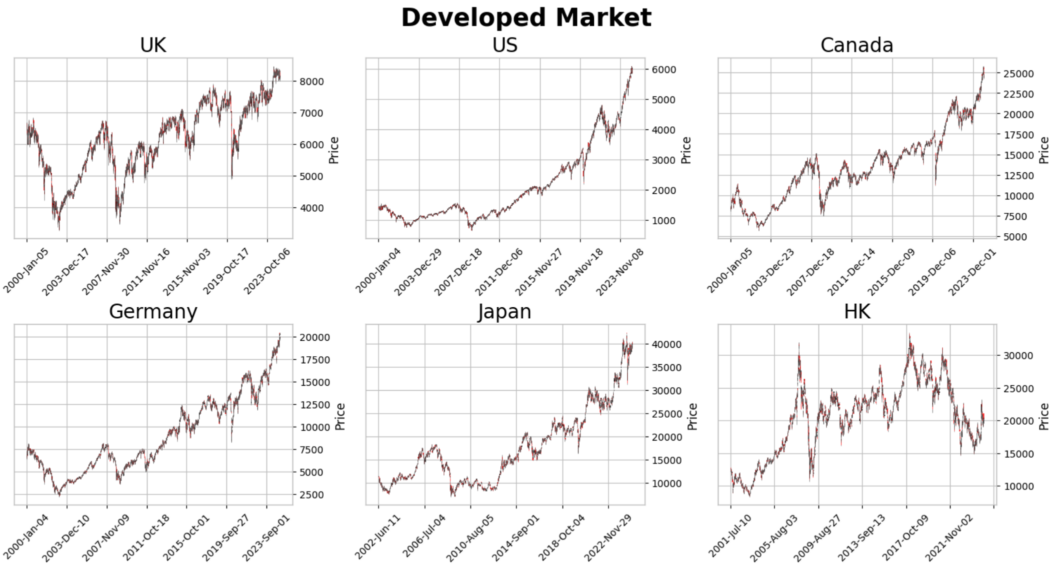


Figure 2. Developed Market Close Price

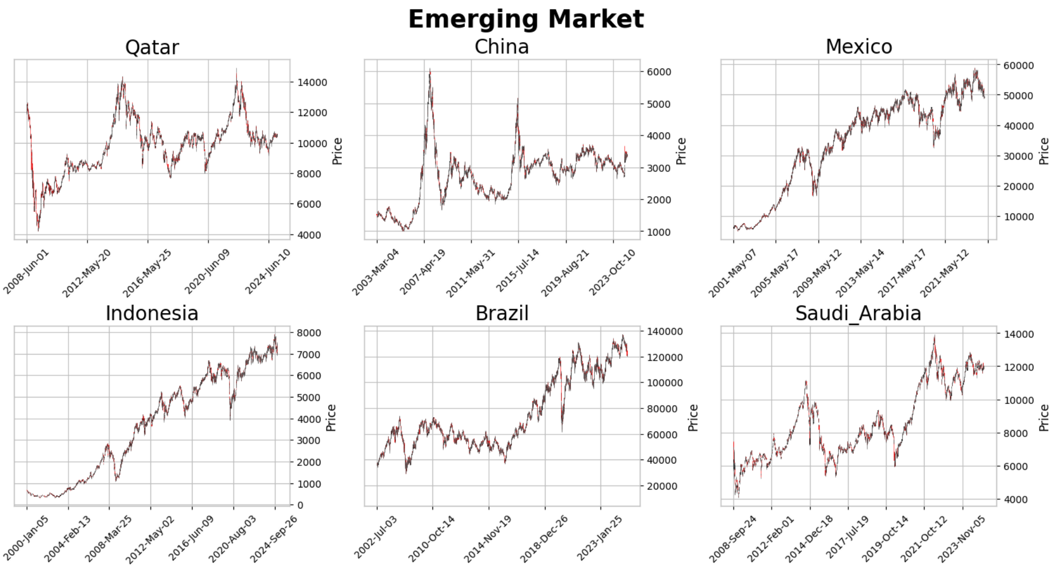


Figure 3. Emerging Market Close Price

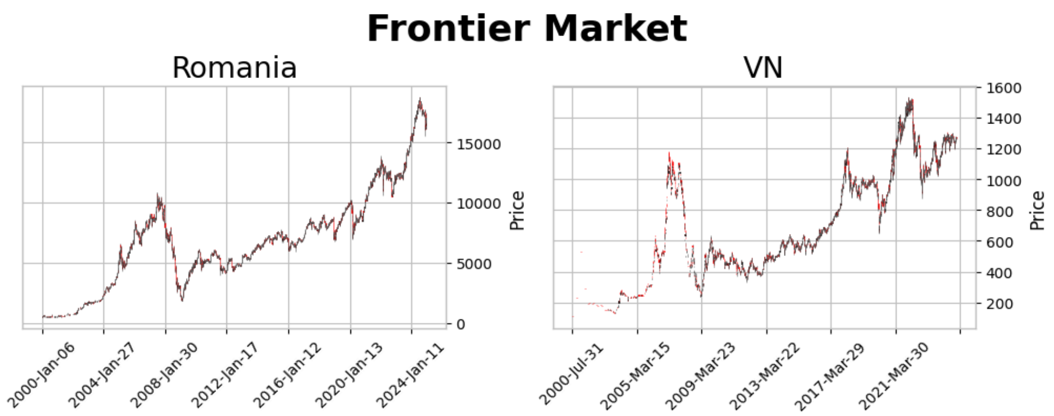


Figure 4. Frontier Market Close Price

Table 4. Market Data

Ticker	Number of days with (+) returns	Number of days with (-) returns	Start Date
SSEC	2781	2520	2003-03-04
BVSP	2307	2166	2002-07-03
FTSE 100	3279	3025	2000-01-05
DAX	3359	2975	2000-01-04
S&P 500	3369	2919	2000-01-04
TSX	3402	2864	2000-01-05
N225	2897	2618	2002-06-11
HSI	2965	2812	2001-07-10
JKSE	3304	2761	2000-01-05
MXX	3106	2812	2001-05-07
QE General	2133	2007	2008-06-01
BET	3355	2890	2000-01-06
TASI	1799	1486	2008-09-24
VNINDEX	3149	2794	2000-07-31

Next, we will look into the descriptive statistic of return. From Table 5, all market indices return mean are close to zero, indicating that, on average, daily returns do not show a significant upward or downward bias over time.

The standard deviation which measures the volatility of returns is around 0.0108 to 0.0148 for most indices. This suggests that daily fluctuations in returns are relatively moderate across markets. However, BVSP (Brazil stock index) exhibits a notable high standard deviation of 0.0397, indicating higher price swings and risk.

Also, all markets have a Kurtosis above 3, indicating returns distribution have a fat tails - more risk and large price move. This implies that extreme price movements - both gains and losses - occur more frequently than would be expected under a normal distribution. BVSP again, has extremely high Kurtosis value of 3005.3654, suggesting the presence of exceptionally large and rare return spikes.

Most of the markets have a negative Skewness level, meaning the distribution of returns has a longer left tail. This suggests that large negative returns (market crashes or corrections) occur more frequently than large positive returns.

The return distribution plots across market are shown in Figure 5 to Figure 7.

Table 5. Descriptive Statistic

Index	Mean	Stdev	Skewness	Kurtosis
S&P 500	0.0003	0.0122	-0.1624	10.1841
TSX	0.0002	0.0108	-0.6305	16.1276
FTSE 100	0.0001	0.0114	-0.1714	8.0258
DAX	0.0003	0.0143	0.0036	6.1567
NIKKEI 225	0.0003	0.0144	-0.2924	7.8242
HSI	0.0002	0.0145	0.2172	7.8460
BVSP	0.0009	0.0397	49.6482	3005.3654
MXX	0.0004	0.0118	0.0058	6.4654
TASI	0.0002	0.0130	-0.7163	14.9651
QE GENERAL	0.0000	0.0119	-0.3339	12.3725
SSEC	0.0003	0.0148	-0.3360	5.0909
JKSE	0.0005	0.0126	-0.4626	7.3338
BET	0.0007	0.0142	-0.3805	10.1513
VNINDEX	0.0005	0.0144	-0.2823	3.3081

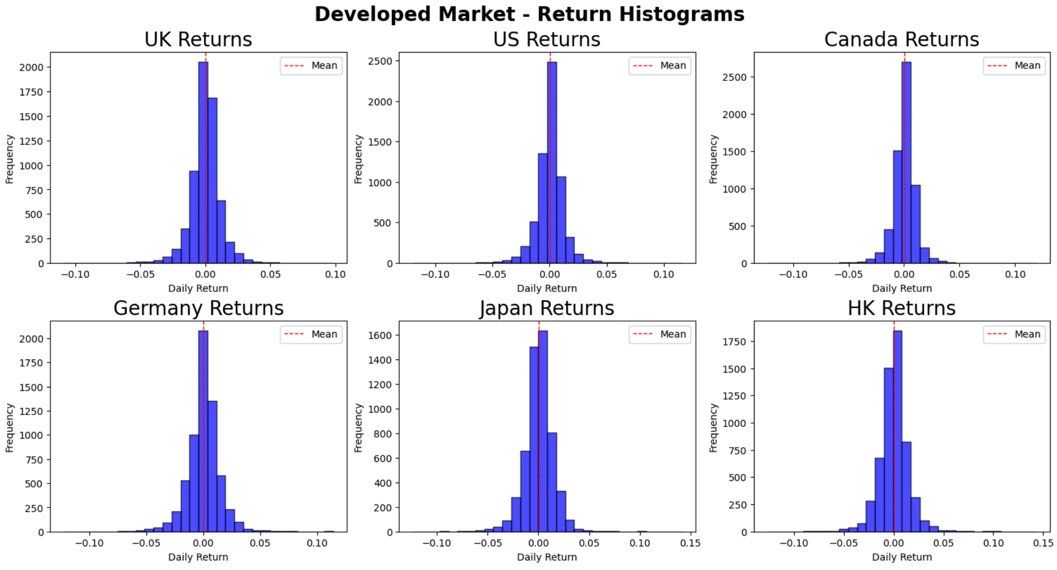


Figure 5. Developed Market Distribution

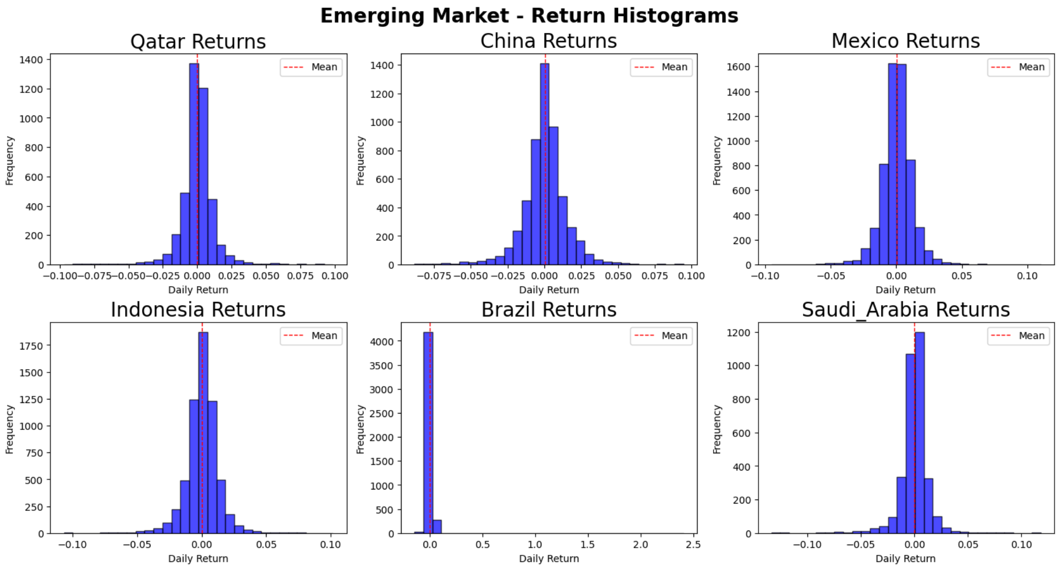


Figure 6. Emerging Market Distribution

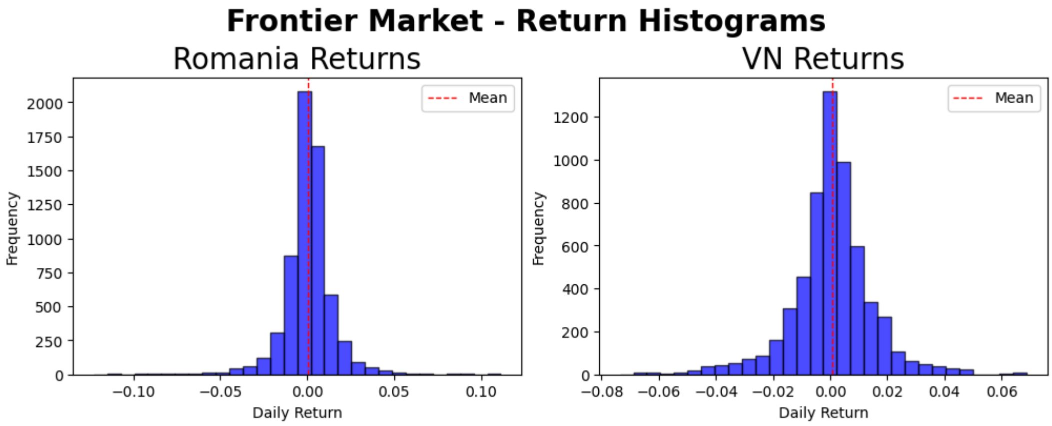


Figure 7. Frontier Market Distribution

4. Methodology

4.1. Prediction Model

The forecasting model selection process builds on previous research and the trend of applying state-of-the-art models in the field of financial forecasting. This process not only ensures superior forecasting accuracy but also focuses on interpretability and practical application value, meeting the strict requirements of complex financial problems. In addition, the Transformer architecture, a SoTA architecture in the field of natural language processing, will also be studied. The models and their parameters are described in Table 4, providing insight into their configurations and performance considerations. The details of the models are presented as follows.

4.1.1. Logistic Regression

LR is a fundamental statistical model used for binary classification problems, making it suitable for predicting buy and sell signals in financial markets. It is computationally efficient, easy to interpret, and serves as a strong baseline for more complex models. The mathematical formulation and derivation of the logistic regression model can be found in (Hosmer et al., 2013).

4.1.2. Support Vector Machine

SVM is a supervised learning algorithm that aims to separate different classes by finding an optimal line in two dimensions or a hyperplane in higher dimensions, optimizing the boundary to achieve the largest possible separation. It is particularly useful for high-dimensional financial datasets and can employ kernel methods to handle non-linearly separable data. Due to its robustness, SVM is often used in financial classification problems. Cortes and Vapnik (1995) provide the theoretical background and mathematical formulation of SVM.

4.1.3. Decision Tree

A DT is a non-parametric, tree-based learning algorithm that recursively splits the data into subsets based on the most informative features. It is simple, interpretable, and useful for quick decision-making but prone to overfitting in complex datasets. The entropy and Gini impurity measures used for node splitting are explained in (Breiman et al., 1984).

4.1.4. Random Forest

Random Forest is an ensemble learning algorithm based on the majority voting mechanism or the wisdom of the crowd. By constructing multiple decision trees and combining their predictions, it enhances robustness and reduces variance in the final output. It introduces randomness through bootstrap sampling and feature selection, making it highly effective in financial applications. Breiman (2001) provides a detailed discussion of Random Forest.

4.1.5. XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced boosting algorithm that builds sequential trees while minimizing loss and regularizing complexity. It has won numerous Kaggle competitions due to its high predictive accuracy and efficiency. A comprehensive explanation of XGBoost can be found in (Chen & Guestrin, 2016).

4.1.6. LightGBM

LightGBM is an optimized gradient-boosting framework designed for high efficiency and speed. It employs a leaf-wise tree growth strategy instead of level-wise growth, improving performance on large datasets while maintaining high predictive power. The technical details of LightGBM are described in (Ke et al., 2017).

4.1.7. Multilayer Perceptron

MLP is a feedforward artificial neural network with multiple layers of neurons using non-linear activation functions. It can learn complex relationships in financial data but requires careful tuning to avoid overfitting. The theoretical foundation of MLP can be found in (Taud & Mas, 2017).

4.1.8. Long Short Term Memory

LSTMs address the vanishing gradient problem in RNNs by introducing gating mechanisms that regulate information flow. They are widely used in financial forecasting to capture long-term dependencies. A comprehensive study of LSTMs is presented in (Hochreiter & Schmidhuber, 1997).

4.1.9. Gated recurrent units

GRU is a simplified LSTM variant that retains long-term dependencies while reducing computational complexity. It is an efficient alternative to LSTM for financial time-series modeling. The theoretical background of GRU can be found in (Cho et al., 2014). Differences in architecture between GRU and LSTM can be found in Figure 8.

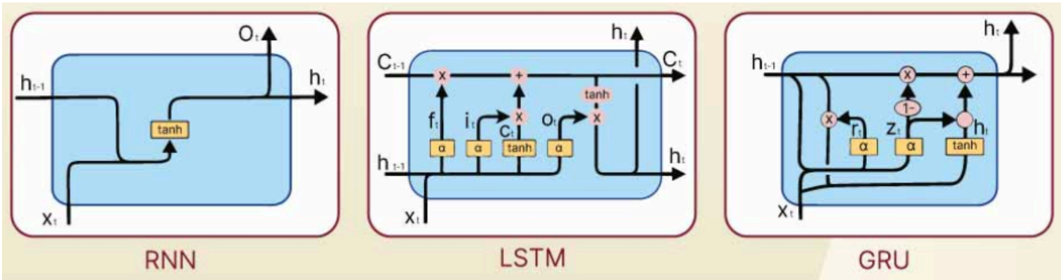


Figure 8. Differences in Architecture of RNN, LSTM and GRU

4.1.10. Convolutional Neural Network

CNNs are deep learning models that apply convolutional layers to extract spatial and temporal features from data. While commonly used in image processing, CNNs have been successfully applied to financial time-series forecasting by identifying patterns in historical price data. They are particularly effective in feature extraction when combined with other architectures such as LSTMs. A comprehensive discussion of CNNs can be found in (O'Shea, 2015).

We employ a CNN with Conv2D layers for stock price prediction. The input is structured as a grayscale image with dimensions 20-day sliding window × technical indicators.

4.1.11. Generative Adversarial Network

GANs consist of a generator and a discriminator trained in an adversarial framework to generate synthetic data. They are particularly useful for market data augmentation, improving model generalization, and simulating realistic financial conditions. The original formulation of GANs is discussed in (Goodfellow et al., 2014).

We adopt the architecture of Lin et al. (2021) for the Wasserstein GAN (WGAN) model, employing a CNN as the discriminator and an LSTM as the generator, replacing the GRU used in Lin's original design. The WGAN (Arjovsky, 2017) improves training stability by using a Wasserstein loss with weight clipping, addressing the vanishing gradient problem commonly encountered in basic GANs.

In our GAN setting, we train the model to predict actual returns as a regression task rather than a classification task. This choice leverages the GAN's ability to generate continuous data, making it more suitable for return prediction.

4.1.12. Transformer

The Transformer model replaces recurrence with self-attention mechanisms, enabling parallel processing of sequential data. It has demonstrated strong performance in capturing long-range dependencies in financial markets and is a promising alternative to

RNN-based architectures. The original Transformer model is described in (Vaswani et al., 2017).

In this work, we follow the lightweight multi-head attention proposed by Nguyen et al, 2023. The transformer model in this study uses positional encoding to capture temporal order, followed by multi-head self-attention layers that learn dependencies across time steps. Each attention output passes through feed-forward networks with ReLU activation, enhanced by residual connections and layer normalization. The final output layer generates binary stock price predictions. The architecture is shown in Figure 9.

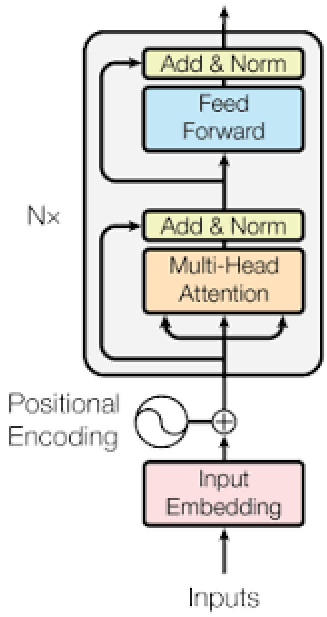
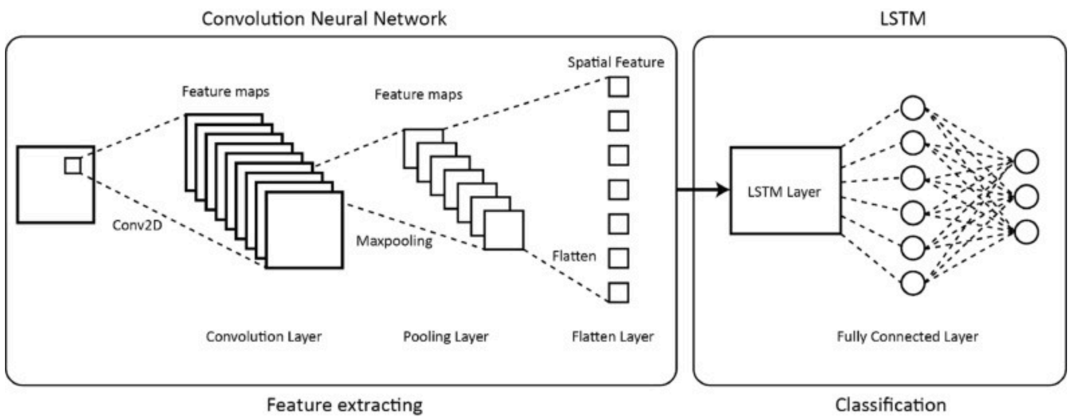


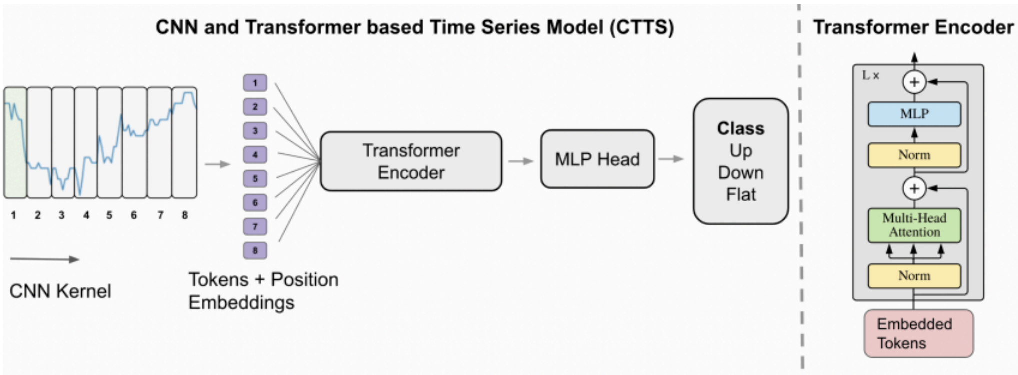
Figure 9. Architecture of Transformer Model

4.1.13. Hybrid Model

Hybrid models are believed to possess superior predictive power by combining the strengths of multiple deep learning architectures. To test this hypothesis, we deploy two hybrid models, which are CNN-LSTM and CNN-Transformer (Zeng et al, 2023). In both of these models, the CNN acts as a feature extraction component. Its output is fed into the LSTM and Transformers architectures, which are best known for their power to capture temporal dependencies. Both models are shown in Figure 10.



(a) CNN-LSTM (Shah et al, 2022)



(b) CNN-Transformer (Zeng et al, 2023)

Figure 10. Architecture of Hybrid Model

Table 6. Model's Parameters

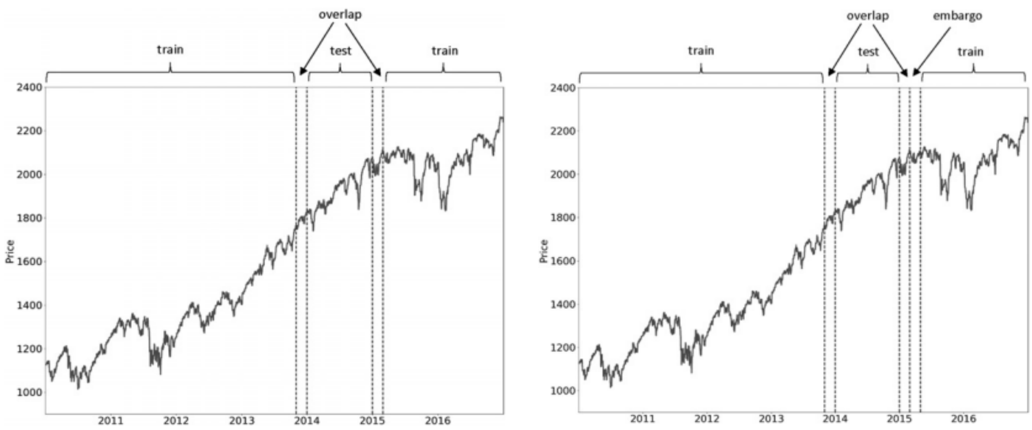
Model	Parameter
LR	'C': [0.1, 0.5, 1.0], 'penalty': ['l1', 'l2'], 'solver': ['liblinear'], 'class_weight': ['balanced', None]
SVM	'C': [0.1, 0.5, 1.0], 'kernel': ['linear', 'rbf', 'poly', 'sigmoid'], 'gamma': ['scale', 'auto']
DT	'max_depth': [3, 5, 7], 'min_samples_split': [2, 3], 'min_samples_leaf': [2, 3], 'criterion': ['gini', 'entropy']
RF	'n_estimators': [50, 100], 'max_depth': [3, 5, 7], 'min_samples_split': [2, 3], 'min_samples_leaf': [1, 2], 'bootstrap': [True, False]
XGB	'n_estimators': [50, 100], 'max_depth': [3, 5], 'learning_rate': [0.01, 0.05], 'subsample': [0.6, 0.8], 'gamma': [0, 1], 'colsample_bytree': [0.6, 0.8], 'reg_alpha': [0, 0.1], 'reg_lambda': [1, 2]
LGB	'num_leaves': [7, 10], 'learning_rate': [0.01, 0.05], 'n_estimators': [100, 150], 'max_depth': [3, 5], 'subsample': [0.6, 0.8], 'colsample_bytree': [0.6, 0.8], 'min_data_in_leaf': [1, 5, 10]
MLP	Adam optimizer, learning_rate = 0.0005, batch_size = 32, epochs = 30
RNN	
LSTM	
GRU	
CNN	
Transformer	
CNN - Transformer	
CNN - LSTM	

GAN	Adam optimizer, learning_rate = 0.0001, epochs = 50

4.2. Cross-Validation Approach

4.2.1. Purging and Embarago

Data leakage is a critical issue in financial modeling, where information from the test set unintentionally influences the training process, leading to overly optimistic performance estimates. To address this, Lopez de Prado (2018) introduced the purging and embargo techniques. Purging removes training samples near the test set to eliminate lookahead bias, while embargo adds a buffer period between the two sets to prevent delayed signal leakage in cases where purging is not able to prevent all leakage. These methods ensure a more realistic model evaluation, as illustrated in Figure11.



(a) **(b)**

(Lopez de Prado, 2018)

Figure 11. Purge and Embargo

4.2.2. Combinatorial Purged Cross Validation (CPCV)

Evaluating the performance of machine learning models for financial trading is challenging due to the non-stationary nature of financial markets, autocorrelation in returns, and potential data leakage. Traditional cross-validation methods, such as k-fold or leave-one-out, fail to account for the temporal structure of time series data, leading to overly optimistic performance estimates.

To overcome this problem, Walk-forward validation (WFV) is widely used to evaluate machine learning trading models (Kamil, 2015; Cao et al., 2003). It mimics real-world trading conditions by training on past data and testing on unseen future data. However, WFV has notable drawbacks. First, it does not fully utilize all available data since each test set is used only once. Secondly, WF tends to overfit due to testing only a single historical scenario (Bailey et al., qtd. in Lopez de Prado 162).

To address these limitations, Lopez de Prado (2018) introduced Combinatorial Purged Cross-Validation (CPCV), a more sophisticated validation method specifically designed for financial datasets. CPCV improves upon WFV by generating multiple overlapping train-test splits while ensuring proper separation to eliminate data leakage. Unlike traditional cross-validation methods, CPCV purges training data points that are too close to the test set, reducing overfitting risks and enhancing the robustness of performance estimation. The algorithm for CPCV can be found in (Lopez de Prado, 2018).

Key Benefits of CPCV:

- Prevents Data Leakage: By purging data points that are temporally close to the test set, CPCV minimizes lookahead bias.
- Better Utilization of Data: Unlike WFV, CPCV creates multiple overlapping train-test splits, allowing for more efficient use of the dataset.
- Robust Performance Estimation: CPCV evaluates models across a range of market conditions, making performance assessments more reliable.

This paper uses ML algorithms to predict stock prices as trading signals. The CPCV algorithm is applied for hyper-parameter tuning for ML models, ensuring robust evaluation.

4.3. Backtesting Framework

To evaluate the predictive models' real-world applicability, backtesting is conducted on historical market data using the vectorbt library in Python. This library offers powerful vectorized backtesting tools, allowing for quick and seamless evaluation of trading strategies. The models generate buy and sell signals, and trading strategies are tested with transaction costs incorporated to mimic realistic trading conditions.

4.4. Evaluation Metric

For modeling, we implement ML algorithms (SVM, CART, XGBoost, LightGBM, RF, LR) and DL models (RNN, CNN, LSTM, MLP, GRU, SAE, GAN, Transformer). These models are trained and optimized using appropriate techniques, and their performance is evaluated using prediction accuracy, and trading profitability to ensure both accuracy and robustness in financial forecasting These metrics provide insights into different aspects of model performance, including classification quality and practical financial implications. Finally, we compare model effectiveness across different market conditions to extract meaningful insights for investment strategies.

4.4.1. Machine learning metrics

To compare the accuracy performance of prediction models, we use traditional metrics, such as Accuracy, Recall, Precision, and F1 score for classification performance evaluation. These metrics are computed as follows:

- Accuracy: Measures the proportion of correctly classified instances among all predictions.
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$
- Precision: Represents the proportion of correctly predicted positive instances among all predicted positives.
$$Precision = \frac{TP}{TP+FP} \tag{2}$$
- Recall (Sensitivity): Indicates the proportion of actual positives that are correctly identified.
$$Recall = \frac{TP}{TP+FN} \tag{3}$$
- F1-Score: A harmonic mean of precision and recall, providing a balanced measure.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

- Matthews Correlation Coefficient (MCC): A more balanced metric that accounts for all four confusion matrix components.

$$MCC = 2 \times \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (5)$$

Where TP stands for True Positive, TN for True Negative, FP for False Positive, and FN for False Negative."

We also consider the Area Under the Curve (AUC-ROC), which measures the ability of the model to distinguish between classes. A higher AUC value indicates better model performance.

$$AUC = \int_{0}^{1} TPR(FPR) dFPR \quad (6)$$

With $TPR = \frac{TP}{TP+FN}$ is the True Positive Rate and $TPR = \frac{FP}{FP+TN}$ is the False Positive Rate.

4.4.2. Trading performance metrics

The trading performance metrics are defined as follows:

- Sharpe Ratio: Evaluates an investment's risk-adjusted return by measuring the amount of return earned per unit of risk taken.

$$SR = \frac{E[R_p - R_f]}{\sigma_p} \quad (7)$$

- Sortino Ratio: While the Sharpe ratio considers both upside and downside risk, the Sortino ratio only considers downside risk.

$$S = \frac{E[R_p - R_f]}{\sigma_d} \quad (8)$$

- Total Return: Represents the overall return generated by an investment over a specified period, including capital appreciation and income.

$$R_{total} = \prod_{t=1}^{T} (1 + r_t) - 1 \quad (9)$$

- Maximum Drawdown: Defines as the largest loss in a trading strategy's cumulative returns.

$$MDD = max_{t \in [0;T]} \left(\frac{Peak_t - Trough_t}{Peak_t} \right) \quad (10)$$

5. Empirical Result

In this paper, we use an iterative training process to train ML and DL models through the CPCV algorithm to ensure the robustness of the results. This section compares results based on some machine learning metrics: Accuracy, Precision, F1-score, Recall, MCC, and AUC. The data is split into six groups, two of which are for testing purposes. This results in 21 simulations of training and testing processes.

For trading performance analysis, we compare the mean daily return generated by the mean of seven backtest paths created by the CPCV algorithm to see how different models perform in different markets. To mimic the real-world trading environment, we set up a

1% slippage and 1% fixed cost per trade as the trading cost. A buy-and-hold (BnH) strategy acts as the benchmark for deciding whether the ML and DL strategies are good.

To determine whether there are statistically significant differences among models in terms of machine learning metrics and trading performance, we employ the Kruskal-Wallis test, a non-parametric statistical test suitable for comparing multiple related samples. If the Friedman test detects a significant difference, we conduct a Nemenyi post hoc test to identify which models differ significantly.

5.1. Developed Market

5.1.1. S&P 500 Index

Table 7. Machine Learning Metrics for S&P 500

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9252	0.9387	0.9423	0.9402	0.8375	0.9177
SVM	0.8270	0.9085	0.8051	0.8365	0.6702	0.8292
DT	0.9094	0.9352	0.9202	0.9271	0.8048	0.9029
RF	0.9132	0.9252	0.9372	0.9311	0.8101	0.9021
XGB	0.9217	0.9266	0.9506	0.9383	0.8286	0.9084
LGB	0.9219	0.9280	0.9486	0.9381	0.8288	0.9101
MLP	0.9493	0.9600	0.9572	0.9584	0.8924	0.9468
CNN	0.9446	0.9466	0.9663	0.9558	0.8814	0.9358
LSTM	0.9089	0.9085	0.9539	0.9298	0.7997	0.8914
GRU	0.9266	0.9338	0.9494	0.9414	0.8409	0.9174
Transformer	0.9008	0.9407	0.9020	0.9160	0.8005	0.9011
CNN - LSTM	0.9258	0.9316	0.9539	0.9416	0.8399	0.9135
CNN - Transformer	0.9072	0.9003	0.9663	0.9303	0.7979	0.8834
GAN	0.5113	0.6417	0.4962	0.5185	0.0546	0.5289

As shown in Table 7, all the models provide good accuracy performance except for the GAN model. This can be due to the different approaches used in the GAN model, which uses a regression task instead of a classification task. We apply the Kruskal-Wallis test and get a p-value of 0.019, which means there is a significant difference in the return between strategies. However, the Nemenyi post hoc test does not recognize any difference.

Table 8 shows the average trading performance of ML and DL models compared to BnH. No model achieves a higher total return than BnH. However, four models, LSTM, GRU, Transformer, and CNN—Transformer, get better SR and S than BnH, suggesting they are better at capturing returns with a smaller risk. In terms of MMD and standard deviation of returns, all models show that they are less prone to risk compared to BnH.

Table 8. Trading Performance for S&P 500

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	86.93	0.26	0.37	37.29	0.0006	0.0064
SVM	66.09	0.19	0.26	38.94	0.0005	0.0054
DT	77.36	0.24	0.33	42.66	0.0005	0.0058
RF	94.09	0.29	0.40	37.65	0.0005	0.0061
XGB	101.23	0.31	0.43	33.03	0.0005	0.0062
LGB	90.87	0.28	0.39	34.84	0.0006	0.0062
MLP	108.14	0.33	0.45	36.95	0.0006	0.0059
CNN	112.87	0.34	0.48	38.48	0.0003	0.0064
LSTM	253.89	0.57	0.80	34.24	0.0003	0.0067
GRU	213.63	0.53	0.74	34.80	0.0003	0.0065
Transformer	193.89	0.51	0.72	34.43	0.0003	0.0062
CNN - LSTM	130.96	0.38	0.53	36.92	0.0003	0.0065
CNN - Transformer	205.54	0.49	0.69	34.33	0.0003	0.0068
GAN	87.35	0.25	0.35	52.79	0.0001	0.0058
BnH	304.56	0.47	0.66	56.77	0.0003	0.0121

We perform a Friedman test on the daily returns and the alternative hypothesis is established, which means the returns generated are different for models. We then apply the Nemenyi test to see which models are different from others. From the Nemenyi test results, the GAN model is the only model different from BnH.

5.1.2. TSX Index

Table 9. Machine Learning Metrics for TSX

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9402	0.9503	0.9512	0.9507	0.8739	0.9368
SVM	0.9360	0.9459	0.9488	0.9473	0.8648	0.9320
DT	0.9111	0.9363	0.9162	0.9259	0.8143	0.9093
RF	0.9227	0.9371	0.9355	0.9362	0.8366	0.9176
XGB	0.9318	0.9420	0.9461	0.9440	0.8557	0.9270
LGB	0.9301	0.9419	0.9430	0.9424	0.8521	0.9257
MLP	0.9640	0.9716	0.9690	0.9700	0.9248	0.9615

CNN	0.9469	0.9496	0.9656	0.9569	0.8893	0.9419
LSTM	0.9213	0.9291	0.9444	0.9359	0.8353	0.9149
GRU	0.9377	0.9419	0.9564	0.9489	0.8686	0.9324
Transformer	0.9261	0.9347	0.9443	0.9390	0.8455	0.9202
CNN - LSTM	0.9402	0.9437	0.9597	0.9510	0.8753	0.9349
CNN - Transformer	0.9272	0.9264	0.9581	0.9413	0.8469	0.9183
GAN	0.5364	0.6252	0.5893	0.5890	0.0564	0.5226

The TSX index's ML metrics result is similar to that of the S&P 500 index. As shown in Table 9, all ML and DL perform well except for GAN. The statistical test results show that CNN is significantly better than DT in terms of accuracy, and MLP significantly outperforms LSTM.

Table 10. Trading Performance for TSX

	Return (%)	**SR**	**S**	**MMD (%)**	**Mean Return**	**Stdev Return**
LR	102.83	0.38	0.52	31.72	0.0005	0.0056
SVM	111.44	040	0.54	31.23	0.0005	0.0057
DT	102.28	0.38	0.53	28.69	0.0005	0.0054
RF	112.47	0.41	0.56	28.81	0.0005	0.0055
XGB	96.38	0.36	0.50	30.12	0.0005	0.0056
LGB	102.83	0.38	0.52	29.23	0.0005	0.0055
MLP	115.90	0.41	0.56	28.48	0.0005	0.0055
CNN	99.84	0.37	0.51	30.21	0.0003	0.0057
LSTM	212.56	0.60	0.83	28.63	0.0003	0.0058
GRU	166.89	0.52	0.72	29.82	0.0003	0.0058
Transformer	112.02	0.40	0.55	30.64	0.0002	0.0059
CNN - LSTM	133.25	0.45	0.62	29.48	0.0003	0.0058
CNN - Transformer	121.33	0.42	0.57	33.13	0.0003	0.0060
GAN	68.57	0.25	0.34	45.36	0.0001	0.0056
BnH	137.03	0.39	0.53	49.99	0.0002	0.0107

Table 10 shows that DL models outperform traditional ML and BnH when applied to the TSX index. We can see that LSTM and GRU have better returns with a smaller amount of risk, while other DL models could yield the same results but at a lower cost of risk. Similar to the S&P 500 index, LSTM and GRU are the best performance models when applied in trading.

5.1.3. FTSE 100 Index

Table 11. Machine Learning Metrics for FTSE 100

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9275	0.9431	0.9249	0.9335	0.8529	0.9261
SVM	0.9254	0.9375	0.9273	0.9321	0.8480	0.9231
DT	0.8937	0.9089	0.8991	0.9037	0.7834	0.8918
RF	0.9044	0.9135	0.9143	0.9138	0.8042	0.9012
XGB	0.9164	0.9241	0.9254	0.9247	0.8291	0.9140
LGB	0.9147	0.9219	0.9246	0.9232	0.8254	0.9119
MLP	0.9487	0.9536	0.9540	0.9534	0.8964	0.9475
CNN	0.9374	0.9459	0.9408	0.9431	0.8727	0.9359
LSTM	0.9095	0.9182	0.9198	0.9183	0.8164	0.9069
GRU	0.9161	0.9189	0.9317	0.9249	0.8288	0.9121
Transformer	0.9072	0.9138	0.9225	0.9170	0.8131	0.9054
CNN - LSTM	0.9247	0.9363	0.9293	0.9320	0.8483	0.9239
CNN - Transformer	0.9079	0.9162	0.9196	0.9173	0.8126	0.9044
GAN	0.5108	0.5851	0.5234	0.5254	0.0208	0.5110

FTSE 100 yields similar results to both S&P 500 and TSX. The statistical test signified that CNN, LSTM and MLP outperformed DT model. More detailed ML metrics can be found in Table 11.

Table 12. Trading Performance for FTSE 100

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	-50.48	-0.33	-0.43	53.16	0.0004	0.0057
SVM	-52.30	-0.35	-0.46	64.91	0.0003	0.0058
DT	-56.70	-0.43	-0.57	69.01	0.0003	0.0056
RF	-55.00	-0.40	-0.53	67.51	0.0003	0.0057
XGB	-50.46	-0.32	-0.44	63.18	0.0003	0.0058
LGB	-48.64	-0.30	-0.41	61.60	0.0003	0.0058

MLP	-47.52	-0.28	-0.38	59.20	0.0003	0.0058
CNN	-32.45	-0.14	-0.20	46.89	0.0003	0.0060
LSTM	4.77	0.08	0.11	37.58	0.0001	0.0061
GRU	-12.63	-0.01	-0.02	42.91	0.0001	0.0061
Transformer	-19.25	-0.06	-0.06	40.28	0.0001	0.0063
CNN - LSTM	-35.07	-0.22	-0.22	49.13	0.0000	0.0061
CNN - Transformer	-23.78	-0.10	-0.10	40.21	0.0001	0.0064
GAN	5.03	0.06	0.08	45.76	0.0000	0.0056
BnH	28.56	0.18	0.24	51.65	0.0001	0.0114

In the case of the FTSE 100, all models except LSTM and GAN can gain positive returns. This shows that when the market does not perform well (28.56% return over 25 years) the ML and DL models can not perform well and the risk is as high as BnH. This is worth noting when doing market selection.

5.1.4. DAX Index

Table 13. Machine Learning Metrics for DAX

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9291	0.9448	0.9328	0.9386	0.8530	0.9272
SVM	0.9243	0.9302	0.9414	0.9353	0.8423	0.9186
DT	0.9035	0.9121	0.9229	0.9173	0.7984	0.8966
RF	0.9062	0.9104	0.9305	0.9202	0.8031	0.898
XGB	0.9156	0.9160	0.9415	0.9284	0.8232	0.9076
LGB	0.9162	0.9177	0.9403	0.9288	0.8243	0.9085
MLP	0.9446	0.9578	0.9456	0.9513	0.8869	0.9457
CNN	0.9451	0.9471	0.9597	0.9527	0.8877	0.9418
LSTM	0.9197	0.9271	0.9347	0.9306	0.8338	0.9155
GRU	0.9262	0.9287	0.9454	0.9367	0.8472	0.9221
Transformer	0.9198	0.9234	0.9408	0.9314	0.8348	0.9153
CNN - LSTM	0.9461	0.9504	0.9563	0.9530	0.8889	0.9437
CNN - Transformer	0.9242	0.9218	0.9513	0.9358	0.8426	0.9165
GAN	0.5199	0.6158	0.5270	0.5560	0.0330	0.5163

The DAX index's ML metrics result looks similar to that of the S&P 500 index. As shown in Table 13, all ML and DL perform well except for GAN. The statistical test reveals that for accuracy, MLP, CNN, and CNN - Transformer significantly outperform traditional DT and RF. Also, the GAN models again show significant differences from others.

Table 14. Trading Performance for DAX

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	49.33	0.20	0.28	42.11	0.0007	0.0076
SVM	43.62	0.17	0.24	43.78	0.0007	0.0075
DT	54.92	0.21	0.30	43.33	0.0005	0.0077
RF	70.72	0.25	0.35	38.69	0.0006	0.0077
XGB	73.89	0.26	0.36	40.73	0.0006	0.0078
LGB	85.16	0.28	0.39	40.12	0.0006	0.0078
MLP	35.84	0.16	0.22	45.57	0.0007	0.0075
CNN	137.06	0.37	0.52	42.15	0.0004	0.0079
LSTM	256.29	0.51	0.72	39.05	0.0003	0.0080
GRU	189.93	0.44	0.63	39.43	0.0004	0.0079
Transformer	149.47	0.39	0.55	45.01	0.0003	0.0082
CNN - LSTM	136.23	0.37	0.53	42.20	0.0004	0.0078
CNN - Transformer	146.47	0.38	0.54	38.33	0.0003	0.0081
GAN	64.05	0.22	0.3	56.17	0.0001	0.0072
BnH	167.23	0.35	0.49	70.84	0.0003	0.0142

Table 14 shows that DL models outperform traditional ML and BnH when applied to the DAX index. LSTM and GRU have better returns with less risk, while other DL models could yield the same results but at a lower cost of risk.

5.1.5. N225 Index

Table 15. Machine Learning Metrics for Nikkei 225

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9173	0.9287	0.9283	0.9284	0.8301	0.9148
SVM	0.9163	0.9271	0.9283	0.9275	0.8283	0.9141
DT	0.8847	0.9124	0.8866	0.8983	0.7664	0.8834
RF	0.9022	0.9127	0.9185	0.9155	0.7989	0.8985

XGB	0.9101	0.9210	0.9235	0.9222	0.8153	0.907
LGB	0.9105	0.9218	0.9233	0.9225	0.8161	0.9076
MLP	0.9476	0.9555	0.9539	0.9543	0.8936	0.9466
CNN	0.9477	0.9595	0.9496	0.9542	0.8937	0.9476
LSTM	0.9151	0.9283	0.9260	0.9268	0.8260	0.9127
GRU	0.9143	0.9346	0.9167	0.9252	0.8255	0.9142
Transformer	0.9117	0.9347	0.9108	0.9218	0.8215	0.9113
CNN - LSTM	0.9277	0.9333	0.9434	0.9377	0.8523	0.9245
CNN - Transformer	0.9186	0.9458	0.9118	0.9280	0.8356	0.9199
GAN	0.5371	0.5821	0.5530	0.5417	0.0628	0.5353

There is no statistically significant difference between models (except for GAN) in terms of accuracy when applied in the N225 index market.

Table 16. Trading Performance for N225

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	32.22	0.08	0.11	52.05	0.0007	0.0076
SVM	40.47	0.12	0.16	52.68	0.0006	0.0077
DT	39.86	0.11	0.15	46.25	0.0006	0.0073
RF	54.66	0.17	0.24	41.29	0.0007	0.0077
XGB	48.94	0.15	0.21	45.90	0.0007	0.0077
LGB	43.91	0.13	0.18	44.90	0.0006	0.0076
MLP	48.92	0.15	0.21	42.11	0.0007	0.0075
CNN	131.69	0.36	0.51	34.90	0.0004	0.0078
LSTM	319.14	0.62	0.88	34.00	0.0005	0.0080
GRU	246.81	0.56	0.79	34.00	0.0003	0.0079
Transformer	275.24	0.60	0.85	30.47	0.0004	0.0080
CNN - LSTM	152.26	0.41	0.58	35.78	0.0003	0.0070
CNN - Transformer	248.58	0.57	0.81	31.02	0.0004	0.0079
GAN	99.90	0.30	0.42	50.98	0.0001	0.0072
BnH	320.76	0.49	0.68	61.37	0.0004	0.0143

In the Nikkei 225 case, the LSTM accumulates similar returns with half of the MMD. Notice that all ML and DL models are exposed to less risk and drawdown compared to simple BnH, but in case the benchmark performs well.

5.1.6. HSI Index

Table 17. Machine Learning Metrics for HSI

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9251	0.9300	0.9305	0.9301	0.8481	0.9239
SVM	0.9048	0.9183	0.9021	0.9063	0.8115	0.9038
DT	0.9023	0.9065	0.9118	0.9088	0.8020	0.8997
RF	0.9107	0.9133	0.9212	0.9170	0.8186	0.9079
XGB	0.9168	0.9145	0.9322	0.9231	0.8311	0.9142
LGB	0.9153	0.9128	0.9311	0.9217	0.8278	0.9127
MLP	0.9521	0.9559	0.9547	0.9552	0.9027	0.9511
CNN	0.9540	0.9573	0.9577	0.9570	0.9071	0.9520
LSTM	0.9160	0.9283	0.9148	0.9211	0.8301	0.9147
GRU	0.9243	0.9362	0.9221	0.9286	0.8470	0.9228
Transformer	0.9162	0.9304	0.9126	0.9209	0.8310	0.9159
CNN - LSTM	0.9409	0.9507	0.9389	0.9443	0.8806	0.9395
CNN - Transformer	0.9244	0.9431	0.9138	0.9276	0.8490	0.9254
GAN	0.5355	0.5583	0.5627	0.5486	0.0652	0.5321

For the case of the HSI index, the DT is significantly outperformed by MLP, CNN and CNN - Transformer models in terms of accuracy. The MLP also gains significantly higher accuracy compared to the RF model.

Table 18. Trading Performance for HSI

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	112.31	0.36	0.52	43.11	0.0006	0.0082
SVM	109.63	0.36	0.51	43.55	0.0006	0.0079
DT	137.95	0.40	0.58	41.40	0.0006	0.0082
RF	124.04	0.38	0.55	42.38	0.0006	0.0083
XGB	147.73	0.42	0.60	42.04	0.0006	0.0084

LGB	167.92	0.44	0.64	40.17	0.0006	0.0084
MLP	112.40	0.36	0.52	45.03	0.0007	0.0082
CNN	129.12	0.39	0.57	43.70	0.0005	0.0082
LSTM	252.96	0.55	0.80	35.63	0.0004	0.0084
GRU	226.07	0.52	0.76	35.33	0.0004	0.0083
Transformer	132.41	0.40	0.58	46.31	0.0003	0.0083
CNN - LSTM	156.49	0.43	0.63	42.03	0.0004	0.0083
CNN - Transformer	145.92	0.41	0.61	38.30	0.0003	0.0082
GAN	51.63	0.17	0.26	56.47	0.0001	0.0081
BnH	74.78	0.27	0.38	65.18	0.0002	0.0144

In the HSI market, all ML and DL outperform BnH strategy in terms of returns and risk except GAN models. Among them, LSTM and GRU again yield the best result. Although the market may not perform well similar to FTSE 100, this time all models show good trading performance.

5.2. Emerging Market

5.2.1. BVSP Index

Table 19. Machine Learning Metrics for BVSP

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9333	0.9454	0.9296	0.9373	0.8657	0.9333
SVM	0.9274	0.9366	0.9279	0.9321	0.8538	0.9272
DT	0.9100	0.9192	0.9140	0.9162	0.8187	0.9090
RF	0.9147	0.9304	0.9099	0.9197	0.8286	0.9143
XGB	0.9280	0.9378	0.9280	0.9327	0.8550	0.9273
LGB	0.9265	0.9355	0.9275	0.9313	0.8518	0.9258
MLP	0.9461	0.9538	0.9458	0.9495	0.8919	0.9459
CNN	0.9421	0.9404	0.9548	0.9469	0.8840	0.9395
LSTM	0.9161	0.9211	0.9222	0.9212	0.8308	0.9142
GRU	0.9237	0.9310	0.9280	0.9287	0.8470	0.9220
Transformer	0.9180	0.9284	0.9201	0.9234	0.8358	0.9175
CNN - LSTM	0.9279	0.9314	0.9380	0.9329	0.8571	0.9257

CNN - Transformer	0.9239	0.9337	0.9254	0.9289	0.8474	0.9231
GAN	0.5597	0.6187	0.5506	0.5679	0.1228	0.5627

BVSP is the first emerging market of analysis. Based on the results of statistical analysis, except GAN model, there is no statistically significant difference between models in terms of accuracy.

Table 20. Trading Performance for BVSP

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	176.18	0.51	0.74	38.18	0.0009	0.0091
SVM	188.61	0.53	0.77	38.65	0.0009	0.0091
DT	161.32	0.49	0.70	37.91	0.0007	0.0093
RF	167.89	0.50	0.50	40.26	0.0008	0.0092
XGB	202.56	0.55	0.80	39.60	0.0009	0.0091
LGB	164.12	0.50	0.72	41.20	0.0008	0.0092
MLP	118.72	0.41	0.59	40.76	0.0009	0.0090
CNN	263.00	0.61	0.89	35.01	0.0006	0.0099
LSTM	496.25	0.80	1.19	30.26	0.0005	0.0101
GRU	355.88	0.70	1.02	37.39	0.0005	0.0098
Transformer	310.79	0.66	0.98	31.44	0.0004	0.0099
CNN - LSTM	248.96	0.60	0.87	35.77	0.0005	0.0098
CNN - Transformer	266.12	0.62	0.92	37.16	0.0005	0.0096
GAN	96.20	0.29	0.42	53.37	0.0002	0.0089
BnH	172.97	0.42	0.59	59.96	0.0004	0.0169

Except for MLP and GAN, the rest of the ML and DL models yield similar to better results compared to the benchmark in the case of BVSP index. LSTM GRU and Transformer yield especially high returns. This might be due to the characteristic of BVSP index, with a high value of skewness and kurtosis leading (more extreme positive return).

5.2.2. MXX Index

Table 21. Machine Learning Metrics for MXX

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9235	0.9345	0.9368	0.9354	0.8391	0.9195
SVM	0.9123	0.9226	0.9290	0.9254	0.8163	0.9073

DT	0.8994	0.9217	0.9062	0.9133	0.7903	0.8951
RF	0.9049	0.9069	0.9332	0.9198	0.7996	0.8966
XGB	0.9146	0.9117	0.9465	0.9285	0.8206	0.9058
LGB	0.9144	0.9153	0.9413	0.9280	0.8198	0.9066
MLP	0.9462	0.9466	0.9658	0.9557	0.8861	0.9406
CNN	0.9477	0.9484	0.9670	0.9571	0.8888	0.9409
LSTM	0.9135	0.9184	0.9411	0.9284	0.8198	0.9051
GRU	0.9181	0.9164	0.9527	0.9334	0.8257	0.9055
Transformer	0.9147	0.9310	0.9263	0.9281	0.8214	0.9116
CNN - LSTM	0.9352	0.9317	0.9635	0.9468	0.8633	0.9276
CNN - Transformer	0.9251	0.9264	0.9514	0.9382	0.8410	0.9146
GAN	0.5359	0.6242	0.5739	0.5763	0.0377	0.5130

When testing accuracy on the MXX index, CNN has a better accuracy compared to DT statistically.

Table 22. Trading Performance for MXX

	Return (%)	**SR**	**S**	**MMD (%)**	**Mean Return**	**Stdev Return**
LR	338.05	0.64	0.93	35.06	0.0007	0.0070
SVM	315.53	0.62	0.90	35.04	0.0007	0.0070
DT	283.19	0.58	0.84	44.36	0.0007	0.0068
RF	322.65	0.62	0.91	38.21	0.0006	0.0072
XGB	325.09	0.62	0.91	37.88	0.0007	0.0072
LGB	288.88	0.58	0.85	40.91	0.0007	0.0072
MLP	296.75	0.60	0.87	40.72	0.0007	0.0070
CNN	328.79	0.61	0.89	35.28	0.0005	0.0072
LSTM	528.52	0.78	1.14	34.33	0.0005	0.0174
GRU	403.39	0.67	0.98	32.38	0.0004	0.0076
Transformer	417.96	0.71	1.05	33.79	0.0004	0.0072
CNN - LSTM	316.96	0.59	0.86	37.07	0.0004	0.0073
CNN - Transformer	347.57	0.61	0.89	35.63	0.0004	0.0074
GAN	404.55	0.56	0.81	41.54	0.0002	0.0064

BnH	641.44	0.67	0.96	48.56	0.0004	0.0117

No models are able to outperform the benchmark BnH in the MXX dataset. However, LSTM, GRU and Transfomer give better results than traditional ML models. Also, GAN works exceptionally well in this case.

5.9. TASI Index

Table 23. Machine Learning Metrics for TASI

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9416	0.9516	0.9489	0.9502	0.8772	0.9381
SVM	0.9314	0.9384	0.9466	0.9422	0.8560	0.9266
DT	0.9018	0.9148	0.9196	0.9165	0.7952	0.8967
RF	0.9201	0.9305	0.9337	0.9320	0.8316	0.9150
XGB	0.9336	0.9455	0.9413	0.9433	0.8606	0.9305
LGB	0.9336	0.9466	0.9398	0.9431	0.8606	0.9307
MLP	0.9594	0.9654	0.9667	0.9657	0.9149	0.9556
CNN	0.9421	0.9562	0.9505	0.9516	0.8812	0.9370
LSTM	0.9283	0.9493	0.9277	0.9379	0.8523	0.9273
GRU	0.9432	0.9498	0.9544	0.9519	0.8812	0.9393
Transformer	0.9425	0.9545	0.9483	0.9509	0.8811	0.9411
CNN - LSTM	0.9609	0.9674	0.9667	0.9668	0.9190	0.9587
CNN - Transformer	0.9469	0.9537	0.9575	0.9550	0.8902	0.9448
GAN	0.5307	0.6285	0.5640	0.5648	0.0387	0.5159

In the TASI index dataset, CNN and CNN - LSTM again outperform the DT models.

Table 24. Trading Performance for TASI

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	271.70	1.13	1.63	22.73	0.0008	0.0060
SVM	234.89	1.04	1.48	26.42	0.0007	0.0059
DT	195.48	0.92	1.32	26.09	0.0006	0.0058
RF	186.04	0.91	1.30	24.72	0.0007	0.0059
XGB	221.48	1.01	1.44	24.39	0.0007	0.0059

LGB	217.89	1.00	1.43	24.70	0.0007	0.0059
MLP	241.48	1.06	1.53	23.01	0.0007	0.0060
CNN	220.13	1.02	1.46	25.35	0.0005	0.0061
LSTM	239.12	1.09	1.54	22.43	0.0005	0.0062
GRU	262.35	1.15	1.66	23.36	0.0005	0.0062
Transformer	242.82	1.10	1.57	23.27	0.0006	0.0061
CNN - LSTM	240.50	1.09	1.55	22.38	0.0006	0.0062
CNN - Transformer	206.86	1.00	1.42	22.45	0.0005	0.0063
GAN	56.29	0.37	0.50	38.72	0.0002	0.0064
BnH	165.52	0.60	0.83	51.1	0.0003	0.0116

Except for GAN, the rest of the ML and DL models gain better results compared to the benchmark. These models not only outperform in return but also in risk. In this case, the ML and DL have roughly the same results, except for the tree-based model DT and RF yield lower total returns.

5.10. QE General Index

Table 25. Machine Learning Metrics for QE General

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9263	0.9313	0.9292	0.9298	0.8496	0.9235
SVM	0.9175	0.9139	0.9324	0.9223	0.8328	0.9162
DT	0.8945	0.8986	0.9031	0.9002	0.7844	0.8902
RF	0.9065	0.9193	0.9026	0.9104	0.8089	0.9038
XGB	0.9151	0.9252	0.9138	0.9189	0.8267	0.9121
LGB	0.9146	0.9213	0.9168	0.9185	0.8256	0.9117
MLP	0.9500	0.9584	0.9452	0.9514	0.8988	0.9494
CNN	0.9307	0.9234	0.9506	0.9354	0.8607	0.9255
LSTM	0.9148	0.9171	0.9281	0.9215	0.8274	0.9112
GRU	0.9442	0.9503	0.9441	0.9470	0.8862	0.9427
Transformer	0.9358	0.9425	0.9358	0.9382	0.8706	0.9338
CNN - LSTM	0.9407	0.9380	0.9512	0.9440	0.8797	0.9382
CNN - Transformer	0.9266	0.9357	0.9242	0.9289	0.8520	0.9248

GAN	0.5682	0.5807	0.6258	0.5924	0.1273	0.5599

In this dataset, the DT's accuracy is significantly lower than DL models like MLP, GRU and CNN - LSTM.

Table 26. Trading Performance for QE General

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	141.16	0.68	1.01	26.50	0.0001	0.0060
SVM	109.72	0.57	0.84	31.49	0.0005	0.0060
DT	123.30	0.62	0.90	26.84	0.0005	0.0059
RF	136.74	0.67	1.00	27.15	0.0005	0.0060
XGB	114.94	0.60	0.89	27.67	0.0005	0.0060
LGB	126.68	0.64	0.95	26.39	0.0005	0.0060
MLP	126.55	0.63	0.92	29.31	0.0005	0.0060
CNN	60.92	0.37	0.52	39.96	0.0002	0.0065
LSTM	76.04	0.43	0.60	42.49	0.0002	0.0065
GRU	140.86	0.66	0.97	29.40	0.0004	0.0062
Transformer	68.13	0.41	0.58	38.96	0.0003	0.0064
CNN - LSTM	70.64	0.42	0.61	36.41	0.0003	0.0062
CNN - Transformer	75.94	0.44	0.63	34.44	0.0003	0.0064
GAN	13.18	0.09	0.13	44.45	0.0000	0.0059
BnH	11.87	0.16	0.21	55.19	0.0001	0.0112

All ML and DL models perform better than the BnH benchmark. For the QE index, traditional ML models even yield better results compared to the DL models, except for GRU models. LSTM is no longer in the top best-performance models. This could be due to the bad performance of the QE index, as same as the FTSE 100 index, which result in bad generalize ability of the LSTM model.

5.11. SSEC Index

Table 27. Machine Learning Metrics for SSEC

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9247	0.9274	0.9251	0.9261	0.8483	0.9240
SVM	0.9129	0.9107	0.9207	0.9149	0.8258	0.9121
DT	0.8959	0.9011	0.8945	0.8974	0.7903	0.8952

RF	0.9081	0.9158	0.9023	0.9089	0.8148	0.9076
XGB	0.9185	0.9274	0.9113	0.9192	0.8356	0.9179
LGB	0.9130	0.9211	0.9073	0.9140	0.8246	0.9126
MLP	0.9544	0.9668	0.9401	0.9526	0.9090	0.9536
CNN	0.9450	0.9414	0.9517	0.9458	0.8907	0.9454
LSTM	0.9321	0.9338	0.9318	0.9324	0.8637	0.9318
GRU	0.9416	0.9473	0.9352	0.9410	0.8824	0.9409
Transformer	0.9270	0.9324	0.9227	0.9269	0.8537	0.9271
CNN - LSTM	0.9397	0.9318	0.9524	0.9412	0.8800	0.9397
CNN - Transformer	0.9264	0.9355	0.9187	0.9260	0.8532	0.9263
GAN	0.5249	0.5592	0.5552	0.5262	0.0551	0.5246

As same as the QE General index, GRU, MLP and CNN - LSTM tend to have better accuracy compared to the DT model. MLP is also better than RF models. These results are statistically significant.

Table 28. Trading Performance for SSEC

	Return (%)	**SR**	**S**	**MMD (%)**	**Mean Return**	**Stdev Return**
LR	346.29	0.62	0.87	41.02	0.0009	0.0091
SVM	227.51	0.51	0.71	45.63	0.0008	0.0089
DT	231.64	0.52	0.72	52.14	0.0007	0.0089
RF	273.07	0.57	0.8+	47.89	0.0008	0.0089
XGB	231.05	0.52	0.73	45.69	0.0008	0.0089
LGB	260.95	0.55	0.77	45.62	0.0008	0.0089
MLP	405.61	0.67	0.94	39.25	0.0009	0.0088
CNN	452.50	0.69	0.98	38.71	0.0006	0.0091
LSTM	618.08	0.78	1.10	37.60	0.0006	0.0091
GRU	589.36	0.76	1.08	38.04	0.0006	0.0091
Transformer	442.73	0.69	0.97	38.82	0.0005	0.0091
CNN - LSTM	438.95	0.68	0.95	41.58	0.0006	0.0091
CNN - Transformer	332.88	0.61	0.85	43.83	0.0005	0.0092
GAN	110.14	0.29	0.41	57.03	0.0002	0.0092

BnH	118.92	0.33	0.46	71.98	0.0003	0.0148

Except for GAN, the rest of the ML and DL models completely outperform the benchmark. In this market, DL tends to perform better than traditional ML models, with the LSTM and GRU results are superior, suggesting DL should be applied when investing in SSEC.

5.12. JKSE Index

Table 29. Machine Learning Metrics for JKSE

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9249	0.9444	0.9305	0.9370	0.8428	0.9219
SVM	0.8997	0.9197	0.9159	0.9160	0.7931	0.8945
DT	0.8872	0.9031	0.9106	0.9064	0.7627	0.8790
RF	0.9038	0.9179	0.9237	0.9205	0.7966	0.8966
XGB	0.9133	0.9237	0.9344	0.9285	0.8173	0.9055
LGB	0.9133	0.9212	0.9366	0.9285	0.8168	0.9052
MLP	0.9315	0.9282	0.9627	0.9442	0.8581	0.9232
CNN	0.9364	0.9583	0.9345	0.9449	0.8712	0.9354
LSTM	0.9033	0.9074	0.9357	0.9208	0.7973	0.8957
GRU	0.9158	0.9263	0.9366	0.9305	0.8239	0.9083
Transformer	0.9089	0.9338	0.9132	0.9227	0.8115	0.9063
CNN - LSTM	0.9180	0.9413	0.9220	0.9298	0.8333	0.9167
CNN - Transformer	0.9262	0.9437	0.9335	0.9379	0.8470	0.9234
GAN	0.5712	0.6610	0.5896	0.5873	0.1321	0.5653

There are no statistically significant differences between models in terms of accuracy in JKSE index.

Table 30. Trading Performance for JKSE

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	1587.88	1.47	2.39	26.49	0.0008	0.0064
SVM	1566.67	1.46	2.37	28.57	0.0008	0.0063
DT	1447.14	1.39	2.20	28.26	0.0007	0.0066
RF	1515.02	1.43	2.30	28.06	0.0007	0.0065

XGB	1484.61	1.42	2.27	26.85	0.0008	0.0066
LGB	1472.37	1.41	2.24	27.26	0.0008	0.0066
MLP	1409.48	1.37	2.16	30.39	0.0008	0.0065
CNN	1328.75	1.34	2.12	33.11	0.0007	0.0066
LSTM	1655.04	1.51	2.45	27.31	0.0006	0.0064
GRU	1519.00	1.45	2.32	29.87	0.0007	0.0064
Transformer	1602.19	1.48	2.42	27.77	0.0006	0.0064
CNN - LSTM	1269.48	1.33	2.09	34.04	0.0006	0.0066
CNN - Transformer	1450.92	1.41	2.25	30.48	0.0006	0.0065
GAN	370.47	0.59	0.85	53.47	0.0003	0.0071
BnH	736.75	0.68	0.96	79.88	0.0005	0.0138

JKSE index shows similar results to the SSEC index, with all models except for GAN. The difference here both ML and DL have equally performed, suggesting a good index to apply ML and DL trading strategies. Most models yield nearly double the returns of BnH strategy.

5.13. VNINDEX

Table 31. Machine Learning Metrics for VNINDEX

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9309	0.9421	0.9329	0.9374	0.8586	0.9294
SVM	0.9125	0.9060	0.9424	0.9233	0.8212	0.9078
DT	0.9098	0.9173	0.9210	0.9189	0.8149	0.9064
RF	0.9179	0.9263	0.9259	0.9260	0.8316	0.9149
XGB	0.9277	0.9336	0.9362	0.9348	0.8517	0.9252
LGB	0.9259	0.9303	0.9366	0.9333	0.8481	0.9233
MLP	0.9415	0.9380	0.9608	0.9483	0.8823	0.9380
CNN	0.9486	0.9543	0.9516	0.9527	0.8960	0.9485
LSTM	0.9308	0.9367	0.9396	0.9377	0.8595	0.9295
GRU	0.9330	0.9362	0.9441	0.9395	0.8645	0.9319
Transformer	0.9249	0.9521	0.9126	0.9256	0.8562	0.9268
CNN - LSTM	0.9343	0.9554	0.9234	0.9386	0.8677	0.9344

CNN - Transformer	0.9218	0.9246	0.9392	0.9301	0.8440	0.9205
GAN	0.5421	0.5975	0.5508	0.5468	0.0919	0.5481

There is no significant difference between models for the VNINDEX dataset.

Table 32. Trading Performance for VNINDEX

	Return (%)	SR	S	MMD (%)	Mean Return	Stdev Return
LR	1822.66	1.19	1.79	25.98	0.0010	0.0074
SVM	1581.04	1.13	1.69	29.15	0.0009	0.0073
DT	1144.44	1.00	1.47	34.66	0.0009	0.0064
RF	1173.04	1.02	1.50	29.97	0.0008	0.0075
XGB	1234.52	1.04	1.53	33.56	0.0009	0.0075
LGB	1212.41	1.03	1.52	34.77	0.0009	0.0076
MLP	1306.74	1.04	1.54	32.45	0.0009	0.0076
CNN	1339.27	1.07	1.60	32.59	0.0007	0.0074
LSTM	1655.29	1.11	1.64	30.77	0.0007	0.0079
GRU	1536.27	1.10	1.64	31.03	0.0006	0.0078
Transformer	1332.79	1.07	1.58	32.03	0.0006	0.0077
CNN - LSTM	1169.11	1.02	1.51	32.23	0.0006	0.0074
CNN - Transformer	1267.71	1.05	1.55	31.39	0.0006	0.0077
GAN	694.74	0.77	1.10	43.52	0.0004	0.0069
BnH	1232.40	0.78	1.08	60.73	0.0005	0.0126

There is a mixed result when applying ML and DL trading models on the VNINDEX market. Some ML models work well like LR and SVM, and others can not beat the index. DL models do slightly better when only CNN - LSTM can not beat the benchmark while others are able to do that. However, the results are not very impressive.

5.14. BET Index

Table 33. Machine Learning Metrics for BET

	Accuracy	Precision	Recall	F1	MCC	AUC
LR	0.9272	0.9405	0.9411	0.9407	0.8448	0.9223
SVM	0.9132	0.9373	0.9211	0.9286	0.8179	0.9109
DT	0.8981	0.9158	0.9193	0.9171	0.7831	0.8890

RF	0.9118	0.9221	0.9355	0.9286	0.8115	0.9026
XGB	0.9188	0.9297	0.9387	0.9340	0.8268	0.9110
LGB	0.9160	0.9233	0.9414	0.9321	0.8206	0.9067
MLP	0.9363	0.9490	0.9522	0.9493	0.8667	0.9305
CNN	0.9078	0.9485	0.9040	0.9224	0.8168	0.9104
LSTM	0.9117	0.9158	0.9492	0.9313	0.8087	0.8971
GRU	0.9183	0.9338	0.9372	0.9350	0.8239	0.9108
Transformer	0.9098	0.9192	0.9410	0.9287	0.8065	0.8965
CNN - LSTM	0.9279	0.9430	0.9435	0.9425	0.8463	0.9220
CNN - Transformer	0.9150	0.9483	0.9143	0.9298	0.8236	0.9145
GAN	0.5306	0.6447	0.5249	0.5490	0.0557	0.5246

There is no significant difference between models for the BET dataset.

Table 34. Trading Performance for BET

	Return (%)	**SR**	**S**	**MMD (%)**	**Mean Return**	**Stdev Return**
LR	3836.86	1.21	1.81	51.82	0.0010	0.0074
SVM	3545.81	1.19	1.79	51.97	0.0009	0.0073
DT	4604.95	1.28	1.94	41.18	0.0009	0.0064
RF	4033.99	1.24	1.86	45.67	0.0008	0.0075
XGB	4266.19	1.25	1.88	45.94	0.0009	0.0075
LGB	4502.54	1.26	1.92	41.53	0.0009	0.0076
MLP	4348.09	1.24	1.86	47.65	0.0009	0.0076
CNN	3390.02	1.22	1.84	40.87	0.0007	0.0074
LSTM	7393.26	1.39	2.12	43.08	0.0007	0.0079
GRU	5563.93	1.32	2.01	40.26	0.0006	0.0078
Transformer	6940.34	1.36	2.06	41.25	0.0006	0.0077
CNN - LSTM	4546.48	1.28	1.96	41.23	0.0006	0.0074
CNN - Transformer	4708.91	1.28	1.93	46.68	0.0006	0.0077
GAN	974.66	0.71	1.01	54.10	0.0004	0.0069
BnH	3077.19	0.89	1.26	82.55	0.0005	0.0126

The results for the BET Index show that deep learning models like LSTM and Transformer delivered the highest returns, significantly outperforming the BnH strategy. Traditional ML models also performed well, while hybrid models showed strong potential. These findings highlight the superiority of advanced ML/DL models in financial forecasting.

6. Conclusion

This study aims to perform a meta-analysis on the effectiveness of various ML and DL models in stock market predictions and trading execution across a wide range of financial markets. By applying these models to different market indices and creating robustness performance through CPCV algorithm and backtesting under realistic trading conditions, we unfold empirical insights into their predictive capabilities and practical utility.

First, despite achieving high accuracy in classification metrics, ML and DL models do not necessarily translate into better trading performance, revealing that predictive accuracy alone is not enough to ensure profitability, especially when trading cost is considered

Second, our results show that DL, particularly LSTM, GRU and sometime, Transformer, consistently outperforms traditional ML models in trading performance in many cases. These results might be due to the ability to capture temporal dependencies, as the input for DL models is data with a 20-day historical lookback window, which helps them recognize sequential patterns and trends in stock price. Additionally, while single-component models like CNN, LSTM and Transformer provide good accuracy and trading results, combining them into hybrid models does not make the results better, some cases depreciate it. We believe that these architectures may be more effective when working with larger feature sets and data or when deeper, more complex models are required.

Third, even though some ML and DL models outperform the benchmark BnH strategy in some cases, their Sharpe ratios remain relatively low (below 1). This indicates that these strategies require further refinement and risk management adjustments before they can be used in live trading. However, all models exhibit lower risk than the benchmark BnH strategy in terms of standard deviation and maximum drawdown, suggesting that they offer good potential for reducing downside risk while maintaining competitive returns.

Fourth, the effect of market structure, liquidity and market efficiency play an important role in the trading performance of models. This could explain why ML and DL models struggle to consistently outperform BnH in developed markets, such as the S&P 500, TSX, FTSE 100, DAX, Nikkei 225, and HSI while showing sound results in emerging and frontier markets (i.e., SSEC, JKSE, VNINDEX, BET, BVSP, MXX, QSI, TASI), where market inefficiencies create exploitable trading opportunities. Since this study relies on technical indicators (TA) as model inputs, the stronger performance of ML/DL in less efficient markets suggests that TA-based approaches may be more effective in environments where patterns and trends persist due to lower market efficiency. Indices from the APAC region (HSI, SSEC, JKSE, VNINDEX) have shown promising results, which may be influenced by cultural and behavioral finance factors. Additionally, previous research (Oprean & Tanasescu, 2014) has shown that BVSP and BET are affected by behavioral finance, and their performance in this study further supports this idea, as they also exhibit promising results.

Fifth, GAN model underperformance could be due to that GAN is trained as a regression task rather than a classification task, making it less suitable for generating trading signals. However, GAN like other complex DL models works best when there is a vast of training data, and requires careful tuning and high computational resources to be effective.

From these key points, we provide some practical implications for investors, traders and researchers as follows:

a. Model selection should be designed based on market conditions:

- DL, particularly LSTM and GRU work best in multiple market conditional, especially in less efficient markets
- For developed markets, consider adding alternative data sources (fundamental ratios, macroeconomic variables, sentiment analysis, …) for better performance
- Traditional ML remains valuable if investors need a transparent decision-making process, and computational efficiency. Also notice that ML models should only be applied in less efficient markets because their ability to recognize patterns is still not as good as DL.

b. GANs models should considered to be used for data augmentation for enhancing robustness and help improvements for other models

c. This study serves as a baseline study for model and market selection. Therefore, despite outperforming the benchmark in some cases, ML and DL models require further refinements and tuning to improve Sharpe ratios and overall risk-adjusted returns, making them more practical for real-world trading.

Finally, future research could explore several ideas: (1) Assess the promising reinforcement learning-based trading models to see how the agent dynamically adapts to different market sets up and environments; (2) Studies could examine the impact of extreme market events and cases when market did not perform well (FTSE 100 Index) or include a regime switch function to see how ML and DL perform under stress conditions and weak market; (3) Incorporating alternative data sources and behavioral finance set up to enhance accuracy and trading performance.

References

[1] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock price prediction using the ARIMA model", Proceedings of the 16th UKSim-AMSS International Conference on Computer Modelling and Simulation, Cambridge, UK, **(2014)** March 26-28, pp. 106-112.

[2] A. Nguyen and S. Ha, "A lightweight multi-head attention transformer for stock price forecasting", Available at SSRN, article ID 4729648, **(2023)**.

[3] A. Peiró, "The distribution of stock returns: international evidence", Applied Financial Economics, vol. 4, no. 6, **(1994)**, pp. 431-439.

[4] A. S. M. Shihavuddin, M. N. Ambia, M. M. N. Arefin, M. Hossain, and A. Anwar, "Prediction of stock price analyzing the online financial news using Naive Bayes classifier and local economic trends", 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 4, **(2010)**, pp. V4-22, IEEE.

[5] A. Subasi, et al., "Stock market prediction using machine learning", Procedia Computer Science, vol. 194, **(2021)**, pp. 173-179.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, … & I. Polosukhin, "Attention is all you need", Advances in Neural Information Processing Systems, vol. 30, **(2017)**.

[7] A. W. Lo, H. Mamaysky and J. Wang, "Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation", The Journal of Finance, vol. 55, no. 4, **(2000)**, pp. 1705-1765.

[8] C. Cortes and V. Vapnik, "Support-vector networks", Machine Learning, vol. 20, **(1995)**, pp. 273-297.

[9] C. F. Tsai, Y. C. Lin, D. C. Yen, and Y. M. Chen, "Predicting stock returns by classifier ensembles", Applied Soft Computing, vol. 11, no. 2, **(2011)**, pp. 2452-2459.

[10] C. Xiang and W. M. Fu, "Predicting the stock market using multiple models", 2006 9th International Conference on Control, Automation, Robotics and Vision, **(2006)**, pp. 1-6, IEEE.

[11] D. Fister, et al., "Deep learning for stock market trading: a superior trading strategy?", Neural Network World, vol. 3, **(2019)**.

[12] D. Lv, S. Yuan, M. Li, and Y. Xiang, "An empirical study of machine learning algorithms for stock daily trading strategy", Mathematical Problems in Engineering, vol. 2019, no. 1, **(2019)**, article ID 7816154.

[13] D. W. Hosmer Jr, S. Lemeshow and R. X. Sturdivant, Applied Logistic Regression, John Wiley & Sons, **(2013)**.

[14] E. B. Deakin, "A discriminant analysis of predictors of business failure", Journal of Accounting Research, vol. 10, no. 1, **(1972)**, pp. 167-179.

[15] E. Tricker, S. Srivastava, and M. Mitchell, "Transaction costs", Retrieved from Graham Capital Management library, **(2017)**.

[16] G. Ke, et al., "LightGBM: A highly efficient gradient boosting decision tree", Advances in Neural Information Processing Systems, vol. 30, **(2017)**.

[17] H. Lin, C. Chen, G. Huang, and A. Jafari, "Stock price prediction using generative adversarial networks", Journal of Computer Science, vol. 17, no. 3, **(2021)**, pp. 188-196.

[18] H. Song and H. Choi, "Forecasting stock market indices using the recurrent neural network based hybrid models: CNN-LSTM, GRU-CNN, and ensemble models", Applied Sciences, vol. 13, no. 7, **(2023)**, article ID 4644.

[19] H. Taud and J. F. Mas, "Multilayer perceptron (MLP)", Geomatic Approaches for Modeling Land Change Scenarios, Cham: Springer International Publishing, **(2017)**, pp. 451-455.

[20] H. Wang, Y. Jiang, and H. Wang, "Stock return prediction based on Bagging-decision tree", 2009 IEEE International Conference on Grey Systems and Intelligent Services (GSIS 2009), **(2009)**, IEEE.

[21] I. Goodfellow, et al., "Generative adversarial nets", Advances in Neural Information Processing Systems, vol. 27, **(2014)**.

[22] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A comprehensive evaluation of ensemble learning for stock-market prediction", Journal of Big Data, vol. 7, no. 1, **(2020)**, article ID 20.

[23] K. Alkhatib, H. Najadat, I. Hmeidi and M. K. A. Shatnawi, "Stock price prediction using k-nearest neighbor (kNN) algorithm", International Journal of Business, Humanities and Technology, vol. 3, no. 3, **(2013)**, pp. 32-44.

[24] K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market", IEEE International Conference on Big Data, **(2015)**, pp. 2823-2824.

[25] K. Cho, et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation", arXiv preprint, arXiv:1406.1078, **(2014)**.

[26] K. O'Shea, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, **(2015)**.

[27] K. Żbikowski, "Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy", Expert Systems with Applications, vol. 42, no. 4, **(2015)**, pp. 1797-1805.

[28] L. Breiman, Classification and Regression Trees, Routledge, **(2017)**.

[29] L. Breiman, "Random forests", Machine Learning, vol. 45, **(2001)**, pp. 5-32.

[30] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting", IEEE Transactions on Neural Networks, vol. 14, no. 6, **(2003)**, pp. 1506-1518.

[31] L. Tabaro, et al., "Algorithmic trading using double deep Q-networks and sentiment analysis", Information, vol. 15, no. 8, **(2024)**, article ID 473.

[32] M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein generative adversarial networks", Proceedings of the 34th International Conference on Machine Learning (ICML'17), Sydney, Australia, **(2017)** August 6-11, pp. 214-223.

[33] M. H. D. M. Ribeiro and L. S. Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series", Applied Soft Computing, vol. 86, **(2020)**, article ID 105837.

[34] M. L. De Prado, Advances in Financial Machine Learning, John Wiley & Sons, **(2018)**.

[35] M. R. Toochaei and F. Moeini, “Evaluating the performance of ensemble classifiers in stock returns prediction using effective features”, Expert Systems with Applications, vol. 213, **(2023)**, article ID 119186.

[36] N. H. Hung, “Various moving average convergence divergence trading strategies: A comparison”, Investment Management and Financial Innovations, vol. 13, no. 2, **(2016)**, pp. 363-369.

[37] N. H. Hung and Y. Zhaojun, “Profitability of applying simple moving average trading rules for the Vietnamese stock market”, Journal of Business Management, vol. 2, no. 3, **(2013)**, pp. 223-231.

[38] R. R. Officer, “The distribution of stock returns”, Journal of the American Statistical Association, vol. 67, no. 340, **(1972)**, pp. 807-812.

[39] R. Rosillo, D. De la Fuente, and J. A. Lopez Brugos, “Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies”, Applied Economics, vol. 45, no. 12, **(2013)**, pp. 1541-1550.

[40] S. B. Imandoust and M. Bolandraftar, “Forecasting the direction of stock market index movement using three data mining techniques: the case of Tehran Stock Exchange”, International Journal of Engineering Research and Applications, vol. 4, no. 6, **(2014)**, pp. 106-117.

[41] S. Hochreiter, “Long Short-term Memory”, Neural Computation, MIT Press, vol. 9, no. 8, **(1997)**, pp. 1735-1780.

[42] S. M. Nor and G. Wickremasinghe, “The profitability of MACD and RSI trading rules in the Australian stock market”, Investment Management and Financial Innovations, vol. 11, no. 4, **(2014)**, pp. 194-199.

[43] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model”, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), **(2017)**, pp. 1643-1647, IEEE.

[44] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system”, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, **(2016)** August 13-17, pp. 785-794.

[45] T. Muhammad, et al., “Transformer-based deep learning model for stock price prediction: A case study on Bangladesh stock market”, International Journal of Computational Intelligence and Applications, vol. 22, no. 3, **(2023)**, article ID 2350013.

[46] W. B. Arowolo, “Predicting stock prices returns using GARCH model”, The International Journal of Engineering and Science, vol. 2, no. 5, **(2013)**, pp. 32-37.

[47] W. Brock, J. Lakonishok and B. LeBaron, “Simple technical trading rules and the stochastic properties of stock returns”, The Journal of Finance, vol. 47, no. 5, **(1992)**, pp. 1731-1764.

[48] W. Huang, Y. Nakamori and S. Y. Wang, “Forecasting stock market movement direction with support vector machine”, Computers & Operations Research, vol. 32, no. 10, **(2005)**, pp. 2513-2522.

[49] W. Mao, P. Liu, and J. Huang, “SF-Transformer: A mutual information-enhanced transformer model with spot-forward parity for forecasting long-term Chinese stock index futures prices”, Entropy, vol. 26, no. 6, **(2024)**, article ID 478.

[50] Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange”, Expert Systems with Applications, vol. 38, no. 5, **(2011)**, pp. 5311-5319.

[51] Z. Zheng, “Boosting and bagging of neural networks with applications to financial time series”, Neural Network, The University of Chicago, **(2006)**.

[52] Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, and M. Veloso, “Financial time series forecasting using CNN and transformer”, arXiv preprint arXiv:2304.04912, **(2023)**.