Data Science and Engineering (DSE) Record, Volume 6, issue 1.

Development of Model to Predict Next Day's Asset Price Movements Using Ensemble Classification Techniques

Natchar Pongsri [1] and Nasi Tantitharanukul [2]

[1] Master's Degree Program in Data Science, Chiang Mai University, Chiang Mai, Thailand
[2] Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand
Natchar_P@cmu.ac.th

Abstract This research presents a predictive model for determining the next-day price direction of EUR/USD in the Binary Options market. The study utilizes technical indicators and price data over a 10,000-day span, collected from TradingView, and applies machine learning techniques particularly an ensemble classification framework combining CNN, LSTM, SVM, and XGBoost models. A total of 23 features were engineered from candlestick data and popular indicators such as RSI, MACD, ATR, and EMA. Statistical analysis ensured data quality and distribution symmetry. Model performance was evaluated using accuracy, F1 score, and ROC-AUC metrics. The resulting ensemble model outperformed individual models in predictive accuracy and stability. This research contributes to the development of automated trading systems and serves as a foundation for further work in financial time series forecasting using machine learning.

Keywords: Binary Options, Ensemble Learning, EUR/USD, Machine Learning

1 Introduction

Financial markets are highly dynamic and influenced by various factors such as global economic shifts, political events, and investor sentiment. Predicting asset price movements, especially in short time frames like a single day is both a practical need and a technical challenge for traders. Binary Options trading requires precise prediction of whether an asset's price will go up or down over a fixed short period. While traditional technical analysis techniques like RSI (Relative Strength Index), EMA (Exponential Moving Average), and MACD (Moving Average Convergence Divergence) are useful, they often lack the ability to adapt to complex and fast-changing patterns in financial time series.With the advancement of machine learning and deep learning, models such as LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) have shown strong capabilities in capturing temporal and nonlinear patterns. Furthermore, ensemble learning by combining the strengths of multiple models can reduce individual model bias and improve prediction accuracy. This research aims to develop an ensemble classification model to predict the next-day direction of EUR/USD prices using a dataset of over 10,000 candlesticks and 23 technical features. The model is intended to assist in decision-making for Binary Options trading by improving predictive performance over traditional approaches.

Fig. 1. Binary Options trading interface for EUR/USD on IQ Option.

Figure 1 Example of a Binary Options trading interface on IQ Option for the EUR/USD currency pair. The platform allows users to predict whether the asset price will rise ("Call") or fall ("Put") within a specified time frame. The chart displays real-time candlestick movements and technical indicators such as moving averages, Parabolic SAR, and RSI, which are commonly used in financial forecasting models.

2 Literature Review

Recent advances in machine learning and deep learning have led to increasingly effective models for financial market forecasting. Models such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Support Vector Machines (SVM) have been widely applied to predict price movements in time-series data. These models can capture complex, nonlinear patterns that are difficult to detect using traditional statistical approaches (Selvin et al., 2017). Bi-Directional LSTM (Bi-LSTM) has been shown to improve prediction accuracy by learning from both past and future sequences in time-series data (Shah et al., 2021). Similarly, hybrid models that combine deep learning and rule-based techniques have also proven effective, especially in volatile markets like cryptocurrency (Uzun et al., 2024). Additionally, ensemble learning techniques—including Stacking and Averaging—have gained attention for their ability to combine multiple models to improve generalization and reduce prediction error. Studies have demonstrated that combining LSTM, CNN, GRU (Gated recurrent unit), and tree-based models like XGBoost leads to more robust and accurate forecasting, particularly for short-term asset movements (Livieris et al., 2020; Karim et al., 2021). This research builds on those findings by integrating a variety of classifiers into a single ensemble model tailored for the Binary Options context, where predicting the next day's price direction (up or down) is critical to trading success.

3 Data and Methodology

3.1 Data Source

The dataset used in this study consists of daily candlestick data for the EUR/USD currency pair, retrieved from FXCM via the TradingView platform. The data spans 10,015 trading days from 1986 to 2025, and includes timestamped price information (open, high, low, close) along with technical indicators. The data was exported in CSV format and thoroughly validated for completeness and consistency prior to model development. A total of 23 features were engineered from the raw data, covering both price-based and indicator-based metrics. These include

1.) Candlestick Prices: Open, High, Low, Close

2.) Trend Indicators: EMA (8, 13, 21, 34, 55, 100, 200), Bollinger Bands (Basis, Upper, Lower)

3.) Momentum Indicators: RSI, RSI-based MA, MACD, Signal Line, Histogram

4.) Volatility Indicators: Average True Range (ATR), Rolling Volatility

5.) Others: Plot (RSI-derived), and timestamp (time)

All features were normalized and inspected for correlation to ensure optimal input representation for the learning models. (see Figure 2 and Figure 3)

time	open	high	low	close	Basis	Upper	Lower	EMA8	EMA13	EMA21	EMA34
2/6/1986	0.84	0.8449	0.8336	0.8416	0.87357	0.913572	0.833568	0.852563	0.85952	0.865595	0.867772
3/6/1986	0.8414	0.8637	0.8414	0.8621	0.87226	0.911962	0.832558	0.854683	0.859888	0.865277	0.867448
4/6/1986	0.8621	0.8668	0.8567	0.8629	0.87045	0.908352	0.832548	0.856509	0.860318	0.865061	0.867188
5/6/1986	0.8627	0.8764	0.8615	0.876	0.86954	0.905963	0.833117	0.86084	0.862559	0.866055	0.867692
6/6/1986	0.8754	0.8817	0.8731	0.8809	0.868915	0.904086	0.833744	0.865298	0.865179	0.867405	0.868447
9/6/1986	0.8809	0.8825	0.8698	0.8701	0.86729	0.898909	0.835671	0.866365	0.865882	0.86765	0.868541
10/6/1986	0.87	0.8838	0.8694	0.8815	0.86696	0.897833	0.836087	0.869728	0.868113	0.868909	0.869282
11/6/1986	0.8815	0.8872	0.8805	0.886	0.86649	0.895867	0.837113	0.873344	0.870668	0.870463	0.870237
12/6/1986	0.8856	0.8899	0.8815	0.8881	0.86638	0.895414	0.837346	0.876623	0.873159	0.872066	0.871258
13/6/1986	0.8881	0.8897	0.8795	0.8848	0.866595	0.896103	0.837087	0.87844	0.874822	0.873224	0.872032

Fig. 2. Example of the first 10 rows of the input dataset, showing columns 1 to 12.

EMA55	EMA100	EMA200	ATR	RSI	RSI-based MA	Plot	Rolling Volatility	Histogram	MACD	Signal
0.863634	0.845566	0.80366	0.011676	33.82096	42.72428208	-3.08584	0.027981253	-0.00502	-0.00886	-0.00384
0.863579	0.845893	0.804241	0.012435	47.36452	42.04025463	-2.19212	0.391397449	-0.00325	-0.0079	-0.00465
0.863555	0.84623	0.804825	0.012268	47.81336	41.59220933	-1.69116	0.263319404	-0.00188	-0.007	-0.00512
0.863999	0.84682	0.805533	0.012456	54.63516	41.99459119	-1.3024	0.357273913	-3.58E-05	-0.00517	-0.00513
0.864603	0.847494	0.806283	0.012181	56.90441	42.37675913	-1.04029	0.347871824	0.00148	-0.00328	-0.00476
0.864799	0.847942	0.806918	0.012218	50.865	42.66671539	-0.90957	0.486900358	0.001706	-0.00263	-0.00433
0.865395	0.848607	0.80766	0.012374	56.15477	43.75964494	-0.74747	0.505026902	0.002525	-0.00118	-0.0037
0.866131	0.849347	0.80844	0.011968	58.07355	45.09612873	-0.59382	0.505181146	0.003228	0.000333	-0.0029
0.866916	0.850114	0.809232	0.011713	58.97582	46.5195465	-0.48722	0.609084579	0.003659	0.001679	-0.00198
0.867555	0.850801	0.809984	0.011605	56.90346	47.79493869	-0.39535	0.523486219	0.003545	0.002451	-0.00109

Fig. 3. Example of the first 10 rows of the input dataset, showing columns 13 to 23.

3.2 Statistical Feature Engineering

To enhance the predictive capacity of the model, several statistical techniques were applied to construct meaningful features from the raw time-series data. These include:

1.) Price Change Calculation

Daily price change and percent change were computed using

$$\text{Price Change} = \text{Close}_t - \text{Close}_{t-1}$$

$$\text{Percent Change} = \frac{\text{Price Change}}{\text{Close}_{t-1}} x100$$

2.) Directional Labeling

A binary classification label was created based on the direction of the next day's closing price.

$$\text{Label} = \begin{cases} 1 & \text{if Close}_{t+1} > \text{Close}_t \\ 0 & \text{otherwise} \end{cases}$$

3.) Bullish and Bearish Candlestick Count

Rolling windows were used to count the number of bullish (Close > Open) and bearish (Close < Open) candlesticks over the past N days to capture short-term trend behavior.

4.) Symmetry and Distribution Checks

Variables such as RSI and price change distributions were examined for symmetry using histogram plots (see Figure 4). This helped confirm the suitability of features for model input and reduced the likelihood of introducing bias during training.

5.) Volatility Estimation

Rolling standard deviation was used to calculate Rolling Volatility, which captures recent fluctuations in price and serves as a measure of market uncertainty.

These features were combined with traditional technical indicators to form a comprehensive input matrix for training the ensemble models.

3.2.1 Feature Distribution and Analysis

To evaluate the suitability of the input features for modeling, the distribution of key variables was analyzed. In particular, the Relative Strength Index (RSI) was selected due to its popularity in momentum-based trading strategies. The histogram and kernel density estimate (KDE) of RSI values revealed a near-symmetrical distribution centered around 50, with well-defined overbought and oversold thresholds at 70 and 30, respectively.

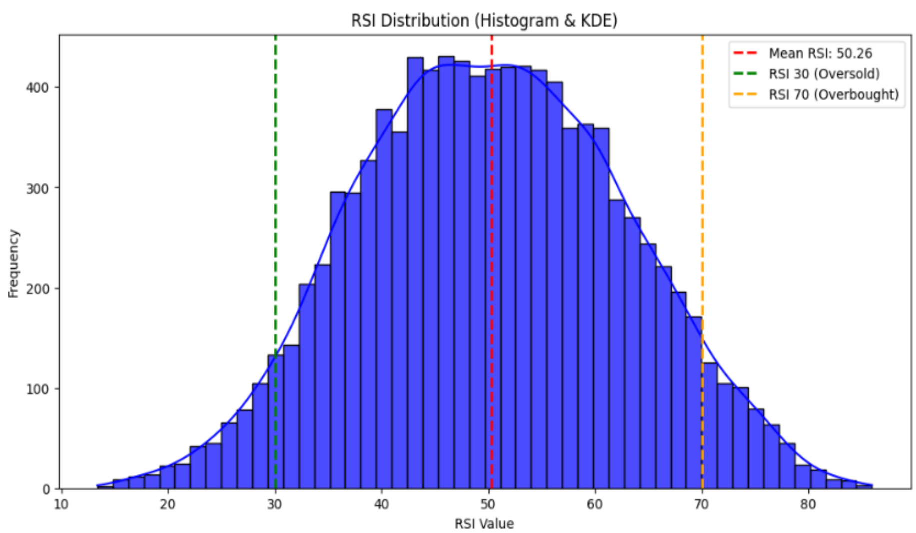

Fig 4. Distribution analysis of the RSI values in the dataset, showing a symmetrical price distribution pattern.

This symmetry supports the assumption that RSI can be used effectively in binary classification tasks related to price direction.

3.2.2 Feature Selection

To identify the most informative features for predicting the next-day price movement, correlation analysis and feature importance scoring were performed:

1.) Pearson and Spearman correlation coefficients were calculated between each feature and the target variable (binary price direction).

2.) Random Forest feature importance was used to rank features based on their predictive contribution in a non-linear ensemble model.

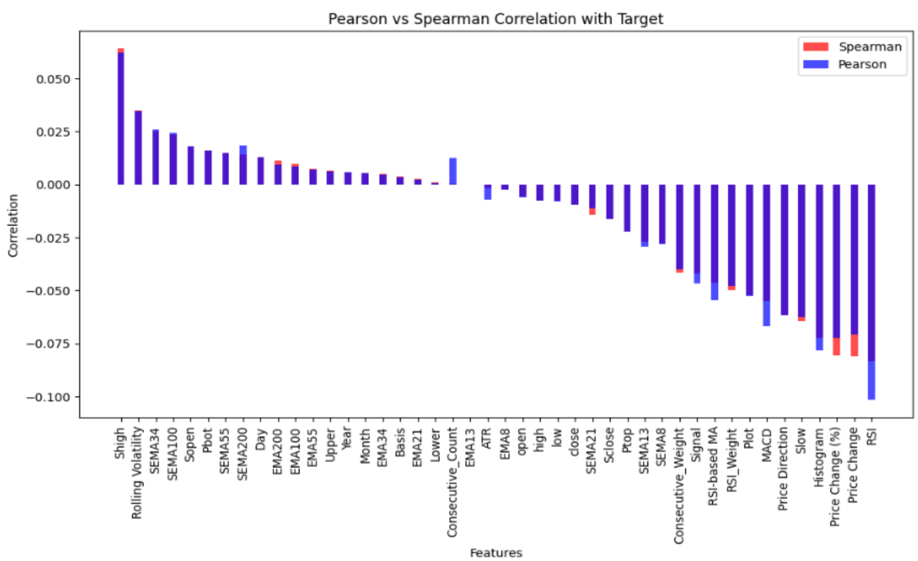

Fig 5. Pearson vs. Spearman correlation with the target variable.

Figure 5 illustrates the correlation between each feature and the target variable using both Pearson (linear) and Spearman (rank-based) methods. Features such as RSI, Price Change, and Histogram show the strongest negative correlations with the target, indicating their high potential in distinguishing upward vs. downward price movements. Meanwhile, Rolling Volatility and Signal show the strongest positive correlations, though overall correlation values remain modest due to market complexity.

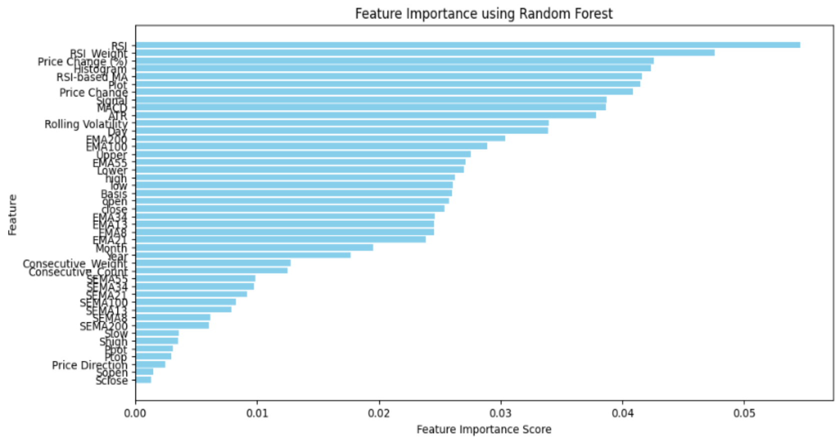

Fig 6. Feature importance ranking based on Random Forest.

Figure 6 shows the feature importance of scores computed using a Random Forest classifier. The most influential features for predicting the next-day price direction include RSI, Price Change (%) and RSI-based MA, indicating their strong contribution to the model's decision-making process. These features play a key role in capturing market momentum, volatility, and trend behavior.

3.2.3 Final Feature Set and Preprocessing

After selection and validation, a refined dataset of 33 columns was constructed (see Figure 5), including both raw prices, technical indicators, engineered features, and the binary Target variable indicating upward or downward price movement.

```
RangeIndex: 10013 entries, 0 to 10012
Data columns (total 33 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   time                10013 non-null  object
 1   open                10013 non-null  float64
 2   high                10013 non-null  float64
 3   low                 10013 non-null  float64
 4   close               10013 non-null  float64
 5   Basis               10013 non-null  float64
 6   Upper               10013 non-null  float64
 7   Lower               10013 non-null  float64
 8   EMA8                10013 non-null  float64
 9   EMA13               10013 non-null  float64
 10  EMA21               10013 non-null  float64
 11  EMA34               10013 non-null  float64
 12  EMA55               10013 non-null  float64
 13  EMA100              10013 non-null  float64
 14  EMA200              10013 non-null  float64
 15  ATR                 10013 non-null  float64
 16  RSI                 10013 non-null  float64
 17  RSI-based MA        10013 non-null  float64
 18  Plot                10013 non-null  float64
 19  Rolling Volatility  10013 non-null  float64
 20  Histogram           10013 non-null  float64
 21  MACD                10013 non-null  float64
 22  Signal              10013 non-null  float64
 23  Day                 10013 non-null  int64
 24  Month               10013 non-null  int64
 25  Year                10013 non-null  int64
 26  Price Change        10013 non-null  float64
 27  Price Change (%)    10013 non-null  float64
 28  Consecutive_Count   10013 non-null  int64
 29  SEMA8               10013 non-null  int64
 30  Consecutive_Weight  10013 non-null  float64
 31  RSI_Weight          10013 non-null  float64
 32  Target              10013 non-null  int64
ltypes: float64(26), int64(6), object(1)
nemory usage: 2.5+ MB
```

Fig 5. Final dataset structure showing all selected features.

To ensure effective model convergence, feature scaling was applied using standardization (z-score normalization) for all continuous numerical features. This normalization step helped align feature ranges and improve training stability across all model types. Additionally, the dataset was split into training and testing sets using an 80:20 ratio, ensuring chronological integrity to preserve the time-series nature of the data. The training set was used for model fitting and cross-validation, while the test set was reserved for final evaluation.

3.3 Model Design Overview

This study aims to develop an ensemble classification model to predict the next-day direction of the EUR/USD price (up/down) in the context of Binary Options trading. The model is designed to leverage the strengths of both traditional machine learning classifiers and deep learning architecture.

The overall structure consists of two main stages

3.3.1 Base Models: Independently trained classifiers that produce prediction probabilities

3.3.2 Meta-Model: A higher-level model that combines the base predictions into final decisions

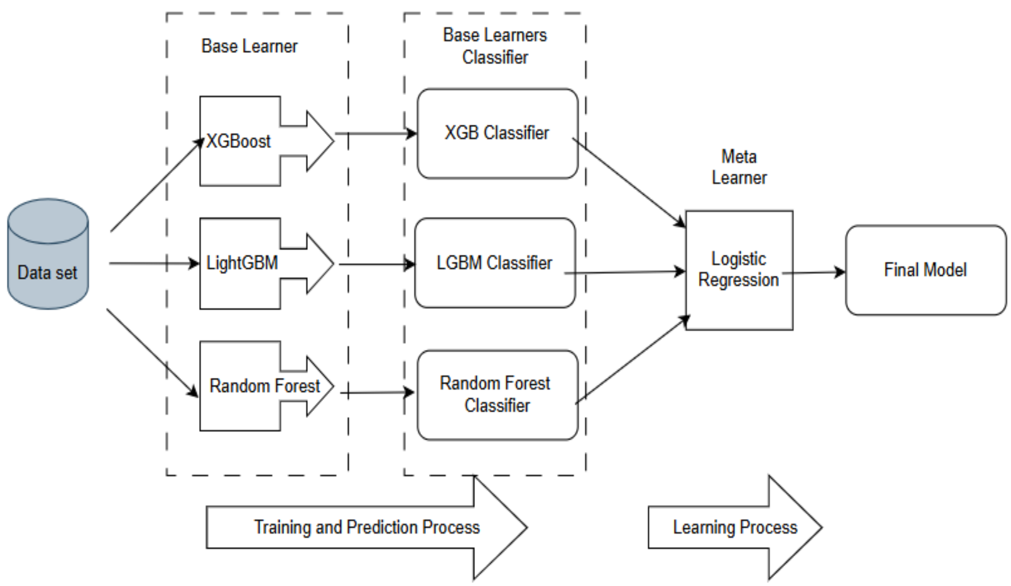

Fig 6. Initial Stacking Architecture without LSTM

Figure 6 illustrates the initial stacking ensemble architecture without LSTM. The dataset is processed through three base learners XGBoost, LightGBM, and Random Forest each producing predictions that are passed to a Logistic Regression meta-learner. The meta-learner combines the outputs to generate the final model decision. This structure emphasizes parallel model training and predictive aggregation.

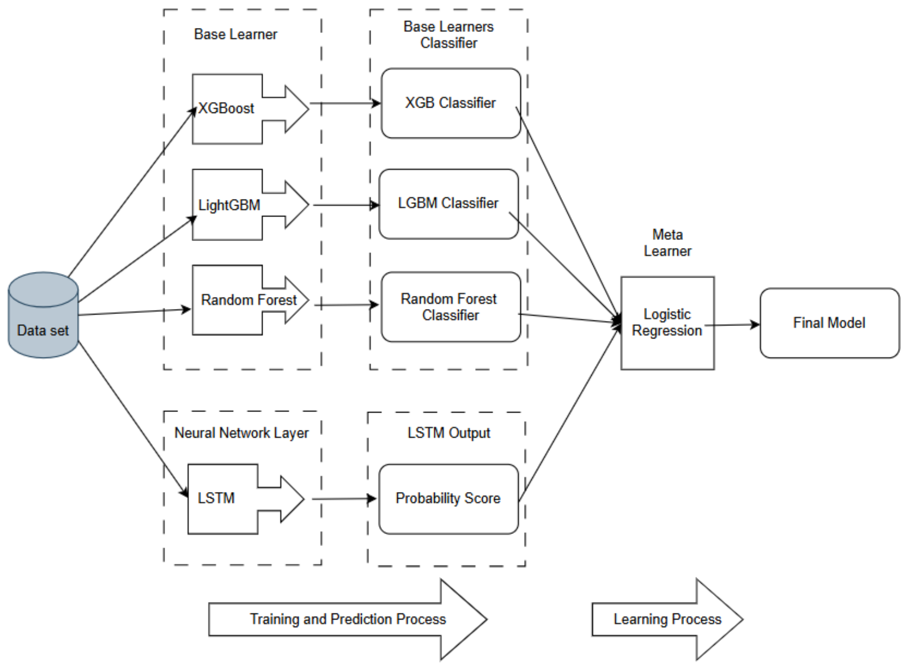

Fig 7. Final Stacking Architecture with LSTM added as base learner

Figure 7 shows the final stacking ensemble architecture with LSTM integrated as an additional base learner. Alongside XGBoost, LightGBM, and Random Forest, the LSTM model processes sequential data and contributes probability scores to the meta-learner (Logistic Regression). This addition enhances the ensemble's ability to capture temporal patterns in financial time series, improving overall predictive performance.

3.4 Base Models

The following models were selected as base learners due to their complementary strengths:

1) XGBoost: gradient boosting framework, well-suited for tabular data
2) LightGBM: fast and efficient boosting model for large datasets

3) Random Forest: ensemble of decision trees, stable and interpretable
4) Support Vector Machine (SVM): margin-based classification with high precision
5) Long Short-Term Memory (LSTM): deep learning model for sequential data, added to capture temporal dynamics

3.5 Meta-Model (Stacking)

A Logistic Regression classifier was used as the meta-learner to integrate predictions from all base models. Base model outputs (class probabilities) were used as inputs for training the meta-model. This stacking strategy improves generalization and reduces bias from individual models.

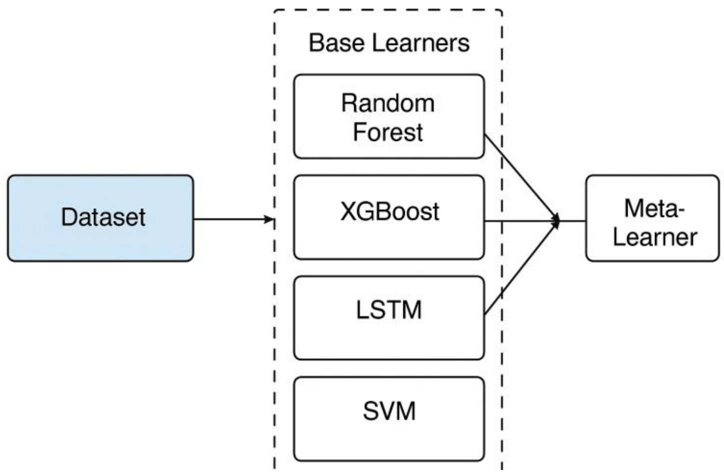

Fig 8. Final Ensemble Flow – Dataset → Base Learners → Meta-Learner → Final Prediction

Figure 8 illustrates the final ensemble model architecture implemented in this research. The workflow begins with a structured dataset, which is preprocessed and then passed to a set of diverse base learners: Random Forest, XGBoost, LSTM, and SVM. Each model independently analyzes the data and produces predictions, leveraging different strengths such as tree-based decision making, gradient boosting, sequence learning, and margin classification.These individual predictions are not used directly, but are forwarded to a meta-learner, typically a Logistic Regression model. The meta-learner is trained to optimally combine the base model outputs into a final prediction. This stacking ensemble design helps reduce model bias, improve generalization, and boost overall accuracy. It also enhances the model's adaptability to both linear and nonlinear patterns in financial time-series data.

3.6 Model Selection and Justification

To support model selection, performance was evaluated at different confidence intervals. Only predictions with confidence ≥ 70% were considered for the final ensemble. Accuracy, prediction volume, and reliability were analyzed separately for Buy and Sell signals.

Table 1. Model Accuracy by Confidence Range (Grouped)

Model Name	Confidence Range	Buy only			Sell only		
		Total buy	Correct	Accuracy (%)	Total sell	Correct	Accuracy (%)
XG Boost	51-60%	150	76	50.66	56	26	46.42
	61-70%	140	79	56.42	47	26	55.31
	71-80%	156	77	49.35	46	26	56.52
	81-90%	140	78	55.71	72	36	50.00
	91-100%	156	112	71.79	99	68	68.68
LightGBM	51-60%	343	177	51.60	87	34	39.08
	61-70%	199	99	49.74	83	44	53.01
	71-80%	87	44	50.74	66	31	46.96
	81-90%	25	14	56.00	38	21	55.26
	91-100%	70	68	97.14	37	29	78.37
Random Forest	51-60%	428	211	49.29	161	80	49.68
	61-70%	210	112	53.33	107	54	50.46
	71-80%	53	25	47.16	34	16	47.05
	81-90%	16	10	62.50	4	4	100.00
	91-100%	59	59	100.00	18	18	100.00
SVM	51-60%	999	522	52.25	341	160	46.92
	61-70%	49	42	85.71	64	25	39.06
	71-80%	2	2	100.00	0	0	NaN
	81-90%	0	0	NaN	0	0	NaN
	91-100%	0	0	NaN	214	107	50.00
Logistic Regression	51-60%	659	352	53.41	96	45	46.87
	61-70%	142	85	59.85	16	13	81.25
	71-80%	31	28	90.32	6	5	83.33
	81-90%	1	1	100.00	0	0	NaN
	91-100%	0	0	NaN	0	0	NaN

Table 1 summarizes model accuracy across different confidence ranges (from 51% to 90%), separated into Buy-only and Sell-only predictions. The results show that accuracy generally improves with higher confidence levels, supporting the use of confidence as a filter in trading decisions. Models like XGBoost, LightGBM, and Random Forest demonstrate steady performance increases as confidence increases.

Table 2. Model Decision Comparison at ≥70% Confidence

Model Name	Buy only			Sell only		
	Total buy	Correct	Accuracy (%)	Total sell	Correct	Accuracy (%)
XGBoost	452	267	59.07	217	130	59.90
LightGBM	182	126	69.23	141	81	57.44
Random Forest	138	99	71.73	60	41	68.33
SVM	2	2	100.00	2	2	100.00
Logistic Regression	32	29	90.62	22	18	81.81

Table 2 compares the accuracy of different models in making Buy and Sell decisions when filtered by confidence levels of 70% or higher. The results highlight that Random Forest achieved the most balanced and accurate performance, while Logistic Regression and SVM showed high precision in smaller subsets. This evaluation guided the selection of base and meta models for the final ensemble.

3.7 Training Process

The model training followed a two-stage procedure

1) Base Model Training: 80:20 chronological split, with z-score normalization applied to numerical features. Cross-validation was used where applicable.
2) Meta-Model Training: Trained in out-of-fold predictions to avoid data leakage.

3.8 Model Evaluation Metrics

To ensure the robustness and practical effectiveness of the model, the evaluation was conducted from multiple perspectives, combining standard classification metrics, confidence-level analysis, and financial performance indicators.

3.8.1 Classification Performance

The model was assessed using

1) Accuracy – Overall correctness of predictions

2) Precision – Accuracy of Buy signals (low FP = higher precision)
3) Recall – Ability to capture actual Buy instances (low FN = higher recall)
4) F1-score – Harmonic mean of precision and recall

These metrics were calculated using a confusion matrix (TP, TN, FP, FN) to measure classification quality.

3.8.2 Confidence-Based Signal Analysis

Predictions were grouped by confidence thresholds (e.g., 0.7, 0.8, 0.9) to

1) Analyze how accuracy varies across different confidence levels
2) Support filtering decisions for high-confidence trading actions

3.8.3 Financial Performance Evaluation

A back testing simulation was conducted using real market scenarios to evaluate economic viability. Key financial metrics included:

1) Profit Factor – Ratio of average profit to average loss
2) Maximum Drawdown – Largest portfolio declines from peak to trough
3) Sharpe Ratio – Return per unit of risk above the risk-free rate

3.8.4 Buy/Sell Model vs. Binary Classification

A comparative analysis was performed between

1) separate Buy/Sell prediction strategy, and
2) A binary classification model predicting direction to determine which yields higher accuracy and stability.

3.9 Expected Return Calculation

To connect prediction quality with decision-making, the expected return was calculated using probability theory

$$E(R) = \sum P_i \cdot R_i$$

In binary trading (fixed reward, full loss), this becomes

$$E(R) = p \cdot R_{win} + (1 - p) \cdot (-1)$$

Where:

p = model's win probability

R_{win} = reward per win (e.g., 0.8), loss = -1

This equation helps determine if the model is statistically profitable

3.10 Remaining Capital Simulation

The final capital after multiple trades was estimated using:

$$\text{Remaining Capital} = C \cdot (1 + r \cdot E(R))^n$$

Where:

C = Initial capital

r = % of capital invested per trade

$E(R)$ = Expected return

n = Number of trades

This simulates the long-term impact of the model's decisions under controlled money management.

4 Results

4.1 Model Performance Evaluation

The performance of the ensemble model was evaluated in both Buy and Sell signal scenarios using classification metrics including Accuracy, Precision, Recall, and F1-score

Table 3. Comparison of baseline models and LSTM-enhanced models on Buy signals

	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score
1	M2 (Stacking + LSTM)	62.858284	68.883249	52.122143	59.341861
0	M1 (Stacking Only)	62.708479	64.095694	64.317265	64.206288

Table 4. Comparison of baseline models and LSTM-enhanced models on Sell signals

	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score
0	M1 (Stacking Only)	63.477479	66.493883	61.108994	63.687817
1	M2 (Stacking + LSTM)	59.342854	63.689230	52.172256	57.358332

From table 3 and table 4 show that M1 outperformed M2 in **Sell signals**, while M2 achieved slightly better **precision in Buy signals**.

4.2 Confidence-Level Analysis and Equity Performance

The model's predictions were segmented by confidence thresholds (≥0.6 to ≥0.9) to analyze how accuracy and profitability change with prediction certainty.

Table 5. Profitability comparison across different confidence levels

	Total Buy Signals	Correct Buy Signals	Buy Accuracy (%)	Total P&L (USD)	P&L (%)	Max Drawdown (USD)	Max Drawdown (%)	Total Trades	Profitable Trades (%)	Profit Factor
0.6	838	532	63.48	26352501.99	2635.25	-8424.05	-0.84	838	63.48	1.38
0.7	518	279	53.86	160545.45	16.05	14889.18	1.49	518	53.86	1.026
0.8	480	247	51.46	-259752.75	-25.98	311063.86	31.11	480	51.46	0.942
0.9	413	218	52.78	-49175.5	-4.92	123407.11	12.34	413	52.78	0.988

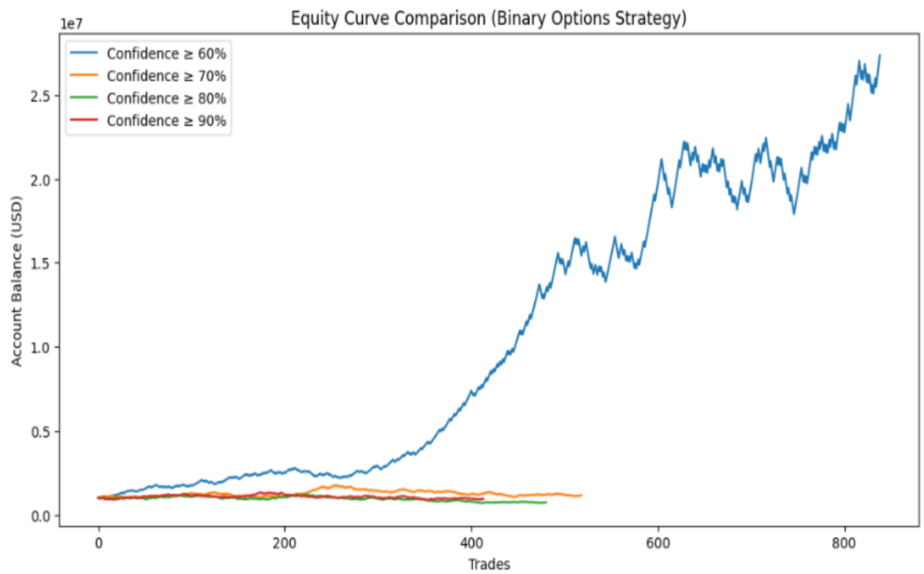

Fig 9. Equity curve comparison across confidence thresholds in Binary Options trading strategy

From Table 5 and Figure 9 show the higher confidence generally leads to more accurate and stable trade. The best trade-off between profit and risk was achieved at Confidence ≥ 60%, with a Profit Factor of 1.633 and the highest total return. At very high confidence levels (≥ 90%), the number of trades was lower, leading to reduced profit despite high reliability

4.3 Trading Strategy Comparison

The study compared different ways of combining Buy and Sell models to find the most profitable strategy.

Table 6. Performance comparison of model combinations for classifying Buy and Sell

Model Usage Strategy	Accuracy (%)	Total P&L (USD)	Maximum Drawdown (USD)	Profit Factor
Unified model for both Buy and Sell	50.66	787,846	-18,000	1.264
Buy M1 + Sell M1	64.48	236,580,925	-18,000	1.510
Buy M2 + Sell M2	66.27	22,110,710	-18,000	1.522
Buy M1 + Sell M2	59.68	22,312,654	-8,424	1.238
Buy M2 + Sell M1	67.55	3,320,315,189	-18,000	1.633

Table 7. Performance metrics at different confidence thresholds

	Total Signals	Correct Signals	Accuracy (%)	Total P&L (USD)	P&L (%)	Annualized Return (%)	Max Drawdown (USD)	Max Drawdown (%)	Total Trades	Profitable Trades (%)	Profit Factor
0.6	1895	1280	67.55	33203151889.200001	3320315.19	46.15	-18000.0	-1.8	1895	67.55	1.633
0.7	1135	621	54.71	1002502.4	100.25	2.56	-18000.0	-1.8	1135	54.71	1.049
0.8	1059	553	52.22	-300294.1	-30.03	-1.29	435407.02	43.54	1059	52.22	0.971
0.9	778	414	53.21	32456.22	3.25	0.12	94687.55	9.47	778	53.21	1.004

This table summarizes key trading performance indicators—including accuracy, total profit/loss, annualized return, drawdown, and profit factor—based on varying model confidence thresholds (≥ 60% to ≥ 90%).

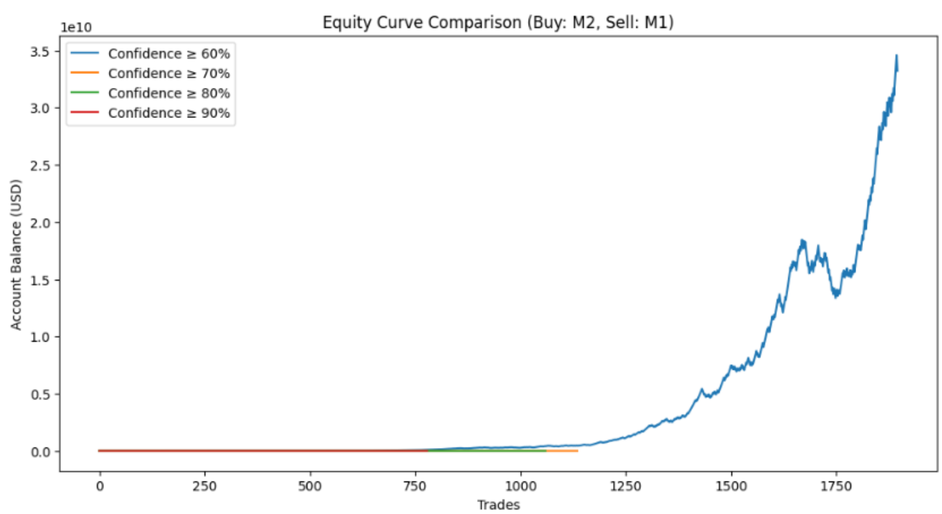

Fig 10. Equity curve comparison by confidence threshold (Buy: M2, Sell: M1)

This graph visualizes the cumulative account balance over time for different minimum confidence levels, demonstrating that moderate thresholds (≥ 60%) yield the strongest and most consistent growth.

4.4 Yearly Performance Stability

Annual returns were analyzed to evaluate the model's performance across different market conditions from 1986–2025. Figure 11 displays the year-by-year return generated by the trading strategy. Most years yielded positive returns (green bars), while only a few years experienced losses (red bars). The strategy shows consistent profitability across decades, with peak performance during the early 2000s and stability in recent years despite market fluctuations.

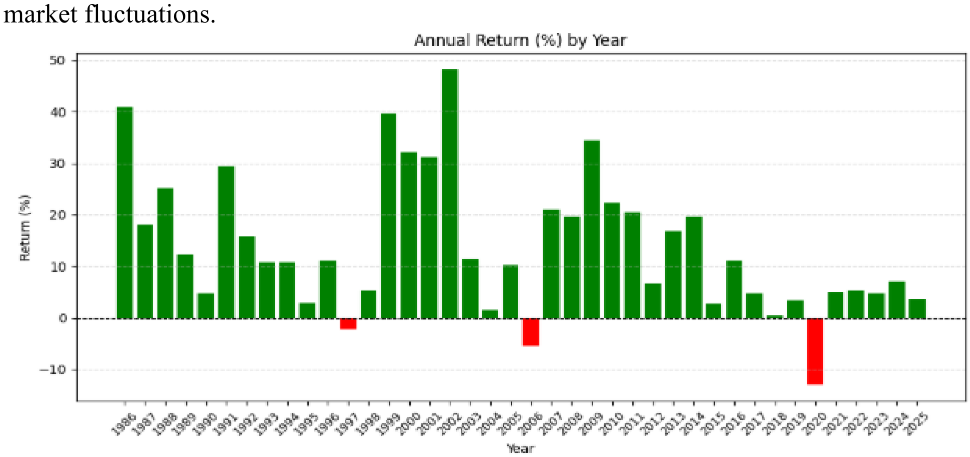

Fig 11. Annual return (%) by year from 1986 to 2025

The experimental results demonstrate that applying a confidence threshold to model predictions significantly improves trading performance. By filtering out low-confidence signals, the strategy can focus on high-probability trades, leading to increased accuracy, profitability, and reduced drawdown. Integrating LSTM into the ensemble architecture enhanced the precision of Buy signals, as LSTM effectively captures temporal patterns in financial time series. However, its contribution to Sell signal accuracy was more limited, indicating that LSTM may be more effective for detecting upward trends than downward ones. Among various model combinations tested, the strategy that combined Buy (M2) and Sell (M1) yielded the best overall performance. This configuration achieved the highest accuracy, profit factor, and total return, making it the most balanced and robust choice for practical deployment.

Finally, long-term backtesting from 1986 to 2025 confirmed that the model is stable across market cycles and performs consistently in both volatile and stable conditions. These findings suggest that the proposed ensemble model is not only technically sound but also adaptable and financially viable for real-world Binary Options trading.

5. Conclusion and Recommendations

5.1 Summary of the Research

This research aimed to develop an ensemble classification model to predict the next-day direction of EUR/USD price movements for Binary Options trading. The final architecture integrates multiple base models XGBoost, LightGBM, Random Forest, and LSTM with a Logistic Regression meta-learner using a stacking approach. Additionally, the strategy separates Buy and Sell signal prediction into independent pipelines to enhance decision precision. The model was evaluated using both statistical classification metrics and financial performance indicators. Results showed that filtering confidence by prediction significantly improved profitability and stability. The best-performing strategy combined Buy (M2: Stacking + LSTM) and Sell (M1: Stacking only), yielding the highest accuracy and return with manageable risk.

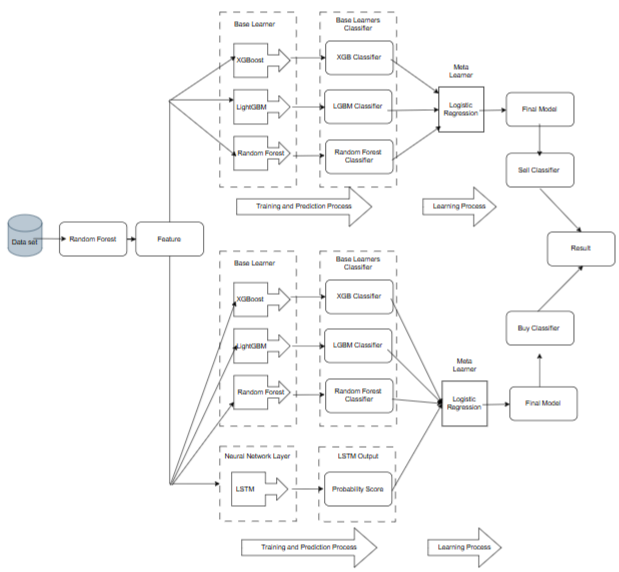

Fig 8. Final model architecture with feature engineering and separate Buy/Sell

This diagram illustrates the full workflow from dataset input, feature generation via Random Forest, parallel base learner training (with and without LSTM), stacking with a meta-learner, and final classification into Buy/Sell decisions.

5.2 Discussion of Findings

The research supports several key conclusions:

1) Stacking ensembles outperform individual models, especially when combined with LSTM for Buy signals.

2) Confidence thresholds serve as an effective filtering mechanism, directly correlating with improved accuracy and profit factor.

3) Separate Buy/Sell classification outperforms binary classification, as it allows each model to specialize in detecting upward or downward movements.

4) The model remained robust across historical data (1986–2025), indicating adaptability to long-term market cycles.

However, the contribution of LSTM to Sell signals was limited, suggesting that downward price patterns may be less temporally dependent or harder to capture via sequential modeling.

5.3 Research Limitations

Despite promising results, this study has several limitations:

1) The model assumes **fixed return rates** in Binary Options trading, which may differ in real platforms.

2) The LSTM model was implemented using basic architecture. More advanced options (e.g., Bi-LSTM, attention mechanisms) were not explored.

3) The study did not incorporate **external or fundamental factors**, such as news or macroeconomic data, which can influence short-term price movements.

4) Computational complexity increases significantly with stacking and deep learning, which may hinder deployment in latency-sensitive environments.

5.4 Recommendations for Future Research

To expand on the current findings, future research could explore:

1) Integrating **news sentiment analysis** or economic event tagging to enrich input features.

2) Testing **real-world deployment** with live data on trading platforms for risk control validation.

3) Comparing with **reinforcement learning** or agent-based systems for adaptive decision-making.

4) Optimizing the architecture of LSTM or incorporating **attention-based models (e.g., Transformer)** to better capture long-term dependencies.
5) Extending the framework to **multi-asset portfolios** or **multi-class prediction**, such as predicting price range or volatility zone.

The author would like to acknowledge the use of **ChatGPT (developed by OpenAI)** as a writing assistant during the drafting of this report. ChatGPT was used to help structure, summarize, and improve the clarity of the text. All interpretations and conclusions remain the sole responsibility of the author.

References

1. Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2021). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, 116, 237–245. https://doi.org/10.1016/j.neunet.2019.04.014
2. Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32, 17351–17360. https://doi.org/10.1007/s00521-020-04867-x
3. Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1643–1647). IEEE. https://doi.org/10.1109/ICACCI.2017.8126078
4. Shah, D., Isah, H., & Zulkernine, F. (2021). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 9(2), 41. https://doi.org/10.3390/ijfs9020041
5. Uzun, T., Akdoğan, H. A., & Yıldız, B. (2024). Rule-based candlestick pattern recognition for cryptocurrency trading. *Expert Systems with Applications*, 239, 120203. https://doi.org/10.1016/j.eswa.2023.120203
6. OpenAI. (2023). *ChatGPT* (Mar 14 version) [Large language model]. https://chat.openai.com/