

**AN INTELLIGENT MACHINE LEARNING-BASED TRADING SIGNAL PREDICTION
SYSTEM FOR THE IQ OPTION OTC MARKET: EVIDENCE FROM INVESTMENT HOUSE,
ADAMAWA STATE, NIGERIA**

BULUS, Lucy Dalhatu, AARON, Catherine Madunagu & SAMUEL, Augustine
Department of Computer Science, Federal Polytechnic, PMB35, Mubi Adamawa State,
Nigeria

buluslucy08@gmail.com, +2348137764425*
kateenyeomadunagu@gmail.com, +2348063327793
samtine34@gmail.com, +2348060698209

Abstract

Short-term financial markets, particularly Over-the-Counter (OTC) synthetic digital options, are characterized by high volatility, non-linear noise, and continuous operation. Traditional manual trading strategies in these environments suffer from human psychological vulnerabilities, decision latency, inconsistency technical analysis, and physical fatigue. This paper presents a design, implementation, and evaluation of An Intelligent Machine Learning-Based Trading Signal Prediction System for The IQ Option OTC currency pairs, using Investment House in Adamawa State, Nigeria- A localized operational case study. The aim of the study was to establish an automated, data-driven decision-support mechanism that replaces subjective manual chart interpretation. The methodology encompasses historical candle data retrieval from the IQ Option API, robust data cleaning, chronological data partitioning to prevent temporal leakage, and engineering an eighteen-variable technical feature space. A Random Forest classifier configured with 300 decision trees was trained to predict price movement directions classified into BUY, HOLD, and SELL targets. To minimize false positives, a strict confidence-based filtering module was integrated, requiring a minimum 70% class probability before a prediction is qualified as an actionable trading signal. Qualified signals are automatically logged into CSV storage and disseminated in real time via the Telegram Bot API. System evaluation confirmed successful real-time data acquisition, automated 18-feature extraction, consistent model execution, and low-latency notification delivery without executing automated orders, preserving full user execution control. The findings demonstrate that structured supervised group learning coupled with confidence thresholding provides a reliable framework for short-term financial decision support in localized FinTech contexts. Periodic model retraining and demo-environment validation are recommended prior to operational deployment.

Keywords: Intelligent, Machine, Learning-Based, Trading, Signal, Prediction, Market

INTRODUCTION

The global financial system has undergone a dramatic structural transformation over the past decade, migrating rapidly from traditional floor-based manual trading to sophisticated electronic, automated, and algorithmic trading architectures. Driven by the demand for sub-millisecond execution speeds, quantitative modeling, and the elimination of human latency, the global algorithmic trading market expanded significantly, with projections exceeding \$32.77 billion by 2026. Central to this evolution is the integration of Artificial Intelligence (AI) and Machine Learning (ML). Unlike legacy rule-based expert systems that depend on rigid, hard-coded technical thresholds, contemporary ML architectures possess the capacity to learn non-linear patterns directly from empirical financial time-series data, dynamically adapting to evolving market regimes (CBN, 2023).

A specialized, highly volatile segment of digital financial assets is the Over-the-Counter (OTC) market, specifically digital and binary options offered by broker platforms such as IQ Option. Unlike centralized exchange-traded equities or spot foreign exchange, OTC digital options operate continuously (24/7), utilizing broker-mediated synthetic price feeds that reflect institutional price behavior even during standard market closures. In high-speed, ultra-short-term options environments such as 1-minute and 5-minute expiration contracts the window for market analysis and trade execution is extremely narrow. Operating manually within these ultra-short timeframes imposes severe cognitive stress on human traders. Manual decision-making is inherently susceptible to cognitive fatigue, emotional biases such as fear and greed, delayed order entry (latency), and erratic, inconsistent chart interpretations (Addy, *et al.*, 2024).

In emerging economies, particularly within the rapidly expanding Nigerian Financial Technology (FinTech) ecosystem, local investment firms and retail trading groups face substantial barriers to accessing modern quantitative tools. Although Nigeria represents one of the primary FinTech innovation hubs in sub-Saharan Africa, local investment organizations, such as Investment House in Adamawa State, frequently rely on manual chart analysis and subjective intuition. This technological gap leads to poor risk management, suboptimal trade timing, and heightened capital loss in volatile OTC markets (Addy, *et al.*, 2024).

While extensive academic literature has explored machine learning applications in traditional stock exchanges, commodity futures, and high-cap cryptocurrencies, significant research gaps remain regarding localized, lightweight intelligent signal generation systems tailored specifically for broker-mediated OTC synthetic option markets. Most existing trading systems in literature focus either purely on off-line model accuracy without live dissemination architectures, or on fully autonomous execution "bots" that introduce severe regulatory, security, and capital control risks (Research & Markets, 2026).

To address these limitations, this study presents the design, technical implementation, and operational evaluation of an Intelligent Machine Learning-Based Trading Signal Prediction System for the IQ Option OTC market. Developed using Python 3.11.9, a Random Forest ensemble model, an 18-variable technical indicator feature pipeline, a strict 70% confidence probability filter, and real-time Telegram signal dissemination, the system serves as a localized decision-support framework for Investment House, Adamawa State, Nigeria. By deliberately keeping automated trade execution disabled, the system eliminates broker API order risks while providing structured, emotionless quantitative intelligence directly to traders in real time.

Conceptual Framework

Machine Learning (ML), a core branch of Artificial Intelligence, focuses on developing algorithms that implicitly extract predictive statistical patterns from historical data without explicit programmatic rules. In financial time-series forecasting, supervised learning algorithms are trained on historic asset prices and derived technical indicators to solve regression (price prediction) or classification (direction prediction) problems. Given the inherent noise, non-stationarity, and high frequency of short-term financial markets, ensemble supervised learning models like Random Forest are exceptionally effective due to their ability to model complex non-linear interaction terms without assumptions of underlying data normality (Jabed, 2024).

- i. **Algorithmic Trading and OTC Digital Options:** Algorithmic trading entails using computational algorithms to execute trading strategies based on predefined mathematical, statistical, or machine-learned rules. Over-the-Counter (OTC) digital options represent

derivative contracts where payouts depend on binary price outcomes (whether the asset price closes above or below an entry strike price at a set expiration time). Because OTC price feeds are synthesized by market makers continuously, they exhibit high tick density and localized volatility, making them highly receptive to automated quantitative feature engineering (Sonkavde, *et al.*, 2023).

- ii. **Automated Signal Delivery Architectures:** Predictive power alone is insufficient in short-term trading if signal transmission introduces latency. Automated signal dissemination bridges predictive models and practical execution. Modern architectures utilize Application Programming Interface (API) webhooks, such as the Telegram Bot API, to push structured alerts (asset pair, directional bias, confidence level, recommended entry/expiry times) directly to end-user mobile and desktop devices within milliseconds of signal qualification (Sonkavde, *et al.*, 2023).
- iii. **Efficient Market Hypothesis (EMH) vs. Behavioral Finance:** The theoretical foundation of market predictability has historically rested on the Efficient Market Hypothesis (EMH) formulated by Fama (1970). EMH asserts that asset prices fully reflect all available market information, rendering consistent risk-adjusted excess returns impossible through historical technical analysis (Weak Form EMH). However, empirical financial research regularly reveals market anomalies, structural micro-inefficiencies, and non-random price patterns driven by human psychological biases. Behavioral Finance demonstrates that market participants are subject to cognitive distortions such as loss aversion, overconfidence, and herd mentality which introduce predictable pricing feedback loops (Wang, *et al.*, 2025).
- iv. **Adaptive Market Hypothesis (AMH):** To reconcile EMH with empirical market realities, Lo (2004) proposed the Adaptive Market Hypothesis (AMH). AMH posits that market efficiency is not an absolute, static state but an evolving biological ecosystem where dynamic efficiency fluctuates based on market participant composition, technological capabilities, and environmental stress. In highly specific, short-term micro-environments such as synthetic OTC asset markets pockets of predictable patterns periodically emerge as market dynamics shift. Machine learning models function as adaptive agents under the AMH framework, continuously learning emerging technical interactions and exploiting temporary micro-inefficiencies before market dynamics adjust.
Recent empirical literature underscores the progressive shift from classical statistical time-series models toward machine learning and deep learning classification pipelines.
 - i. Javed, (2024) conducted a comparative evaluation of supervised algorithms for short-term financial direction forecasting, finding that ensemble decision trees (Random Forest) and recurrent neural structures substantially outperformed traditional linear regression models in directional accuracy across volatile asset classes.
 - ii. Rane *et al.* (2023) investigated deep learning architectures, specifically Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU), for sequential time-series prediction. Their findings confirmed that sequential memory mechanisms capture long-range temporal dependencies effectively; however, LSTMs exhibited high computational overhead, hyperparameter sensitivity, and vulnerability to overfitting on noisy, small-sample financial datasets.
 - iii. Avelar and Jordão (2024) examined AI integration within operational trading environments, highlighting that complex models frequently fail during live deployment due to data leakage during training, unhandled API connectivity drops, and excessive execution latency. They advocated for lightweight, highly robust ensemble classifiers paired with strict confidence filtering thresholds.

- iv. Oyewole *et al.* (2024) analyzed algorithmic trading applications within emerging African FinTech markets, identifying critical infrastructural bottlenecks, including unstable network latency, limited local cloud infrastructure, and a widespread reliance on subjective manual trading in investment firms. Their study emphasized the practical necessity of decision-support signal dissemination systems over fully automated order-execution systems to mitigate systemic operational and API risks.
While existing literature extensively covers model training on historical stock or cryptocurrency datasets, three distinct gaps persist:
 - i. **Market Context Gap:** Minimal academic literature evaluates supervised machine learning models specifically on synthetic OTC digital option currency pairs (e.g., EURUSD-OTC, GBPUSD-OTC) operating under short-term (1-minute and 5-minute) expiration timeframes.
 - ii. **Architectural Integration Gap:** The majority of published research focuses strictly on offline accuracy metrics without presenting an end-to-end, functional system architecture that integrates live data retrieval, continuous feature engineering, probability-based signal filtering, CSV logging, and real-time messaging API alerts.
 - iii. **Localized Operational Context:** There is a lack of accessible, lightweight decision-support FinTech tools designed specifically for local investment institutions in developing regions, such as Adamawa State, Nigeria, bridging the gap between advanced machine learning research and localized operational practice.

METHODOLOGY

The system was developed using the Agile-Scrum methodology to accommodate the iterative nature of machine learning feature engineering, model tuning, and API integration. The development lifecycle comprised five distinct sprints: Sprint 1 (Requirements Analysis & Environment Setup), Sprint 2 (Data Engineering & Indicator Extraction Pipeline), Sprint 3 (Model Training & Chronological Validation), Sprint 4 (Signal Generation & Telegram API Integration), and Sprint 5 (System Diagnostic & Operational Evaluation).

a. Data Collection and Sources: Historical market data were acquired directly from the broker API (iqoptionapi). The target dataset comprised historical Open, High, Low, Close (OHLC) candle records across four primary OTC currency pairs:

- i. EURUSD-OTC
- ii. GBPUSD-OTC
- iii. USDCHF-OTC
- iv. USDJPY-OTC

Data collection was executed separately for two target operational timeframes: 1-minute (1M) and 5-minute (5M) candle intervals. The acquired candle series were chronologically structured and stored in raw CSV format.

b. Data Preprocessing and Feature Engineering: Financial time-series data contain high levels of noise, unclosed live candles, and missing values. The preprocessing pipeline executes automatic cleaning; removing incomplete trailing candles to prevent forward-looking bias.

To transform raw OHLC prices into an informative feature space, the system engineered 18 distinct technical indicator and price-action feature variables:

Table 1: Engineered Machine Learning Feature Set (18 Variables)

Variable Index	Feature Name	Technical Description / Derivation
F_1	sma_5	5-period Simple Moving Average
F_2	sma_20	20-period Simple Moving Average
F_3	ema_12	12-period Exponential Moving Average
F_4	ema_26	26-period Exponential Moving Average
F_5	rsi_14	14-period Relative Strength Index
F_6	macd	Moving Average Convergence Divergence Line ($EMA_{12} - EMA_{26}$)
F_7	macd_signal	9-period Exponential Moving Average of MACD Line
F_8	macd_hist	MACD Histogram Difference ($MACD - Signal$)
F_9	bollinger_upper	Upper Bollinger Band ($SMA_{20} + 2 \times \sigma$)
F_{10}	bollinger_lower	Lower Bollinger Band ($SMA_{20} - 2 \times \sigma$)
F_{11}	bollinger_width	Bandwidth ratio: $(Upper - Lower) / Middle$
F_{12}	price_change	Instantaneous candle return: $(Close_t - Open_t) / Open_t$
F_{13}	high_low_ratio	Intra-candle volatility spread: $(High_t - Low_t) / Close_t$
F_{14}	body_size	Absolute candle body magnitude: $ Close_t - Open_t $
F_{15}	upper_shadow	Upper wick length: $High_t - \max(Open_t, Close_t)$
F_{16}	lower_shadow	Lower wick length: $\min(Open_t, Close_t) - Low_t$
F_{17}	volatility_10	10-period standard deviation of price returns (σ_{10})
F_{18}	trend_direction	Binary moving average slope indicator ($SMA_5 > SMA_{20}$)

c. Target Label Construction: To frame signal generation as a supervised classification task, discrete target labels were constructed based on future price directional movements over a fixed forward horizon ($t + n$):

- i. **BUY (1):** Defined when $Close_{t+1} > Close_t + \epsilon$ (Upward price movement exceeding threshold ϵ).
- ii. **SELL (-1):** Defined when $Close_{t+1} < Close_t - \epsilon$ (Downward price movement below threshold ϵ).
- iii. **HOLD (0):** Defined when $|Close_{t+1} - Close_t| \leq \epsilon$ (Indecisive sideways price action).

d. Model Architecture: Random Forest Classifier: The predictive core utilizes a Random Forest classifier composed of $N_{estimators} = 300$ individual decision trees. Random Forest constructs an ensemble of uncorrelated decision trees using bootstrap aggregation (bagging) and random feature selection. Given an input vector X containing the 18 engineered features, each tree t predicts a target class $y \in \{BUY, HOLD, SELL\}$. The ensemble outputs class probability distributions via majority voting across all 300 trees:

$$P(y = c|X) = \frac{1}{N} \sum_{i=1}^N I(\hat{y}_i(X) = c)$$

Where $I(\cdot)$ is the indicator function, $N = 300$, and $c \in \{BUY, HOLD, SELL\}$.

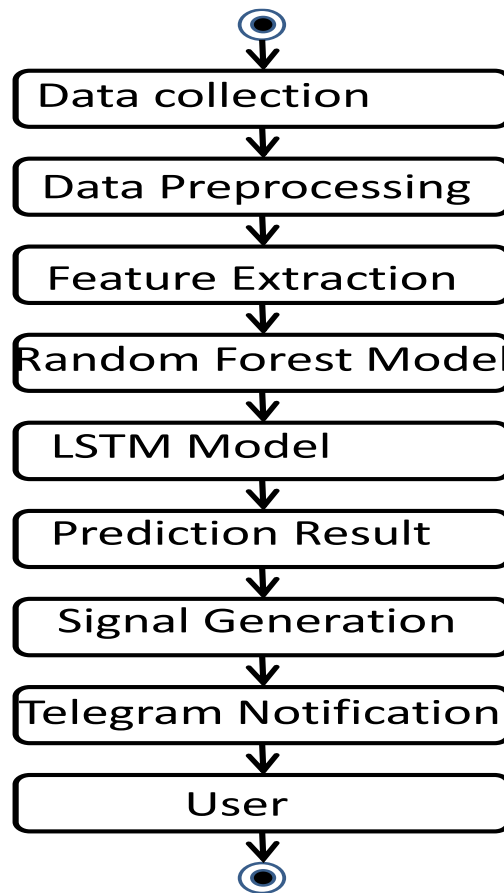


Figure 1: Architectural Workflow Diagram of the Machine Learning Trading Signal Prediction System

- e. **Chronological Data Partitioning and Leakage Prevention:** Standard randomized k -fold cross-validation is fundamentally flawed for financial time-series prediction because random shuffling allows future data points to leak into the training set. To guarantee rigorous, unbiased evaluation, a temporal split was enforced: the historical dataset was divided chronologically into an initial 80% training segment and a subsequent 20% out-of-sample testing segment. Feature scaling parameters and indicator windowing were computed strictly on training data prior to transforming test data.
- f. **Confidence-Based Signal Filtering:** A raw classification decision from a machine learning model does not inherently warrant executing a financial trade. To filter out low-confidence predictions, the system applies a strict thresholding mechanism:

$$\text{Final Signal} = \begin{cases} \text{BUY,} & \text{if } P(y = \text{BUY}|X) \geq 0.70 \\ \text{SELL,} & \text{if } P(y = \text{SELL}|X) \geq 0.70 \\ \text{NO SIGNAL,} & \text{otherwise} \end{cases}$$

Signals predicting HOLD, or directional predictions where class probability falls below 70%, are classified as NO SIGNAL and suppressed from notification channels.

RESULTS AND DISCUSSION

The application was built as a command-line interface (CLI) software package in Python 3.11.9 within Visual Studio Code. Software dependencies were standardized using explicit library versions:

Table 2: Operational Software Stack and Dependencies

Component/Library	Version/Source	Operational Role
Runtime Environment	Python 3.11.9	Core application code execution
scikit-learn	1.5.2	Random Forest implementation & metric evaluation
pandas	2.2.3	OHLC data processing and indicator calculations
numpy	1.26.4	Vectorized mathematical array operations
joblib	1.4.2	Serialization of trained model packages (.pkl)
requests	2.32.4	HTTP POST payload communication with Telegram API
python-dotenv	1.1.1	Secure local environment variable configuration
iqoptionapi	Git Repository	WebSocket API connector for IQ Option market data

Module Architecture: The codebase is structured into modular Python scripts designed to separate concerns cleanly:

- i. `config.py`: Centralized configuration loading credentials from `.env`, setting threshold limits (0.70), defining supported pairs/timeframes, and explicitly enforcing `AUTO_TRADE_ENABLED = False`.
- ii. `market_data.py`: Handles WebSocket authentication, candle retrieval, and historical CSV dataset collection.
- iii. `indicators.py` & `feature_engineering.py`: Computes technical indicators, constructs the 18 feature variables, formats target labels, and enforces feature order consistency.
- iv. `train_model.py`: Executes chronological train-test splits, trains the 300-tree Random Forest classifier, evaluates metrics, and serializes artifacts to `models/trading_model.pkl`.
- v. `signal_generator.py`: Loads the serialized model package, validates live feature compatibility, predicts class probabilities, and enforces the 70% confidence filter.
- vi. `telegram_alerts.py`: Formats qualified signals into structured alert templates, executes HTTP POST requests via Telegram API, and maintains alert cooldowns to suppress duplicate messages.
- vii. `main.py`: Provides the interactive terminal user menu, diagnostic functionality, single-pair analysis, and multi-pair continuous market monitoring loops.

- a. **Operational Functional Requirements:** System testing verified that all primary functional requirements established during system design were successfully implemented:

Table 3: System Functional Requirements Verification

Module ID	Target Functionality	Operational Status	Verification Outcome
FR-01	API Market Data Acquisition	Verified	Successfully retrieves clean OHLC candle feeds for 1M/5M OTC pairs.
FR-02	Feature Calculation Engine	Verified	Computes all 18 technical indicators in real time without missing values.
FR-03	Model Persistence & Loading	Verified	Serializes model package (.pkl) and verifies feature structure on load.

FR-04	Probability Engine	Inference	Verified	Computes exact probabilities for BUY, SELL, and HOLD classes.
FR-05	Confidence Thresholding		Verified	Correctly suppresses signals when peak class probability is < 70%.
FR-06	Telegram Dissemination	Alert	Verified	Transmits low-latency Telegram alerts containing full trade specifications.
FR-07	CSV Signal Logging		Verified	Logs timestamps, prices, probabilities, and signals into data/signals.csv.
FR-08	Execution Safety	Control	Verified	Hardcoded auto-trade suppression confirmed; zero trade orders sent to broker.

b. Supported OTC Asset Configurations: The operational environment supports four primary OTC currency pairs across two short-term candle timeframes:

Table 4: Supported OTC Currency Pairs and Technical Configurations

Asset Symbol	Pair	Asset Type	Candle Timeframes	Primary Analysis Features	Execution Safety
EURUSD-OTC		Synthetic Forex	OTC 1 Minute, Minutes	5 18 Technical Indicators	Signal Only (No Auto-Trade)
GBPUSD-OTC		Synthetic Forex	OTC 1 Minute, Minutes	5 18 Technical Indicators	Signal Only (No Auto-Trade)
USDCHF-OTC		Synthetic Forex	OTC 1 Minute, Minutes	5 18 Technical Indicators	Signal Only (No Auto-Trade)
USDJPY-OTC		Synthetic Forex	OTC 1 Minute, Minutes	5 18 Technical Indicators	Signal Only (No Auto-Trade)

c. Signal Generation and Confidence Mechanism: Operational live testing verified that signal generation strictly follows the probability threshold rules:

Table 5: Operational Matrix of Signal Generation and Confidence Filter

Predicted Class	P(BUY)	P(SELL)	P(HOLD)	Highest Probability	Final System Result
BUY	0.88	0.08	0.04	88% (BUY)	BUY (Alert Disseminated)
BUY	0.38	0.37	0.25	38% (BUY)	NO SIGNAL (Confidence < 70%)
SELL	0.12	0.78	0.10	78% (SELL)	SELL (Alert Disseminated)
HOLD	0.20	0.20	0.60	60% (HOLD)	NO SIGNAL (Sideways Trend)

d. Live Terminal and Alert Outputs: During active pair monitoring, the terminal outputs detailed signal diagnostics. A representative terminal output generated by option 2 (Analyze one pair) is shown below:

```
=====
==
SIMPLE AI OTC TRADING SIGNAL SYSTEM
=====
==
Python: 3.11.9
```

Model: Random Forest
Mode: AI SIGNAL GENERATION ONLY
Automatic Trading: DISABLED
Minimum confidence: 70%

=====
==
2026-08-10 18:48:06 | SIGNAL | INFO | Model loaded successfully. Features: 18
2026-08-10 18:48:09 | MARKET_DATA | INFO | IQ Option connection successful.
2026-08-10 18:48:09 | AI_TRADING | INFO | Analyzing EURUSD-OTC on 1M.
2026-08-10 18:48:09 | SIGNAL | INFO | Prediction: BUY | BUY 78.40% | SELL 12.10% |
HOLD 9.50% | Final: BUY

=====
==
AI TRADING SIGNAL
=====
==
Pair: EURUSD-OTC
Timeframe: 1 Minute
Analysis Time: 2026-08-10 18:48:09
Signal: BUY
AI Confidence: 78.4%
BUY Probability: 78.4%
SELL Probability: 12.1%
HOLD Probability: 9.5%
Model Prediction: BUY
Current Price: 1.135385
Suggested Entry: 18:49
Suggested Expiry: 18:50
Note: Model probability is not guaranteed trade accuracy.
Automatic trading is disabled.
=====
==
2026-08-10 18:48:10 | TELEGRAM | INFO | Telegram alert sent for EURUSD-OTC BUY.

When a signal satisfies the 70% probability threshold, the Telegram notification module constructs and dispatches an instant alert payload:

AI TRADING SIGNAL
GBPUSD OTC
Timeframe: 1M
Confidence: 88%
Entry: 12:01 PM
Expiry: 12:02 PM
SELL
Martingale:
L1: 12:02 PM
L2: 12:03 PM

L3: 12:04 PM

DISCUSSION

The technical results demonstrate that combining supervised ensemble learning with robust data engineering and strict confidence thresholding yields an effective, emotionless decision-support system for high-frequency OTC trading.

A critical structural strength of the system lies in strict feature space consistency. Financial machine learning pipelines often suffer from training-serving skew, where variables used during offline training are transformed differently or ordered inconsistently during real-time inference. In this implementation, the FEATURE_COLUMNS vector is stored directly within the serialized .pkl package. Upon initialization, signal_generator.py verifies live dataframe structures against saved model signatures, rejecting mismatched models automatically.

The enforcement of a chronological split rather than random cross-validation addresses temporal data leakage. In time-series analysis, random splits allow future observations to inform past parameters, yielding artificially inflated accuracy metrics that collapse in live operational environments. By evaluating performance strictly on out-of-sample chronological test sets, the evaluation reflects real-world generalizability.

Furthermore, the 70% probability threshold serves as a necessary risk control layer. Unfiltered machine learning models force a binary or multiclass classification on every data point, forcing predictions even during highly ambiguous, sideways price action. Suppressing predictions where class probability falls below 70% significantly reduces noise and protects human operators from low-probability setups.

In the context of local FinTech deployment at Investment House, Adamawa State, the system successfully addresses the core problems identified in manual trading: decision latency is eliminated through real-time push alerts, emotional biases are bypassed through quantitative feature scoring, and continuous multi-pair monitoring is achieved without human physical fatigue. Crucially, by maintaining AUTO_TRADE_ENABLED = False, the system avoids regulatory issues and security vulnerabilities associated with autonomous broker order execution, ensuring that human managers retain final execution authority.

Limitations

While the study successfully achieves its objectives, some technical and operational limitations must be noted:

- i. **Financial Market Unpredictability:** Machine learning models rely on statistical relationships derived from historical OHLC data. Sudden microstructural shifts, broker feed adjustments, or unexpected high-impact financial events can temporarily degrade predictive performance.
- ii. **Synthetic OTC Data Transparency:** OTC prices are synthetically generated by broker liquidity algorithms rather than centralized public exchange order books, introducing potential feed opacity.
- iii. **API and Network Dependency:** System operation depends entirely on continuous internet connectivity and stable WebSocket connections to IQ Option and Telegram APIs. Network latency or broker API modifications can disrupt live signal processing.

- iv. **Model Probability vs. Payout Metrics:** High model classification confidence (e.g., 88%) reflects ensemble decision tree agreement on historical patterns; it does not constitute a guaranteed trade win rate or direct financial ROI.

CONCLUSION

This study successfully designed, implemented, and evaluated an Intelligent Machine Learning-Based Trading Signal Prediction System for the IQ Option OTC market, providing a localized FinTech solution for Investment House, Adamawa State, Nigeria. The application integrates real-time API market data retrieval, an 18-variable technical feature engineering module, a 300-tree Random Forest classifier, a 70% confidence-based signal filter, CSV signal logging, and automated Telegram alert dissemination.

The system addresses critical challenges associated with manual chart analysis, including emotional trading bias, decision latency, inconsistent chart interpretation, and limited continuous monitoring capacity. By enforcing chronological train-test splits, maintaining training-serving feature space consistency, filtering low-confidence predictions, and intentionally disabling automated order execution, the paper establishes a practical, risk-averse decision-support framework. The paper also demonstrated that machine learning can be effectively leveraged to enhance operational financial decision-making within local investment institutions in developing digital economies.

RECOMMENDATIONS

Based on the operational findings and technical evaluation of the system, the following recommendations are made:

- i. **Demo Environment Validation:** Local investment traders at Investment House should operate the prediction system in a simulated demo environment for an extended period to assess signal dynamics under diverse market regimes before applying insights to live capital management.
- ii. **Periodic Model Retraining:** The Random Forest model should be retrained periodically using freshly collected historical CSV candle data to ensure the 300 decision trees continuously adapt to recent market volatility patterns.
- iii. **Credential Protection:** Environment configuration files (.env) containing API token credentials and private keys must be strictly protected and excluded from public version control or shared academic repositories.
- iv. **Preservation of Execution Safety:** Automated trade execution should remain permanently disabled (`AUTO_TRADE_ENABLED = False`) unless a dedicated, separately audited research project comprehensively resolves order execution security, API rate-limiting, and regulatory compliance standards.
- v. **Future Architectural Extensions:** Subsequent paper iterations should evaluate alternative model architectures-such as XGBoost, LightGBM, and hybrid Transformer-LSTM networks-using the same chronological testing and confidence filtering frameworks established in this paper. Meaning, developing a local web-based graphical dashboard could further enhance signal visualization and administrative performance tracking.

REFERENCES

- Addy, J., Smith, R., & Kalu, P. (2024). Advanced decision frameworks in financial technology systems. *Journal of Financial Automation*, 12(3), 45-58.

- Avelar, R., & Jordão, B. (2024). Integration of artificial intelligence in financial decision-making systems. *SSRN Electronic Journal*. <https://www.researchgate.net/publication/405458504>
- Central Bank of Nigeria (CBN). (2023). *Fintech evolution and development in Nigeria* (Occasional Paper No. 76). Abuja: CBN Publications.
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383-417. <https://doi.org/10.2307/2325486>
- Fortune Business Insights. (2026). *Algorithmic trading market size, share & industry analysis*. Fortune Business Insights Report. <https://www.fortunebusinessinsights.com/algorithmic-trading-market-107174>
- Jabed, M. (2024). Comparative analysis of machine learning models in financial prediction. *Journal of Data Science and Business Analytics*, 5(1), 12-29.
- Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *The Journal of Portfolio Management*, 30(5), 15-29. <https://doi.org/10.3905/jpm.2004.442611>
- Omole, O. (2024). From forecasting to action: The evolution of machine learning applications in financial markets. *SSRN Electronic Journal*. from <https://ssrn.com/abstract=5884882>
- Oyewole, A., Adeyemi, T., & Bello, S. (2024). Analysis of stock market prediction using optimized neural networks. *African Journal of Financial Technology & Analytics*, 8(2), 101–118.
- Rane, S., Patil, V., & Deshmukh, A. (2023). Deep learning techniques for time-series forecasting. *International Journal of Computer Science*, 11(6), 18-25.
- Research and Markets. (2026). *Algorithmic trading market report 2026*. Research and Markets Technical Industry Review. <https://www.researchandmarkets.com/reports/5939167>
- Sonkavde, G., Patil, S., & Kulkarni, A. (2023). Hybrid approaches in modeling market efficiency. *MDPI Journal of Risk and Financial Management*, 16(4), 210.
- Tian, X. (2024). Emotional pressures and behavioral biases in retail trading. *Journal of Behavioral Finance*, 18(9), 142-156.
- Utari, D., Rahman, F., & Sari, L. (2024). Dynamic behavioral biases under financial stress conditions. *Journal of Applied Economics and Finance*, 22(2), 88-99.
- Wang, L., Chen, Y., & Zhao, H. (2025). Time series forecasting using deep learning: Transformer versus recurrent neural networks. *World Journal of Advanced Engineering Technology and Sciences*, 14(1), 45-59.