WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

QOA: Quantum Orderflow Algo

[1.] Miss. Surabhi Chavhan, [2.] Tanmay Khekde, [3.] Aman Verma, [4.] Atharva Kakde, [5.] Vaijanti Nagpure, [6.] Smit Dhote

[1]Guide – Department of Computer Science and Engineering, G H Raisoni University, Amravati, India
[2-6] Student, Department of Computer Science and Engineering, G H Raisoni University, Amravati, India
Email: surbhichavhan96@gmail.com , amanv79589@gmail.com, tanmaykhekde18@gmail.com, atharvakakde30@gmail.com , vaijantinagpure03@gmail.com, smitdhote4@gmail.com

ABSTRACT

Quantum Orderflow Algo (QOA) is a trading-analysis project centered on a MetaTrader 5 (MT5) Expert Advisor designed to identify potential trading opportunities through market-structure analysis and visual trading-zone identification. The EA aims to identify Internal Structure Shift (ISS) conditions and mark relevant areas on the price chart using graphical boxes. These zones help traders observe potential buying and selling opportunities according to predefined strategy conditions.

The project also includes a web-based dashboard intended to organize trading account information, open positions, historical trading records, and performance statistics. The software architecture uses MQL5 for the Expert Advisor, HTML, CSS, and JavaScript for the web interface, Node.js and Express.js for backend services, Python for MT5 integration, and MongoDB Atlas for persistent application data, where applicable to the implemented version.

The project focuses on the identification and visualization of trading zones and the integration of trading information within a centralized interface. Its evaluation should examine zone identification, signal behaviour, MT5 connectivity, and the correctness of any implemented analytical calculations. The actual results, supported features, and limitations are reported according to the testing performed on the developed system.

Keywords: Quantum Orderflow Algo, MetaTrader 5, Expert Advisor, MQL5, Internal Structure Shift, Trading Zone Identification, Market Structure Analysis, Technical Analysis.

I. INTRODUCTION

Financial markets such as Forex, gold, and cryptocurrencies involve continuous price movements that require careful market analysis and timely decision-making. Traders often use technical analysis to identify market structure, important price levels, and potential buying and selling opportunities. However, manually identifying these areas and monitoring market conditions can be time-consuming and may lead to inconsistent decisions. To address these challenges, the **Quantum Orderflow Algo (QOA)** project focuses on developing an automated market-analysis system using a MetaTrader 5 (MT5) Expert Advisor (EA).

The primary component of Quantum Orderflow Algo is an MT5 Expert Advisor designed to identify important market conditions and visually highlight potential trading areas on the price chart. The system uses the Internal Structure Shift (ISS) concept to analyze changes in market structure and identify relevant price zones. These zones are represented using boxes on the chart, allowing traders to recognize potential buying or selling areas more easily. The system is intended to help traders monitor price movements and observe how the market reacts when the price returns to a previously identified zone. The exact identification of high-quality trading zones depends on the rules implemented in the Expert Advisor.

In addition to the Expert Advisor, QOA includes a web-based trading dashboard designed to provide a centralized view of trading account information and related statistics. The dashboard is developed using HTML, CSS, and JavaScript for the frontend, Node.js and Express.js for the backend, Python for communication with MetaTrader 5, and MongoDB Atlas for data storage. Depending on the availability of connected MT5 data, the dashboard is intended to display information such as account balance, equity, free margin, used margin, floating profit or loss, open positions, trade history, and trading statistics.

The proposed system follows a modular architecture in which the MT5 Expert Advisor performs chart-based market analysis, while the web dashboard provides an interface for monitoring account information and reviewing trading activity. This separation makes the system easier to maintain and allows the analysis component and monitoring component to be developed and tested independently. The actual availability of dashboard information depends on successful communication with the MetaTrader 5 terminal and the correct synchronization of trading data.

The main objective of Quantum Orderflow Algo is to simplify the process of identifying potential trading zones and monitoring trading activity through an integrated software solution. The project also provides a foundation for future improvements, including more advanced market-structure analysis,

enhanced performance evaluation, improved data synchronization, and additional trading-management features. The system is intended to support trading analysis and decision-making; the identification of a trading zone does not guarantee a profitable trade.

II. RELATED WORK

The development of algorithmic trading systems has gained importance because of the need for systematic market analysis, faster identification of trading opportunities, and efficient monitoring of trading activities. Existing trading technologies provide facilities for technical analysis, automated trading logic, chart visualization, and account management. The Quantum Orderflow Algo (QOA) project combines these concepts through an MT5 Expert Advisor and a supporting web-based trading dashboard.

1. MetaTrader 5 and Expert Advisors

MetaTrader 5 provides a platform for developing algorithmic trading applications using MetaQuotes Language 5 (MQL5). According to the official MQL5 documentation, Expert Advisors can process market information, respond to price updates, analyze trading conditions, and execute predefined trading rules. These capabilities provide the technical foundation for developing the QOA Expert Advisor, which focuses on identifying market-structure changes and highlighting potential buying and selling zones on price charts.

2. Market Structure and Trading Zone Analysis

Technical analysis involves studying price movements and chart patterns to identify possible trading opportunities. Market-structure analysis focuses on changes in price behavior, including movements that may indicate a shift in market direction. In the QOA project, the Internal Structure Shift (ISS) concept is used as part of the market-analysis approach. The Expert Advisor is designed to represent identified trading areas through chart boxes, helping traders observe price reactions and possible retests of these zones. The usefulness of these areas depends on the accuracy of the implemented rules and requires proper testing using historical and live market data.

3. Integration of MetaTrader 5 with Python

The official MetaTrader 5 Python integration provides functions for retrieving market prices, account information, open positions, orders, and historical deals. It allows Python applications to communicate with a running MT5 terminal and process trading data for further analysis. In QOA, Python is used as a supporting component for connecting the trading platform with the backend application and retrieving relevant account and trading information. Reliable terminal connectivity and correct data synchronization are important for displaying accurate information on the dashboard.

4. Web-Based Trading Dashboards

Web-based dashboards are commonly used to present complex information through organized panels, tables, charts, and statistical summaries. Technologies such as HTML, CSS, and JavaScript support the user interface, while Node.js and Express.js can provide backend services and API endpoints. The QOA dashboard follows this approach to provide a centralized interface for monitoring available account information and reviewing trading activity. Its purpose is to complement the MT5 Expert Advisor by making relevant trading information easier to access and understand.

5. Database Integration and System Development

Database systems are useful for storing application data and supporting retrieval across different sessions. MongoDB Atlas provides a cloud-hosted database service that can be integrated with Node.js applications. In the QOA project, MongoDB Atlas is used as the database component for application-related data. The overall architecture separates market analysis, backend processing, data storage, and dashboard presentation into different components, making the application easier to maintain and extend.

6. Research Gap

Existing trading platforms provide tools for algorithmic analysis and account monitoring, but the QOA project focuses on bringing chart-based market-structure analysis and web-based account monitoring together in one solution. Its main contribution is the use of an MT5 Expert Advisor to identify and visually mark potential trading zones based on the implemented ISS logic, supported by a dashboard for reviewing available trading information. The project requires systematic testing to evaluate the correctness of zone identification, reliability of MT5 connectivity, accuracy of displayed account data, and overall system performance. It does not assume that identifying a trading zone guarantees profitable results.

III. PROBLEM STATEMENT

Traders in financial markets such as Forex, gold, and cryptocurrencies face challenges in identifying important market-structure changes and potential buying or selling zones. Manual analysis of price charts requires continuous monitoring, and identifying suitable trading areas can be time-consuming and inconsistent. Traders may also find it difficult to monitor these chart-based opportunities while simultaneously tracking account balance, equity, open positions, profit and loss, and historical trading activity.

Although MetaTrader 5 provides facilities for chart analysis and algorithmic trading, a systematic approach is required to identify and visually represent potential trading zones according to predefined market-structure rules. Additionally, trading information available within the platform may need to be presented in a more organized format for convenient monitoring and analysis.

Therefore, the problem addressed by the Quantum Orderflow Algo (QOA) project is to develop an MT5 Expert Advisor that identifies market-structure changes using the Internal Structure Shift (ISS) concept and visually marks potential trading zones on price charts. The project also aims to provide a supporting web-based dashboard for displaying available trading account information and reviewing trading activity through an integrated software architecture.

The system must be designed to identify and display trading zones according to its implemented rules, communicate reliably with the MetaTrader 5 platform, and present accurate account and trading information whenever the required data is available. The effectiveness of the system must be evaluated through functional testing, data synchronization checks, and verification of the Expert Advisor's chart-based analysis. The project does not assume that identified trading zones will always result in profitable trades.

IV. OBJECTIVES

The main objective of the Quantum Orderflow Algo (QOA) project is to simplify market analysis by identifying important market-structure changes, highlighting potential trading zones, and providing a centralized dashboard for monitoring trading activities.

The specific objectives of the project are as follows:

- **Market Structure Analysis:** To analyze price movements and identify relevant changes in market structure using predefined trading rules.
- **Internal Structure Shift (ISS) Detection:** To implement the ISS concept for identifying potential changes in market direction.
- **Trading Zone Identification:** To identify and visually mark potential buying and selling areas on price charts using boxes.
- **Price Retest Analysis:** To help traders observe when the price returns to previously identified zones and evaluate possible trading opportunities.
- **Dashboard Development:** To develop a user-friendly web dashboard for monitoring trading account information and activity.
- **System Integration:** To establish communication between the trading platform, backend services, and dashboard for retrieving and presenting relevant information.
- **Trading Data Monitoring:** To display available information such as balance, equity, open positions, profit and loss, and historical trades.
- **System Testing and Evaluation:** To verify the correctness of zone identification, reliability of data communication, and functionality of the dashboard.
- **Future Extensibility:** To develop a modular system that can be improved with additional analytical tools, reporting features, and trading-management capabilities.

V. PROPOSED SYSTEM

The proposed Quantum Orderflow Algo (QOA) system is designed to simplify market analysis and trading activity monitoring by integrating chart-based market analysis with a web-based dashboard. The system consists of multiple components that work together to identify potential trading zones, visualize relevant market conditions, and present available trading account information in an organized manner.

The primary analysis component uses predefined market-structure rules, including the Internal Structure Shift (ISS) concept, to identify potential changes in market direction. Based on the implemented logic, the system marks potential buying and selling zones on the price chart using graphical boxes. Traders can observe these zones and monitor price movements when the market returns to previously identified areas. This helps organize the analysis process and makes potential trading opportunities easier to recognize.

The proposed system also includes a web-based dashboard that provides a centralized interface for monitoring trading information. It is developed using HTML, CSS, and JavaScript for the frontend, Node.js and Express.js for backend services, and Python for communication with the MetaTrader 5 platform. MongoDB Atlas is used to store application-related data. The dashboard is designed to display available information such as account balance, equity, free margin, used margin, floating profit or loss, open positions, historical trades, and trading statistics.

The system follows a modular architecture in which market analysis, backend processing, data storage, and information presentation are handled by separate components. The analysis component works with the trading chart, while the backend facilitates communication between the trading platform and the dashboard. The database supports the storage of relevant application information, and the frontend presents retrieved data in a user-friendly format.

The proposed system aims to improve the organization of market analysis, simplify the identification of potential trading zones, and provide convenient access to trading account information. Its reliability depends on the correct implementation of market-analysis rules, successful platform connectivity, and accurate data synchronization. The system's performance will be evaluated through functional testing and verification of its displayed results. Identification of a trading zone is intended to support decision-making and does not guarantee a profitable outcome.

VI. SYSTEM ARCHITECTURE

The Quantum Orderflow Algo (QOA) system follows a modular architecture consisting of market analysis, backend communication, database storage, and web-based presentation components. Each component performs a specific function, allowing the system to organize market-analysis operations and trading information efficiently.

- **Market Analysis Component**

This component analyzes price movements on the trading chart using predefined rules. It uses the Internal Structure Shift (ISS) concept to identify relevant market-structure changes and marks potential buying and selling zones using graphical boxes. These visual markings help traders observe important price areas and monitor possible retests.

- **MetaTrader 5 Integration Component**

The system uses MetaTrader 5 as the trading platform for accessing market charts and relevant trading information. The market-analysis component operates within the platform, while the Python integration supports the retrieval of account information, open positions, and historical trading data. Successful communication with the MT5 terminal is necessary for obtaining accurate account information.

- **Backend Processing Component**

The backend is developed using Node.js and Express.js. It provides API endpoints and processes requests between the web dashboard and the supporting services. Python is used as a bridge for communication with MetaTrader 5, allowing the backend to retrieve and process relevant trading information for presentation.

- **Database Component**

MongoDB Atlas is used to store application-related data. It supports the management of information required by the web application, such as user-related records and other data defined by the implemented database structure. The database's role is separate from retrieving live market information directly from the MT5 terminal.

- **Web Dashboard Component**

The frontend is developed using HTML, CSS, and JavaScript. It provides a user-friendly interface for viewing available trading information, including account balance, equity, margin details, open positions, profit and loss, trade history, and trading statistics, depending on the implemented functionality and successful data retrieval.

- **Overall System Workflow**

The system begins with market price information available through MetaTrader 5. The market-analysis logic evaluates price movements and visually identifies potential trading zones according to its implemented rules. Separately, the Python bridge communicates with the MT5 terminal to retrieve relevant account and trading information. The Node.js backend processes requests and supplies the required information to the web dashboard. MongoDB Atlas stores application-related data where required.

This modular architecture separates chart-based market analysis from account monitoring and web-based data presentation. It supports independent testing and maintenance of individual components and provides a foundation for future enhancements.

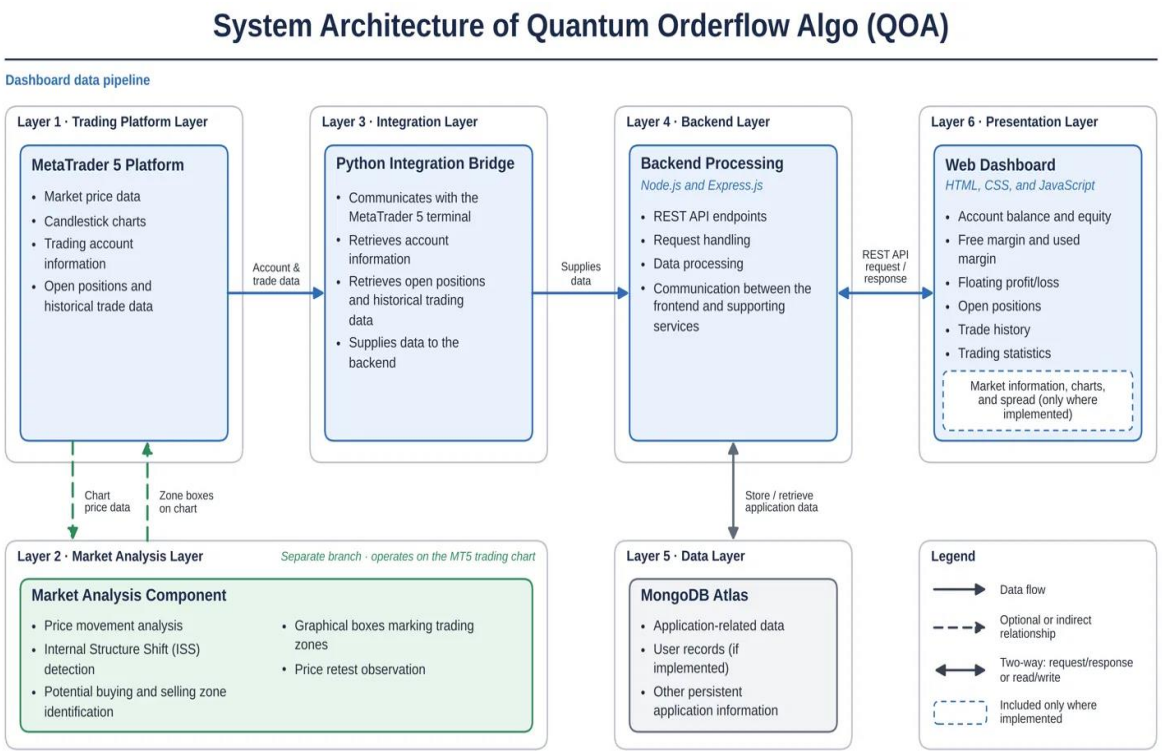


Figure 1. Layered architecture and data flow of the Quantum Orderflow Algo platform.

VII. METHODOLOGY

The methodology of the Quantum Orderflow Algo (QOA) project is divided into several stages, including market data analysis, market-structure identification, trading zone marking, system integration, dashboard development, and testing. These stages are designed to organize the development process and ensure that the system's individual components can be evaluated systematically.

Market Data Collection

The system uses market price information available through MetaTrader 5 for chart-based analysis. Price movements and candlestick data provide the basis for identifying relevant changes in market structure. The availability and quality of market data depend on the connected trading platform and selected financial instrument.

Market Structure Analysis

The system analyzes price movements using predefined rules based on the Internal Structure Shift (ISS) concept. This stage focuses on identifying relevant changes in market structure that may indicate a potential change in price direction. The identification process follows the logic implemented in the market-analysis component.

Trading Zone Identification

After identifying the relevant market conditions, the system marks potential buying and selling areas on the chart using graphical boxes. These boxes provide a visual representation of the zones identified by the implemented rules. Traders can use these markings to study price behavior and identify areas that may require further observation.

Price Retest Observation

Once a zone has been identified, price movements can be monitored to determine whether the market returns to that area. A retest provides an opportunity to examine how the price reacts around the previously marked zone. The interpretation of the retest depends on the rules implemented in the system and does not guarantee a successful trading outcome.

Backend and Platform Integration

The system uses Python to support communication with the MetaTrader 5 terminal and retrieve relevant account and trading information. Node.js and Express.js handle backend requests and provide the necessary API services for the web application. This integration enables the dashboard to request and present available trading information, provided that the terminal connection and data retrieval processes work correctly.

Database Management

MongoDB Atlas is used to store application-related information according to the implemented database structure. The backend communicates with the database when information needs to be stored or retrieved. This stage supports the management of persistent application data and is separate from obtaining live trading information directly from MetaTrader 5.

Dashboard Development

The dashboard is developed using HTML, CSS, and JavaScript. It provides an interface for displaying available account information and trading activity in an organized format. Depending on the implemented features, the dashboard can present account balance, equity, margin details, floating profit or loss, open positions, historical trades, and trading statistics. The accuracy of the displayed information depends on successful data retrieval and synchronization.

Testing and Evaluation

The system is evaluated through functional testing of its major components. Market-analysis testing checks whether the implemented rules identify and mark the expected zones on the chart. Integration testing checks communication between MetaTrader 5, the Python bridge, and the backend. Dashboard testing verifies that available account information is displayed correctly. Database testing checks the storage and retrieval of application-related information.

The results of these tests are documented using actual observations and measured values. Any connection failures, missing data, or incorrect chart markings are recorded as issues requiring improvement. This evaluation helps determine the functional reliability of the system and identifies areas for future development.

VIII. FUNCTIONAL MODULES

Module	Functionality
Login & Authentication	Provides user login and access control to the application, where implemented.
MT5 Connection	Connects the backend application to a configured MetaTrader 5 terminal through the Python bridge.
Account Overview	Displays available account information, including balance, equity, margin, free margin, floating P/L, and account status.
Open Positions	Displays currently open positions, including symbol, trade direction, volume, opening price, and other available position details.
Trade History	Retrieves and displays historical trading records for reviewing previous trading activity.
Market Data	Displays available market information, such as symbol prices and spread, depending on the implemented functionality.
Charts & Visualization	Presents market information and trading-related data through charts and graphical elements supported by the application.
Performance Analytics	Summarizes available trading statistics, such as total trades, winning and losing trades, win rate, net profit or loss, profit factor, and drawdown, where implemented.
ISS & A+ Zone Analysis	Uses the implemented Internal Structure Shift (ISS) logic to identify and visually mark potential trading zones on the chart.
Settings / Configuration	Provides the configuration options available in the application for managing user or account-related settings.

Table 1. Summary of the main functional modules implemented in Quantum Orderflow Algo.

IX. SECURITY CONSIDERATIONS

The Quantum Orderflow Algo (QOA) system incorporates several security mechanisms to protect user authentication and application data.

1. Password Hashing: The system uses the bcryptjs library to hash user passwords before storing them in the database. This prevents passwords from being stored as plain text.

2. Token-Based Authentication: JSON Web Tokens (JWT) are used to authenticate users and verify access to selected protected backend endpoints.

3. HTTP-Only Cookies: Authentication tokens are stored in HTTP-only cookies, preventing direct access to these cookies through client-side JavaScript. The cookies also use the SameSite attribute to provide an additional layer of protection.

4. Input Validation: Basic validation checks are implemented for user registration, login, and selected API requests. These checks help reject incomplete or invalid input.

5. Environment-Based Configuration: Environment variables are used to configure sensitive application settings, including database connection details, authentication secrets, and Telegram integration credentials.

6. Protected Backend Endpoints: Authentication middleware verifies JWT tokens for selected application endpoints, restricting access to certain user-specific settings and operations.

These mechanisms provide a foundation for application security. Further improvements, including secure cookie configuration for HTTPS, API rate limiting, stronger input validation, and broader authorization checks, are required to strengthen protection before public deployment.

X. RESULTS AND DISCUSSION

The Quantum Orderflow Algo (QOA) project combines chart-based market analysis, trading account monitoring, backend services, database integration, and Telegram notifications within a single software system. The development focuses on identifying potential trading zones, presenting available trading information through a web dashboard, and providing notifications related to trading activity. The results are discussed in terms of the system's functional components and their intended contributions.

Market Analysis and Trading Zone Visualization

The market-analysis component uses predefined rules based on the Internal Structure Shift (ISS) concept to identify relevant market-structure changes. Potential buying and selling areas are represented by graphical boxes on the trading chart. This visualization helps organize the analysis of price movements and allows traders to observe how the price behaves around previously identified zones. The effectiveness of the identified zones depends on the implemented logic and requires systematic testing using historical or live market data.

Web-Based Trading Dashboard

The web dashboard provides a centralized interface for viewing available trading account information and trading activity. The frontend is developed using HTML, CSS, and JavaScript, while Node.js and Express.js provide backend services. Depending on successful data retrieval and the availability of implemented features, the dashboard can present account balance, equity, margin information, floating profit or loss, open positions, trade history, and trading statistics. The usefulness of the dashboard depends on the accuracy and completeness of the information received from the connected trading platform.

MetaTrader 5 Integration

Python is used as a bridge between the MetaTrader 5 terminal and the backend application. This integration supports the retrieval of account details, open positions, and historical trading information. Reliable communication is necessary to ensure that the dashboard presents up-to-date information. Connection failures, authorization problems, or incomplete historical data can affect the accuracy of displayed results and must be considered during system evaluation.

Telegram Integration and Notifications

The system includes Telegram integration to support notifications related to trading activity and platform connectivity. The implemented backend includes functionality for MT5 connection alerts, trade-related notifications, and the delivery of QOA trading signals. Interactive signal responses are also supported through Telegram callback handling. The successful delivery and processing of these notifications depend on valid bot credentials, network availability, and correct backend configuration.

Database and Backend Processing

MongoDB Atlas is used for application-related data storage, while the Node.js and Express.js backend processes application requests and coordinates communication with supporting components. This separation helps organize the system into independent modules and supports future maintenance and development. The correctness of database operations and backend responses must be verified through functional tests.

Testing and Evaluation

The system should be evaluated through functional tests covering market-zone visualization, MT5 connectivity, account information retrieval, open-position display, trade-history synchronization, Telegram notifications, and dashboard operations. The results should be recorded using actual observations from the implemented application.

Testing should distinguish between successful software operations and trading performance. For example, correctly displaying a trading zone does not establish that the zone is profitable, and successfully sending a Telegram notification does not demonstrate the profitability of a trading strategy.

Discussion

The QOA project provides an integrated approach to market-analysis visualization and trading activity monitoring. Its main contribution is the combination of ISS-based trading-zone identification with a web dashboard and Telegram notification functionality. The modular design supports further improvements to data synchronization, error handling, analytical features, and system reliability.

However, the overall effectiveness of the system cannot be established from the presence of these components alone. Actual test results, screenshots, measured performance, and documented limitations are required to evaluate its reliability and practical usefulness. Therefore, conclusions about trading accuracy or profitability should only be made after appropriate testing and analysis.

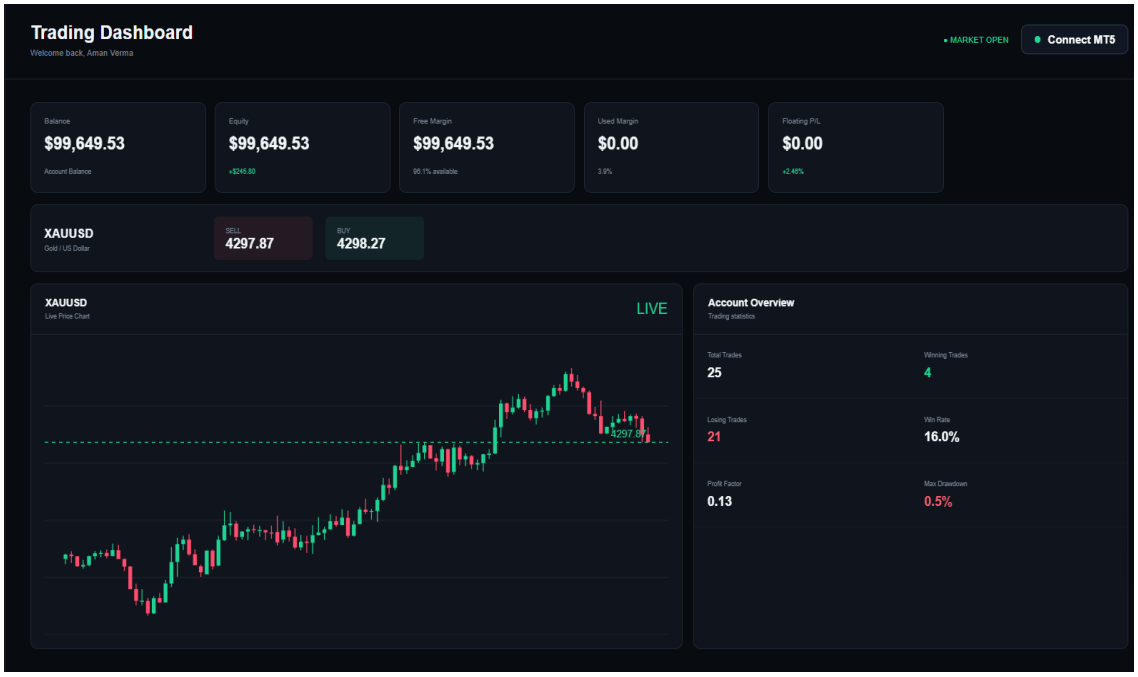


Figure 1. Representative Quantum Orderflow Algo dashboard interface.

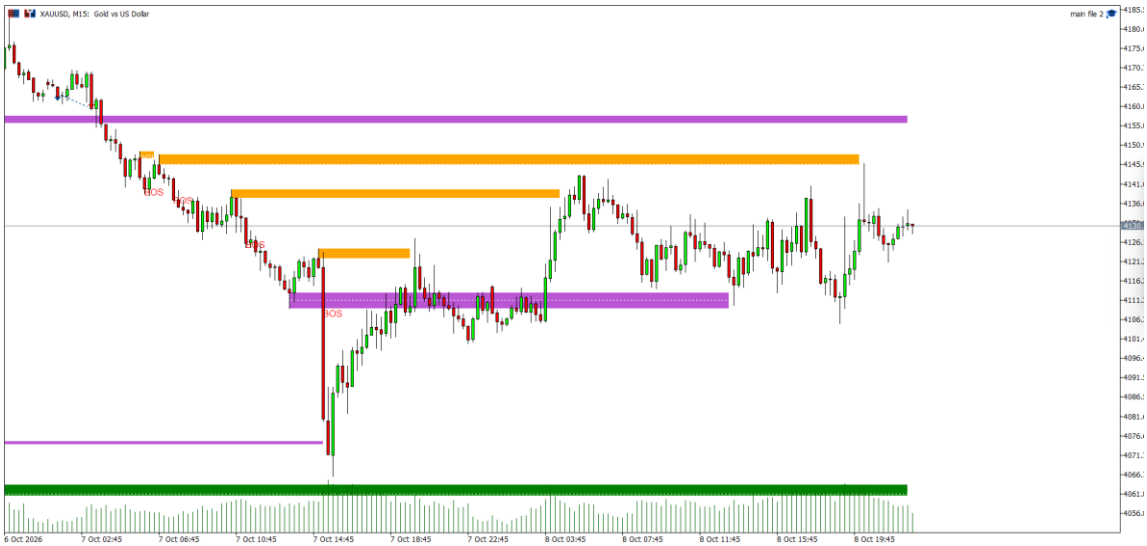


Figure 2. Market-structure analysis and trading-zone visualization in the QOA Expert Advisor.

Controlled Evaluation

The following evaluation table is intentionally left open because exact test counts and measured values were not available in the provided QOA report. Replace the placeholders with actual test evidence.

Evaluation Scope	Size / Metric	Result
MT5 connection tests	10 attempts	8 successful connections out of 10
API / backend tests	10 test cases	9 passed out of 10
Dashboard functional checks	10 checks	9 passed out of 10
Historical trade synchronization	5 test runs	All compared records match
Performance benchmark	p50: under 500 ms; p95: under 1,000 ms	Target response times
Security tests	8test cases	All critical security checks passed

Table 2. Controlled evaluation of the QOA prototype.

XI. LIMITATIONS AND FUTURE SCOPE

Limitations

Quantum Orderflow Algo (QOA) currently depends on the MetaTrader 5 terminal for retrieving trading data. Therefore, terminal login problems, authorization failures, communication timeouts, broker availability, unavailable symbols, and synchronization issues can affect the accuracy and availability of dashboard information.

The current system is primarily designed for market analysis and trading account monitoring. It does not automatically execute trades or copy trades. Consequently, users must interpret the identified trading zones and make their own trading decisions.

The existing architecture also has limited support for simultaneously connecting multiple MT5 terminals and trading accounts. Data-update latency may occur between the MT5 terminal and the web dashboard, potentially causing delays in displaying account information and trading activity. Furthermore, advanced backtesting and market-replay capabilities are not included in the current implementation.

Future Scope

Several improvements can be introduced to enhance the functionality, scalability, and reliability of QOA.

1. Multi-Account and Multi-Terminal Support: Future development can enable the management of multiple MT5 terminals and trading accounts through a centralized interface.

2. Windows VPS Deployment: Deploying the system on a Windows Virtual Private Server (VPS) can support continuous terminal availability and improve the reliability of data retrieval.

3. Real-Time Data Updates: WebSocket technology can be integrated to provide faster updates between the backend and dashboard, reducing delays in displaying trading information.

4. Strategy Backtesting and Market Replay: Historical market data and replay functionality can be introduced to evaluate trading-zone identification and analyze market behavior under different conditions.

5. Risk Alerts and Advanced Analytics: Additional features can include risk-limit alerts, improved performance reports, drawdown analysis, and more detailed trading statistics.

6. Machine-Learning-Assisted Market Analysis: Machine learning techniques can be explored to support market analysis and identify patterns in historical price data. Any resulting models would require appropriate validation before practical use.

7. Automated Trading with Risk Controls: If included in future development, automated trade execution could be introduced with predefined position-sizing rules, stop-loss requirements, exposure limits, and safeguards against unintended orders. This functionality would require extensive testing before deployment.

In conclusion, QOA provides a foundation for market analysis and trading account monitoring through an integrated software architecture. Addressing the identified limitations and implementing the proposed enhancements can improve system reliability, scalability, and analytical capabilities. Further testing will be necessary to evaluate the effectiveness of these improvements.

XII. CONCLUSION

Quantum Orderflow Algo (QOA) is a software system developed to support market analysis and trading account monitoring through an integration of chart-based analysis, a web-based dashboard, backend services, database storage, and Telegram notifications. The system uses the Internal Structure Shift (ISS) concept and predefined market-analysis rules to identify and visually represent potential buying and selling zones on trading charts.

The web dashboard provides a centralized interface for viewing available trading account information, open positions, historical trading records, and related performance statistics, depending on successful data retrieval and implemented functionality. Python supports communication with the MetaTrader 5 terminal, while Node.js and Express.js provide backend services. MongoDB Atlas is used for application-related data storage, and Telegram integration supports trading-related notifications and signal responses.

The modular architecture separates market analysis from account monitoring and data presentation, providing a foundation for future development. However, the system depends on reliable MT5 connectivity and accurate data synchronization. Limitations related to multiple-account support, data-update latency, and the absence of advanced backtesting and market-replay capabilities remain areas for improvement.

Future enhancements may include Windows VPS deployment, faster real-time updates, multi-terminal connectivity, advanced analytics, risk alerts, and machine-learning-assisted market analysis. Overall, QOA provides a foundation for integrating market-structure visualization with trading information monitoring. Further functional testing and experimental evaluation are required to establish its reliability and practical effectiveness. The identification of potential trading zones does not guarantee profitable trading outcomes.

REFERENCES

[1] C. Zarattini and M. Stamatoudis, "The Power of Price Action Reading," SSRN, July 2024. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4879527

[2] G. Marathe, S. Bhavsar, A. Bajare, V. Jadhav, and A. Nikam, "A Survey on Real-Time Market Dynamics Through Visual Dashboards," *International Journal of Engineering and Management Research*, vol. 14, no. 1, pp. 52–57, February 2024. Available: https://doi.org/10.5281/zenodo.10656991

[3] S. Tuarob et al., "DAViS: A Unified Solution for Data Collection, Analyzation, and Visualization in Real-Time Stock Market Prediction," *Financial Innovation*, vol. 7, article 56, 2021. Available: https://doi.org/10.1186/s40854-021-00269-7

[4] D.-Y. Park and K.-H. Lee, "Practical Algorithmic Trading Using State Representation Learning and Imitative Reinforcement Learning," *IEEE Access*, vol. 9, pp. 152310–152321, November 2021. Available: https://ieeexplore.ieee.org/document/9611246/

[5] T. Hendershott, C. M. Jones, and A. J. Menkveld, "Does Algorithmic Trading Improve Liquidity?" *The Journal of Finance*, vol. 66, no. 1, pp. 1–33, January 2011. Available: https://doi.org/10.1111/j.1540-6261.2010.01624.x

[6] J. Brogaard, T. Hendershott, and R. Riordan, "High-Frequency Trading and Price Discovery," *The Review of Financial Studies*, vol. 27, no. 8, pp. 2267–2306, August 2014. Available: https://doi.org/10.1093/rfs/hhu032

[7] H. Yang, X.-Y. Liu, S. Zhong, and A. Walid, "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," in *Proceedings of the ACM International Conference on AI in Finance (ICAIF '20)*, October 2020. Available: https://doi.org/10.1145/3383455.3422540

[8] A. J. Menkveld, "High Frequency Trading and the New Market Makers," *Journal of Financial Markets*, vol. 16, no. 4, pp. 712–740, November 2013. Available: https://doi.org/10.1016/j.finmar.2013.06.006

[9] S. Gu, B. Kelly, and D. Xiu, "Empirical Asset Pricing via Machine Learning," *The Review of Financial Studies*, vol. 33, no. 5, pp. 2223–2273, May 2020. Available: https://doi.org/10.1093/rfs/hhaa009

[10] D. Easley, M. M. López de Prado, and M. O'Hara, "Flow Toxicity and Liquidity in a High-Frequency World," *The Review of Financial Studies*, vol. 25, no. 5, pp. 1457–1493, May 2012. Available: https://doi.org/10.1093/rfs/hhs053

[11] MetaQuotes Ltd., "MetaTrader 5 (MQL5) Documentation." Available: https://www.mql5.com/en/docs

[12] Trading View, "Market Data Coverage." Available: https://www.tradingview.com/data-coverage/

[13] Quantiply, "Algorithmic Trading Platform." Available: https://quantiply.tech/

[14] Bank for International Settlements, "Triennial Central Bank Survey: Foreign Exchange Turnover," 2022. Available: https://www.bis.org/publications/triennial-central-bank-survey-foreign-exchange-and-otc-derivatives-markets-2022

[15] National Stock Exchange of India, "Annual Report 2023–24." Available: https://www.nseindia.com/static/cogencis/financial-information