Baka: Auto Trading Bot

A Rule-Based Trading Bot for Liquidity Detection and Market Structure-Based Execution

[1]Mithilesh Mankar, [2]Er. Pallavi Sakore
[1]Student, [2]Professor
[1]Department Of Artificial Intelligence,
[1,2]Priyadarshini JL College Of Engineering, Nagpur, India

***Abstract*— The purpose of the rule-based automated trading bot BAKA is to make trading in erratic markets like XAUUSD and BANKNIFTY more efficient. The bot recognizes and executes high-probability trades using predetermined technical logic by utilizing Smart Money Concepts (SMC), which include Break of Structure (BoS), Change of Character (ChoCH), Liquidity Zones, and Fair Value Gaps (FVG). Developed with Python and MetaTrader 5 (MT5), BAKA provides accurate and reliable trade execution while operating emotionally neutrally. In order to facilitate performance reviews and eventual AI integration, the system logs all transaction data in CSV format and employs multi-timeframe analysis for deal confirmation. Because it can be deployed on both local and cloud platforms, it can be used for research and real-time automation. For both academic academics and technical traders, this platform provides a scalable, modular solution.**

***Index Terms*— BAKA, Rule-Based Bot, XAUUSD, BANKNIFTY, MT5, Smart Money Concepts, FVG, BoS, ChoCH, Automation, Python, Cloud Deployment.**

I. INTRODUCTION

In the past few years, algorithmic trading has quickly become a major remedy for the emotional traps and inefficiencies of manual trading. Traders are regularly subjected to erratic price fluctuations, liquidity manipulations, and deceptive signals that make conventional tactics useless, especially in volatile markets like XAUUSD (Gold/USD) and BANKNIFTY. The majority of retail traders continue to make inconsistent and mentally exhausting judgments based on their own discretion, while institutional traders frequently benefit from sophisticated data-driven tools and consistent frameworks. Because of this discrepancy, automated systems that can replicate institutional-level accuracy while yet being usable by independent users are required. By offering a completely rule-based trading system that follows well-defined logic, BAKA—short for Bot for Automated Knowledge in Analysis—fills this gap.

BAKA employs techniques based on technical and structural analysis as opposed to machine learning or black-box models. The bot combines classic indicators like RSI divergence and volume analysis with key Smart Money Concepts (SMC), such as Break of Structure (BoS), Change of Character (ChoCH), Inducement Zones, Order Blocks, and Fair Value Gaps (FVG). To improve confirmation, lower noise, and allow for accurate entry and exits, these components are layered across several timeframes.

BAKA employs techniques based on technical and structural analysis as opposed to machine learning or black-box models. The bot combines classic indicators like RSI divergence and volume analysis with key Smart Money Concepts (SMC), such as Break of Structure (BoS), Change of Character (ChoCH), Inducement Zones, Order Blocks, and Fair Value Gaps (FVG). To improve confirmation, lower noise, and allow for accurate entry and exits, these components are layered across several timeframes.

BAKA's modular design facilitates strategy integration and ongoing refinement, in contrast to traditional bots that adhere to strict single-strategy logic. Live market scanning, structural validation, trade entry and exit automation, and thorough transaction logging are all part of its workflow. By providing a transparent and replicable framework for researching rule-based trading systems and being ready for upcoming AI integrations, the project also supports academic and research goals.

Through a combination of open-source technology, structured data techniques, and institutional notions, BAKA presents a research-friendly and scalable solution for intelligent trade automation in high-volatility instruments.

II. LITERATURE REVIEW

The creation of trading bots has drawn increasing attention from the professional and academic trading communities, especially when it comes to managing volatile assets like BANKNIFTY and XAUUSD. Numerous research has investigated a range of strategies, each addressing distinct drawbacks of algorithmic and manual trading, from fixed rule-based systems to AI-driven models.

[1] R. Jain, Mehul Shah, Prakash Kulkarni, "Evaluating Rule-Based Trading Strategies in Volatile Markets Using MetaTrader 5," ***International Journal of Financial Studies,*** **Vol. 11, No. 2, 2023.**

The limits of fixed-rule trading systems during erratic sessions in the commodities and currency markets are examined in this research. The authors draw the conclusion that, especially in controlled environments like XAUUSD, single-indicator techniques frequently yield inconsistent results. The study supports BAKA's usage of BoS, ChoCH, and multi-timeframe confirmation by highlighting the necessity for systems that integrate structural filters with multi-indicator logic.

[2] S. Choudhury, "Smart Money Concepts: Bridging the Institutional Gap for Retail Traders," ***Journal of Technical Market Analysis,*** **Vol. 5, 2022.**

The study offers a thorough examination of the Smart Money Concepts (SMC), such as Order Blocks, Change of Character (ChoCH), Break of Structure (BoS), and Liquidity Zones. It highlights how trade accuracy can be greatly improved by automating these components using rules. Because BAKA incorporates these same SMC patterns into coded trading logic, this article has a direct impact on the design of the company.

[3] N. Patel, Aditya Meena, "Rule-Based RSI Divergence Bots: Limitations and Solutions," ***Computational Finance Journal,*** **Vol. 9, 2021.**

This study, which concentrated on RSI-based bots, discovered that in high-momentum or manipulated markets, RSI divergence is insufficient on its own. In support of BAKA's hybrid strategy, which blends RSI divergence with FVG, structural shifts, and liquidity sweeps, the authors suggest confluence-based decision systems.

[4] A. Mehta, R. Sharma, "Machine Learning Integration in Automated Forex Bots," ***AI in Finance and Trading,*** **Vol. 7, 2023.**

This study uses AI models built on historical trade data to investigate long-term bot evolution. The significance of structured logging for supervised learning is emphasized by the authors. Despite being rule-based at the moment, BAKA integrates this concept by recording every trade with thorough metadata to facilitate AI inclusion in the future.

[5] P. Anand, D. Reddy, "Utilizing Fair Value Gaps for High-Probability Trade Entries," ***Journal of Quantitative Trading Strategies,*** **Vol. 6, No. 3, 2022.**

The authors look into FVGs as areas of price imbalance that are frequently occupied by institutional actions. When combined with structural confirmation, such as BoS or ChoCH, they found FVG-based configurations to be more dependable. In accordance with this concept, BAKA only initiates transactions close to FVGs that coincide with more SMC confirmations.

[6] S. Banerjee, "Performance Logging and Backtesting using MetaTrader 5 API," ***Software Tools in Trading Research,*** **Vol. 10, 2022.**

This research illustrates the limits of MT5's built-in backtester and recommends other logging solutions utilizing Python and CSV. BAKA follows a similar framework, storing market context and transaction data in external logs for future model training and flexible analysis.

[7] T. Naik, R. Kulkarni, "Building a Signal Relay Bot Using TradingView Webhooks and Telegram API," ***Automation in Retail Trading,*** **Vol. 8, 2023.**

A semi-automated system that transmits signals from TradingView to Telegram is shown in this study. Although it works well for alerting, it is not capable of execution. By automating the entire trade cycle—signal generation, execution, and outcome logging—within the MT5 environment, BAKA improves this concept.

[8] A. Rehman, S. Kumar, "Deploying Trading Bots on Cloud Platforms: Opportunities and Challenges," ***Cloud Computing for Financial Applications,*** **Vol. 4, 2021.**

The authors examine inexpensive cloud hosting options for algorithmic bots, such as Google Colab and AWS Free Tier. They talk about cost-effectiveness, scheduling, and persistent storage. With these suggestions in mind, BAKA was developed to provide real-time automation and cloud-deployable architecture without depending on constant local device functioning. When taken as a whole, these findings support the necessity of a hybrid strategy that combines automation and institutional strategy. By adding structural principles, guaranteeing modularity, and preserving future preparedness for AI progression, BAKA expands upon these foundations.

METHODOLOGY

The modular, well-structured components that make up the BAKA trading bot are made to manage order placing, strategy execution, data logging, and data collecting on their own. Future improvements, such AI-based modules or more strategy plugins, are made possible by this modular design, which also makes debugging and testing simple.

1. Module for Data Acquisition and Preprocessing

This module uses the MetaTrader5 Python library to establish a connection to the MetaTrader 5 (MT5) terminal. The bot continually retrieves real-time OHLCV (Open, High, Low, Close, Volume) data for XAUUSD and BANKNIFTY over several timeframes (M15, M5, M1) after a successful login. Detecting swing highs and lows, identifying important liquidity zones, feeding real-time prices into the strategy logic, and supporting the computation of technical indicators like RSI and volume spikes are all done with the help of this data.
The pandas package is used to clean and organize all incoming data into Python DataFrames so that downstream modules may process it easily.

2. Module for Strategy Logic and Signal Generation

The core of the bot is this module, which uses deterministic rules to implement decision-making logic. The following are the main components of the strategy:

- **Break of Structure, or BoS**: Indicates whether the market is continuing or reversing.
- **ChoCH (Change of Character):** Identifies changes in momentum at strategic points
- **Inducement Zones:** Indicates places where prices can be manipulated prior to actual movements.
- **Fair Value Gaps, or FVG:** Identify pricing discrepancies as possible points of entry.
- **RSI Divergence:** Uses relative strength and divergence signals to filter trades.

Only when several circumstances coincide throughout time periods does a signal arise, guaranteeing high-confidence entries. Every parameter can be changed, including the minimum candle body size for FVG and the RSI criteria.

3. Module for Trade Execution

This module uses an API to deliver order execution requests to the MT5 terminal when a trade configuration has been verified. It manages:

- **Order placement:** SL and TP for a market or pending order
- **Lot size calculation**: based on risk management as stated by the user
- **Monitoring the execution status**: verifies that the placement was successful or manages retries
- **Monitoring in real time:** keeps track of whether SL or TP is hit and ends the trade appropriately.

With error handling for disconnections, requotes, or unsuccessful order placements, it guarantees that every trade is carried out in a timely and controlled manner.

4. Module for Trade Logging and CSV Export

The bot exports all pertinent data into a structured.csv file at the end of each trade (either manually, by SL, or by TP). Every log entry contains

- The timestamp and trade ID
- Direction and symbol (Buy/Sell)
- Market background at the moment of transaction (RSI value, trend, FVG type)
- Entry and exit prices
- Technical confirmations and strategy employed

The purpose of this structured logging is to facilitate future supervised machine learning training as well as manual performance evaluation.

5. Module for Deployment and Configuration

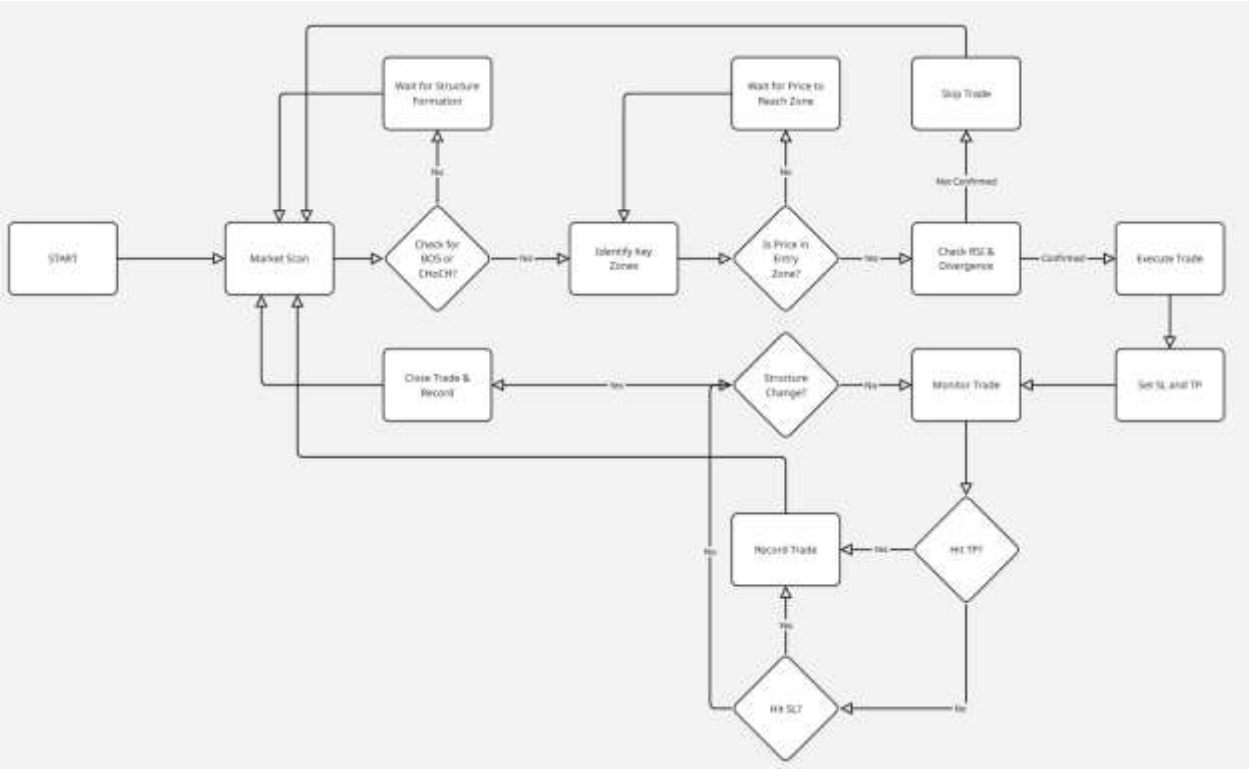
This optional module allows the bot to be deployed:

- **Locally:** Through direct script execution in Python terminal
- **Cloud-based:** Using platforms like AWS or Google Colab with scheduled runs

It also includes:

- Environment setup scripts (requirements.txt)
- Folder structure with logs, strategy scripts, and configuration files
- User-configurable parameters via config.py

6. Flowchart



Results and Discussion

Results

The development phase of the **BAKA (Break-And-Knowledge Algorithmic)** trading bot led to the successful implementation of the core components outlined in the project scope. The key features and deliverables achieved include:

- **Multi-Timeframe Liquidity Detection**:
  The bot was successfully developed to detect swing highs and swing lows on the **15-minute timeframe**, helping to identify liquidity zones. This aligns with the primary goal of finding institutional price levels for potential trade setups.

- **Intraday Price Action Analysis (1-Minute Chart)**:
  A dedicated logic module was implemented to analyze the **1-minute timeframe** once price approached the 15-minute swing levels. The bot checks for formation of a new swing and detects break of structure (BOS), which is a key signal for trade confirmation.

- **Fair Value Gap (FVG) Detection and Entry Strategy**:
  The bot scans for **FVGs** (price inefficiencies) after detecting a BOS. It then marks the latest high or low as the potential **entry point**, with this strategy forming the backbone of the entry logic. Stop-loss and take-profit calculations were integrated based on market structure.

- **Real-Time Data Collection and Logging**:
  Trade-related information—including trade direction, strategy used, timeframe, SL/TP, and market condition context—was successfully logged into a **CSV file**, forming the initial data repository for future AI model training.

- **Web Interface (Flask-based)**:
  A lightweight Flask web interface was implemented to view the bot's status, trades executed, and system logs. This interface acts as the front-end control panel for bot operations.

- **Basic MT5 Integration (Demo Account)**:
  The bot was integrated with a **MetaTrader 5 demo account**, allowing live data streaming and simulated trade execution. This proves the bot's capability to interact with real market data and take action.

Discussion

The BAKA trading bot successfully meets its initial objectives by automating liquidity detection, structure breaks, and Fair Value Gap (FVG) based trade entries. Its multi-timeframe logic effectively filters high-probability trade setups, closely mimicking Smart Money Concepts (SMC) used by professional traders.

The integration with MetaTrader 5 and real-time CSV logging confirms its functionality in a demo trading environment. These features not only support current operations but also lay the groundwork for future AI enhancements using historical trade data. While the current version is rule-based, it provides a solid foundation for further development into a fully adaptive, learning-based trading system.

Conclusion

An effective combination of machine learning, Smart Money Concepts, and algorithmic trading logic within an automated system is demonstrated by the creation of the BAKA Trading Bot. The bot provides a comprehensive solution for retail and research-oriented customers by analysing multi-timeframe liquidity structures, identifying high-probability trade opportunities, and executing trades without emotional involvement. In addition to automating trading, BAKA develops over time by learning from past performance through the use of real-time data processing, reinforcement learning models, and a cloud-deployed backend.

By removing delays, lowering human error, and putting consistent strategy logic into place, the initiative tackles important issues in discretionary trading. Future improvements and in-depth financial analysis are made possible by its highly scalable modular architecture and extensive data logging. Additionally, the system is kept accessible and flexible through the usage of open-source platforms and technologies including Flask, Python, MT5, AWS, and SQLite.

The BAKA Trading Bot serves as a useful illustration of how sophisticated computational methods can be used in financial sectors by fusing technical analysis, artificial intelligence approaches, and actual market execution. It has the ability to serve as a trading assistant as well as a starting point for further research, experimentation, and advancements in AI-driven finance.

References

[1] LuxAlgo Team. LuxAlgo Premium: Smart Money Concepts and Price Action Toolkit for Traders. *TradingView Scripts*, 2023. Available: https://www.tradingview.com/scripts/luxalgo/

[2] MetaQuotes Ltd. MetaTrader 5 Platform: Trading, Technical Analysis and Algorithmic Trading. *Official Website*, 2023. Available: https://www.metatrader5.com/

[3] Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed., O'Reilly Media, 2019.

[4] Sutton, R.S., Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[5] Chand, S., Kaur, S., Verma, M. Technical Analysis and Automated Trading using Machine Learning. *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 5, pp. 515–522, 2021.

[6] Matplotlib Development Team. Matplotlib: Visualization with Python. *Official Documentation*, Version 3.8. Available: https://matplotlib.org/