Financial Advisor Bot: SmartForexBot

Tan Wei Cheng

September 2025

Contents

1 Introduction

1.1 Project Concept

The project that is selected is based on the Artificial Intelligence Idea 2: Financial Advisor Bot template. The project is called SmartForexBot, which is an AI-driven financial advisor bot which is designed to work in the Forex market, where live data and historical data would be retrieved and the bot will use indicators to decide when and how the user would enter the market. It's main indicator to be used would be the fibonacci retracement pattern,using different time periods and candlestick data. And basing of the analysis, the bot will provide recommendations in natural language to users through a web interface. The main goal for this project is to simplify trading decisions for non-technical users, whilst trying to make trading decisions transparent.

This project focuses specifically on the forex market due to its volatility and its constant activity, running 24hrs 5 days a week. This type of active and quick moving market combined with a need and reliance on technical indicators makes it hard for newer investors to get into the market. It is also perfect for Machine learning to help make sense of the data and perform quick turnovers and alerts.

2 Motivations

In forex trading, many forex traders struggle with the complexity and subjectivity of using technical indicators for pattern detection and market movement, both of which are time-consuming. Paired with most existing advisor bots relying on static rules based systems or hidden behind paywalls, makes such products less adaptable and harder to trust. At present, there are no accessible tools that bring together technical indicator pattern detection, machine learning based predictor validator and human readable training recommendation explanation. This is a clear gap in the market where non technical users who seek explainable and data supported trading decisions. SmartForexBot aims to bridge this gap by offering an automated and user-friendly solution that simplifies analysis while maintaining transparency.

3 Literature Review

The goal of this literature review would be to look at other related works and to learn from so that we can have a better understanding of how to more effectively implement a financial trading advisor bot, and how we can take certain components of other projects and put a spin of them for implementation or to see where we can improve from. Some examples would include other projects detailing an implementation process of an AI trading bot and other already existing products.

3.1

The first literature to be reviewed is an article on an intraday trading bot that focuses on foreign exchange markets using python, which is closely related to the aims of the project. The authors present a forex trading bot with a strategy based on a combination of technical indicators, such as RSI, EMA, and MACD, and utilizing Python and MetaTrader. Finding price action within a given time window during the New York session, referred to as the definition range (DR), is the central idea of trading logic. According to their explanation, "the bot will enter the trade on retest of the Imbalance Gap" and "we have observed an 85

The model's dependence on performance optimization through historical tuning is a major cause for concern. "Strategies were optimized using parameter tuning to maximize profit over historical windows," the authors write, exposing a risk of overfitting, in which strategies work well on historical data but fall short in real-world settings because of market volatility. In addition, their own research shows that "the accuracy of our algorithm may plummet to as low as 44

Furthermore, while the study includes back testing and comparisons with benchmarks such as gold and the S&P500, it lacks a systematic evaluation of model interpretability. Users are not informed why signal generated, only, only that they are. This lack of transparency may lead to poor decision-making or reduced trust in the system, particularly among technical users.

The project to be implemented would take inspiration from the approach of this project using signals and time-based entry logic, but would also include advice on how the bot would determine when it enters the market, addressing the lack of transparency in what signals the bot would be looking for. SmartForexBot integrates Fibonacci retracement analysis with a lightweight ML classifier to assess pattern reliability across different conditions. By also providing plain-language explanations of each trade decision,

3.2

The next article is about a trading bot that is based on the crypto currency market. Peng et al. [2022] developed a cryptocurrency trading bot that uses deep reinforcement learning (DRL) to automate buy and sell decisions. Their system is built on a combination of two types of machine learning models, CNN and LSTM, which work together to identify patterns in market data. CNN focuses on spotting useful features from the input data, while LSTM tracks how these patterns change over time. According to the authors, this setup helps the bot respond more effectively to the fast-moving nature of the crypto market, especially since 'temporal correlations' play a big role in price behavior [Peng et al. 2022]. The bot is trained using 100 hour segments of market information, including technical indicators such as RSI, ATR, and OBV, and its performance is measured by how much it increases or decreases the user's net worth - "each predicted action directly impacts the agent's net worth" [Peng et al. 2022].

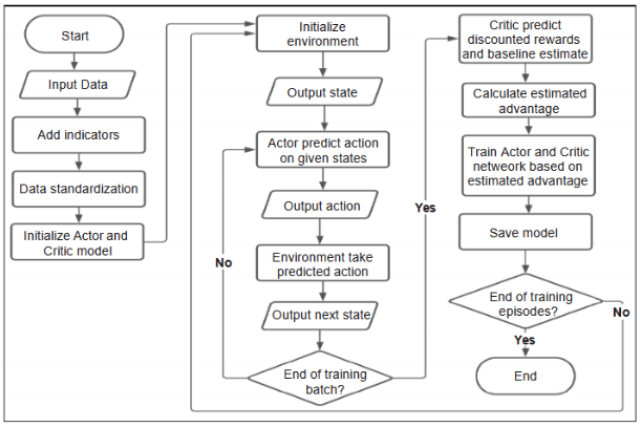

Figure 1: Caption

The results are impressive, with reported annual returns of over 10,000% on Bitcoin trades. However, the paper also points out the risks of using this approach. Because the model is constantly updating its strategies based on past outcomes, there's a risk it will make unstable or overly aggressive changes if the input data is noisy. Even though the PPO algorithm is designed to limit these shifts, the authors admit that problems with the "noisy value function" still led to "major updates to the policy network" (p. 2696). Another concern is that the decision-making process of the bot isn't transparent. Unlike rule-based systems where you can see exactly why a trade happened, this DRL model doesn't provide clear explanations, which could be a problem in situations where trust and accountability matter—especially in regulated financial environments. It's also worth noting that the model was trained and tested mainly on Bitcoin, which has historically trended upward. As the authors admit, this could skew results: the model's high returns might not hold in less favourable market conditions (p. 2699).

For SmartForexBot, this research is both a helpful reference and a reminder of the risks. The idea of combining technical indicators with machine learning is encouraging, but instead of fully relying on DRL, SmartForexBot takes a more cautious route. It uses a transparent, rule-based strategy—Fibonacci retracement—along with a simple machine learning model to evaluate pattern reliability. This way, users can understand the bot's decisions while still benefiting from data-driven insights. Peng et al.'s work opens the door to potentially using DRL in the future, perhaps to help rank or prioritize trade signals. But for now, the need for explainability and trust makes a more interpretable approach the better choice.

In addition to academic research papers, it's also important to consider the tools currently available to traders. Platforms like MetaTrader and TradingView, while widely used, expose clear gaps that SmartForexBot aims to address.

Since MetaTrader is one of the most popular platforms for algorithmic trading, it was examined in order to investigate rule-based automation techniques. With MetaTrader, users can create, implement, and operate "Expert Advisors" (EAs), which are automated trading bots that are programmed in MQL4 or MQL5 to execute trades in response to pre-established technical signals.

These bots are completely deterministic in their design. A typical EA strategy might say, for example, "Buy if RSI ¡ 30 and MACD crossover is positive." When the requirements are satisfied, the bot automatically follows these rules. Although this provides speed and consistency, a major drawback is that the user is not given any feedback or an explanation of the reasons behind a trade. Users must examine code or logs to determine why a trade was made because there is no interpretability layer.

Furthermore, MetaTrader bots don't learn from errors or enhance signal quality over time unless specifically changed. This reduces their usefulness in extremely volatile markets like forex, where changing macroeconomic

conditions can cause signal performance to deteriorate over time. Additionally, they are susceptible to false positives from distracting technical signals, particularly in markets that are choppy or sideways.

On the other hand, SmartForexBot would aim to expand on this idea of organized, rule driven execution by adding natural language explanations and ML-driven pattern scoring. SmartForexBot enhances interpretability and dependability by assisting users in comprehending the why of decisions rather than just the what.

The next platform I wanted to look into would be TradingView. TradingView is one of the most widely used platforms for technical charting, which provides a range of visual tools, including over 100 indicators and the ability to write custom scripts using Pine Script.. It allows traders to manually test ideas through backtesting, set alerts, and identify trends.

Although the platform has advanced visual and analytical features, it lacks an automated advisory system and recommendation system. In other words, it does not advise users on how to use the data; instead, it leaves the trader's interpretation completely up to them. "Scripts can generate alerts but not execute orders or provide trading advice," according to its documentation - https://www.tradingview.com/pine-script-docs/

TradingView's philosophy as a tool for technical traders rather than a decision-making engine is reflected in this design decision. Because of this, novice or non-technical users might find it difficult to decipher complicated signals or comprehend when and why a setup is advantageous. Additionally, there is no learning mechanism to highlight or modify better-performing indicators in light of past results.

TradingView's clarity and pattern visualization serve as an inspiration for SmartForexBot, going one step further by evaluating Fibonacci retracements, evaluating signal dependability, and providing understandable explanations. By doing so, it helps users who might lack the knowledge to decipher raw charts make informed decisions by bridging the gap between professional charting and intelligent decision support.

In summary, this literature review explored a diverse set of approaches to algorithmic trading, ranging from academic implementation using indicators and deep learning to established trading platforms like Meta trader and Trading View. Each example help to contribute key insights into developing and designing a reliable financial advisor bot.

4 Project Design

4.1 Project overview

A web-based financial advisor bot for the forex market, SmartForexBot combines technical analysis, machine learning (ML), and natural language generation to deliver easily comprehensible, data-driven investment advice. This project tackles a major issue facing retail forex traders: the intricacy of analyzing market trends.

The project aims to make forex trading open to more people by building an intelligent advisor bot that helps users make informed investment decisions without needing technical expertise.The bot operates by analysing market behavior through a blend of Fibonacci retracement pattern recognition and machine learning models, then delivering trade recommendations via natural language explanation.

4.2 Target Domain and User

The project targets the domain of forex trading, a globally accessible and highly liquid market that is challenging to get into especially for newcomers due to its volatility and reliance on having the technical knowledge and expertise to do technical analysis. There are many products out there on the market that are designed for professionals, or experts who know what they are looking for, allowing for fine tuning, or on the other hand products that are made for users who have no knowledge at all and allows for automated "AI Trading" with a fee.

So with that in mind, the target user in mind for this project would be users who lack time to learn the market or lack financial literacy to interpret complex data who are likely aged 20 to 40 years old. These users would prefer to have a more intuitive and easy to navigate platform with little to no learning curve, allowing it to be a platform where users can get advice and see why or how the bot makes the decision.

4.3 Design Choice

The goal of SmartForexBot is to provide reliable, understandable investment advice, and its design is informed by the needs of its target users, which are retail forex traders without the technical expertise usually needed to analyze market data. A modular architecture has been chosen to address this challenge, guaranteeing that intricate procedures such as data analysis and machine learning are managed in the backend while the frontend concentrates on usability and simplicity.

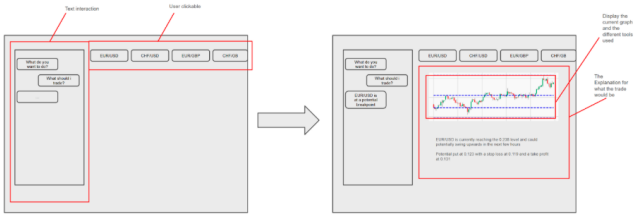

Figure 2: Caption

Fibonacci retracement is a well-known technical analysis tool that graphically depicts probable support and resistance levels based on recent price action, which is why it was used. When properly explained, this makes it an obvious choice for users to understand market turning points. The credibility of the recommendations will be increased by employing a lightweight machine learning model to validate retracement-based decisions using past price patterns, thereby increasing the accuracy of these signals. Explainability is a fundamental component of the system. To convert technical outputs into explanations in plain language that are appropriate for the user's comprehension level, a natural language module is integrated. This promotes user trust and simplifies trading decisions.

Lastly, the system will be mobile-friendly and web-based. With a simple design and a focus on interaction, the front-end will remain user-friendly, enabling users to view charts, ask questions of the bot, and get feedback with ease. Future growth, like mobile apps or integration with third-party brokers, will be made possible by the backend architecture's API-driven design.

4.4 Project Features

SmartForexBot is built around four key features: a Fibonacci retracement detection module, a machine learning-based signal validation engine, an explanation generation module, and an interactive web interface.

For the retracement detection module, the idea is to identify the critical support and resistance levels by calculating Fibonacci retracement percentages (e.g., 23.6%, 38.2%, 50.0%, and 61.8%) from recent highs and lows in currency pair prices. The bot monitors price action in 4-hour candlestick intervals over a 3-month window to identify potential reversal zones. The retracement levels serve as initial trading signals, alerting the system when the price approaches a historically reactive zone.

To implement this, finance API would be used to retrieve live and historical price data and then parsed to calculate the fibonacci ranges. There would also be an alert that would be triggered when the price approaches a retracement level, and sent through the user interface.

The next module would be a machine learning signal validation, where we would use a classifier to predict if the signal would likely lead to a successful trade, and to get an estimate of how high the success rate would be. The model would use the previous feature of fibonacci, and other indicators such as RSI or trends to provide context around the signal.

We would use historical data to train the model for detecting retracement patterns and their success rates, and integrate a confidence score for each signal.

The natural language explanation would play hand in hand with the web interface, where users would mainly interact with. One of the main aspects of this project is the ability to explain recommendations in human readable text pairing with an easy way to interact with the search and result. The component would use a local language model to convert the results received in data points into a sentence or paragraph that is readable, enabling users to understand why the recommendation was made. The front end serves as the user's main point of interaction. It includes a dashboard with candlestick charts, current Fibonacci levels, trade recommendations, and the ability to ask the bot questions. The design will be minimalist, mobile-first, and optimized for speed.

4.5 Work Plan

This chart details the work to be done for the final year project. The items in red are to be completed. The initial prototype in week 9 is to test certain features to ensure that the project will be able to run, and that the main functions are able to be implemented. Following that in week 11 to 12, with the foundation in place, the weekly coding sprint begins, each focusing on different key parts of the project. In week 12 to 14, a machine learning classifier was trained using historical retracement data and other indicators, playing the role of a validator to ensure the quality of trading signals and improve recommendations reliability. After which, testing is conducted on all

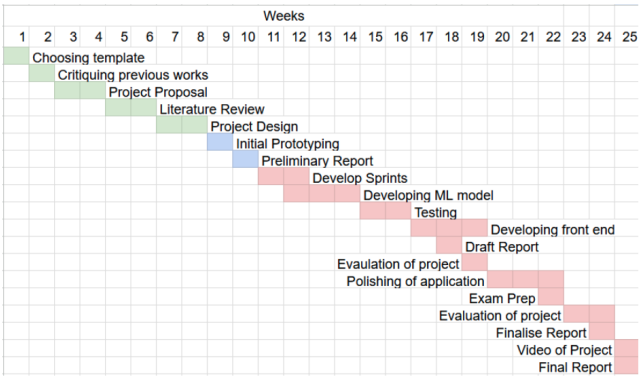

Figure 3: Caption

the core modules, making sure that all of them works before putting them through to the next step of front end development.

In week 17 to 19, The User interface of the project will be developed to connect all the core modules and to enable communication by the users. This step would focus on clarity and ease of use, ensuring that it is able to fulfil it's purpose to show and explain trade recommendations. During the development of the front end, a draft report will also be made alongside to update the content and testing outcomes. Followed by an initial evaluation of the project to see if it hits the target.

Week 20 to 22 would mainly consist of finetuning of the project and any other design features to be implemented, and to focus on the examination.

And lastly week 23 to 25 would be the final sprint to conclude the final evaluation of the project and completion of the report. Video of the final product would also be filmed to showcase the bot's core functionality and submission of the report due on the 15th of Sept.

4.6 Project Evaluation

To see if SmartForexBot meets its goals, the project will be tested in three main areas: how well the bot performs technically, how easy it is for users to use, and how likely users are to keep using it. A mix of testing methods will be used, including looking at past trading data and asking users for feedback.

The first part of the evaluation will focus on how accurate and useful the bot's trading advice is. This will be done by running backtests using historical forex data, such as from EUR/USD or GBP/USD currency pairs. The bot's signals will be compared to a basic "buy and hold" strategy to see which performs better. Important things to measure include how often the bot's suggestions are correct, how many of those suggestions turn out to be winning trades, and how much profit the strategy would have made overall. If the bot can give correct signals at least 70 percent of the time and show better results than a simple strategy over a three-month period, this part of the project will be considered successful.

Next, the project will look at how easy and understandable the bot is for users. After using the system, participants will be asked to fill out a short System Usability Scale (SUS) survey, which is commonly used to check how user-friendly something is. They'll also be asked to do a few simple tasks, like finding a trade recommendation, understanding why it was given, and asking the bot a follow-up question. These activities will help show how smooth and clear the user experience is. On top of that, some users will be interviewed to get more detailed thoughts on what they liked, what confused them, and how confident they felt using the bot. This part will be considered a success if most users find the explanations clear and if the system scores 75 or higher on the SUS survey.

The final area is about whether people actually enjoy using the bot and want to keep coming back. During testing, usage data will be collected to see how long users stay on the app, how often they return, and how they interact with features like charts and explanations. If users are spending at least five minutes per session and coming back at least three times a week, that will be a good sign that the bot is useful and engaging.

Overall, the project will be considered successful if it can give accurate trading advice, explain it in a way that users understand, and keep users coming back. All parts of the system—retracement detection, machine learning, natural language explanations, and the web interface—should work together smoothly. User feedback will be a big part of deciding whether the bot is actually helping people feel more confident in their trading decisions.

5 Implementation

6 Evaluation

7 Conclusion

References

MetaQuotes Software Corp.. 2024. *MetaTrader 5 — Trading Platform*. Online. Accessed: 5 June 2025. (2024). https://www.metatrader5.com/en.

A. Nigam, R. Singh, V. Kumar, and S. Narayan. 2023. *Intraday Forex Trading Bot Using Python and Technical Indicators*. SSRN. Accessed: 5 June 2025. (2023). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4492355.

A. Peng, S. L. Ang, and C. Y. Lim. 2022. *Automated Trading Strategies Using Reinforcement Learning: A Comparative Analysis*. ResearchGate. Accessed: 5 June 2025. (2022). https://www.researchgate.net/publication/364025503_Automated_Cryptocurrency_Trading_Bot_Implementing_DRL.

TradingView Inc.. 2024. *Pine Script Language Reference Manual — Version 5*. Online. Accessed: 5 June 2025. (2024). https://www.tradingview.com/.

Twelve Data Pte. Ltd.. 2025. *Twelve Data — Stock, Forex, and Crypto Market Data APIs*. Online. Accessed: 16 June 2025. (2025). https://twelvedata.com/.