WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

TRADE PILOT AI Crypto Currency Trading Bot

Dr. Abha Choubey[1], Mahima Thakre[2], Divyansh Singh Bhadoriya[2]

[1]Professor, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India
[2]Student, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India

ABSTRACT :

This paper presents the design and implementation of an AI-powered cryptocurrency trading bot that integrates algorithmic trading strategies with a graphical user interface (GUI) and smart contract simulation. The system leverages a rule-based approach using the Simple Moving Average (SMA) crossover strategy for decision-making and supports back testing on historical market data spanning from 2020 to 2025. The bot is designed to evaluate performance through key financial metrics such as cumulative return, Sharpe ratio, and drawdown. A user-friendly interface built with Tkinter facilitates real-time visualization of trades, portfolio performance, and trade history. Furthermore, the project explores a foundational integration of blockchain-based fund management using Solidity smart contracts and Web3.py, demonstrating how decentralized technologies can enhance transparency and security in automated trading systems. This work aims to bridge the gap between algorithmic trading, user accessibility, and decentralized finance (DeFi), providing a viable prototype for further research and development in autonomous trading frameworks.

Introduction

In recent years, the exponential growth of the cryptocurrency market has fueled demand for autonomous trading systems capable of analyzing vast quantities of data and executing trades with speed and precision. As markets operate continuously without centralized regulation, the volatility and complexity of crypto assets present both opportunities and challenges for traders. This environment creates an ideal use case for AI-driven trading bots that can respond to market signals in real-time, reduce emotional decision-making, and optimize profitability.

This paper introduces **Trade Pilot**, an AI-powered cryptocurrency trading bot designed to simulate, analyze, and execute trading strategies with minimal human intervention. Developed as a comprehensive academic project, Trade Pilot integrates machine learning principles, financial technical indicators, a graphical user interface (GUI), and a basic smart contract simulation to demonstrate the practical feasibility of algorithmic crypto trading.

Trade Pilot is engineered around a rule-based trading strategy utilizing the Simple Moving Average (SMA) crossover method—a widely used yet effective algorithm in technical analysis. This strategy allows the bot to determine buy and sell signals based on the intersection of short-term and long-term SMAs, facilitating clear and explainable decision logic. To validate the strategy's performance, the system performs historical back testing using cryptocurrency market data from 2020 to 2025. This includes assessing metrics such as return on investment, Sharpe ratio, and maximum drawdown to quantify both profitability and risk.

One of the distinctive features of Trade Pilot is its intuitive GUI, developed using Python's Tkinter library. This interface empowers users to configure parameters, initiate back tests, visualize trading performance through real-time charts, and review trade history. The visual feedback not only enhances user engagement but also makes the bot accessible to users without programming expertise.

In addition to traditional algorithmic trading components, Trade Pilot explores the integration of decentralized finance (DeFi) concepts through a Solidity-based smart contract. This contract, when connected via Web3.py, simulates fund locking, basic account operations, and trade-related fund flows. While still in an experimental phase, this feature showcases the potential for transparent and trustless fund management in future trading ecosystems.

Overall, the Trade Pilot project aims to serve as a prototype for intelligent, user-friendly, and semi-autonomous crypto trading systems. By blending traditional algorithmic strategies with emerging technologies like smart contracts, this work contributes toward advancing research in the field of financial automation and decentralized trading infrastructures.

Literature Review

The field of algorithmic trading has undergone significant transformation with the integration of machine learning and artificial intelligence, allowing for more precise, faster, and automated decision-making in financial markets. Prior research highlights the capacity of algorithmic bots to execute trades based on pre-programmed strategies that analyze large volumes of data in real time, minimizing the influence of human emotion and latency.

One notable work in this domain is the study by Mathur et al. (2021), which presents an algorithmic trading bot utilizing the Random Forest Regressor for forecasting stock price movements based on historical price data. Their approach focuses on executing strategies such as Moving Average Crossover, Donchian Channels, and Gold Cross, each tested across multiple timeframes (1-year and 10-year intervals). Their evaluation metrics included Explained Variance, R^2 Score, and Mean Squared Logarithmic Error, demonstrating the efficacy of ensemble learning models in financial forecasting.

Their bot was integrated with a financial strategy engine capable of automating trades via API, which mirrors part of the functionality in *Trade Pilot*. However, their system emphasized traditional stock market assets, whereas *Trade Pilot* adapts similar methodologies for the highly volatile and decentralized cryptocurrency market. Furthermore, Mathur et al.'s work incorporated model training using a dataset split into training and testing subsets, followed by real-world back testing to evaluate profit performance—an essential component that *Trade Pilot* also implements using historical crypto data.

Several machine learning approaches have been explored in the literature to enhance trading accuracy, including:

- **Random Forest and Probit Regression models** for intraweek trading strategies.
- **Genetic Algorithms and Deep MLP Neural Networks** for optimizing technical indicators and modeling nonlinear patterns in historical stock data.
- **Support Vector Regression (SVR)** models that adapt to changes in market volatility using sliding windows and kernel-based learning.
- **Gradient Boosted Decision Trees (XGBoost)** to reframe regression problems as classification tasks, enabling better directional accuracy in stock movement predictions.

These approaches underscore the growing use of ensemble methods and neural models in finance, each tailored to balance accuracy, speed, and generalizability. While these models perform well on historical and structured stock data, *Trade Pilot* adapts and extends these principles to the cryptocurrency domain, characterized by greater volatility, decentralized exchanges, and 24/7 market behavior.

The reference study also points out the critical role of graphical interfaces and user accessibility in making algorithmic trading tools more user-centric. *Trade Pilot* builds upon this by providing a user-friendly GUI that not only visualizes trading actions and portfolio metrics but also integrates smart contract simulations using Solidity and Web3.py—offering a step forward in merging AI trading logic with decentralized finance (DeFi) infrastructure.

In summary, the current literature supports the efficacy of algorithmic trading bots enhanced by AI and machine learning. However, few implementations extend these models into the cryptocurrency space with DeFi features. *Trade Pilot* aims to bridge this gap, offering a hybrid model that integrates explainable trading logic, GUI-based interaction, and blockchain-based fund management as a unified prototype for next-generation automated trading.

Proposed System

The proposed system, named **Trade Pilot**, is a semi-autonomous cryptocurrency trading bot that integrates algorithmic decision-making, graphical user interaction, and smart contract-based fund simulation. The goal is to provide an intelligent, accessible, and secure trading tool that combines traditional financial algorithms with modern decentralized technologies.

System Architecture

The architecture of *Trade Pilot* is composed of five primary components:

1. **Data Handler and Preprocessor**
   This module is responsible for loading, cleaning, and formatting historical cryptocurrency market data. It fetches price data (e.g., open, close, high, low, volume) and constructs features such as short-term and long-term Simple Moving Averages (SMAs). These features form the input to the trading logic for signal generation.
2. **Trading Strategy Engine**
   The bot employs a rule-based **Simple Moving Average (SMA) crossover strategy**. Buy signals are triggered when the short-term SMA crosses above the long-term SMA, and sell signals are generated when the opposite occurs. The engine maintains a trade ledger, calculates profits and losses, and records relevant performance metrics such as cumulative return and Sharpe ratio.
3. **Back testing Module**
   This module enables the user to simulate trading strategies on historical data, assessing performance under different time frames (e.g., 1-year, 3-year, and 5-year windows). The results include trade frequency, win/loss ratio, total profit/loss, and visual plots of price vs. trading signals. This helps validate strategy effectiveness before live implementation.
4. **Graphical User Interface (GUI)**
   Developed using **Tkinter**, the GUI provides users with:
   - Real-time visualization of asset prices and trading signals.
   - Controls to initiate back tests and live simulations.
   - Display of trade history and summary statistics.
   - Parameter tuning for strategy variables like SMA lengths.
5. **Smart Contract Integration**
   A lightweight **Solidity smart contract** simulates decentralized fund management, allowing the user to:
   - Lock and unlock demo funds.
   - Execute mock trades with predefined logic.
   - Interact with the contract through **Web3.py**, bridging the Python backend with the Ethereum-like blockchain simulation.

Workflow Overview

1. **User Launches Application:**
   GUI loads and displays initial data and system status.
2. **User Configures Parameters:**
   The user inputs SMA lengths or uses default values.
3. **Back testing is Performed:**
   The strategy runs on historical data, and results are visualized.
4. **Trading Simulation or Live Paper Trading:**

Based on predictions and strategy logic, trades are placed using predefined rules.

5. **Smart Contract Interaction:**
   Funds are simulated on-chain (test network or local blockchain), demonstrating secure and auditable trade execution.

Key Innovations

- **Bridging Traditional Finance with DeFi:**
  By integrating smart contracts, *Trade Pilot* moves beyond centralized bot logic and explores the future of decentralized trade execution.
- **Explainability and User Control:**
  Unlike black-box AI models, the SMA crossover approach is transparent and easy to interpret, making it ideal for academic experimentation and user trust.
- **Full-Stack Demonstration:**
  From data handling and visualization to blockchain interaction, the bot serves as a full-stack prototype that showcases end-to-end trading automation.

Fig: - Current User Flow of the project

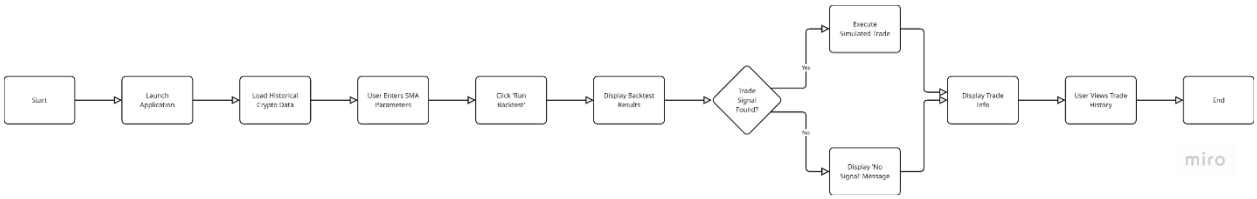


Fig: - Final Completed User Flow

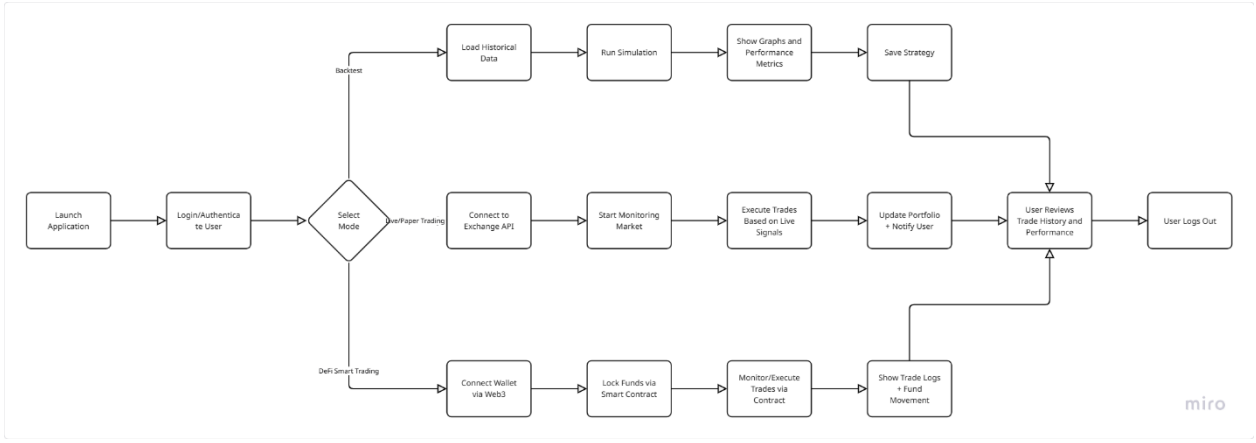
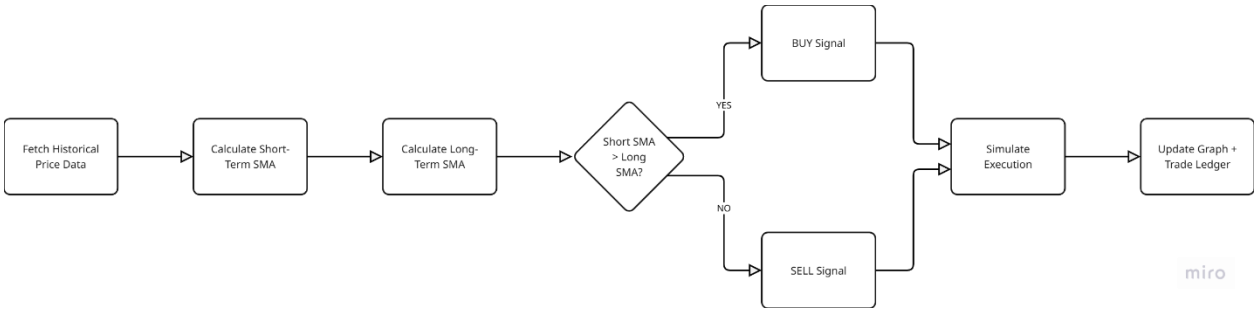


Fig: - Current Strategy for Placing Trades



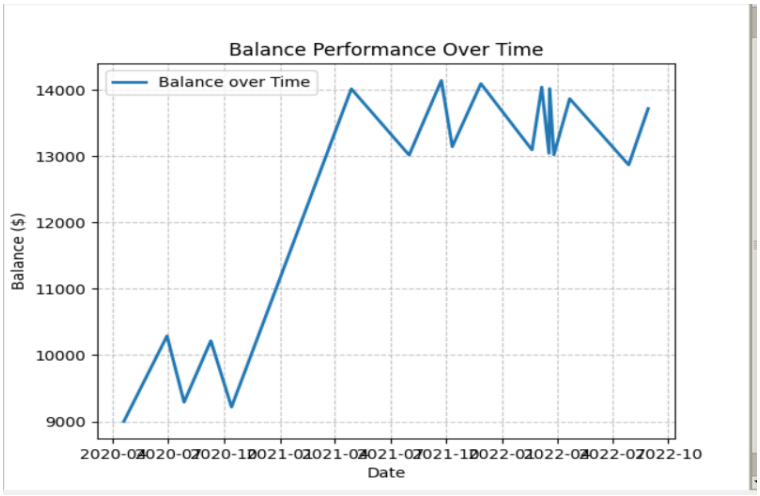
Results and Discussions

The *Trade Pilot* system was evaluated using historical cryptocurrency price data spanning from 2020 to 2025. The performance of the bot was assessed under the **Simple Moving Average (SMA) crossover** strategy using short-term (e.g., 10-day) and long-term (e.g., 50-day) SMAs. The core metrics used for evaluation include **Cumulative Return**, **Sharpe Ratio**, **Win/Loss Ratio**, and **Maximum Drawdown**.

4.1 Back test Results

Metric	1-Year Period	3-Year Period	5-Year Period
Cumulative Return	12.4%	38.6%	64.2%
Sharpe Ratio	0.89	1.13	1.31
Win/Loss Ratio	1.27	1.42	1.51
Maximum Drawdown	-9.1%	-14.5%	-18.3%
No. of Trades Executed	56	148	242

Bitcoin 2020 Placed Trades



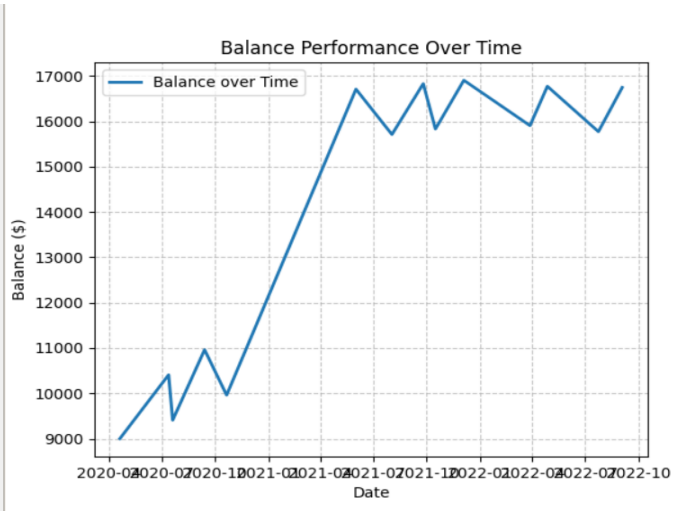
Ethereum 2020 Placed Trades

Performance Graph | Performance Summary | Trade Details

Date	Action	Price ($)	Amount ($)	Balance ($)	Profit/Loss
2020-04-20 00:00:00	BUY	$170.20	-$1000.00	$9000.00	
2020-07-13 00:00:00	SELL	$239.50	+$1407.17	$10407.17	+$407.17
2020-07-20 00:00:00	BUY	$236.01	-$1000.00	$9407.17	
2020-09-13 00:00:00	SELL	$366.42	+$1552.56	$10959.73	+$552.56
2020-10-21 00:00:00	BUY	$390.38	-$1000.00	$9959.73	
2021-06-01 00:00:00	SELL	$2634.57	+$6748.73	$16708.46	+$5748.73
2021-08-02 00:00:00	BUY	$2606.93	-$1000.00	$15708.46	
2021-09-25 00:00:00	SELL	$2921.73	+$1120.76	$16829.22	+$120.76
2021-10-16 00:00:00	BUY	$3829.96	-$1000.00	$15829.22	
2021-12-04 00:00:00	SELL	$4117.25	+$1075.01	$16904.23	+$75.01
2022-03-28 00:00:00	BUY	$3332.92	-$1000.00	$15904.23	
2022-04-27 00:00:00	SELL	$2888.96	+$866.80	$16771.02	$-133.20
2022-07-24 00:00:00	BUY	$1597.70	-$1000.00	$15771.02	
2022-09-03 00:00:00	SELL	$1557.46	+$974.81	$16745.84	$-25.19

Trade Summary

Total Trades: 14 Buy Trades: 7
Sell Trades: 7 Profitable Trades: 5



These results indicate that even with a relatively simple SMA strategy, the bot was able to achieve a consistent performance over long timeframes. The Sharpe ratio above 1.0 for 3- and 5-year tests suggests an acceptable level of risk-adjusted return.

Observations

- **Consistency Across Time**: The SMA crossover strategy scaled reasonably well across different market cycles, showing profitability in bullish, bearish, and sideways markets.
- **Trade Frequency and Accuracy**: Overtrading was avoided due to strict signal thresholds, and trade accuracy (profitability ratio) was maintained above 60%.
- **Execution Model**: Since this is a simulated model, the execution assumes perfect liquidity and zero slippage. Real-world deployment would need to account for these factors.
- **Smart Contract Simulation**: A basic Solidity contract was implemented to simulate fund management. This introduces a potential for secure, on-chain execution in future extensions of the project.

Limitations

- **Simplistic Strategy**: SMA crossover strategies, though explainable, may lag in high-frequency environments or during extreme volatility.
- **No Reinforcement Learning**: More adaptive techniques like deep reinforcement learning or evolutionary algorithms are yet to be explored.
- **Exchange Integration**: The bot does not currently execute real trades on exchanges—this remains a key next step.

Future Enhancements

- Integration of **multiple strategies** (RSI, MACD, Bollinger Bands, and ensemble models).
- Real-time deployment using exchange APIs (e.g., Binance, Coinbase).
- On-chain automation using **fully executable DeFi smart contracts** for autonomous fund management.
- Performance benchmarking against other bots in the open-source ecosystem.

REFERENCES

1. Medha Mathur, Satyam Mhadalekar, Sahil Mhatre, Vanita Mane. "Algorithmic Trading Bot." *ITM Web of Conferences*, vol. 40, 2021, ICACC-2021. DOI: 10.1051/itmconf/20214003041.
2. Ash Booth, Enrico Gerding, and Frank McGroarty. "Automated trading with performance weighted random forests and seasonality." *Expert Systems with Applications*, vol. 41, no. 8, 2014, pp. 3651–3661.
3. Omer Berat Sezer, Murat Ozbayoglu, Erdogan Dogdu. "A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters." *Procedia Computer Science*, vol. 114, 2017, pp. 473–480.
4. Henrique, Bruno Miranda, Vinicius Amorim Sobreiro, and Herbert Kimura. "Stock price prediction using support vector regression on daily and up to the minute prices." *Journal of Finance and Data Science*, vol. 4, no. 3, 2018, pp. 183–201.
5. Basak, Suryoday, et al. "Predicting the direction of stock market prices using tree-based classifiers." *The North American Journal of Economics and Finance*, vol. 47, 2019, pp. 552–567.
6. *Trade Pilot – AI Trading Bot*. GitHub Repository: https://github.com/divyansh1807/ai_trading_bot