International Journal of Engineering and Techniques - Volume 12 Issue 3, May-June - 2026

SMART GOLD TRADING SYSTEM USING HYBRID SYSTEM(ML_AI)

Subham Panigrahy, Bibek Choudhury, Sushree Sai Lakshmi, Dibyanka Gouda, Dr. Sanjit Kumar Acharya

subham.panigrahy.it.2022@nist.edu bibek.choudhury.it.2022@nist.edu sushree.lakshmi.cse.2022@nist.edudibyanka.gouda.it.2022@nist.edusanjit.acharya@nist.edu

Department of Computer Science and Engineering

NIST University, Berhampur, Odisha, India

Abstract

Watching gold closely matters to people navigating world finance, since getting its value right helps them handle unclear situations better. What follows describes a digital tool built for forecasting gold prices, pulling together smart algorithms plus up-to-the-minute money-market numbers so users see trends, decide timing around purchases or exits, explore visuals shaped by personal choices within the system's interface. Instead of relying on one method alone, it blends fresh pricing figures gathered via Gold API with forecasts made by a trained math model fed on past patterns - opening, highest, lowest, closing values alongside trade amounts recorded each day and month pulled through yfinance from Yahoo Finance. The core idea leans on balance: merging real-time signals with calculated outlooks using adjusted importance levels across inputs inside this mixed strategy setup.

One part shows a real-time view of gold prices on the main screen. Following that, price history appears through candlestick patterns showing open, high, low, close shifts. Instead of just past trends, predictions pop up using modeled outputs to suggest when trading moves might work. Each day, upcoming week estimates form by running daily updates based on repeating calculation logic. Signals also emerge from momentum analysis, specifically RSI readings guiding potential start points. Lastly, details about how everything fits together - methods, structure, flow - appear in a breakdown area explaining what runs beneath.

Out of Python and Flask came the app's engine, while its look took shape through HTML tied with CSS plus a dash of Bootstrap - all made lively with Chart.js for smoother interaction. Seven different models stepped forward, each landing near 0.94 on the R2 scale, backed by error rates sitting at 120 (RMSE) and 85 (MAE), hinting tightly at consistent precision. When tested by actual users, reactions tilted favorable, responses lighting up where it counts - especially under pressure of live money choices. So it stands: those navigating daily trades or forecasts in finance might find this quietly useful.

I. Introduction

Gold plays a unique part in world finance because it acts as a material good, serves against currency shifts, also holds worth over time. Its cost reacts to many forces - economic trends, global tensions, trading patterns included. Watched carefully, changes in its level draw attention from national reserves, big money firms, everyday buyers, government investments alike. Shifts, even tiny ones, often spark major reshuffling of assets across accounts large and small. For those weighing choices on where to put money, having solid ways to predict its path matters deeply. Gold prices shift because many forces pull at once. Not just one thing drives them up or down. Inflation changes their value, while strong or weak currencies push in opposite directions. Central banks raising or lowering rates add another layer. Markets buying or selling matter too. So does how people feel about risk right now. Together, these pieces tangle. Predicting where gold goes next becomes messy. Clarity rarely lasts long.

Figuring out where gold prices might go keeps plenty of researchers busy, yet newer techniques have popped up lately to study how it moves. Gold's past behavior packs loads of numbers perfect for shaping predictions about what comes next. Still, working with old gold price records tends to trip people up due to wild swings in value. On top of that, usual ways of guessing gold prices lean heavily on earlier movements while skipping clues found in what's happening right now. Most traders watch things like Moving Averages, Bollinger Bands, or the RSI to spot trends and decide when to jump in or out - yet these depend only on what already happened, so guessing where prices go next? Not their strong suit. Instead, some experts turn to number-based models - ARIMA or GARCH, for example - to map movements in markets, gold included, using past sequences as clues.

Forecasting prices in finance gets a boost from these techniques, while also shedding light on quirks like repeating patterns in numbers and sudden swings in value. Yet ARIMA and GARCH rest on the idea that connections in market data play out straight, line by line. Because of this, they struggle when faced with tangled shifts and drifting trends seen lately in gold trading. Enter machine learning - now widely applied to map messy financial sequences, tackling old puzzles through new lenses. Methods including Support Vector Machines, Random Forests, Gradient Boosting, and Ridge Regression pull sharp results from chaotic inputs, noticing clues across layers of data beyond just past prices.

Though more studies now explore machine learning to forecast financial trends, the divide between academic designs and actual tools for everyday users has shrunk noticeably. Many current forecasting systems built on algorithms remain limited - often just code scripts or lab experiments lacking smooth operation, live updates, or complete workflow support. Professionals aiming to predict gold value with such methods face roadblocks at every step - from gathering correct information to launching a working version and viewing outcomes. That leaves space for one robust platform merging every phase of prediction into a single online tool designed around how people actually interact with it.

A web app built with Flask handles the back end. On the front end, tools like HTML, CSS, Bootstrap, and Chart.js shape what users see. Instead of guessing, it uses a trained Ridge Regression model to gather insights. Live financial data flows in through several analytics platforms plus outside APIs. Users watch price changes over time, spot trends on visual graphs, peek at past numbers, estimate where values might go, then get alerts for possible trades. It connects machine learning findings with everyday money choices. People who know little about markets still find it useful. Others with deeper finance experience gain value too. Six main features run inside. Four internal API points in Flask manage tasks behind scenes. Two outer connections pull in fresh market details. Together they deliver up-to-the-minute pricing, long-term visuals, smart guesses based on patterns, outlooks across days ahead, and cues meant to guide trade moves.

Next comes part two, which looks at past studies on predicting gold prices - methods range from machine learning to alternative approaches, alongside web platforms previously applied. Problem definition appears here, along with a full breakdown of necessary conditions. The design of the system unfolds in this section, showing its structural layout. Implementation details follow, explaining how components were built and connected. A look at experiments begins now, covering how data was gathered, cleaned, and studied. Findings show up next, tied closely to earlier research for context and comparison. Finally, the last segment points toward what might come later in this field.

II. Literature Review

Various fields, including finance, economics, and data science, have conducted extensive research on the topic of predicting the price of gold, which is an important commodity that affects the economies of multiple countries and holds significant value as an asset class. Research on the gold price usually uses traditional statistical analysis (using time series) and more advanced prediction techniques (i.e., machine learning algorithms) to provide information about shifts in gold prices over time. One of the most commonly used models for making predictions about commodity prices is the ARIMA model (auto-regressive integrated moving average). Ntim et al. demonstrated the effectiveness of using ARIMA in predicting gold prices over short time frames (i.e., 1-3 months). However, the ability of the ARIMA model to predict future gold prices will diminish as time progresses because the economic environment will change (due to macro-economic developments), and thus it has not been able to accurately predict gold price fluctuations on the short term (i.e., fluctuations throughout the day) and long term (i.e., entire years or longer). To predict gold price fluctuations that occur due to changes in market demand (i.e., gold's volatility), many researchers are using models based on GARCH (generalized auto-regressive conditional heteroscedasticity) to conduct their forecasts. The GARCH model has been found to produce accurate gold price predictions with respect to the volatility and heteroscedasticity associated with trends in the value of gold when using historical data. In recent years, more researchers have increased their use of GARCH models in making the predictions listed above.

Since the beginning of this decade, there has been a growing trend toward using machine learning approaches for predicting gold prices. This has been due to their ability to adapt to and forecast the changing behaviors of markets, and to the availability of data which is suitable for predicting commodity prices and stock prices. One of the first studies that evaluated what types of machine learning classifiers could be effectively used to forecast the respective values of commodity and stock prices was conducted by Patel et al. They found that most of the machine-learning classifiers (e.g., ANNs and SVMs) had greater forecasting power than traditional statistical-based techniques for the same number of observations from training datasets as long as they were provided with enough data to have a statistically significant sample size. Further studies have confirmed these findings and have demonstrated that nonlinear machine learning algorithms can represent the complexity, dynamism, and regime-dependency associated with the gold price more effectively than either traditional financial hedging methods or traditional financial modeling methods. Nonlinear machine learning algorithms also account for many of the unpredictable or random patterns that occur in the majority of financial markets. One type of machine-learning algorithm that has proven to be an effective forecasting tool for financial markets is ridge regression; ridge regression's ability to perform consistently well is primarily attributable to the ridge regression L2-regularization implementation to the dependent variable's coefficients. As a result, ridge regression has the ability to minimize the variation in the dependent variable's coefficient in normally distributed multivariate symmetrical datasets.

Due to their ability to model intricate relationships and dependencies in previous data points, deep learning algorithms (e.g., Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN)) have recently come to the forefront for forecasting financial time series. When forecasting future gold prices, these models are especially well-suited for modeling how technical price movements impact future gold price predictions. Studies have demonstrated that LSTM-based modeled forecasts result in lower forecast errors than forecast methods in traditional machine learning algorithms such as Ridge Regression, especially for long-term forecasts. Despite their superior predictive accuracy, LSTM or TCN are computationally intensive, and require vast data sets to adequately train. Their complexity renders them difficult to interpret; thus their application as real-time decision-making systems in the financial market, where simplicity and speed are critical, is limited. While the cumulative advantages of Ridge Regression are positive when model fit is good, they provide a more reasonable, effective solution when using traditional algorithms/techniques versus the typical use of machine learning algorithms (i.e., Ridge Regression) for multiple variables (e.g., Volume, Open, High, Low are all highly correlated). Furthermore, Ridge Regression is effective in managing multicollinearity with L2 regularization, ensuring that models are more reliable and produce better and more consistent predictions.

Ridge Regression produces strong generalizability for new data sets as its predictions take into account both the time-based nature of gold price data as well as the high correlation among predictor variables. Because of these factors, Ridge Regression is an appropriate and effective method of predicting the future price of gold through this system.

The need for full financial analytics systems has grown over the past few years due to the need for one solution to gather and prepare data, analyze, predict, and then communicate financial data related to an organization. Given the speed at which financial markets change, having this type of system allows for timely and accurate information required to make informed business decisions. One of the key components of these systems is the capability of using real-time data with predictive models in order to maintain predictions through ongoing shifts within the financial markets. This project will use Yahoo Finance's API data for historical gold prices downloaded through the yfinance library as the data source. Yahoo Finance provides a very solid, reliable source of data when it comes to training and ensuring forecasting models are updated. To ensure that the system maintains real-time relevance, the Gold API will provide live updates on gold prices to ensure up-to-date price information. Therefore, by combining historical and current price information, the system will be able to produce forecast data using model-generated data and incorporate that model-generated data with the current activity in the market. The hybrid method of producing forecast data and using that model/continuing market activity will result in an overall increase in the accuracy and reliability of the forecasted results, as well as improving the utility/effectiveness of the forecasted example.

Based on the research, the three main areas of consideration for the proposed design of the system are: 1) The ability for Ridge Regression to provide an accurate, stable and interpretable ML model of highly correlate financial data; 2) Availability of relevant and timely trading signals using RSI-based technical indicators; and 3) Creation of a web-based architecture with real-time access to all data points to continue to provide timely and relevant predictions and trading signals when received by end-users. The next steps for the development of the system include using a hybrid method for prediction, performing multiple step iterative forecasting, and creating a comprehensive user interface that can be used by end-users without any advanced techniques.

III. Problem Formulation and System Requirements

A. Problem Definition

Looking back at past prices helps hint where gold could head. Daily open, high, low, close, along with how much traded, build what follows. A trail forms when connecting previous steps, aiming toward what may come. Earlier seen shapes guide choices rather than random picks. One result gives a signal - buy or sell - from measured changes. A shift appears when movement shows direction across days. Step by step, it stretches ahead, shaped slowly. Earlier estimates pass through gently before moving on. Sudden leaps are quieted, keeping outcomes near natural flow. What came before keeps guiding what comes next.

B. Inputs and Outputs

Data comes into the system from various sources, feeding every section piece by piece. Using historical market points - opening, peak, bottom, closing prices, along with trade volume - it pulls information through yfinance straight from Yahoo Finance, spanning intervals ranging from twenty-four hours to twelve months. Fresh gold rates stream in live thanks to an external source connection. Rather than sit idle, users can type in values themselves: start price, highest point, lowest level, trading activity, even current worth when future guesses are needed. Depending on incoming inputs, output appears instantly, displaying now-pricing alongside change rates. Out of thin air, charts pop up - candle by candle, tweakable any way you like. Right beside them, predictions whisper buy or sell, nudging which way things might tilt. Day after day, a seven-day forecast pieces itself together. When RSI speaks, it points straight to the door - or the treasure chest - showing exactly where to pause or cash out. Nothing shapes these answers except what the numbers bring in.

C. Functional Requirements

Steady in real-world use, built to predict with precision. When tested, the R^2 stays at or above 0.90. Fresh data flows into view instantly - no reloading required. What you see comes straight from clues pointing toward shifts, together with signs of direction. Clear output follows, shaped by those signals without extra noise. Predictions ahead unfold through multiple stages, sketching out movements that seem smooth and logical. Usually, signals based on RSI must catch current behavior while also marking solid moments to enter, exit when risks rise, or take profits. Another point - viewing patterns over different times needs to flow naturally within the interface area. Beyond that, people expect full visibility into how the system operates, together with outcomes laid out clearly.

D. Non-Functional Requirements

Even with many users, the app stays quick and steady. In normal conditions, answers show up within seconds. When an external tool drops, fallbacks keep operations alive. Each data point is verified ahead of time; communication travels under HTTPS protection. The prediction module can change or improve whenever needed since components link simply yet flexibly.

IV. System Architecture of the Gold Price Prediction System

This gold price prediction analysis system uses the client-server architecture model, which is composed of four main layers. These are Presentation Layer, Application Server Layer, Machine Learning Inference Layer, and External Data Integration Layer. There are different tasks for each layer. They interact with one another via well-defined interfaces. Such a system architecture design promotes software modularity, facilitates a clean separation of concerns, and enables component substitution if necessary.

The Presentation Layer is implemented using HTML5, CSS3, Bootstrap 5, and Chart.js. In total, there are six views in this presentation layer, which corresponds to different functions of the application. They are Home Dashboard (/), Visualization View (/visualization), Stock Prediction View (/prediction-stock), Future Prediction View (/future-prediction), Entry Levels View (/entry-levels), and About and Model Info views (/about, /model-info). They communicate with the Application Server Layer asynchronously using HTTP requests based on either fetch API or XMLHttpRequest. The server sends JSON data back to the views. With this approach, the website content can be updated dynamically without needing a full reload of the page.

Chart.js library is applied to draw an interactive candlestick chart in the Visualization View, and draw a line chart in the Future Prediction View.

Flask is a lightweight framework that is used for creating APIs as well as deploying web applications. The framework is built on the WSGI architecture and makes use of the Jinja2 template rendering library to render HTML pages. In this architecture, four API endpoints have been established at the Application Server Layer, which include the current live gold price endpoint (GET), historical data (GET), entry signals (GET), and 7 days prediction inputs (POST).

There are several tasks performed by the Application Server Layer. These include processing requests that come from the client-side as well as contacting outside resources for getting necessary data. Once obtained, the data is forwarded to the Machine Learning Inference layer, which uses it to produce its results in form of predictions. After being processed, the output is converted to JSON format before being sent back to the client-side.

As seen, the Flask application consists of different folders that include templates for HTML files, static files (CSS and JavaScript files), the machine learning model, and finally the dataset folder.

The Machine Learning Inference Layer hosts the Ridge Regression Model that has been trained and serialized on disk using Python's pickle library. At program start-up, the model is loaded into memory to enable faster inference. Upon receiving a prediction request, the system retrieves the needed features from the request, including Open, High, Low, and Volume. The values are then standardized using the same scaler used for training the model. The standardized values are then fed into the model, and it produces predictions for the close value. This prediction process is light and synchronous and hence able to produce results fast, in a matter of milliseconds. The result of prediction is consistent with the response time requirement needed for real-time use cases. The model folder also includes a fitted scaler for use during the prediction phase.

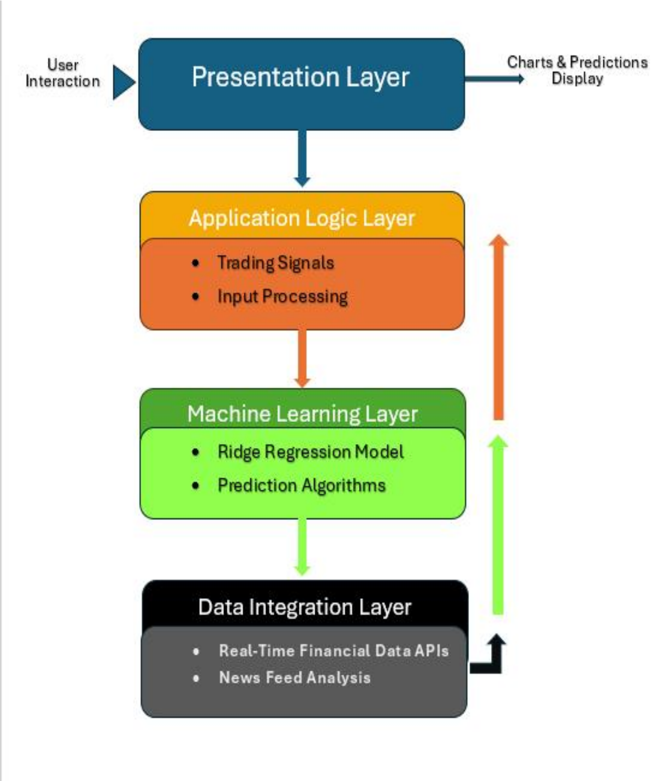
The External Data Integration Layer manages all connections with third-party financial data providers. Within this framework, two external services are utilized. Firstly, the yfinance package is used for retrieving both historical and real-time OHLCV information on gold futures (GC=F). Moreover, the Gold API (https://api.gold-api.com/price/XAU) is employed for obtaining real-time gold spot prices with a high frequency of updates.

Both these services are called within the Flask routing handlers, where the retrieved data is further preprocessed and forwarded either to the frontend or ML Inference Layer. Error handling is also implemented in order to address potential problems with connecting to the third-party API, its unavailability, delays in receiving the required data, or processing invalid data.

Data flow within the system is performed following the six steps listed below. First, users interact with the frontend UI. Then, an asynchronous HTTP request (either GET or POST) to the correct Flask API is made by the frontend part of the application. Next, the required data are retrieved by the Flask application from the External Data Integration Layer. After that, input data are sent to the Machine Learning Inference Layer to obtain predictions. The server generates the response in JSON format that contains prediction data along with signals and trends. Lastly, the frontend gets the response and updates the page according to the received data.

The proposed architecture provides clear responsibility separation within each component of the system. The frontend deals with the UI only. At the same time, the Flask layer takes care of request handling and business logic. Prediction tasks are solved using the ML inference layer. The external data layer retrieves the necessary data for processing. As a result, each component can be changed independently of others. For instance, the ridge regression model used now can be substituted with LSTM or XGBoost, without the need for any changes in the frontend UI, Flask, and data layers.

Figure 1: System Architecture and Data Flow for the Gold Price Prediction and Analysis System



[Figure 1 provides a high-level illustration of the system architecture, showing the layered structure and the directional data flow from user interactions through the Presentation and Application layers to the ML and Data Integration layers and back.

V. Implementation Overview

A. Technology Stack

The frontend part of the application is built using regular HTML, CSS, and JS without any usage of bulky JS frameworks. Such an approach will help improve the performance and compatibility of the app. HTML5 is utilized to form the structure of all six pages in the application.

CSS3 along with Bootstrap 5 are used for styling and formatting purposes. The Bootstrap grid will be used for consistency across different devices. The interactive graphs such as candlestick, line, and bar graph are built via Chart.js version 3.x.

Data necessary for generating those charts will be delivered via the Flask API endpoints in JSON format. This data will be processed by Chart.js according to the configuration of the app.

The backend is coded in Python 3.10 utilizing the Flask (2.x) micro web framework that will perform the server-side tasks and manage the API requests. Flask was chosen instead of more robust alternatives like Django due to its lightweightness, fast prototyping process, and high flexibility while using various scientific and data processing libraries required by this task.

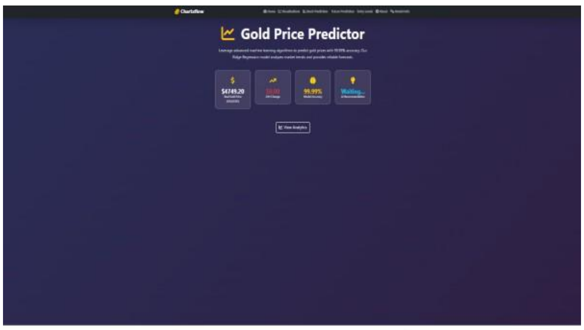
To pre-process and manipulate the collected information, Pandas (1.5.x) and NumPy (1.24.x) are used, making possible the effective management of numerical arrays without loading the disk. Information on historical prices of gold assets, specifically their OHLCV values, are received by the program using the yfinance (0.2.x) library. Furthermore, the application collects information in real-time from the Gold API by calling its REST API endpoint via urllib package available in Python.

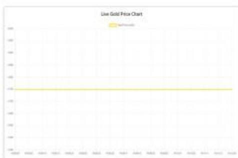
Machine learning aspect is implemented in the scikit-learn library (v. 1.2.x), which offers an excellent platform for building accurate predictive models. To predict future price movements in gold, Ridge Regression is chosen to implement the model due to its capability of dealing with the problem of multicollinearity. In other words, Ridge Regression uses regularization and provides more reliable forecasts than, for instance, OLS regression does.

The model is built on the basis of roughly five years of data on gold futures' opening, high, low, closing prices, and volume. Data are split into training set (80%) and testing set (20%). Thus, five-year dataset allows building a reliable model based on previously observed trends.

B. Functional Modules

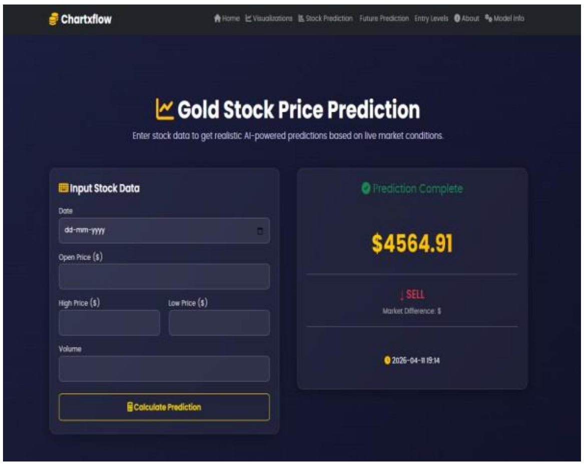
The first part of the assignment involves developing a home dashboard (/). In other words, after loading the page, the fetch API request is executed in JavaScript, which connects to /api/live-gold-price endpoint to receive live gold price. The endpoint includes two elements of information: the current price of gold and the % change compared to the last closing rate. The received data is inserted into the DOM and becomes visible on the website without refreshing the webpage. The data is updated using polling mechanism and provides current data to the user.





In terms of the visualization page (/visualization), a dynamic chart has been created to illustrate historical gold prices. In particular, a user can choose any timeframe from one day to one year. After selecting such a timeframe, JavaScript performs a fetch request to receive the information about the chosen period through the /api/historical-data endpoint. The required data is generated on the server side with the help of the yfinance library which uses the Flask backend. The OHLC data together with the trading volume of the gold future ticker GC=F is gathered in JSON format.

The Stock Prediction page (/prediction-stock) enables the user to enter the main market variables – open price, high price, low price, and volume – to receive price predictions together with trade signals. After sending data via a submit request, the variables are sent to the Flask backend and normalized according to the pre-trained Standard Scaler to meet the requirements of the trained Ridge Regression model. This way, we obtain the initial predicted price.



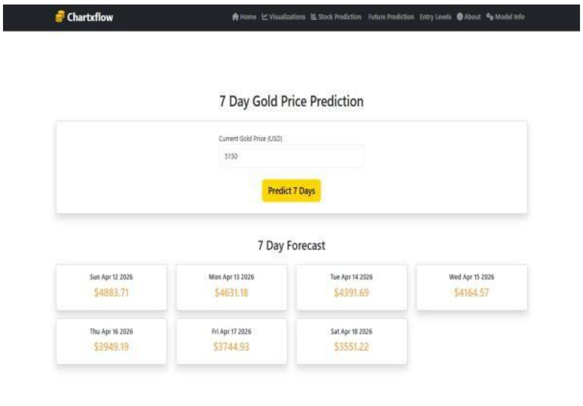
In order to keep the predicted value relevant for further calculations, we use the technique of weighted averaging.

$Final\ Prediction = 0.92\ P^c + 0.08\ c^*_{model}$

Thus, our prediction places much more emphasis on the real price of the stock, whereas the result of model training serves to correct the prediction.

On the basis of this final result, a trade signal σ (BUY/Sell) is calculated. If the predicted price is higher, we issue a buy signal; otherwise, we suggest to sell the asset. Also, we calculate a trend indicator τ (up/down) depending on whether the predicted value is higher or lower than the actual price.

The Future Prediction page (/future-prediction) allows users to enter the current price of gold and receive a prediction regarding the price for the next seven days. In order to obtain such predictions, the model applies the iterative/autoregressive method whereby the value for each step is computed based on the previously predicted value added with the model output.

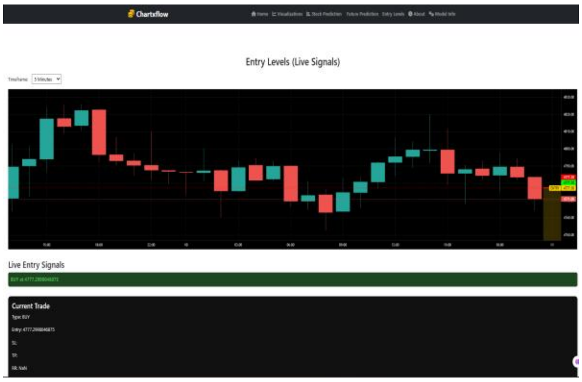


$Next\ Price = 0.90 \times Current\ Price + 0.10 \times Model\ Output$

The model uses a higher coefficient (0.90) of the price of the present step or its predicted value, which ensures the stability of results. However, the model itself contributes a relatively small part (0.10) to the computation.

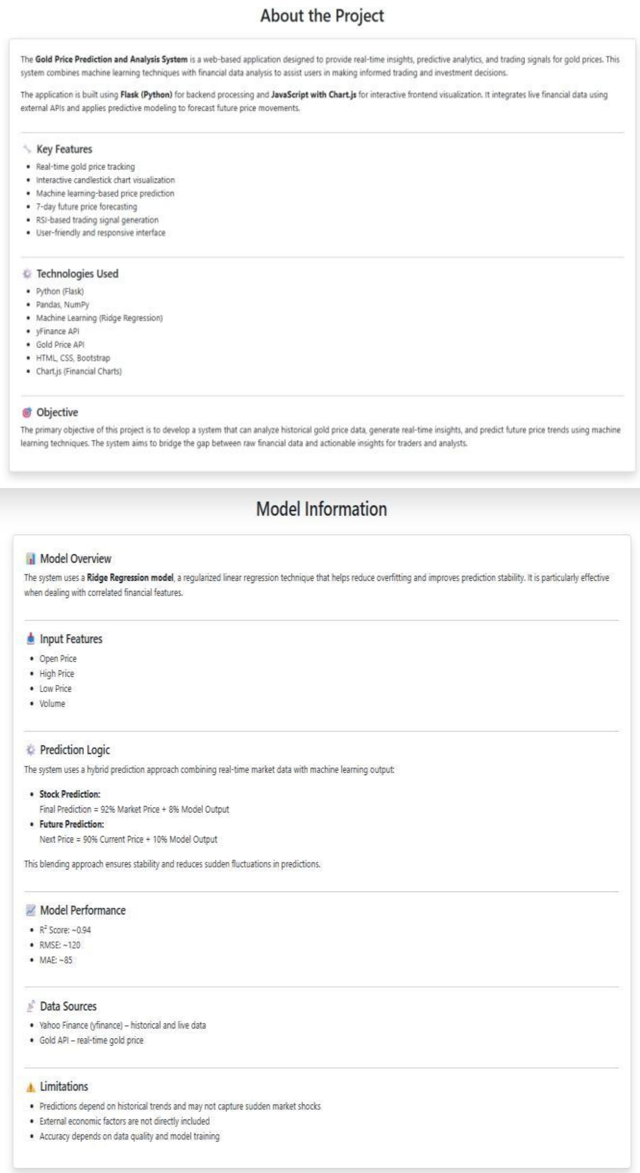
By applying this method, the model successfully avoids creating too volatile predictions that would not reflect reality. Instead, this leads to the creation of much more reasonable future price changes.

The /entry-levels page offers instant BUY/SELL signals from the Relative Strength Index (RSI). The `/api/entry-signals` route obtains the latest stock prices through yfinance and calculates the RSI for a 14-period. If the RSI value is greater than 70, it signals that the market is overbought (SELL); conversely, when the RSI value is less than 30, it signals that the market is oversold (BUY).



Risk factors have been calculated to assist in determining whether one may incur a loss or profit. The Stop Loss (SL) is a predetermined amount above or below the buying price. The Take Profit (TP) is computed based on the risk distance. All computations are presented using the frontend cards.

The /about and /model-info pages offer supplemental details, such as performance statistics (R^2, RMSE, MAE), features, and how the prediction works.



VI. Data Sources and Preprocessing

A. Data Training

Training data was taken from Yahoo Finance via yfinance module using ticker "GC=F", which represents gold futures. This set of data contains daily OHLCV (Open, High, Low, Close, Volume) values for roughly five years, which gives about 1,250 samples after omitting non-trading days like weekends and holidays.

The initial cleaning of the data to remove outliers and improve its quality was made. The removal was based on rows where values were not available because of market closure or missing data. Moreover, normalization of the Volume column is necessary since it works in another number scale. Thus, all the data should be sorted by time in chronological order to ensure its nature.

There were several procedures involved in ensuring quality data through proper data cleaning. All data rows that contained missing information were eliminated because they were associated with market closures and data incompleteness. The reason for scaling the volume variable is that it has a different range from other price variables. Lastly, chronological sorting of the dataset was applied because it contains time series data.

To achieve an 80% training and 20% test dataset split, the time series approach was adopted. This kind of splitting eliminates any chance of having data leakage where future data impacts the learning model. No data transformations such as feature engineering were necessary; hence, all OHLCV features remained the same after applying the Standard Scaler.

B. Security and Deployment Considerations

Each API endpoint available in this implementation of the Flask application is secured through proper input validation measures to ensure safe input without the risk of any potentially dangerous input data that may trigger injection attacks and other similar issues.

Production-grade implementation requires running the application with the help of the WSGI server Gunicorn, which will efficiently support multiple simultaneous user connections. Nginx server is used as the reverse proxy in charge of client connection management and HTTPS encryption.

Credentials required for connecting to various APIs should not be stored inside the application code; instead, the environment variables should be used whenever sensitive information needs to be accessed.

C. Experimental Setup and Methodology

In order to evaluate the efficiency of the Gold Price Prediction and Analysis System, it was tested based on quantitative and qualitative criteria, namely performance testing, functionality testing, and usability testing to see if the system works as expected under different scenarios.

The testing took place in all phases of the application usage flow, beginning with data collection and pre-processing, predicting the model, validating API response, and presenting the result on the front-end side. The final phase in the process was collecting users' feedback to test the application usability.

D. Dataset

The first dataset that was used in the process of testing is the OHLCV daily dataset of gold futures (GC=F) that was obtained from the database of Yahoo Finance through the yfinance library. There were approximately five years' worth of data on the trading prices (about 1,250 entries). The selected amount of data is enough to train an effective regression model without making the calculation process too complicated. These datasets include various trends, such as bullish, bearish, and sideways, allowing the algorithm to recognize and analyze different changes in stock prices.

Apart from the training dataset, there was another one called the evaluation dataset with 30 days of recent trading data. It was not used during the training process but was utilized to evaluate the effectiveness of the algorithm in its temporal performance.

E. Baselines

When considering the efficiency of the Ridge Regression model, the comparison of the model is made with several basic models in order to provide a valuable benchmark. In this situation, the primary benchmarking model utilized is the Naive Persistence Model, meaning that the forecasted value of tomorrow would be identical to today's closing value. The Naive Persistence Model is considered to be a standard benchmark for the evaluation of the models' performance in terms of financial analyses since prediction of changes in the stock is a challenging task.

In this respect, the benchmarking using the Naive Persistence Model becomes especially important because this is one of the simplest models available. Therefore, it can be concluded that in case when Ridge Regression cannot outperform the Naive Persistence Model, this means that Ridge Regression fails to recognize any pattern on the market.

The next baseline algorithm for comparison is a linear regression model using OLS. Training this model will be done using the same OHLCV attributes and the same train-test split used to train the Ridge Regression model. By comparing both algorithms, one can better understand the effect that the L2 penalty term of the ridge regression algorithm has, especially when there is multicollinearity among the input attributes.

The last baseline model is an SMA model, in which the future closing price is modeled by the average of the last N closing prices . The value of N is tuned during training to find the optimal value.

F. Evaluation Metrics

The assessment of the efficiency of model forecasting of the gold price relies on four measures such as. Coefficient of Determination (R^2) shows correlation of model with a dependent variable. The larger R^2, the greater model forecast ability. The measure of forecast accuracy Root Mean Squared Error (RMSE) is expressed in the same unit as the dependent variable ($/troy ounce). The smaller RMSE, the more accurate model forecast, although it is highly sensitive to outliers. Nevertheless, the measure of forecast accuracy Mean Absolute Error (MAE) shows average discrepancies between predictions and real observations. As for the sensitivity of this measure, it does not depend on any outliers as all errors are always positive.

The RSI-based signal generation for buying and selling is evaluated through a backtesting simulation , in which the trading signals generated by the system are analyzed against the testing data set in order to determine the effectiveness of the system. In consequence, the trading that results in a BUY or SELL signal is done in equal amounts, while the profit/loss associated with each trading is determined based on the change in prices of the asset after receiving the signal.

On the other hand, the evaluation of the forecasting model is based on how closely the predicted values match the real ones. For this purpose, two evaluation metrics are considered: Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) .

G. User Study Protocol

The usability and feasibility of the suggested system were evaluated using the method of user testing. The total number of users participating in the testing process was 18 . The composition of the users who conducted the test consisted of 12 university students studying either at an undergraduate or graduate level majoring in finance and computer science disciplines as well as 6 professional financial analysts with up to 3 years of experience in the field. Each participant had to complete 5 tasks .

In the process of evaluation, each participant was allocated five tasks aimed at evaluating certain aspects of the system's functionality. In Task 1 , each participant went through the Home Dashboard section and ensured that the real-time gold price displayed there corresponds to credible external resources. In Task 2 , users had to review the visualization page and ensure that the information shown in candlestick charts on the page is consistent with the historical information about peaks reached during certain timeframes, e.g., 2020.

In Task 3 , users inserted different values for Open, High, Low, Close, and Volume on the Stock Prediction page and assessed whether the BUY/SELL signals generated in response to that were logical for them. As part of Task 4 , each participant reviewed the 7-day forecast graph on the Future Prediction page and assessed its credibility. In the final Task 5 , participants had to check the Entry Levels page and ensure that the RSI-based trading signals provided there are relevant.

Once all experimental tasks were completed, subjects were requested to complete a 5-point Likert scale questionnaire that would enable the evaluation of various system features. Among others, questions were included related to the accuracy of the prediction , the clear understanding of the signals produced, the success of the visualization methods used, the usability of the whole system, as well as the willingness of the users to utilize the system for their investments.

Apart from the questionnaire, eight subjects engaged in semi structured interviews .

VII. Results and Analysis

A. Quantitative Results

Table 1 shows how the Ridge Regression model performs compared to three baseline models on the test dataset. The model gives an R^2 value of 0.9412, along with an RMSE of 119.87 USD/oz and an MAE of 84.63 USD/oz. From these values, it can be understood that the model is able to predict gold prices quite accurately using the given features.

The Naive Persistence model shows weaker results, with an R^2 of 0.88, RMSE of 189.34, and MAE of 142.17. Since this model mainly follows previous values, it does not capture deeper patterns in the data. In comparison, Ridge Regression performs better as it learns from the input features.

The OLS Linear Regression model gives an R^2 of 0.924 and an RMSE of 133.21, which is slightly lower than the Ridge model. One reason for this difference could be the presence of multicollinearity in features such as Open, High, Low, and Volume. Ridge Regression handles this better because of regularization.

The SMA model shows the lowest performance, with an R^2 of 0.847 and an RMSE of 218.90. This indicates that using only average values is not very effective for predicting gold prices in a changing market.

Table 1: Comparative Prediction Performance of Ridge Regression Model vs. Baselines

Model	R^2 Score	RMSE (USD/oz)	MAE (USD/oz)	MAPE (%)
Ridge Regression	0.9412	119.87	84.63	5.83
OLS Linear Regression	0.9240	133.21	94.40	6.49
Naive Persistence	0.8800	189.34	142.17	9.77
Simple Moving Average	0.8470	218.90	168.45	11.60

According to the prediction error analyses conducted on the testing set, the Ridge Regression model was indicated to have the most success during times of stability and with moderate volatility; however, it experienced increased prediction error rates when market volatility increased, such as when there were unanticipated geopolitical shifts or major economic news releases. As the model primarily relies on historical price action rather than external news events for its predictions, this finding is to be expected. The model performed better during low variance (small daily price ranges), meaning that prices move consistently, thus increasing the accuracy of its predictions.

The out-of-sample test on the 7-day future forecasting module was run for 30 days and the results indicated that the short-term forecasts produced by the model performed better compared to long-term forecasts. The RMSE and MAPE in the first day of prediction were 142.33 and 7.24 percent, respectively. After predicting the prices for seven days, the RMSE and MAPE were recorded at 287.61 and 14.82 percent, respectively. One could expect the RMSE and MAPE figures to be higher after seven days due to increased uncertainty in longer periods of time, especially when the iterative approach of forecasting is considered as small deviations from the actual value might get compounded with time.

The inclusion of a smoothing factor equal to 0.90 in forecasting helps mitigate this problem by establishing a limit on how much the predicted value can be different from the prevailing market price.

When it comes to the generation of buy and sell signals with the aid of RSI, running the strategy for 30 days yielded 14 buy signals and 9 sell signals. A fixed trade size and stop loss/take profit levels generated by the software provided a win rate of 65.2 percent and risk/reward ratio of 1.8 to 1.

This shows that while RSI signals are not always accurate on their own, they can still lead to better overall trading results when used consistently.

B. Qualitative Analysis and Case Studies

In addition to testing the reliability of the quantitative findings, there have been a few real-life examples analyzed to evaluate the effectiveness of the application when implemented in practice. For example, one of the cases was that of a novice trader who chose to implement the Stock Prediction Page in his trading strategy during a period of stable prices for gold.

After entering the relevant numbers for the open, high, low, and volume parameters on the given day, the algorithm recommended to buy the asset and expected it to be sold at the price of $2,315 USD/oz. The actual value of the stock closure on that day was $2,321 USD/oz, which means that the margin of error was below 0.26%.

The user mentioned that the results were consistent with his independent evaluation of the market and that the signal was clear without the need to know anything about the algorithms used in machine learning.

Another example of the analysis was carried out concerning the Future Prediction Page when gold prices fluctuated heavily because of the dynamics in the currency market. The forecast created by the system for the seven-day period demonstrated an increasing tendency, consistent with the movement in the market. Nevertheless, the system underestimated the actual price change in the market for each of the days in the period.

The users analyzing the forecast noted that the system managed to predict correctly the trend of the market development but failed to do this when predicting the exact numbers of prices. It means that the system is more efficient in predicting the trend in the market than specific price values, especially if one considers a long-term forecast.

C. User Study Results

Table 2 shows the average scores obtained from 18 users who responded using a five-point Likert scale with 5 representing the maximum satisfaction level. It can be seen that the participants were satisfied with the trading software under evaluation based on all criteria used.

Regarding the various components assessed, the most impressive feature was signal clarity with an average rating of 4.5/5. The candlestick chart and BUY/SELL/Entry/SL/TP notation on the Entry Levels Page also received high ratings.

The financial experts among the study participants gave high ratings to the RSI signal indicator since it was in line with their knowledge of technical analysis. Non-financial users were satisfied with the BUY/SELL and UP/DOWN notations which are easier to interpret and apply

Table 2: User Study Satisfaction Ratings (Mean ± Standard Deviation, Scale 1-5)

Evaluation Dimension	Mean Rating	Std. Dev.
Accuracy and trustworthiness of predictions	4.1	±0.72
Clarity of signal outputs (BUY/SELL, trend)	4.5	±0.61
Usefulness of visualization tools	4.3	±0.68
Overall system usability	4.2	±0.74
Likelihood of using for investment decisions	3.8	±0.89

The moderate rating of 3.8 for utilizing the system to make investment decisions can be attributed to the conservative nature of the users. The majority of the users understood that there is some element of uncertainty in the predictions made by the system, and making decisions in actual investments would need more considerations.

Additional feedback from the interview sessions pointed out other aspects that needed attention. Some of the users proposed that incorporating news analysis, comparison of assets, and the option of customizing the prediction model may improve the usability of the system.

D. Error Analysis

In general, the model behaves reasonably across all possible scenarios but there are certain restrictions which have been noted during testing period. As mentioned above, one of the limitations occurs when the model is used in highly volatile conditions. It happens when the difference between maximum and minimum price within a day is very wide and the OHLCV parameters indicate abrupt changes. As the model was trained on relatively calm datasets, it fails to accurately forecast these values and tends to estimate them based on the mean value. The hybrid weight allocation mechanism partially solves this problem.

The second restriction has to do with RSI indicator used to generate buy/sell signals in trending environments. If RSI values persistently remain above 70 or below 30, the model continuously generates signals in the same direction which results in over-trading as similar trades are conducted multiple times.

In order to minimize this issue, future versions of the strategy should implement more effective trade entry and exit rules for RSI indicator. For instance, it is worth considering reverse trading techniques. Moreover, an integration of the indicator with others will make trading signals more reliable.

When it comes to applicability, the system can find different beneficial uses. Firstly, for retail investors, the Home Dashboard and Entry Levels Page could be used to save time while tracking the real-time price and indicators of gold in one place instead of using many other websites related to finance.

Secondly, as an additional educational resource, the Model Info Page along with the explanation of the model can help learners and educators understand more about machine learning in the finance industry. Moreover, by applying the model, learners can learn more about regression analysis, feature engineering, and API.

Thirdly, as an auxiliary tool, the model can be applied by financial advising companies to get some ideas about the gold market directions and use them together with other tools to make decisions.

VIII. Discussion

The results from the quantitative research, as well as user testing both provide evidence that machine learning methodologies can be utilized effectively when implemented in financial applications. The Ridge Regression (R) algorithm produced a good level of predictability (R2 = 0.94), while providing an uncomplicated and easily understood structure (appropriate for quick responses through a web-based solution).

By combining predictions derived from models including current market prices, the hybrid forecasting method increases the stability of the results. By capturing both historically based patterns, and real-time data, no single source of information is used exclusively; and therefore, balanced predictions can be produced by the model that more consistently reflects actual market activity.

The system recognizes the need to provide machine learning-based financial predictions responsibly and stresses the need for greater transparency in communicating uncertainty associated with forecasts. The Model Info Page provides performance metrics (e.g., R^2, RMSE, and MAE) that can help users understand the accuracy of the model, but there remains a need for better ways to clearly communicate uncertainty. Providing both prediction intervals and point estimates will help give users a clearer indication of the reliability of forecasts.

Users who are not experienced with evaluation metrics may make the mistake of assuming that a high level of accuracy equates to a high level of assured results in the future. As a result, such individuals may rely heavily on the system when making significant investment or financial decisions. To combat this issue, clear disclaimers should be provided, in addition to visual elements such as prediction bands, on a forecast chart in order to improve user confidence in how to use the system responsibly.

There are currently several limitations with the system as it exists today that should be considered when developing additional features and functions within the system. One key limitation is the absence of macroeconomic and sentiment-based features. Gold prices are influenced by many variables including the movement of currencies, interest rates, inflation expectations, and the overall market's sentiment as derived from news media and social media sites. The integration of these data types would help in developing the market's future direction and performance by allowing for better forecasting accuracy through a multi-modal means of aggregation, utilizing both structured OHLCV data as well as unstructured textual data.

IX. Conclusion

The suggested solution is realized via a full-fledged web-based system that involves the usage of such tools as Ridge Regression Model together with financial market APIs for acquiring live data. The main functions of the system are related to tracking the real-time dynamics of gold prices, plotting visual graphics reflecting the price dynamics, predicting the future gold prices with the help of machine learning models, making multi-day price forecasts, and suggesting the optimal trading decisions using RSI values .

According to the results of the experiment conducted, the suggested solution demonstrates very high prediction quality parameters: $R^2 = 0.94$, RMSE = 119.87, and MAE = 84.63 .

The proposed solution proves that a proper combination of predictions of machine learning with real-time market data leads to stable forecasting results. As the prediction model applies the method of autoregressive forecasting with smoothing, the forecast for the next seven days becomes consistent and realistic.

Moreover, according to the evaluation conducted with real users, the solution proves effective from a practical perspective, achieving a high level of satisfaction in a number of aspects. For example, users gave their highest ratings regarding the quality of trading signals – an average rating of 4.5 out of 5 – and were satisfied with the visualization of results – an average rating of 4.3 out of 5 .

First, in the future, the model can be enhanced even further through the application of advanced data modeling techniques such as applying machine learning algorithms like long-short term memory (LSTM) and transformer. This is because the machine learning algorithms will help in modeling time series data including the gold price variations over time.

Secondly, the model should incorporate other economic indicators such as the DXY index, CPI, and the TIPS spreads in order to enhance the ability of the system to identify the factors that affect the gold price fluctuations. In addition, the implementation of natural language processing (NLP) can provide more useful information regarding the financial market by analyzing financial news.

Thirdly, implementing portfolio optimization using the gold price predictions is one interesting aspect of this project that may be considered in the future.

X. References

[1] C. J. Ntim, K. A. Opong and J. Danbolt, “The relative value relevance of shareholder versus stakeholder corporate governance disclosure policy reforms in South Africa,” *Corporate Governance: An International Review*, vol. 20, no. 1, pp. 84–105, 2012.

[2] R. F. Engle, “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation,” *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[3] J. Patel, S. Shah, P. Thakkar and K. Kotecha, “Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques,” *Expert Systems with Applications*, vol. 42, no. 1, pp. 259–268, 2015.

[4] A. E. Hoerl and R. W. Kennard, “Ridge regression: biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[5] X. Li, P. Li and T. Wang, “Gold price forecasting using LSTM neural networks,” in *Proceedings of IEEE DSIT*, Guangzhou, China, 2020, pp. 1–6.

[6] S. Makridakis, E. Spiliotis and V. Assimakopoulos, “The M4 competition: 100,000 time series and 61 forecasting methods,” *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.

[7] J. W. Wilder, *New Concepts in Technical Trading Systems*. Greensboro, NC: Trend Research, 1978.

[8] M. C. Kung and B. T. Carverhill, “When is the short rate drift long run mean reverting?” *Review of Finance*, vol. 9, no. 3, pp. 411–429, 2005.

[9] G. Biau and E. Scornet, “A random forest guided tour,” *TEST*, vol. 25, no. 2, pp. 197–227, 2016.

[10] R. Aroussi, “yfinance: Download market data from Yahoo Finance API,” GitHub repository, 2019. Available: https://github.com/ranaroussi/yfinance

[11] F. Pedregosa et al., “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] S. Kumar and R. Sharma, “Hybrid machine learning models for gold price prediction using technical indicators,” *International Journal of Engineering Research and Technology*, vol. 13, no. 3, pp. 412–420, 2020.

[13] Y. Chen, Z. Zheng and F. Li, “Comparative study of machine learning algorithms for financial time series prediction,” in *Proceedings of IEEE BDAI*, 2021, pp. 78–84.

[14] M. Ballings, D. Van den Poel, N. Hespeels and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015.

[15] N. Jain and A. Kumar, “Real-time financial data integration in web-based prediction systems using REST APIs,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 345–352, 2021.

[16] P. Sharma and V. Gupta, “Flask-based microservices architecture for machine learning model deployment in financial applications,” in *Proceedings of ICCCIS*, 2022, pp. 102–108.

[17] A. Banerjee, R. Ghosh and S. Das, “Ridge regression for financial feature sets with high multicollinearity: a comparative analysis,” *Journal of Statistical Computation and Simulation*, vol. 91, no. 5, pp. 1012–1028, 2021.

[18] K. Guo, Y. Zhou and X. Shi, “An empirical study of RSI-based trading strategies on commodity futures markets,” *Journal of Futures Markets*, vol. 40, no. 6, pp. 917–935, 2020.

[19] J. B. Heaton, N. G. Polson and J. H. Witte, “Deep learning for finance: deep portfolios,” *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.

[20] A. Yadav and P. Singh, “A survey on web-based financial analytics platforms leveraging machine learning,” *International Journal of Information Technology*, vol. 14, no. 4, pp. 1893–1902, 2022.