Design of a Machine Learning-Based Platform for Currency Market Prediction: A Fundamental Design Model

K. Gordillo-Orjuela, P. A. Gaona-García*, C. E. Montenegro-Marín*

Faculty of Engineering, Universidad Distrital Francisco José de Caldas, Bogotá (Colombia)

* Corresponding author: cemontenegrom@udistrital.edu.co (C. E. Montenegro-Marín), pagaonag@udistrital.edu.co (P. A. Gaona-García)

Received 22 February 2024 | Accepted 22 October 2024 | Published 25 November 2024

Abstract

Prediction models in foreign exchange markets have been very popular in recent years, and in particular, through the use of techniques based on Machine Learning. This growth has made it possible to train several techniques that increasingly allow us to improve predictions according to the criteria that each algorithm supports and can cover. However, the development of these models and their deployment within computer platforms is a complex task, given the variety of approaches that each researcher uses based on the training process and therefore by definition of the model, which leads to the consumption of high computing resources for its training, as well as various processes for its deployment. For this reason, the following article focuses on designing a technological platform oriented to micro services, which minimizes the consumption of resources and facilitates the integration of various techniques and the analysis of various criteria, which improves their analysis and validation in a Web environment.

Keywords

Forex, Machine Learning, Micro Services, Reduction Model, Software Architecture, Support Vector Machine (SVM).

DOI: 10.9781/ijimai.2024.11.002

I. Introduction

In recent years, artificial intelligence and in particular the area of Machine Learning (ML) have given rise to a large amount of research on price prediction in financial markets, such as hydrocarbons, precious metals, currencies, among others [1]. Currently, a series of models and algorithms have been proposed that seek to predict the value of different currencies in the foreign exchange market, also known as "Forex".

With the growing importance of artificial intelligence, prediction models based on machine learning have been developed, within which we can find seven broad categories: regression methods, optimization techniques, support vector machines (SVMs), neural networks, chaos theory, pattern-based methods, and other methods that include natural language processing [2].

Despite the development and rise of studies based on prediction techniques and models to analyze behaviors and characteristics of algorithms based on machine learning, the process of choosing, analyzing and implementing them within an application scenario is a complex process given the variety of resources, criteria and computational aspects that are required to simulate them within a work environment. Therefore, the design of a computer platform that allows the comparison of machine learning techniques could be of great value and usefulness to carry out the choice of models that allow the optimization and prediction of the value of currencies within specific scenarios and, in turn, allow the combination of some of these models to obtain even more accurate results.

The objective of the following article is to propose the design of a software platform that allows the execution of several models based on Machine Learning, to carry out prediction processes in the prices of the foreign exchange market, in order to identify elements that facilitate their implementation within various scenarios and improve the process of deployment and measurement of these models in various scenarios.

The rest of the article is divided as follows: in Section II the background is addressed where the different methods, techniques and mechanisms that have been developed for prediction within the foreign exchange market will be analyzed, in order to identify which of them can be implemented within a computational platform. Section III presents the methodology that was used to carry out the design of the computational platform, Section IV proposes the design of the computational platform based on a model based on software architecture. Section V presents the architectural approach. Section VI presents the results of this platform design and some results that were obtained from its development. Finally, section VII presents the findings and future work.

Please cite this article as:
K. Gordillo-Orjuela, P. A. Gaona-García, C. E. Montenegro-Marín. Design of a Machine Learning-Based Platform for Currency Market Prediction: A Fundamental Design Model, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 1, pp. 162-172, 2024, http://dx.doi.org/10.9781/ijimai.2024.11.002

II. Background

A. Forex Currency Study

The foreign exchange market, also known as “FOREX”, is a transactional mechanism where a part of the population interested in this mechanism can acquire units of one currency to buy a proportional amount in another. There are two types of analysis on this market: fundamental analysis and technical analysis. In this paper, we will focus on technical analysis, which, in most cases, is based on statistical graphs and machine learning techniques [3].

Other proposals for price prediction in the foreign exchange market focus on improving existing machine learning algorithms through the use of optimization techniques [4]. Some research also uses genetic algorithms, such as particle swarm optimization (PSO). For example, in the study conducted by Pradeep kumar [5], the PSO algorithm was used to train a quantile regression neural network, allowing predicting the value of the forex market for USD/INR, EUR/USD, and USD/JPY. The research yielded good results based on the EM and MSE, although only for certain currencies and not for all currencies used in the study in general.

B. Analysis Tools and Algorithms Based on Machine Learning

Although numerous studies have been conducted on this topic, each with promising results, little research has been done on the development of a platform that allows several of these models to be collected and run in an environment that allows for statistical comparison. Within the literature we can find several proposals where the authors focus on the exchange and execution of models based on Machine Learning in a general way, but not in a specific context such as within the FOREX market. Fig. 1 describes 4 related works that are close to the approach that is being carried out through proposals based on Machine Learning.

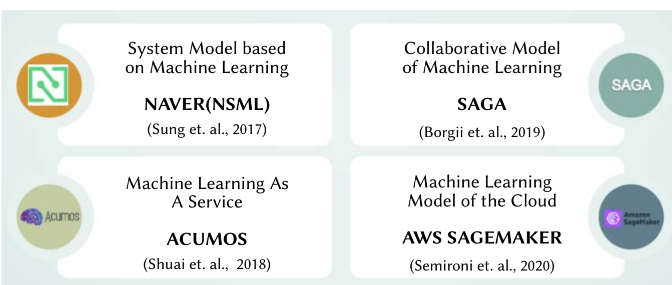


Fig. 1. Related Jobs Based on Machine Learning.

The NAVER Smart Machine Learning (NSML) platform, introduced by [6], aims to help researchers focus on modeling ML systems, rather than performing low-level tasks such as monitoring the training state of neural networks or mapping graphics processing units. The system consists of three main parts: a task scheduler, which efficiently allocates memory to execute ML tasks; a container system that provides a runtime and storage environment for models running on the system; and a graphical interface that acts as a mediator between the outputs generated by the system and the input flows provided by users.

Although there are models and studies on forex market prediction, there are few initiatives that focus on collecting, running, and comparing such models on a software platform. One such initiative is SAGA, an open-source platform presented by Borgli [7]. SAGA allows you to share ML models, training techniques for neural networks, and datasets. In addition, the platform offers the possibility of mixing several models and including extensions to improve the training of neural networks. This initiative promotes collaboration and knowledge sharing in the field of forex market forecasting.

According to Zhao [8], machine learning models are often developed for specific tasks using different frameworks, which makes it difficult to reuse and improve them. To address this issue, they introduce the open-source platform called ACUMOS, which allows ML models to be packaged into micro services containers and shared through the platform’s catalog. Although this platform facilitates the reuse of models by treating them as “black boxes” and focusing on communication, it does not allow model optimization and leaves this task to the developers.

Some models focus on improving or combining ML machine learning techniques with natural language processing. This is the case of the work carried out by Semiromi [9] where the authors combine the use of textual information from news from the economic calendar with indicators and historical information from the foreign exchange market to train ML models such as extreme gradient boosting (XGB), random forest (RF) and SVM. At the end of the study, statistical measurement accuracy was above 60%, and in some cases 64%, for predictions in the following thirty minutes. According to the authors, this shows that text mining combined with ML techniques can improve prediction accuracy in FOREX.

Based on this panorama and in order to compare and allow the selection of a model based on Machine Learning that best adapts to the prediction of the currency selected for our case study, a platform is proposed that facilitates the execution of several of these prediction algorithms and provides certain useful metrics that provide relevant information for decision-making regarding the purchase of a certain currency.

In this sense, there are authors who propose an approach to build an ensemble classifier using sentiment in news at sentence level and technical indicators to predict stock trends [10], but because it is unstructured data, extracting the most important features is difficult. Moreover, positive or negative news does not always affect stock prices in a certain way, so it is a purpose not viable to this work.

Finally, the time series approach could be used for forecasting, as done is some works [11]: autoregressive integrated moving average (ARIMA) and seasonal autoregressive integrated moving average (SARIMA) to forecast the impact of COVID-19 on cigarette sales. This could be the basis to apply it to currency market prediction, for now we will focus only on Machine Learning-based techniques.

III. Research Methods

This research aims to propose the software design of a platform that allows the integration of multiple models of price prediction of the foreign exchange market. Fig. 2 shows the phases that were addressed for the proposal.

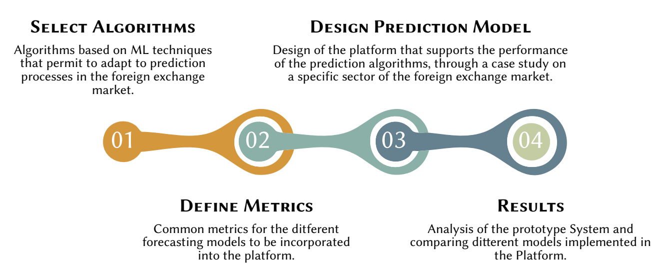


Fig. 2. Research methods.

Fig. 2 shows 4 phases of work. Phase 1, oriented to the selection of algorithms, will be based on a literature review addressed in the background section of the most common uses algorithms with the best results for price prediction in the foreign exchange market. This will make it possible to establish the inputs for the design of the application, the input and output values of the models, as well as to identify the statistical measures necessary to compare the performance of these algorithms.

In the next phase (definition of metrics), metrics will be identified to compare the performance of different ML models for prediction in the forex market. This involves defining the output interfaces of each algorithm and the mechanisms for storing and calculating errors in the predictions, in line with the literature identified in the background section. Since each model behaves differently, it will be necessary to perform a preliminary design of the communication mechanism between the application and the models.

The third phase of the project will focus on the design of the software platform that will allow the execution and comparison of the selected algorithms at a statistical level. In this phase, the definition of the architecture to be used, the selection of the necessary software components, the definition of the communication interfaces, the design of the graphical user interface and the creation of the mechanism to support the execution of the algorithms, which can be developed using multiple frameworks, will be carried out.

For the last phase, the implementation of a prototype of the designed solution will be carried out and then the results of selected algorithms will be presented. For software development, the benefits and adaptations of the Kanban agile development methodology will be used [12] due to its characteristics of flexibility and adaptability to changes.

IV. Platform Design

A. Analysis and Selection of Algorithms

For practical purposes of prototype design, the use of five Machine Learning techniques was determined, following the approach adopted in other studies such as those of Bansal [13] and Biswas [14]. Below, the mode of operation to carry out its implementation within the prototype of the platform is briefly described.

1. SVR-Wavelet Adaptive Model for Forecasting Financial Time Series

This work was carried out by Raimundo [15]. The study focuses on the use of regression support vectors (SVRs) and the Wavelet transform to make predictions in financial series over time, specifically in the foreign exchange market (FOREX). SVRs are a mathematical model used in Machine Learning that allows regression analysis and data classification. This model, according to the authors, has proven to perform better than the ARIMA and ARFIMA models in terms of predicting the price of the Australian dollar against the Japanese yen. To compare the models, several metrics were used, including ME (Mean Error), RMSE (Root Mean Square Error), MAE (Mean Absolute Error), MPE (Mean Percent Error), and ASM (Mean Absolute Percentage Error).

2. Bayesian Compressed Vector Auto Regression for Financial Time-series Analysis and Forecasting

This work published by Paponpat [16] has its origins in the previous work published by G. Koop, D. Korobilis and D. Petternuzzo [17], in which the autoregressive vector model (VAR) is combined with the Bayesian compression model (BC). VAR models are commonly used for prediction in financial markets. However, when the number of variables increases, the computational load also increases. This problem, known as the "dimensionality problem," occurs when the number of predictors in the VAR model equations is greater than the number of observations. In other words, "high-dimensionality methods" are used to refer to models that deal with this problem.

3. CNN–LSTM Model for Gold Price Time-series Forecasting

Proposal put forward by Livieris [18] where neural networks are used to predict the price of gold in the stock market. Specifically, two machine learning models are combined: convolutional neural networks (CNNs) and short-long term memory (LSTM) recurrent neural networks. CNNs are often used in image processing, as their layered architecture allows image data to be broken down into smaller components and analyzed using filters. Unlike CNN networks, recurring networks allow you to process data streams such as videos, written text, voice, time series, among others, as this data makes sense when analyzed together. For example, when analyzing a text, it is not enough to interpret word by word or letter by letter, but it is necessary to analyze how these words/letters are concatenated to make sense of a sentence or phrase.

To analyze the correlation between the data, recurrent neural networks not only use the activation of the current neuron, but also use the activation of the previous iteration. In other words, such networks implement a kind of "memory" that allows them to analyze the results of a previous activation.

As shown in Fig. 3, the neurons in an LSTM network are composed of three gates that control the flow of information to and from the "C" state cell, which can be understood as a band that stores the relevant information of the network. These gates act as valves that allow the state cell to be modified, either to forget, add, or give an output. The first gate is the "Forget gate", which allows you to remove unnecessary information from the status cell. The second is the Input gate, which allows you to add new information to the state cell. Finally, the Output gate allows us to generate the output of the LSTM neuron.

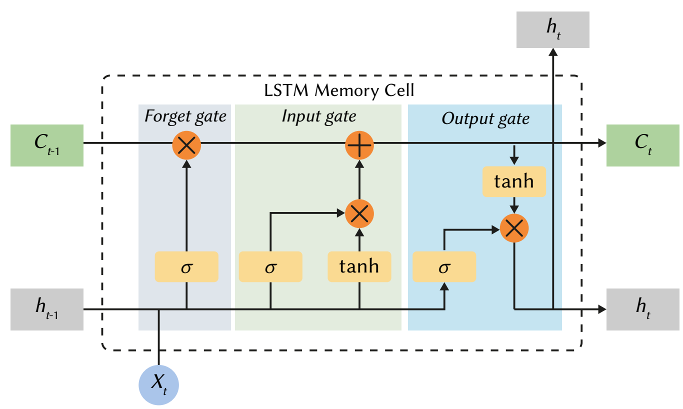


Fig. 3. Unit of an LSTM neuron (Source: [19] under license CC BY https://creativecommons.org/licenses/by/4.0/) .

4. Stock Price Forecasting Via Sentiment Analysis on Twitter

In this work [20], a large number of tweets from different dates related to the price of a certain stock market stock are collected. Then, using sentiment analysis techniques and the use of SVM (Support Vector Machine), a prediction is made of the price of the stock being analyzed.

The module works relatively simply: first, a message cleanup is performed to remove special characters and non-relevant information, such as emojis. Then, each message is scored based on the emotion it expresses, using keywords. For example, if a message contains words such as "bad," "hate," or "disgust," it is considered to have a negative emotionality score. On the other hand, if a message uses words like "love," "opportunity," "excellent," or "good," it is considered to have a positive emotionality. Connectors, such as "to", "from", "the", among others, are omitted. In the end, each message is weighted and labeled as positive, negative, or neutral. This information is put into a matrix, along with the date of the tweet and moves on to the next phase, which is processing and analysis.

When you have the sentiment score matrix and the time series matrix, you proceed to merge both into a single characteristic matrix. This matrix of characteristics is constructed with the following parameters for each of the days prior to the prediction: percentage of positive sentiments, percentage of negative sentiments, percentage of neutral sentiments, closing price of the stock/currency, HLPCT (High-Low percentage), PCTchange (Percentage Change), volume of the stock/currency.

5. Price Forecasting for Agricultural Products Based on BP and RBF Neural Network

In this work carried out by Yu, Shouhua and Ou, Jingying [21], neural networks based on radial functions (RBF) and the backpropagation technique (Backpropagation) are used to predict the value of certain agricultural products in the Chinese market, which was tested between January and December 2011.

B. Criteria for Analysis

Within the literature review carried out by Islam (2020) [2], seven main categories can be identified in which the current advances in machine learning applied to prediction in the foreign exchange market can be classified. What all of these works have in common is the use of statistical measures to establish the accuracy of their predictions. Although the measures used may vary depending on the model approach, the following statistical measures are generally employed: MAE (Mean Absolute Error), MSE (Mean Square Error), RMSE (Root-meansquare Error), and ASM (Mean Absolute Percentage Error). All these measures seek to show how far the prediction is from the actual value in the market.

Based on the algorithm review presented in the previous section and considering the conceptual framework, three main metrics have been selected for the comparison of the five selected algorithms: MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and ASM (Mean Absolute Percentage Error).

It was decided to use these three measures because, according to the literature review described in the background section, they are the most widely used to present the results and allow the degree of error of each algorithm in the prediction to be determined. In addition, authors such as Alexei Botchkarev [22] highlight that these metrics have been the most popular in the last 25 years. The mathematical formulation of each of these measures is presented below.

1. MAE

MAE seeks to give a measure of how far the calculated data is from the observed data. In other words, how far the prediction would be from reality. Its formula is given by (1):

$$MAE = \sum_{i=1}^{n} \frac{|y_i - x_i|}{n} \tag{1}$$

Where: y_i, is the value of the prediction and x_i is the actual or observed value.

2. RSME

The Root Mean Square Error (RMSE), like the MAE, allows you to measure how far the predicted values are from the observed or actual values. The main difference between these two metrics is that the RMSE shows us the squared difference between the estimated value and the observed value.

In simple words, the RMSE tells us how accurate a model is at representing the phenomenon of reality. Its mathematical formulation tells us how far the data being predicted is off the ground from the actual data expressed in (2)

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(Q_i - y)^2}{n}} \tag{2}$$

Where n is the number of observations. $\hat{y}_i$, is the predicted value at instant i. y_i, is the real value at instant i.

The MSE alone penalizes errors that are larger in the model. On the other hand, the RMSE, by taking the square root of the MSE, allows us to have a better perspective in terms of scoring when there are some very large errors that, at a general level, should not impact the performance of the model.

3. ASM

The Mean Absolute Percentage Error (ASM) is a measure that allows us to express the absolute error in percentage terms. It gives us an estimate of the accuracy of the prediction method we are using. It is more intuitive to understand than absolute error since error is expressed in percentage terms for a given time series.

For example, if we have an ASM value of 5%, it is interpreted to mean that the difference between the predicted value and the actual value is approximately 5%. Its formulation is simple and is expressed in (3).

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{3}$$

Where n is the number of measurements, A_t it's a real number. F_t, is the value predicted by the algorithm.

Based on these criteria, the following section focuses on a description of the architectural approach made.

V. Architectural Approach

A. Conceptual Model

From the development perspective, the technical team will be in charge of adding new algorithms to the platform, conceptually following the process described in Fig. 4, where the possibility of the platform taking the code from a repository and performing the provisioning and cataloging of said module within the system is raised.

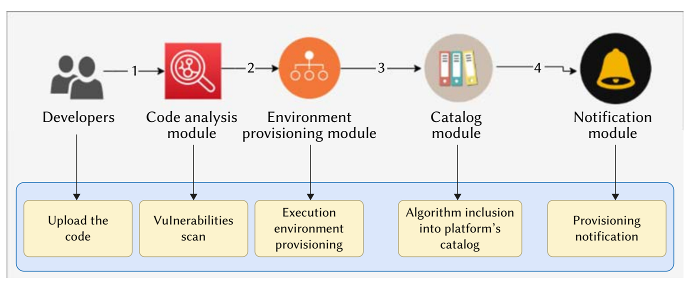


Fig. 4. Conceptual model of the process that a developer follows to link a new algorithm, prediction with ML, to the platform.

The steps to link a new algorithm are: (1) The developer uploads the code to the repo and starts the algorithm linking process. (2). Once the vulnerability scan is completed, the environment provisioning module is triggered. (3). After the provisioning module is done, there are an initial call to the algorithm to check is functionality. (4). Once the algorithm is added to the catalog, there is a notification sent to the developer.

However, from the perspective of a user who needs to analyze the performance of the different algorithms in the catalog, the platform should allow the selection of algorithms from the catalog to which the user has access and run them with different data sets to show the results in relation to three metrics: MAE, RMSE and ASM. This is how the person who requires it, makes a prediction of "n" days in the future for the time series that is entered as a parameter. This behavior is described in Fig. 5.

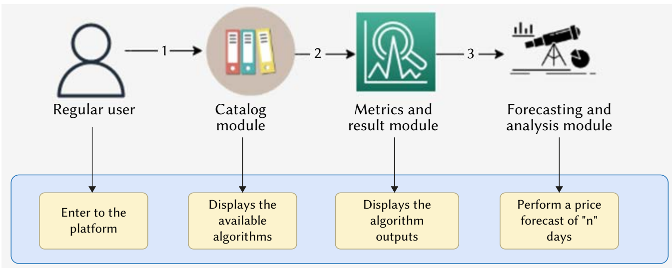


Fig. 5. Conceptual model of the process that a regular user follows to use and analyze prediction models with ML within the platform.

The steps for the conceptual model of the process that a regular user follows are: (1). The regular user selects one or more algorithms from catalog to compare. (2). The metrics module displays the results of each algorithm for the analysis. (3). Once the results are displayed, the user can select any of the algorithms and ask for a price forecast.

B. Process Model

For the incorporation of a new algorithm to the platform's catalog, the business process shown in Fig. 6 is followed.

The diagram presents the request to add a new algorithm to the platform via a link to the code repository. The system then performs a scan for potential vulnerabilities in the code. If any vulnerabilities are detected, the user is notified. Otherwise, the runtime environment is provisioned and an initial execution test is performed. Finally, if the execution was successful, the algorithm is added to the catalog and the requestor is notified.

C. Proposed Architecture

This section focuses on defining the application architecture and how it works at a high level. Both, the architecture follows the architectural pattern of publisher/subscriber and the use of machine learning models we include in this repository https://github.com/kgordillo-hub/SVM-Wavelet-forecasting-Financial-Time-Series.

Table I presents a description of the code developed for the assembly of the prototype. The web services were written in Java and the user interface in ReactJS.

TABLE I. Repository Used to Develop and Assembly the Platform

Name	Link
Code analyzer	https://github.com/kgordillo-hub/AnalizadorCodigo
Environment generator	https://github.com/kgordillo-hub/GeneradorEntornos
API Parameters manager	https://github.com/kgordillo-hub/GestorParametrosAPI
Metrics	https://github.com/kgordillo-hub/Metricas
Catalog manager	https://github.com/kgordillo-hub/GestorCatalogo
Results manager	https://github.com/kgordillo-hub/GestionResultados
Code linker	https://github.com/kgordillo-hub/VinculadorCodigo
Frontend	https://github.com/kgordillo-hub/material-kit-react
Wavelets+SVR	https://github.com/kgordillo-hub/1.Wavelets_SVM
BCVAR	https://github.com/kgordillo-hub/2.BCVAR
LSTM+CNN	https://github.com/kgordillo-hub/3.LSTM_CNN
SA + SVR	https://github.com/kgordillo-hub/4.SA_Twitter
RBF + NN	https://github.com/kgordillo-hub/5.RBFNN

By using the event bus, each service subscribes and receives messages as needed. In this way, the asynchronous operation of the services is guaranteed, and they are decoupled, this deployment is presented in Fig. 7.

As can be seen in Fig. 7, from left to right, the user accesses the platform from a terminal and communicates with the identity provider to obtain a JWT token, entering a username and password, following the OpenID Connect (OIDC) protocol and OAuth2.0. Once the client obtains the token, it includes it in the header of the HTTPS request to access the application. All incoming requests, both to the web layer and the composition services layer, will be made through a gateway. Each component that was defined for its construction is described below.

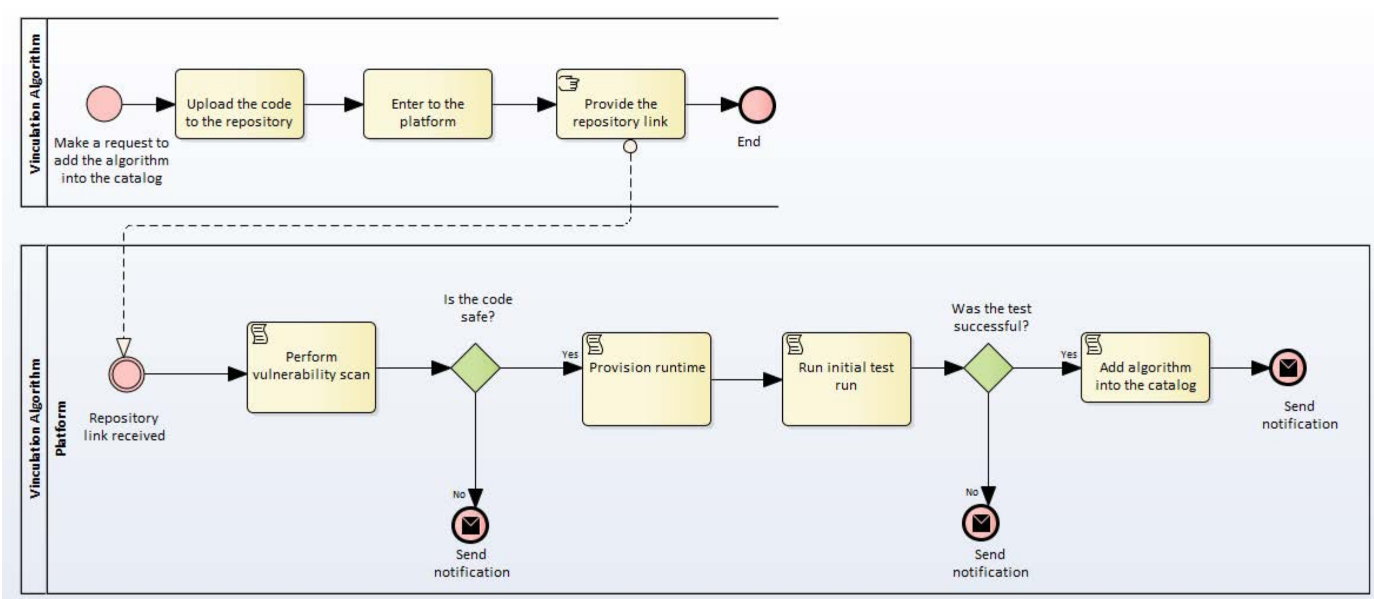


Fig. 6. Business process model for linking a new ML algorithm to the platform.

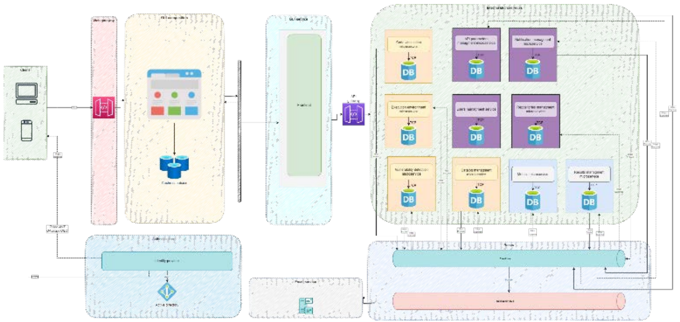


Fig. 7. Architecture proposed for the prototype using event bus and micro services.

1. Security and Authentication

Since the code is going to be included in the platform, a mechanism for scanning for possible vulnerabilities must be established. It is necessary to have a service in charge of performing a static analysis of the code in search of possible security breaches and storing only the code that does not present vulnerabilities. For this, there are multiple providers that offer security analysis tools, each with its own integration parameters and format of the rules that are loaded. Some of the functionalities that the administrator will have access to are adding, editing and deleting code analysis providers. Additionally, you can manage the code analysis rules that are loaded for each provider. There may be cases in which code analysis is not necessary or not possible. For such cases, there is the possibility of disabling/enabling mandatory code scanning by user groups. For these aspects, a microservice called Code Vulnerability Detection Microservice was developed, which can be identified in orange in Fig. 7.

Related to the authentication layer there are mainly two components. The first component is the identity provider, which will be responsible for validating the user's identity and returning a JWT token using the OAuth2.0/OIDC authentication protocol. This token will have the permissions of the services that the user can access. This repository will store the data of the users, the information of the groups to which they belong and the permissions they have to consume the different services of the system. Fig. 7 presents this service in green color.

2. IGU Composition Service

The web part or "frontend" will have its own cached database to maintain the states that will be displayed to the user. On the other hand, in the "GUI (Graphical User Interface) compositing services" layer, there will be a single component that will allow the composition of the visual part. These components are described in Internal Micro services, which allows to support the logic of the application or "backend" will be composed of micro services that communicate through an event bus. Each micro service will have its own database to maintain the state and configuration of the service it provides. Below is a description of each of the functionalities of the services to be developed:

- **Code linking and execution micro services**: Within the architecture of Fig. 7, the services presented in orange are responsible for the management, configuration and execution of the algorithms to be linked on the platform, as well as the maintenance of the algorithms within the execution process.
- **Platform management micro services**: Likewise, the services presented in purple in Fig. 7 are responsible for the management of the platform's configuration, for example, the configuration of notifications or the catalog of algorithms, the parameters to be used for the consumption of each of them, successful linking and completion of algorithms used, among others.
- **Result micro services:** The services presented in light blue in Fig. 7 are responsible for the use of metrics and management of results produced by the different ML algorithms linked to the platform.

D. Structural Model

As shown in Fig. 8, the class diagram is sketched for the micro service in charge of analyzing the code to be linked for vulnerabilities. This service will be implemented using the "Spring Boot" framework. The structure will consist of a "Controller" class in charge of receiving and dispatching requests through the HTTP protocol, a class that implements the logic of the service that is offered, and a logical layer that contains the implementation of the code analysis.

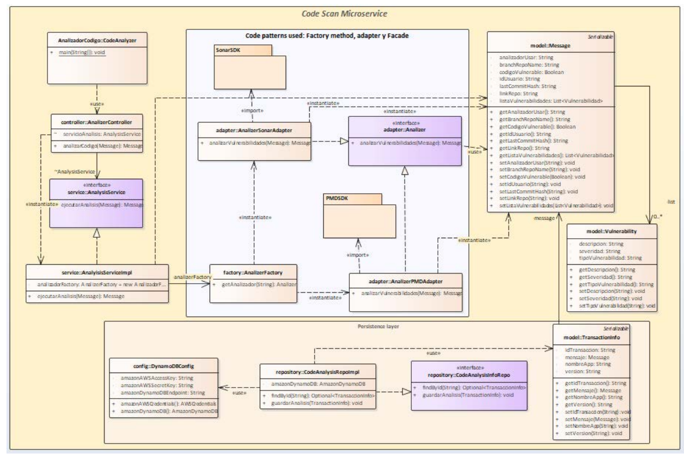


Fig. 8. Class diagram for the vulnerability scanning microservice in code.

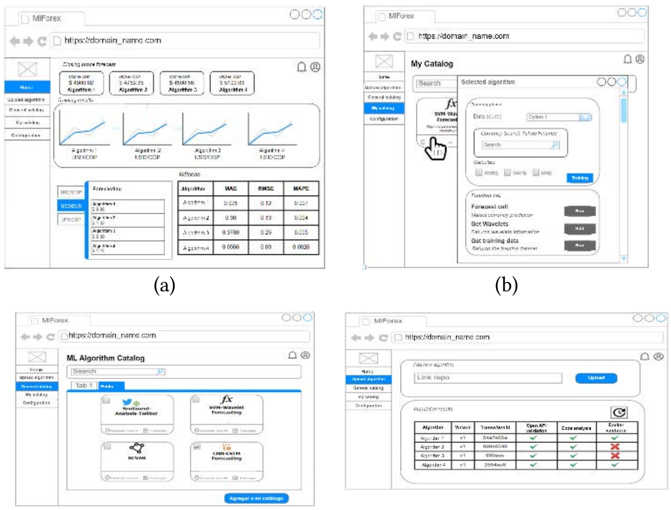
On the other hand, this service uses the façade pattern to hide the call to the different code analysis providers, such as PMD and Sonar. Because each vendor has different logic for each call to their implementations, the object adapter pattern is used to take the input message, transform it to the needs of each SDK (software development kit), and return a response to the service layer. Finally, the Factory Method pattern is implemented to obtain the different analysis adapters depending on the value that arrives in the Message object.

For the implementation of the services, Java was selected as the programming language and the "Spring framework" framework was selected to speed up the development process. On the other hand, for the implementation of the graphical user interface, the JavaScript programming language and the "ReactJS" framework were used. For the implementation of the databases and data bus, two technologies provided by Amazon Web Services (AWS) are used: DynamoDB and Simple Notification Service (SNS).

E. Front-end Graphic Prototype

Regarding graphic design, the following models were proposed to carry out its implementation, as shown in Fig. 9.

Fig. 9(a) shows the Web Schema of the page that allows you to display the general catalog of the system, in particular it allows to add algorithms to the personal catalog to analyze it. The interface in Fig. 9(b) allows you to interact with the algorithms added to the personal catalog to call the training method and the other methods offered by the selected ML algorithm. Fig. 9(c) presents the proposed Web schema that allows the main results of the algorithms to be displayed. Finally, Fig. 9(d) shows the status of the algorithms' linkage.



(a) (b) (c) (d)

Fig. 9. Architecture proposed for the prototype using event bus and micro services.

VI. Results

A. Architectural Software

The interface shown in Fig. 9 allows the user to enter the algorithm information to be linked, verify the link to the repository, model name, description, and version. Then, clicking “Add” allows you to perform model validation.

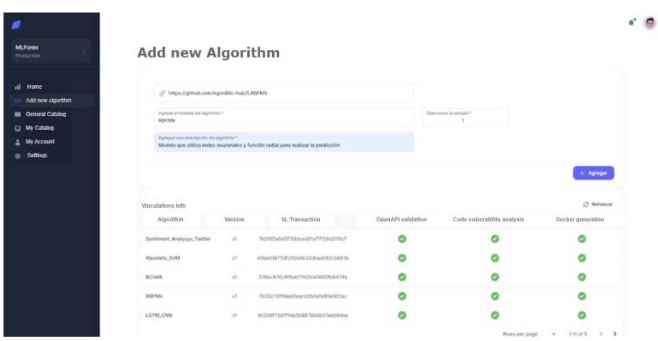


Fig. 10. Algorithm linking screen to the platform.

As presented in Fig. 10, the main flow of linking algorithms to the platform was implemented. To make use of this screen, the user must first specify the public URL of the repository where the algorithm to be run is hosted. Second, you need to add the name of the algorithm, its description, and the version to be linked, and click the “Add” button. After that, you must wait for the API specification structure validations, image generation, and vulnerability scanning to complete. Once the linking has been done, the algorithm will appear in the “General Catalog” section, as shown in Fig. 11. In this section, the user can click on “Add to my catalog” to add the algorithm to their list of available algorithms. This platform is designed for multiple users to link multiple algorithms and can be used collaboratively.

With the interface shown in Fig. 10, the user can view and select the models to use, which were described in the background section. Finally, Fig. 12 shows the results pane. This panel has two main sections: one that shows the estimated price for each algorithm for a specific currency, and another that shows the training result of each model and their respective comparison metrics.

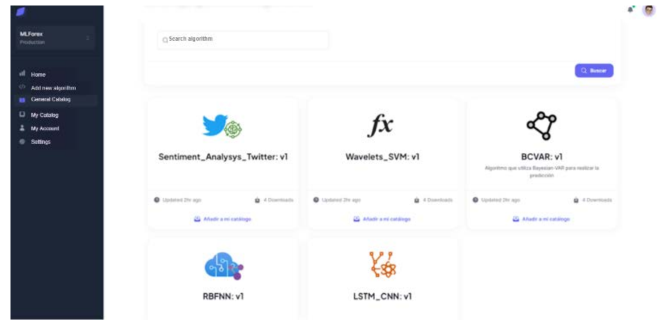


Fig. 11. General catalogue of algorithms linked to the platform.

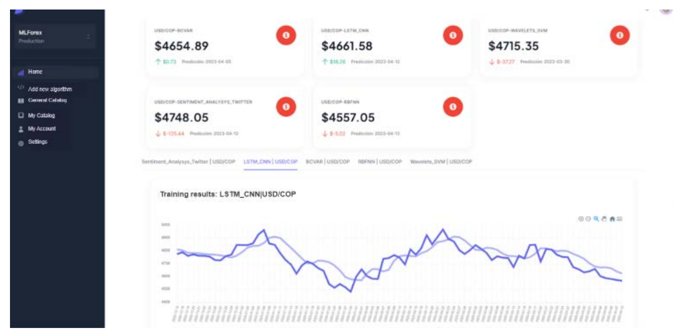


Fig. 12. Prediction results main screen.

B. Comparison of the Algorithms Selected

Following the first phase of the proposed methodology, the results of the five algorithms selected in the methodology section were replicated using the Python programming language; specifically, using Jupyter Notebooks as a tool. Once the algorithms were obtained, their performance was compared using the three metrics mentioned above (MAE, RMSE and MAPE), which, as Botchkarev [22] points out, allow us to understand how far the estimates are from the observed values, and have been the most used statistical measures since 1982. Below are the results of each algorithm for the following currency pairs: USD/EUR, USD/JPY, USD/COP and EUR/COP. The data evaluated correspond to the period from January 1, 2022, to August 1 of the same year (01/01/2022 - 01/08/2022).

TABLE II. Analysis of Results

Date	Open	High	Low	Close	Adj Close	Volume
2022-01-03	4557.200195	4571.700195	4557.200195	4557.200195	4557.200195	0
2022-01-04	4571.700195	4571.700195	4539.399902	4571.700195	4571.700195	0
2022-01-05	4539.399902	4568.399902	4539.399902	4539.399902	4539.399902	0
2022-01-06	4568.399902	4568.399902	4504.100098	4568.399902	4568.399902	0
2022-01-07	4504.100098	4510.799805	4504.100098	4504.100098	4504.100098	0
...	...	...	...	...	...	...
2022-07-26	4519.600098	4519.600098	4485.399902	4519.600098	4519.600098	0
2022-07-27	4485.399902	4485.399902	4462.200195	4485.399902	4485.399902	0
2022-07-28	4462.200195	4462.200195	4434.200195	4462.200195	4462.200195	0
2022-07-29	4434.200195	4434.200195	4426.399902	4434.200195	4434.200195	0
2022-08-01	4426.399902	4426.399902	4345.000000	4426.299902	4426.399902	0
151 rows x 6 columns						

The percentage of training data was 75%, leaving the remaining 25% as test data. That is, the time series from 01/01/2022 to 08/06/2022 was used as training information for each model, and the remaining time series, from 09/06/2022 to 01/08/2022, was used as test data to measure the accuracy of each algorithm, in accordance with the data sets presented at https://finance.yahoo.com/. An example of this data

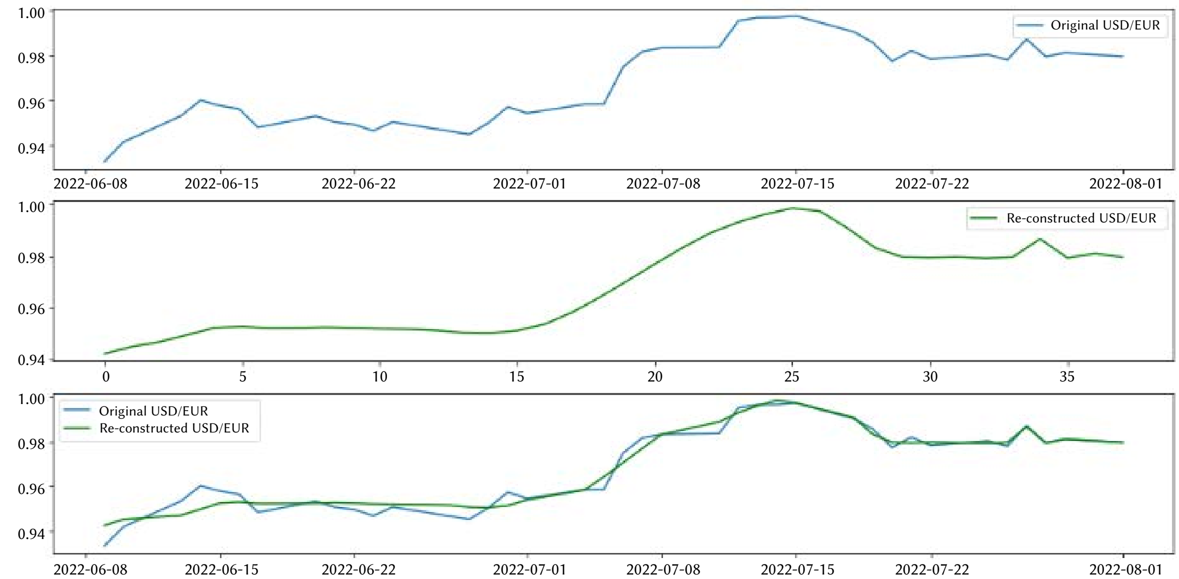


Fig. 13. Comparison graphic between actual and estimated values for USD/EUR using the SVR+Wavelet algorithm.

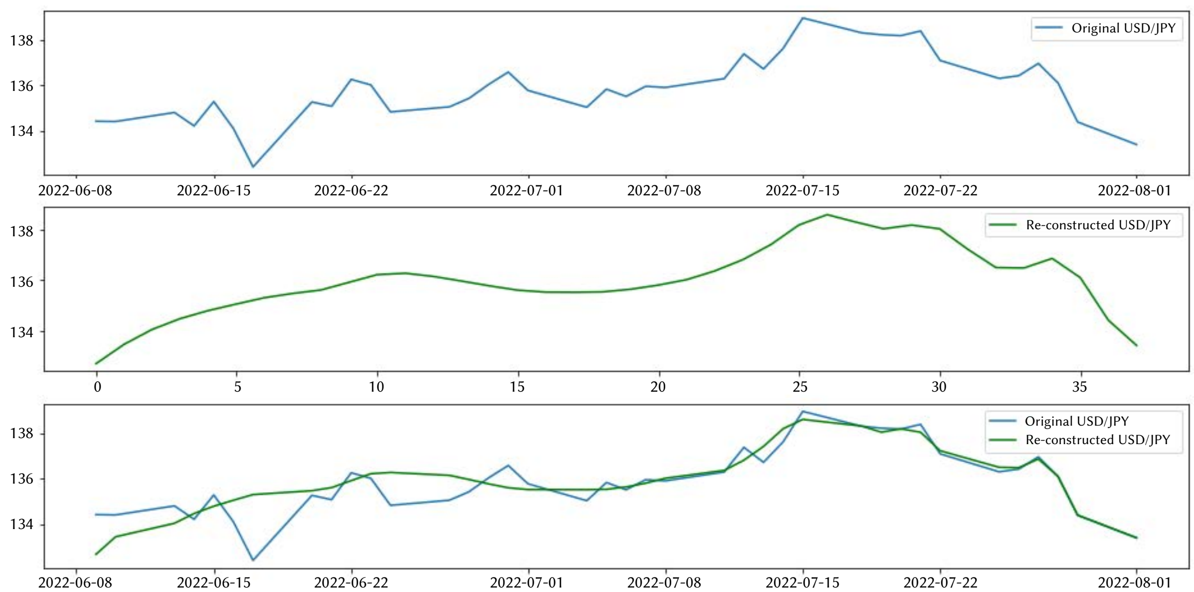


Fig. 14. Actual vs estimated value chart for USD/JPY using SVR+Wavelet algorithm. MAE: 0.485 RMSE: 0.746 MAPE: 0.35%.

is shown in Table II for the conversion of Euro (EUR) to Colombian peso (COP). It shows the opening prices, highest price, lowest price and closing price.

The following section present in detail the most promising results obtained of best algorithms selected, based on a case study defined for each currency pair during the period established above. At the end, a comparative table summarizing the performance of all algorithms is presented, in order to obtain an overview of all algorithms selected.

1. Performance: SVR-Wavelet Adaptive Model

The SVR-Wavelet algorithm, as described above, aims to decompose the time series into smaller components to remove noise and then take the cleanest component and perform prediction using SVR.

a) USD/EUR

The average value of the selected period of the training data, for the conversion of US dollars to euros, was 0.924 EUR for each USD. Performing the prediction, using the test data, the measurements shown in Fig. 13 are found.

The Y-axis shows the value of the currency on a scale of 0 to 1, while the X-axis shows the date on which the value was recorded. MAE: 0.0029 RMSE: 0.0039 MAPE: 0.30%

b) USD/JPY

The average value of the selected period for the training data, for the conversion of US dollars to Japanese Yen, was 125.00 YEN for every 1 USD. Performing the prediction, using the test data, the measurements shown in Fig. 14 are found.

As can be seen in Table III, this algorithm presented the lowest percentage of error (MAPE) when predicting the exchange rate from US dollars (USD) to euros (EUR). As for the worst result, it occurred when predicting the exchange rate from euros (EUR) to Colombian pesos (COP) with an error of 0.67%.

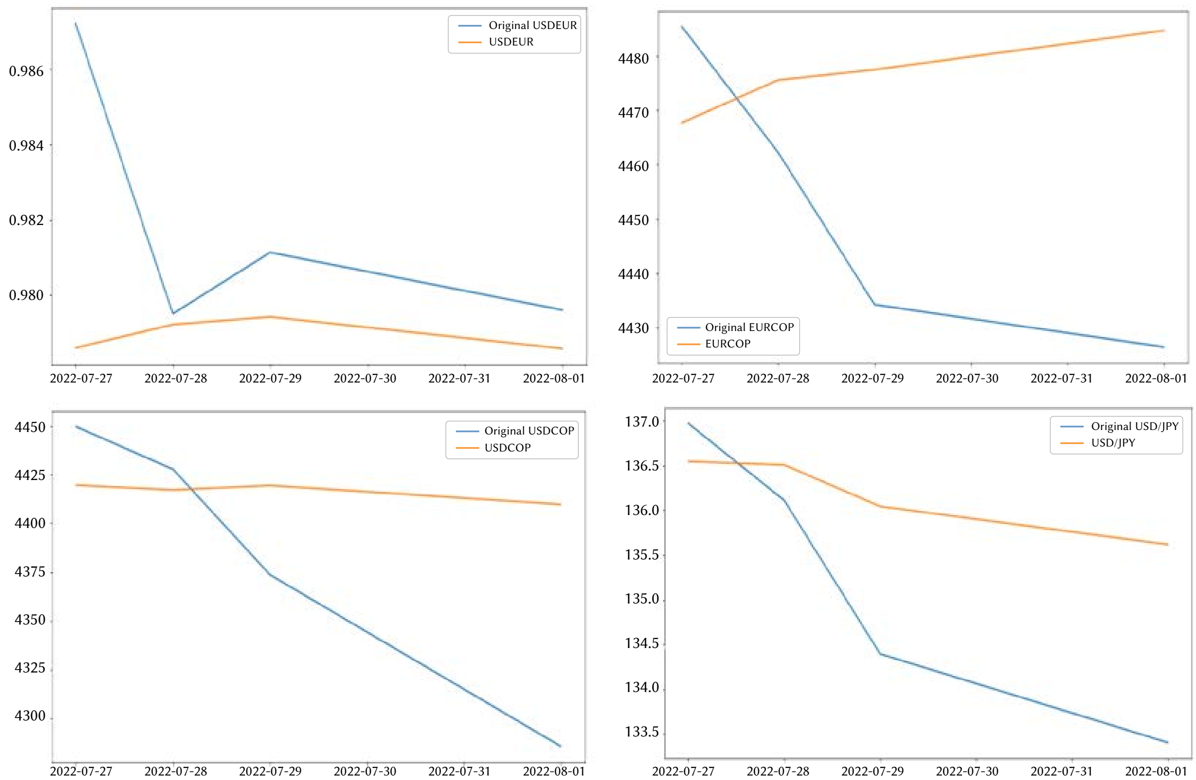


Fig. 15. Actual vs estimated value chart for USD/EUR, USD/JPY, USD/COP and EUR/COP using BCVAR algorithm with h=4 (days ahead).

TABLE III. Comparative Table SVR-WAVELET

Comparative table SVR-WAVELET				
	USD/EUR	**USD/JPY**	**USD/COP**	**EUR/COP**
MAE	0.0029	0.4850	26.7700	28.3900
RMSE	0.0039	0.7460	37.1700	37.0700
MAPE	0.30%	0.35%	0.65%	0.67%

TABLE IV. Comparative Table BCVAR

Comparative table BCVAR				
	USD/EUR	**USD/JPY**	**USD/COP**	**EUR/COP**
MAE	0.0007	1.8674	82.0599	11.0077
RMSE	0.0007	1.8674	82.0599	11.0077
MAPE	0.07%	1.40%	1.91%	0.25%

2. Performance: BCVAR Model

The BCVAR algorithm, described above, works when there are many variables within the system to be predicted and it becomes computationally difficult to find the relationship between them. The logic behind the algorithm is to randomly generate compression matrices and then apply the VAR model to predict the value of the currency “h” days in the future, using these matrices along with the input variables.

In this particular case, to predict the value of USD/EUR, USD/JPY, USD/COP and EUR/COP, other currencies were included in the input matrix in order to determine their possible influence on the value of the target currency. These additional currencies are USD/CAD, USD/AUD, EUR/AUD, EUR/CAD, CAD/JPY, EUR/JPY and EUR/MXN, which correspond to the Canadian dollar, the Australian dollar, the Australian euro, the Canadian euro, the Canadian dollar against the Japanese yen, the euro against the Japanese yen and the euro against the Mexican peso, respectively. The prediction was made by days, and based on criteria defined by h=1, 4 and 12 days in the future, based on criteria defined in the model by T. Paponpat [16].

Performing the estimate with 1 day in the future (h=1), the results shown in Table IV are found.

An estimate was made with 4 days in the future (h=4), obtaining the results shown in Fig. 15.

3. Resume of All Algorithms

Finally, Table V presents a summary of the comparison of the five algorithms implemented using as a case study the results obtained with US/COP currencies.

TABLE V. Chart of US/COP Currency Results

	MAE	**MAPE**	**RMSE**
BCVAR	113.96	2.43%	134.43
RBFNN	10.05	0.22%	11.20
ASTWITTER	151.06	3.27%	186.71
LSTM+CNN	58.49	1.24%	71.62
WAVELETS+SVM	55.93	1.18%	73.66

4. Currency Comparison US/COP

According to the results obtained, it can be indicated that the algorithm that uses radial functions (RBF) to improve the training of a classical neural network is the one that shows the best performance in terms of accuracy. On the other hand, it was found that the algorithm that combines sentiment analysis on Twitter information with support vector regression (SVR) obtained the worst performance. Although the incorporation of natural language processing to the analysis can enrich the input information to the model, it is considered that the substitution of the SVR technique by a neural network could improve the performance of this algorithm. Finally, the RBF NN algorithm has the best performance. Sentiment analysis with SVM is not as accurate. BCVAR is complex, however it is not that precise.

VII. Conclusions

The following article proposed the design of a platform through the use of algorithms based on Machine Learning for currency prediction, in a timely manner. A software architecture was defined that allows to address the problem of currency prediction by incorporating several algorithms identified within the literature review, with the purpose of comparing different techniques and algorithms used to make a comparison of their performance. The results of this proposal can be used in the context of those components and requirements necessary in a computer equipment that allows the development of a platform and the incorporation of new prediction models. Another interesting feature of the developed prototype is its ability to be configured to use other currencies from the foreign exchange market, as well as the stock market. This is because the input defined for each model is a time series and the platform can communicate with Yahoo Finance. It would only be necessary to add a list of the other currencies or stock prices to be consulted and pass them as inputs to the models. This demonstrates the versatility and scalability of the prototype, making it a useful and adaptable tool for predicting different financial markets. For the approach of the architecture, the use of microservices was chosen, which allows to have a modularized system in small services, as well as facilitates the maintenance of each of them by different development teams. In addition, since each service is independent, if one of them fails, the entire system will not be affected and will continue to function with the components that are working properly. Finally, the implemented algorithms have the ability to complement each other according to the needs and currency scenarios.

The literature review found that the most common metrics used by authors to measure the performance of different algorithms and make comparisons are: MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Error). These prediction error measures provide a good statistical idea of how far the estimates are from the real values, which allows comparing the performance of algorithms developed or improved by other authors. Other models, although they also use complex mathematical elements such as BCVAR, did not present results as promising as the RBF NN algorithm. It is possible that BCVAR is optimized for stock price prediction rather than for currency price calculation, which could explain the results obtained.

As critical implications, three challenges were identified in the design of the platform. First, the integration of multiple algorithms using different technologies and libraries, which was addressed by container virtualization. Second, because many of these algorithms require a prior training stage that can last several minutes, it was decided to decouple the system into small services that communicate asynchronously and report their status through an event manager. While this decoupled design solves the problem of high processing times, it also introduces a small delay in the responses perceived by the user, which is not critical in this case since it is a prediction system.

As future work, modules can be proposed that allow the combination of compatible algorithms and the introduction of new measurement parameters, in order to improve the performance of the predictions in accordance with the foreign exchange market to be worked on. The platform has the ability to be extended to include more data sources as needed, meaning it could be used to make predictions on other time series, such as stock price in the stock markets. In future iterations, a module could be added to incorporate additional data sources to those described in this document.

References

[1] A. Hernandez-Aguila, M. García-Valdez, J. -J. Merelo-Guervós, M. Castañón-Puga and O. C. López, "Using Fuzzy Inference Systems for the Creation of Forex Market Predictive Models," IEEE Access, vol. 9, pp. 69391-69404, 2021, doi: 10.1109/ACCESS.2021.3077910.

[2] Md. S. Islam, E. Hossain, A. Rahman, M. S. Hossain, and K. Andersson, "A review on recent advancements in FOREX currency prediction," Algorithms, vol. 13, no. 8, p. 186, Jul. 2020, doi: 10.3390/a13080186.

[3] S. W. Sidehabi, Indrabayu and S. Tandungan, "Statistical and Machine Learning approach in forex prediction based on empirical data," 2016 International Conference on Computational Intelligence and Cybernetics, Makassar, Indonesia, 2016, pp. 63-68, doi: 10.1109/CyberneticsCom.2016.7892568.

[4] S. R. Das, D. Mishra, y M. Rout, "A hybridized ELM-Jaya forecasting model for currency exchange prediction", Journal of King Saud University - Computer and Information Sciences, vol. 32, núm. 3, pp. 345–366, 2020. doi: 10.1016/j.jksuci.2017.09.006.

[5] D. Pradeepkumar and V. Ravi, "Soft computing hybrids for FOREX rate prediction: A comprehensive review," Computers & Operations Research, vol. 99, pp. 262–284, May 2018, doi: 10.1016/j.cor.2018.05.020.

[6] N. Sung et al., "NSML: a machine learning platform that enables you to focus on your models," arXiv (Cornell University), Jan. 2017, doi: 10.48550/arxiv.1712.05902.

[7] R. J. Borgli, H. K. Stensland, P. Halvorsen and M. A. Riegler, "Saga: An Open Source Platform for Training Machine Learning Models and Community-driven Sharing of Techniques," 2019 International Conference on Content-Based Multimedia Indexing (CBMI), Dublin, Ireland, 2019, pp. 1-4, doi: 10.1109/CBMI.2019.8877455.

[8] Z S. Zhao, M. Talasila, G. Jacobson, C. Borcea, S. A. Aftab and J. F. Murray, "Packaging and Sharing Machine Learning Models via the Acumos AI Open Platform," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 841-846, doi: 10.1109/ICMLA.2018.00135.

[9] H. N. Semiromi, S. Lessmann, and W. Peters, "News will tell: Forecasting foreign exchange rates based on news story events in the economy calendar," The North American Journal of Economics and Finance, vol. 52, p. 101181, Mar. 2020, doi: 10.1016/j.najef.2020.101181.

[10] C.-H. Chen, P.-Y. Chen, J. Chun-Wei Lin, "An Ensemble Classifier for Stock Trend Prediction Using Sentence-Level Chinese News Sentiment and Technical Indicators," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 7, no. 3, pp. 53-64, 2022, doi: 10.9781/ijimai.2022.02.004.

[11] A. Andueza, M. Á. Del Arco-Osuna, B. Fornés, R. González-Crespo, J.-M. Martín-Álvarez, "Using the Statistical Machine Learning Models ARIMA and SARIMA to Measure the Impact of Covid-19 on Official Provincial Sales of Cigarettes in Spain", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 8, no. 1, pp. 73-87, 2023, doi: 10.9781/ijimai.2023.02.010.

[12] H. Alaidaros, M. Omar, and R. Romli, "The state of the art of agile kanban method: challenges and opportunities," Independent Journal of Management & Production, vol. 12, no. 8, pp. 2535–2550, Dec. 2021, doi: 10.14807/ijmp.v12i8.1482.

[13] M. Bansal, A. Goyal, and A. Choudhary, "Stock Market Prediction with High Accuracy using Machine Learning Techniques," Procedia Computer Science, vol. 215, pp. 247–265, Jan. 2022, doi: 10.1016/j.procs.2022.12.028.

[14] M. Biswas, A. Shome, M.A. Islam, A.J. Nova, S. Ahmed, "Predicting Stock Market Price: A Logical Strategy using Deep Learning," in 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 218-223). Penang, Malaysia: IEEE, 2021, doi: 10.1109/ISCAIE51753.2021.9431817.

[15] M. S. Raimundo and J. Okamoto, "SVR-wavelet adaptive model for forecasting financial time series," 2018 International Conference on Information and Computer Technologies (ICICT), DeKalb, IL, USA, 2018, pp. 111-114, doi: 10.1109/INFOCT.2018.8356851.

[16] P. Taveeapiradeecharoen, K. Chamnongthai and N. Aunsri, "Bayesian Compressed Vector Autoregression for Financial Time-Series Analysis and Forecasting," IEEE Access, vol. 7, pp. 16777-16786, 2019, doi: 10.1109/ACCESS.2019.2895022.

[17] G. Koop, D. Korobilis, and D. Pettenuzzo, "Bayesian compressed vector

autoregressions," Journal of Econometrics, vol. 210, no. 1, pp. 135-154, 2018, doi: 10.1016/j.jeconom.2018.11.009.

[18] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Computing and Applications, vol. 32, no. 23, pp. 17351–17360, 2020, doi: 10.1007/s00521-020-04867-x.

[19] H. Fan, M. Jiang, L. Xu, H. Zhu, J. Cheng, and J. Jiang, "Comparison of long short term memory networks and the hydrological model in runoff simulation," Water, vol. 12, no. 1, p. 175, 2020, doi: 10.3390/w12010175.

[20] J. Kordonis, S. Symeonidis, and A. Arampatzis, "Stock price forecasting via sentiment analysis on twitter", in Proceedings of the 20th Pan-Hellenic Conference on Informatics, 2016, pp. 1–6. doi: 10.1145/3003733.3003787.

[21] S. Yu and J. Ou, "Forecasting Model of Agricultural Products Prices in Wholesale Markets Based on Combined BP Neural Network -Time Series Model," 2009 International Conference on Information Management, Innovation Management and Industrial Engineering, Xi'an, China, 2009, pp. 558-561, doi: 10.1109/ICIII.2009.140.

[22] A. Botchkarev, "A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms", Interdisciplinary Journal of Information, Knowledge, and Management, vol. 14, pp. 45-76, 2019, doi: 10.28945/4184.

Kevin Gordillo Orjuela

He has been employed as a software engineer in the private sector since 2016 and has been an active member of the GIIRA research group since 2019. Holding a Master's degree in Information Science and Communication from Universidad Distrital Francisco José de Caldas (Bogotá, Colombia - 2023), he also earned a Bachelor's degree in Systems Engineering from the same university in 2017. His research interests span various domains including web science, machine learning, data analysis, software architecture, information visualization, and visual analytics.

Paulo Alonso Gaona García

He earned a Ph.D. in Information and Knowledge Engineering from University of Alcalá in 2014 and working as a collaborator researcher in the Information Engineering Research Unit at University of Alcalá since 2012. Is full professor at Engineering Faculty of Universidad Distrital Francisco José de Caldas, Bogotá – Colombia since 2008. He is director of Multimedia Research Group and active member of GIIRA research group since 2008. He has a Master in Information Science and Communication from Universidad Distrital Francisco José de Caldas (Bogotá - Colombia - 2006). He is Systems Engineer at Universidad Distrital Francisco José de Caldas (2003). His research interest includes web science, semantic web, network and communications, e-learning, information visualization and visual analytics.

Carlos Enrique Montenegro Marin

PhD in Computer Science from Oviedo University (2012). Master in Web Site Management and Engineering at the International University of the Rioja - UNIR (2013). Master in Information Science at the Universidad Distrital Francisco José de Caldas (2006). Systems Engineer at the District University (2003). He was a Dean of Engineering Faculty (December 2018 - 2019) and full professor, attached to Engineering Faculty of District University "Francisco José de Caldas" since 2006. He was Coordinator of committee of accreditation in the bachelor's in systems engineering (2007 to 2010), He was Coordinator of the bachelor's in systems engineering (2012 to 2014).