Applying prediction model to the type of MXN-USD to spot exchange.

Héctor Martinez, Student, UASLP

I. Introduction

The value of the Mexican peso against the dollar is of great importance, especially for a country like the Mexican Republic, which is considered to have good macroeconomic management and promotes free trade. All of this is made possible by the country's exchange rate regime. Mexico had to opt for a system in which the exchange rate with other currencies could be taken advantage of due to strong economic ties, especially with the United States of America.

The use of technologies has increased due to technological advancements, and with this, our capabilities to develop systems that allow us to predict situations that require in-depth and complex studies, therefore, this work aims to develop a system in conjunction with artificial intelligence that enables us to predict the value of the dollar on a specific date.

II. Objective

We aim to create a prediction model using differents systems based on artificial intelligence and machine learning. There are different methods for building predictive models, so we will analyze some of the methods that naturally work with the problem we are presenting.
Due to the behavior exhibited by the problem we are working on, we will seek to use methods that provide us with higher precision in our results.

III. Methods

A. Linear regression.

Linear regression is the preferred mathematical procedure for predicting parameters in the future using quantified and known data from the past. The basic functioning of the linear regression system involves finding the relationship between the dependent variable (Y) and the independent variable (X) that we previously proposed in our system. When graphing our quantified data, it would be impossible to draw a straight line that passes through all the points representing the values. Therefore, regression aims to draw a line that follows the behavior of the values and has the least possible distance to each of the plotted data points[Fig. 1].

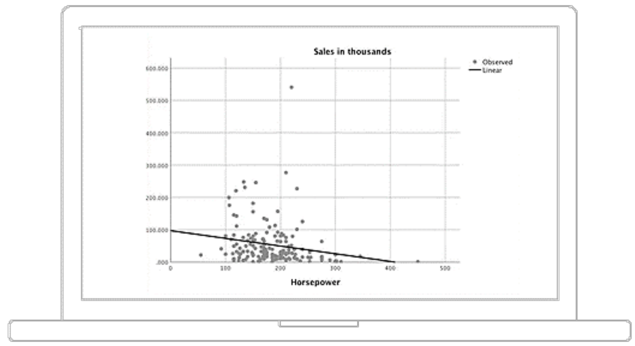

Fig. 1. Example of a linear regression.

B. Neuronal Networks.

Neural networks seek to simulate the behavior of the human brain, simplifying information so that computer programs can recognize patterns and thus solve common problems within machine learning and other branches of artificial intelligence. Neural networks are composed of layers of nodes, an input layer, multiple hidden layers, and an output layer. Each node is connected to all nodes in the next layer and so on for each node [Fig. 2]. All nodes have a weight and a threshold. Information is transmitted between nodes, so if the output node has a threshold value lower than the specified threshold, it becomes an active node with important and relevant information, and it sends its information to the next layer of the network.

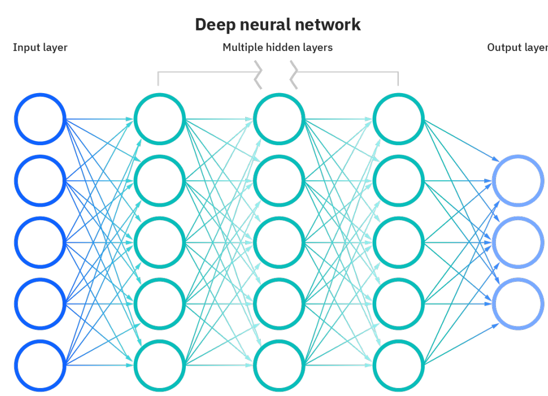

Fig. 2. Example of a Neuronal Network.

Due to the nature of the problem at hand, we chose to use a GRNN (General Regression Neural Network), which is a neural network system used, among other purposes, in generating regression models that will help us achieve our objective, where time is an important and challenging variable to analyze. The decision to use this procedure was based on the analysis presented in the work *"Analysis of Time Series with Artificial Neural Networks"* (González-Grimaldo & Cuevas-Tello, 2008).

As the main tool, we will need software that meets the requirements of our project, specifically software capable of creating a model and applying machine learning techniques. The first option proposed is the use of the Python programming language and self-generated information from another language such as R. Alternatively, we can use a database or software that can provide us with the required data format, such as MySQL or a spreadsheet application like LibreOffice.

The aforementioned GRNN is a neural network that bases its activation and prediction model on the radial basis function. This function focuses on determining the distance between a center generated in a part of the plane and measures the distance of the points from this center to generate models and make predictions.

There are extensive sources of information from which we can obtain the necessary data for the process we need. In this case, we will retrieve a list with the values of the variables, specifically the price of the dollar, from the official web of the Mexican government's website, the Secretariat of Governance of the Mexican Republic.

IV. Precedents

Some research projects (and their explanations) with the same objective as our work are as follows:

- *Prediction of peso-dollar exchange rate using Artificial Neural Networks (ANN)* (Zapata Garrido & Díaz Mojica, 2008).

This project involved the construction of neural networks using the EasyNN-Plus software developed by Neural Planner Software. The input variables were used to predict the exchange rate value in order to find the artificial neural network that best predicts future exchange rates.

Each neural network stored the following information in a matrix:

- Closing prices of the dollar with respect to the peso (COP/USD)
- Day of the week for each closing price
- Trading day for each closing price
- Inflation rates of Colombia and the United States
- Intervention rates of the Central Bank in Colombia and the United States
- Gross Domestic Product of Colombia and the United States
- International reserves of Colombia
- General Index of the Colombian Stock Exchange
- Amount of dollar transactions in Colombia carried out through the Electronic Trading System
- Monetary aggregate M1 of Colombia

Subsequently, 18 tests were conducted on neural networks with different input variables, and 3 acceptable variables were obtained with fewer cycles of learning and validation. This resulted in reliable data with a fairly acceptable accuracy, confirming the success of the prediction system.

- *ARIMA Model applied to the peso-dollar exchange rate in the 2016-2017 period using sliding time windows* (Ayala Castrejón & Bucio Pacheco, 2020).

This project worked with an ARIMA prediction model, which is an autoregressive integrated moving average model. It is used in the field that has been the focus of this work, predicting a future event. This model considers various situations, including variations in data over time and even structural changes, making the obtained data more solid.

This model is a modification of ARMA models, where a differencing operator is applied to remove possible polynomial trends.

As mentioned earlier, the aim was to create a model that can handle temporal and structural changes, which go hand in hand. Specifically, the following variables can be considered prone to change: trend (systematic change in the pattern of data), seasonal variation (repeating behavior pattern in periods equal to or less than a year), cycle (repetitive pattern with a period longer than a year), and random fluctuation.

To make the system less prone to failure due to the aforementioned changes, dummy variables were added to the formulas, such as the passage of time (t).

In summary, the mentioned work presented an analysis and explanation of the obtained data. Out of the 30 generated records, 12 fell within a realistic range of the peso-dollar exchange rate value, all of which had a 95 percent confidence interval, successfully completing the project.

V. Body

The next step in the development of our dollar price prediction model is the creation of the database with the necessary information for analysis and study of the problem. This information was obtained from the official website of the Bank of Mexico (Banxico) (*SIE Mercado Cambiario*, 2023). We will use data that was measured at the time, and we will also generate new data from these existing data that will help us perform operations and calculations to showcase our results.

Alongside this step, a dataframe was created in the programming language we are using, which is Python, along with some functions that allow us to access specific information from our database. Talking about the data, we applied methods to generate information in our tuples where there were empty spaces. This database contains information from January 1, 2010, to April 10, 2023. The library to be used to implement our GRNN will be "pyGRNN". It will allow us to create our network and, more importantly, give us the flexibility to modify necessary parameters to make our prediction model fit our needs.

The network was trained with information from the same dataset, but information was extracted from different levels of the database for testing purposes. The tests were conducted by comparing the values predicted by the network with the actual recorded values and calculating their percentage of error. Going into detail, the implementation of our GRNN will be described. For sigma, a value of 0.1 was used, which yielded good results in various tests. The negative mean squared error (MSE) was used as the evaluation metric, and a cross-validation value of ”3” was used. After several iterations, it was determined that a cross-validation with values of 4 or 5 led to overfitting in our model, while values less than three created a model that was too general and less specific for making predictions. The methodology for training our network was as follows:

Three different years from our database were taken, and the goal was to predict the value of the dollar price in 7 days. These same seven days would be used to calculate by training our network with the previous 60, 90, and 120 days. The provided plot shows one of the conducted analyses and its behavior.

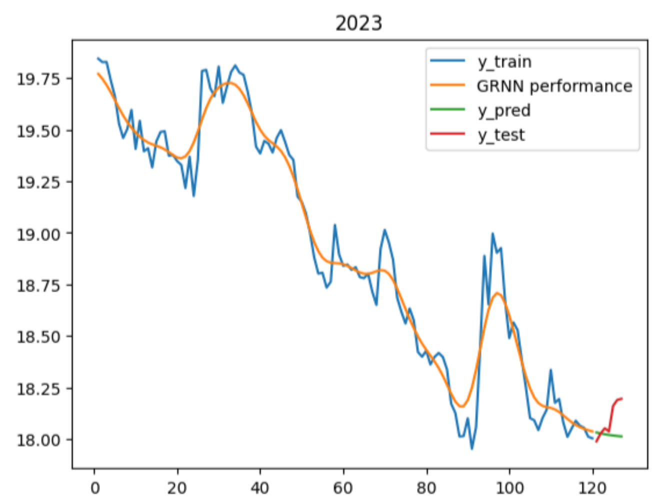

Fig. 3. Plot 1.

The orange line represents the prediction of data generated using the information that was used to train the network. The blue line represents the data that was used for training, the green line represents the data that was predicted without any training data, and the red line represents the actual data that was sought to be found.

Percentage of error of the most important values predicted

	1	2	3	4
120	0.23818778	0.00119233	0.1590877	0.09360398
90	0.29755148	0.06136677	0.09810789	0.03221407
60	0.08837318	0.12881231	0.26862535	0.18500391

TABLE I
PERCENTAGE ERROR TABLE - 2023

	1	2	3	4
120	0.70712474	1.34473287	2.73456318	2.87009342
90	0.94339153	1.62044049	3.01801501	3.15007691
60	0.70712474	1.34473287	2.73456318	2.87009342

TABLE II
PERCENTAGE ERROR TABLE - 2019

	1	2	3	4
120	1.09562022	0.12774061	0.01617095	0.04448959
90	1.02863921	0.11631878	0.33681561	0.36968752
60	0.71529478	0.30064658	0.43440015	0.41970398

TABLE III
PERCENTAGE ERROR TABLE - 2015

VI. CONCLUSIONS

It was concluded that our model is able to predict data very close to the end date of training, approximately three days. It achieves a fairly low percentage of error during these days, but from the fourth day onward, the percentage of error tends to be quite high. It is worth noting that these low error percentages are achieved when the data has a certain consistent behavior. If the data is highly dispersed, the error increases significantly.

Another conclusion reached is that creating a model of this type (GRNN) is good for prediction. However, the selected problem to which it is applied is quite challenging to work with, as the prices of the dollar tend to be highly volatile and can be influenced by numerous variables that affect their behavior.

References

Ayala Castrejón, R. F., & Bucio Pacheco, C. (2020). Modelo arima aplicado al tipo de cambio peso-dólar en el periodo 2016-2017 mediante ventanas temporales deslizantes. *Revista mexicana de economía y finanzas*, *15*(3), 331–354.

González-Grimaldo, R. A., & Cuevas-Tello, J. C. (2008). Analysis of time series with artificial neural networks. In *2008 seventh mexican international conference on artificial intelligence* (pp. 131–137).

Sie mercado cambiario. (2023). Retrieved 2023-10-04, from `https://www.banxico.org.mx/tipcamb/llenarTiposCambioAction.do?idioma=sp`

Zapata Garrido, L. A., & Díaz Mojica, H. F. (2008). Predicción del tipo de cambio peso-dólar utilizando redes neuronales artificiales (rna). *Pensamiento & Gestión*(24), 29–42.