ISSN 2007-9737

Modeling of Financial Systems with Neural Networks

Christopher Gonzalez Linares[1,*], Rubén Belmont Zuñiga[2], Angel Pedroza Sierra[3], Donaldo Garcia[4], Luis Enrique Andrade Gorjoux[4], Humberto Sossa[4]

[1] Instituto Tecnológico de La Piedad,
Tecnológico Nacional de México,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Física y Matemáticas,
Mexico

[3] Universidad Autónoma De Ciudad Juárez,
Instituto de Ingeniería y Tecnología,
Mexico

[4] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{ ing.christophergl, ruben.belzu.esfm, AngelPedrozaSierra02, donigj00, luis.gorjoux, humbertosossa}@gmail.com

Abstract. This study investigates the use of hybrid neural networks for modeling financial systems to address the challenges of market complexity and nonlinear behavior where conventional methods fail. The study uses financial closing price data from three different datasets covering the period from 2010 to 2024. The first dataset includes BBVA, Banorte, Inbursa, the Mexican Stock Exchange (BMV); the second includes ALFA, GAPB, Kimberly-Clark, Inbursa; and the third includes ALSEA, CEMEX, GCC, Grupo CARSO. The hybrid models were built using multilayer perceptrons (MLP), recurrent neural networks (RNN), long term memory networks (LSTM) and transformer architectures in both standard and variational autoencoder configurations. The results show that these networks successfully capture complex patterns, provide accurate predictions, improve generalization and reduce errors, highlighting the potential of hybrid deep learning models for financial time series prediction.

Keywords. Hybrid neural networks, financial modeling, market forecasting, TensorFlow, autoencoder.

1 Introduction

The financial system is a key component of the global economy. It enables the flow of resources and the allocation of capital across markets and institutions. Financial modeling is critical in this process, offering tools for forecasting and market analysis that support strategic decision-making in economic and financial sectors.

However, traditional financial models face significant challenges in capturing the complexity and non-linear behavior of financial markets. These models often struggle to predict unexpected events or identify intricate patterns, revealing the need for more advanced and reliable approaches [1].

This paper introduces several hybrid neural networks that offer significant improvements in financial modeling. By integrating elements of traditional neural network architectures, these models can process large datasets and adapt to

ISSN 2007-9737

various market conditions with high accuracy. The main contributions of this study are:

1. The implementation of hybrid architectures for financial system modeling.
2. An evaluation of these architectures to identify the best-performing network in terms of generalization.
3. A reduction in the number of training parameters across the models.

The theoretical framework and state-of-the-art section describe the components of traditional neural networks used in the proposed hybrid models. It also provides context by reviewing other approaches to financial system modeling. The hybrid model's structure and mathematical operation are explained in detail. Additionally, the databases and variables employed in the study are thoroughly described. Lastly, the training process and performance of the hybrid models are discussed, with results presented in tables that highlight their effectiveness in predicting financial system behavior.

2 Background and State of the Art

The financial system consists of institutions and markets that create and manage monetary instruments to promote economic growth. These institutions channel unused financial resources into productive investments, playing a key role in driving economic development [2].

A critical element of financial systems is the use of stock market indices. A stock index is a standardized indicator used to measure the performance of a group of assets. These indices can represent a narrow or broad range of assets and can be tailored to reflect various types of securities, such as stocks, bonds, or other instruments [3].

In Mexico, the main stock market benchmark is the Index of Prices and Quotations (IPC). The IPC provides a summary of the stock market's performance during each trading session on the Mexican Stock Exchange (BMV), offering valuable insights into market trends [4].

Understanding a country’s stock market index is essential when investing funds, as it helps evaluate the performance of a portfolio. Comparing a portfolio's performance with a relevant index allows investors to assess its success and identify long-term investment opportunities. The index, calculated based on the prices of its constituent shares, reflects the overall performance of the market [4].

One widely used technique in financial forecasting is the Autoregressive Integrated Moving Average (ARIMA) model. This model is designed for predicting future values in time series data. It combines autoregressive, moving average, and differencing components to capture trends and cyclical patterns in the data. Unlike methods that rely on artificial intelligence, ARIMA provides a robust approach for time series analysis based purely on statistical principles [5].

Techniques such as Support Vector Machines (SVM) and the Random Forest (RF) algorithm, as explored in the work of [14], have proven effective in identifying patterns and trends in market data. These methods are especially valuable for their ability to process large datasets and their resistance to overfitting. This is critical for stock price forecasting, as they provide more accurate results than traditional methods. With these techniques, investors can make better-informed decisions and optimize their portfolios, particularly during times of economic uncertainty.

Neural networks have also become increasingly important in the financial sector. These systems consist of interconnected units that adapt by mimicking the behavior of biological nervous systems [6]. Deep learning, in particular, has shown great potential for predicting time series, such as in cryptocurrency trading. Architectures like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have been used to enhance prediction accuracy in these scenarios [7].

Convolutional Neural Networks (CNNs) are highly effective for processing data with a lattice structure, such as images. As mentioned in [8], convolutional layers are designed to extract key features, making CNNs ideal for tasks involving visual pattern recognition. Additionally, dense or fully connected layers play a crucial role in the final stages of a neural network. In these layers, each neuron connects to all neurons in the previous layer, enabling the complete integration of features extracted earlier in the network.

Long Short-Term Memory (LSTM) networks, on the other hand, excel at modeling long-term dependencies in sequential data, such as financial time series. Unlike conventional Recurrent Neural Networks (RNNs), LSTMs are better at capturing these dependencies, making them particularly suited for complex and dynamic analyses [9].

Another promising technique is the use of autoencoders (AE), which are designed to learn compact and efficient representations of input data. Autoencoders reduce dimensionality while preserving essential information. Variational Autoencoders (VAE) go a step further by generating continuous distributions in the latent space. This allows for sampling new points, creating new data samples, and exploring the latent space more effectively [10, 11].

Hybrid networks, which combine linear and non-linear approaches as well as different neural network architectures, provide enhanced predictive capabilities. A common method involves using exponential smoothing to model the linear components of a time series, while neural networks capture the non-linear aspects. This combination leverages the strengths of both techniques, making it especially effective for financial forecasting [6].

Convolutional Recurrent Neural Networks (CRNN) are hybrid deep learning models that combine the strengths of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). As highlighted in [12], their main advantage lies in their ability to automatically learn complex patterns and capture both linear and non-linear dependencies in data. These models do not rely on prior domain knowledge and instead leverage the richness of the training data. This capability can be efficiently implemented using high-level programming frameworks like TensorFlow and PyTorch, especially when paired with graphics processing units (GPUs) [13].

In this context, the work of [15] uses CNNs as the foundation for building a hybrid network. This model is enhanced with a self-attention (SA) mechanism and a new gated unit (NGU) to predict the closing price of silver, which is known for its high volatility and non-linearity. The study concludes that hybrid neural networks, such as the CNN-SA-NGU, significantly improve feature extraction and prediction accuracy compared to traditional models.

In [17], a hybrid neural network model called CNN-STLSTM-AM is presented, which combines a CNN with a special Tanh long-term memory (STLSTM) and an attention mechanism (AM) and concludes that this hybrid model significantly improves the prediction accuracy compared to other traditional methods and simple neural network models such as support vector regression (SVR) and LSTM.

[16] in turn compares different hybrid approaches, in particular combinations of CNN, LSTM, gated recurrent units (GRU) and bi-directional short and long term memory (Bi-directional LSTM), whose results indicate that these hybrid neural networks achieve better stock market prediction performance compared to simpler models, suggesting that the combination of different network types can better capture the complex non-linear relationships and temporal dynamics in financial data.

In a similar vein, the HyRNN network, proposed in [18], combines architectures such as GRU, LSTM, and Bi-LSTM for stock market prediction. This model integrates financial data with sentiment analysis from financial news, demonstrating that incorporating multiple features significantly enhances the model's predictive performance compared to other deep learning approaches. This highlights the ability of hybrid neural networks to effectively capture the complexity and non-linear behavior of market movements.

Additionally, the study in [19] confirms the benefits of integrating diverse variables, such as sentiment data from social networks and earnings conference reports, into hybrid models. For instance, combining LSTM networks with the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model has shown notable improvements in prediction accuracy.

Hybrid networks are particularly powerful because they combine the strengths of different neural network types. For example, a CNN can extract visual features from an image, which an LSTM network can then process to capture temporal dependencies in a video. This approach has proven highly effective for tackling complex tasks, including financial forecasting [8, 9].

The study in [20] presents a hybrid model combining CNN and LSTM networks for financial time series prediction. Using stock index data from four Asian markets—Shanghai, Japan, Singapore, and Indonesia—the research concludes that the multivariate CNN-LSTM model outperforms standalone CNN and LSTM models in both accuracy and efficiency. This is demonstrated by a lower root mean square error (RMSE) value. The findings suggest that integrating relationships between variables into a prediction model significantly enhances the ability to forecast parallel movements in interrelated financial variables, making it a valuable tool for investors.

Similarly, [21] introduces a model that combines Gated Recurrent Units (GRU) and Convolutional Neural Networks (CNN) to improve short-term electricity demand forecasting. GRUs process sequential data and capture long-term dependencies, while CNNs handle high-dimensional spatio-temporal data, extracting key features from smart sensors. This hybrid approach achieves higher accuracy than traditional models such as BPNN, standalone GRU, and CNN, as measured by metrics like MAPE and RMSE. The method is particularly effective for managing large datasets in power distribution systems, improving prediction accuracy by efficiently handling complexity.

In [22], a hybrid model combining convolutional neural networks (CNN) and long-term memory networks (LSTM) is presented to predict the Bitcoin price. The CNN (which extracts spatial patterns in the data) processes relevant features from transaction data, while the LSTM (a network that captures temporal dependencies in data sequences) uses these features to predict the short-term price. This approach improves accuracy compared to traditional models such as BP, CNN or individual LSTMs and performs better on metrics such as MAE (mean absolute error), RMSE (root mean square error) and MAPE (mean absolute percentage error).

The study in [23] proposes a hybrid CNN-LSTM multiscale model for short-term electricity load forecasting, incorporating real-time electricity prices. The CNN extracts features at different scales from key variables such as historical load, prices, and weather conditions, capturing both local details and global trends. These extracted features are then combined into a vector that integrates both continuous and discontinuous aspects of the data. Finally, the LSTM network processes temporal dependencies to improve prediction accuracy, outperforming traditional models like ARIMA and Support Vector Machines.

Similarly, [24] presents a hybrid CNN-LSTM model for predicting stock market index trends using multi-scale feature learning. In this approach, the CNN extracts short-, medium-, and long-term daily price patterns, while the LSTM networks capture temporal dependencies at each time scale. Fully connected layers then combine these learned features to generate the final prediction. By integrating different time frames into a single optimized forecast, this multi-scale method improves accuracy compared to conventional models such as SVMs, standalone CNNs, and simple LSTMs.

In [25], a hybrid CNN-LSTM model for domain name generation is presented to improve algorithmically generated domain recognition (AGD). The CNN extracts relevant features from domain names, while the LSTM network captures sequential dependencies and maintains the consistency of the generated sequences. The domain names generated by the model are difficult to distinguish from legitimate names, as only 10% of them have been recognized as AGD by existing recognition systems, showing the limitations of current methods and their potential to improve computer security.

The use of hybrid neural networks CNN-LSTM-RNN represents an unexplored method in a case study such as our country's stock market index, which, like IPC, could improve the prediction of financial time series. Its ability to capture both spatial patterns and temporal dependencies in the data makes it superior to conventional approaches, and its implementation can provide investors with an accurate and efficient tool for analyzing complex and volatile markets.

Recent advances in hybrid neural networks for stock market index prediction continue to show promising results. In [26], a hybrid model that combines LSTM and GNN is presented and shows superior performance in predicting stock prices by effectively integrating temporal patterns and relationships between stocks.

In addition, researchers are exploring the benefits of integrating signal decomposition techniques such as Empirical Mode Decomposition (EMD) with deep learning models [27]. EMD decomposes complex financial signals into simpler intrinsic mode functions, allowing the model to learn patterns associated with different frequency components, which can lead to more accurate predictions. The EMD-TI-LSTM model, which combines EMD, technical indicators (TI) and LSTM, has shown better performance than the conventional LSTM model and other state-of-the-art methods on a variety of financial datasets.

The inclusion of attention mechanisms in [29], such as, which introduces GRU-enhanced attention in a CNN-LSTM framework, has been shown to improve prediction accuracy by dynamically focusing on relevant time steps [28].

Moreover, the LSTM-mTrans-MLP model presented in [30], a hybrid ensemble of LSTM, a modified transformer and MLP, has demonstrated exceptional prediction capabilities in various financial datasets.

Another significant hybrid model is that of [31], which presents a CLT (CNN-LSTM-Transformer) inspired by the DeepONet architecture, combining a transformer for feature encoding, a one-dimensional convolutional neural network (1D CNN) for local feature extraction, and an LSTM for capturing temporal dynamics. The transformer is responsible for identifying important features in the financial data, the CNN extracts local patterns within these features, and the LSTM models the temporal evolution of these patterns. This combination of techniques enables a more comprehensive analysis of financial data, taking into account different aspects of its complexity.

The combination of [32] is the hybrid CNN-Transformer model proposed for forecasting stock price movements. This model has shown improved accuracy in intraday stock price prediction by effectively modeling both short-term and long-term dependencies. The CNN processes the input time series to identify relevant local patterns, and these extracted features are fed to the transformer, which models the long-term dependencies and performs the final prediction of the stock price movement. Experimental results suggest that this hybrid approach can outperform standalone models and traditional methods in financial forecasting.

In addition, the combination of transformers with GRUs for predicting cryptocurrency prices was investigated in [33]. In these models, the transformer is used to capture long-term patterns in the cryptocurrency market data, while the GRU models sequential trends and short-term fluctuations, important features of this highly volatile market. The ability of the Transformer to learn general market trends and longer cycles, combined with the GRU's sensitivity to immediate price movements and volatility, results in a hybrid model that is able to account for the dynamic nature of cryptocurrencies.

Although the focus of this report is on financial forecasting, it is worth noting that in [34] transformer-based models, including Temporal Fusion Transformers (TFTs), have been successful in forecasting retail demand. This area has similarities with financial forecasting as both involve analyzing time series to predict future values. The superior performance of transformers in retail demand forecasting highlights their potential and versatility in analyzing sequential data in different areas.

[35] shows that the integration of sentiment analysis with hybrid models such as RNN-LSTM improves prediction accuracy by incorporating market sentiment from news articles.

In addition, [36] found significant improvements in capturing both short-term and long-term market dynamics by combining technical indicators with hybrid deep learning models such as LSTM-CNN.

In [37] Ensemble models such as ARMA-CNNLSTM, which combine linear and nonlinear modeling techniques, have also achieved improved prediction accuracy and robustness.

In [38], the CNN-CBAM-LSTM model is presented, which incorporates a convolution block attention module and enables improved prediction of stock returns by effectively evaluating long- and short-term information.

In addition, a hybrid model integrating CNN and Bi-LSTM was used in [39] to evaluate the impact of news events and sentiments on the prediction of stock trends.

Finally, in [40], the application of a hybrid ANN-LSTM model has shown better prediction accuracy than standalone ANN, LSTM and physically informed neural networks.

ISSN 2007-9737

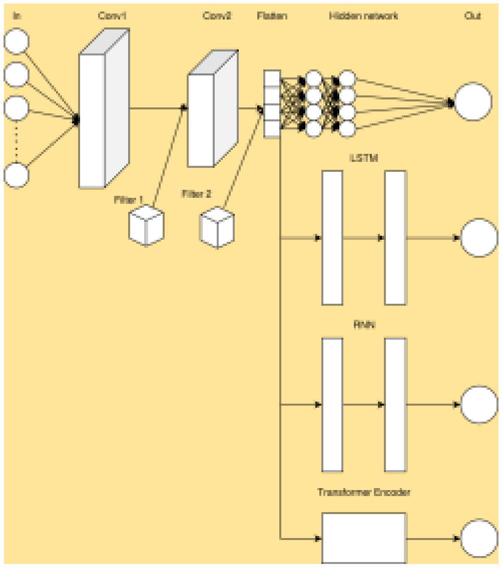

Fig. 1. Architecture of hybrid models

When forecasting financial time series, the increasing prevalence of hybrid deep learning models is striking [41]. By strategically combining different neural network architectures, integrating external data and using advanced techniques such as signal decomposition and ensemble learning, these models push the limits of prediction accuracy and provide valuable tools for investors and financial institutions to cope with the complexity of modern financial markets.

3 Design of the Hybrid Model

Our hybrid neural network models take a matrix as input, which is transformed into a smaller space through the first convolutional layer of the encoder. A second convolution further reduces the dimensionality, and finally, the data is vectorized.

This approach is applied to three of the proposed models, as we incorporate elements of the AutoEncoder architecture for data input and dimensionality reduction.

Fig. 1 illustrates the three proposed architectures at the output stage (after vectorization), where we employ Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Transformer for prediction.

Equations (1), (2), (3), and (4) describe the results of the hybrid networks proposed in Fig 1, as follows:

$$a_1 = W_4 \cdot (W_3 \cdot Flatten(W_2 \times (W_1 * X)) + b_3) + b_4, \quad (1)$$

where a_1 is the output of the convolutional encoder with MLP, X is the input of the network W_1 and W_2 are the weights of the first and second convolutional layers, respectively, $Flatten$ is the vectorization operation, W_3 is the weight of the first dense layer, b_3 is its bias, W_3 is the weight of the output layer, and b_4 is its bias.

$$a_2 = W_o \cdot (LSTM_2 \cdot (LSTM_1 \cdot Flat(W_2 \times (W_1 \times X)))), \quad (2)$$

where a_2 is the output of the convolutional encoder with LSTM, X is the network input, W_1 and W_2 are the weights of the convolutional layers, Flat means $Flatten$ and it is the vectorization operation, $LSTM_1$ and $LSTM_2$are the first and second long short-term memory layers, W_o is the weight of the output layer, and $b_o$is its bias.

$$a_3 = W_o \cdot (RNN_2 \cdot (RNN_1 \cdot Flatten(W_2 \times (W_1 \times X))))+b_o, \quad (3)$$

where a_3 is the output of the convolutional encoder with RNN, X is the network input, W_1 and W_2 are the weights of the convolutional layers, $Flatten$ is the vectorization operation, RNN_1 and RNN_2 are the first and second recurrent network layers, W_o is the weight of the output layer, and $b_o$is its bias.

$$a_4 = W_o \cdot (Transformer \cdot Flatten(W_2 \times (W_1 \times X)))+b_o, \quad (4)$$

where a_4 is the output of the convolutional encoder with Transformer encoder, X is the network input, W_1 and W_2 are the weights of the convolutional layers, $Flatten$ is the vectorization operation, Transformer is the transformer encoder , W_o is the weight of the output layer, and $b_o$is its bias.

The error ε_r of the above architectures is defined by Equation (5) as follows:

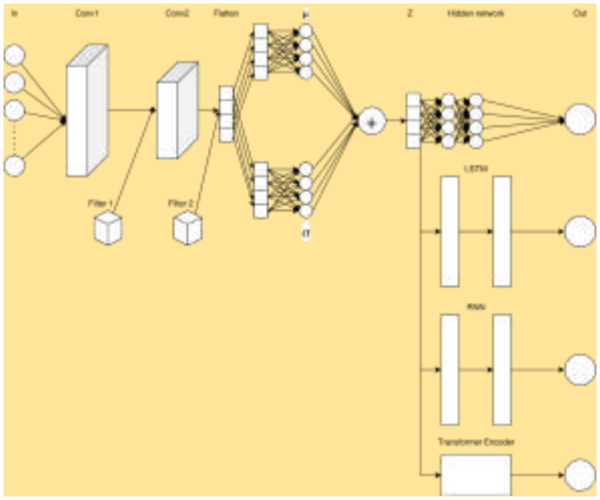

Fig. 2. Architecture of hybrid models with variational part

$$\varepsilon_r = \frac{1}{2}(a_k - t_k)^2, \quad (5)$$

where ε_r represents the network error, a_k is the network output, and t_k is the target value.

Fig. 2 illustrates the three proposed architectures that incorporate elements of the variational autoencoder (VAE) architecture. This approach involves adding a mean and a standard deviation to the models described above, creating a latent space with the required dimensionality. In the output layer (after vectorization), we use the same architectures: Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Transformer.

Equations (6), (7), (8), (9), and (10) describe the outputs of the hybrid networks proposed in Fig. 2, as follows:

$$X_2 = Flatten(W_2 \times (W_1 \times X)) + b_3),$$
$$z = (W_\mu \cdot (X_2) + b_\mu) + exp(\frac{1}{2} \cdot (W_\sigma \cdot (X_2) + b_\sigma)). \quad (6)$$

where X is the lattice input, W_1 and W_2 are the weights of the first and second convolutional layers, respectively.

Flatten is the vectorization operation, W and b_3 are the weights and bias of the latent space transformation, and z represents the latent space.

$$a_5 = W_4 \cdot ((W_3 \cdot z) + b_3) + b_4, \quad (7)$$

where a_5 is the output of the variational encoder with MLP, z is the latent space, W_3 is the weight of the dense layer, b_3 is the bias of the dense layer, W_4 is the weight of the output layer, and b_4 is the bias of the output layer.

$$a_6 = W_o \cdot (LSTM_2 \cdot (LSTM_1 \cdot z)) + b_{o,} \quad (8)$$

where a_6 is the output of the variational encoder with LSTM, z is the latent space, $LSTM_1$ and $LSTM_2$ are the first and second layers of long short-term memory, W_o is the weight of the output layer, and b_o is the bias of the output layer.

$$a_7 = W_o \cdot (RNN_2 \cdot (RNN_1 \cdot z)) + b_o, \quad (9)$$

where a_7 is the output of the variational encoder with RNN, z is the latent space, RNN_1 and RNN_2 are the first and second recurrent network layers, W_o is the weight of the output layer, and b_o is the bias of the output layer.

$$a_8 = W_o \cdot (Transformer \cdot z) + b_o, \quad (10)$$

where a_8 is the output of the convolutional encoder with Transformer encoder, z is the latent space, Transformer is the transformer encoder , W_o is the weight of the output layer, and $b_o$is its bias.

The error of the above architectures, ε_k, is defined by Equation (11) as follows:

$$\varepsilon_k = \frac{1}{2}(a_k - t_k)^2 + \beta(-\frac{1}{2}\sum \left(1 + \sigma - \mu^2 - e^\sigma\right), \quad (11)$$

where ε_k is the network error, a_k is the network output, t_k is the network target, z is the output of a dense network, and β is a regularization coefficient.

The metric used to evaluate the accuracy of all proposed prediction models is the root mean square error (RMSE), defined by Equation (12):

$$RMSE = \sqrt{\frac{1}{n}\sum (a_k - t_k)^2}, \quad (12)$$

where n is the number of observations, a_k is the predicted value, and t_k is the true value.

Table 1. Summary of the variables of the first database

Var	Samples	Mean	Std. Des.	Min	Max
Close Banorte	3566	98.16	31.40	37.95	189.17
Close BBVA	3566	127.68	27.41	54.70	244.83
Close BMV	3566	32.06	7.15	15.48	51.95
Close Inbursa	3566	30.15	7.07	13.55	55.16
weekly average IPC	3566	44073.144	5999.11	30586.08	58234.91
Day	3566	3.01	1.40	1	5
Month	3566	6.46	3.43	1	12
Close IPC	3566	44067.20	6024.41	30368.08	58711.87

Table 2. Summary of the variables of the second database

Var	Samples	Mean	Std. Des.	Min	Max
Close ALFA	3566	19.40	8.13	5.85	42.82
Close GAPB	3566	156.35	86.49	36.64	370.00
Close Kimberly	3566	33.45	5.68	18.39	46.34
Close Inbursa	3566	30.15	7.07	13.55	55.16
weekly average IPC	3566	44067.21	6024.42	30368.08	58711.87
Day	3566	3.01	1.40	1	5
Month	3566	6.46	3.43	1	12
Close IPC	**3566**	**44067.20**	**6024.41**	**30368.08**	**58711.87**

4 Database

The first database includes the closing prices of three banks: BBVA, Banorte, and Inbursa as well as the Mexican Stock Exchange (BMV) and the Price and Quote Index (IPC). This data was sourced and compiled from Yahoo Finance and Investing.com, covering the period from January 4, 2010, to March 15, 2024. The variables considered are:

– Banorte closing price,
– BBVA closing price,
– BMV closing price,
– Inbursa closing price,
– IPC Weekly average,
– Day,
– Month,
– IPC closing price.

Table 1 shows the total data, the mean value, the standard deviation, the minimum and the maximum.

The second database includes the closing prices of the shares of ALFA, GAPB, Kimberly-Clark, Inbursa, and the closing value of the IPC. This data was sourced and compiled from Yahoo Finance and Investing.com, covering the period from January 4, 2010, to March 15, 2024. The variables considered are:

– ALFA closing price,
– GAPB closing price,
– Kimberly-Clark closing price,
– Inbursa closing price,
– IPC Weekly average,
– Day,
– Month,
– IPC closing price.

Table 2 presents the total number of data points, the mean, standard deviation, minimum, and maximum values for the variables in the second database.

Table 3. Summary of the variables of the third database

Variables	Samples	Mean	Std. Des.	Min	Max
Close ALSEA	3566	40.68	18.47	9.37	74.43
Close CEMEX	3566	11.08	3.29	2.91	19.11
Close GCC	3566	83.35	43.88	34.16	204.61
Close CARSO	3566	66.15	25.30	19.66	189.22
weekly average IPC	3566	44067.21	6024.42	30368.08	58711.87
Day	3566	3.01	1.40	1	5
Month	3566	6.46	3.43	1	12
Close IPC	3566	44067.20	6024.41	30368.08	58711.87

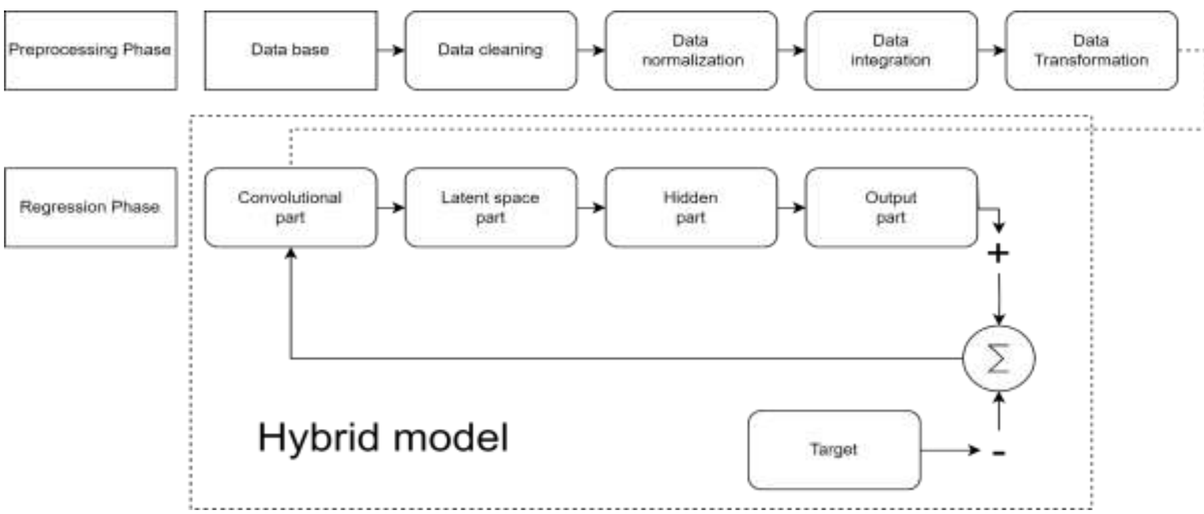

Fig. 3. General diagram for IPC modeling

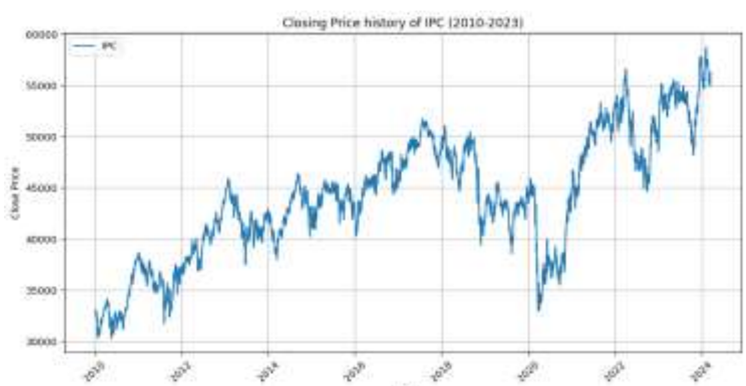

Fig. 4. IPC

The third database contains the closing prices of the shares of ALSEA, CEMEX, GCC, Grupo CARSO, and the closing value of the IPC. This data was collected and compiled from Yahoo

Table 4. Results obtained from the training of the hybrid networks in the prediction of the IPC using the first database

Arq.	Parameter	RMSE Training error	RMSE Test error	R2 Training error	MASE Training error
Encoder Convolutional + MLP	14	1.5%	1.9%	99.2%	5.6%
Encoder Convolutional + LSTM	26	1.6%	2.9%	99.0%	6.4%
Encoder Convolutional + RNN	**11**	**1.4%**	**1.9%**	**99.3%**	**5.5%**
Encoder Convolutional + Transformer	102	7.9%	18.7%	79.4%	31.3%
Encoder Variational + MLP	18	1.5%	2.5%	99.2%	6.2%
Encoder Variational + LSTM	28	1.6%	2.7%	99.0%	6.5%
Encoder Variational + RNN	19	1.6%	2.6%	99.1%	6.3%
Encoder Variational + Transformer	102	7.9%	19.0%	79%	31.3%

Finance and Investing.com, covering the period from January 4, 2010, to March 15, 2024. The variables considered are:

- ALSEA closing price,
- Closing price CEMEX,
- Closing price GCC,
- Closing price Grupo CARSO,
- IPC Weekly average,
- Day,
- Month,
- IPC closing price.

Table 3 presents the total number of data points, the mean, standard deviation, minimum, and maximum values for the variables in the second database.

Fig. 3 illustrates the data preparation process for analysis. The data is cleaned of unnecessary information, normalized between 0 and 1, integrated into a single vector, and finally transformed into a two-dimensional array for input into the hybrid model.

Fig 4 presents a graphical representation of the time series for the price and the stock price index (CPI) over the analyzed period. This visualization is essential for identifying trends and patterns in the financial data, providing valuable insights for the hybrid neural network model.

5 Financial Systems Prediction and Training of Hybrid Neural Networks

For the prediction of financial systems and the training of hybrid neural networks, TensorFlow 2.15.0 was used to design and train the models.

The training process was conducted with a learning rate of 0.002, over 150 to 250 epochs, using mini-batches of size 128 and the Adam optimizer. The equations (5) and (11) were applied to adjust the network parameters, while model performance was evaluated using the root mean square error (RMSE) metric, R2Score metric and the mean absolute scaled error (MASE) metric, which measures the accuracy of the predictions.

Table 5. Results obtained from the training of the hybrid networks in the prediction of the IPC using the second database

Arq.	Parameter	RMSE Training error	RMSE Test error	R2 Training error	MASE Training error
Encoder Convolutional + MLP	14	1.5%	1.9%	99.2%	5.6%
Encoder Convolutional + LSTM	26	1.5%	2.8%	99.1%	6.3%
Encoder Convolutional + RNN	**11**	**1.4%**	**1.9%**	**99.3%**	**5.5%**
Encoder Convolutional + Transformer	102	7.7%	11.0%	80.7%	31.0%
Encoder Variational + MLP	18	1.5%	1.8%	99.1%	5.9%
Encoder Variational + LSTM	28	1.5%	2.7%	99.2%	5.8%
Encoder Variational + RNN	19	1.8%	3.3%	98.0%	7.2%
Encoder Variational + Transformer	102	7.8%	11.3%	79.5%	33.1%

Table 4 presents the results obtained after the training process for the first database.

Table 5 presents the results obtained after the training process for the second database.

Table 6 presents the results obtained after the training process for the third database.

Across all three databases, Encoder Convolutional + RNN consistently achieves the lowest RMSE, highest R^2, and lowest MASE, handling temporal patterns effectively, making them more suitable for IPC prediction tasks. It also offers the best performance/complexity ratio, achieving solid results with a compact architecture.

Figure 5 presents a graphical comparison between the values predicted by our best hybrid model (Convolutional Encoder + RNN) using the first database and the actual values of the Price and Quotation Index (IPC) since November 2022.

Similarly, Figure 6 shows a comparison using the second database, where the predicted values from the hybrid model are evaluated against the actual IPC values over the same period.

Additionally, Figure 7 displays a similar comparison using the third database, highlighting the accuracy of the model's predictions when applied to a different set of financial assets and confirming the model's strong generalization ability.

The results presented in Fig. 5 and 6 demonstrate that the Encoder Convolutional + RNN hybrid model accurately predicts the Price and Quotation Index (IPC) across the datasets. The predicted values closely follow the actual trend of the index, highlighting the model's effectiveness.

Minor deviations observed in some areas may be attributed to the inherent unpredictability of financial markets. These discrepancies suggest potential improvements in the model's training process. Nevertheless, the model has shown strong pattern recognition capabilities over time,

Table 6. Results obtained from the training of the hybrid networks in the prediction of the IPC using the third database

Arq.	Parameter	RMSE Training error	RMSE Test error	R2 Training error	MASE Training error
Encoder Convolutional + MLP	14	1.5%	2.0%	99.2%	5.9%
Encoder Convolutional + LSTM	26	1.9%	4.6%	98.7%	7.4%
Encoder Convolutional + RNN	11	1.4%	1.9%	99.2%	5.6%
Encoder Convolutional + Transformer	102	8.5%	13.3%	75.3%	33.5%
Encoder Variational + MLP	18	1.5%	2.7%	99.2%	5.7%
Encoder Variational + LSTM	28	1.6%	2.5%	99.0%	6.5%
Encoder Variational + RNN	19	1.7%	6.3%	99.0%	6.6%
Encoder Variational + Transformer	102	8.6%	12.2	75.9%	33.7%

making it a valuable tool for forecasting financial market behavior.

5 Conclusion

This paper presents hybrid neural network models for predicting the price and quotation index (IPC) of the Mexican Stock Exchange. Architectures combining convolutional encoders with MLP, RNN, LSTM and Transformers networks have been developed, both in standard versions and with variational autoencoders (VAE).

The results show that these models are able to learn complex and nonlinear patterns from the market. The most effective model was the convolutional encoder with RNN, which achieved a test error of only 1.9% and an R^2 of 99.3%. It was also possible to reduce the number of parameters, improving efficiency without losing accuracy.

Although the models with transformers did not outperform the other architectures in terms of accuracy, their inclusion demonstrates the potential of these techniques to capture more complex temporal structures, especially when combined with autoencoders.

These advances suggest that hybrid networks are valuable tools for financial analysis. To further improve the results, it is recommended to include economic and sentiment variables, use Bayesian optimization techniques and investigate new transformer designs.

To summarize, the developed hybrid models provide robust and scalable predictions. With future adaptations, they can become a reliable

Fig. 5. Comparison of the IPC with the neural network Convolutional Encoder + RNN for the first database

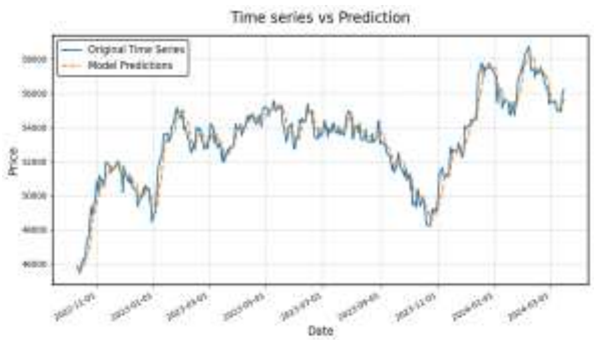

Fig. 6. Comparison of the IPC with the neural network Convolutional Encoder + RNN for the second database

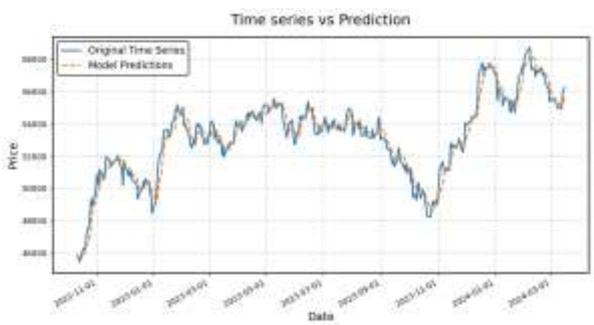

Fig. 7. Comparison of the IPC with the neural network Convolutional Encoder + RNN for the third database

basis for decision making in dynamic financial markets.

Acknowledgments

We would like to thank the Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT), the Instituto Politécnico Nacional, the Centro de Investigación en Computación and the Programa Delfin for their support in carrying out this project.

References

1. **Merton, R. C., Venegas-Martínez, F. (2020).** Tendencias y perspectivas de la ciencia financiera: Un artículo de revisión. Revista Mexicana de Economía y Finanzas, 16(1), 1–15. doi: https://doi.org/10.21919/remef.v16i1.581.
2. **Rodríguez Milián, G. de J. (2020).** Introducción al sistema financiero: especificidades en Cuba. Editorial Universitaria (Cuba).
3. **GBM (2022).** ¿Qué es un índice de referencia? GBM Academy. https://gbm.com/academy/que-es-un-indice-de-referencia/.
4. **GBM (2023).** ¿Qué es el Índice de Precios y Cotizaciones (IPC) de México? GBM Academy. https://gbm.com/academy/que-es-el-indice-de-precios-y-cotizaciones-ipc/.
5. **Khan, S., Alghulaiakh, H. (2020).** ARIMA Model for Accurate Time Series Stocks Forecasting. International Journal of Advanced Computer Science and Applications, 11(7). doi: https://doi.org/10.14569/ijacsa.2020.0110765.
6. **Gao, P., Zhang, R., Yang, X. (2020).** The Application of Stock Index Price Prediction with Neural Network. Mathematical and Computational Applications, 25(3), 53. doi: https://doi.org/10.3390/mca25030053.
7. **Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., Pintelas, P. (2021).** An Advanced CNN-LSTM Model for Cryptocurrency Forecasting. Electronics, 10(3), 287. doi: https://doi.org/10.3390 /electronics10030287.
8. **Aggarwal, C. C. (2018).** Neural Networks and Deep Learning: A Textbook. Springer.
9. **Rivera Zárate, I., Hernández Bolaños, M. y Pérez Romero, P. (2020).** Redes neuronales LSTM para la generación automática de texto a partir de novelas codificado en Python. Humanidades, tecnología y ciencias del Instituto Politécnico Nacional, 12(23).
10. **Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., Xu, Y. (2024).** Autoencoders and their

applications in machine learning: a survey. Artificial Intelligence Review, 57(2). doi: https://doi.org/10.1007/s10462-023-10662-6.

11. **Gulli, A., Pal, S., Kapoor, A. (2019).** Deep Learning with TensorFlow 2 and Keras. Second Edition.
12. **Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2020).** Using deep learning for price prediction by exploiting stationary limit order book features. Applied Soft Computing, 93, 106401. doi: https://doi.org/10.1016/j.asoc.2020. 106401.
13. **Jiang, W. (2021).** Applications of deep learning in stock market prediction: recent progress. Expert Systems with Applications, 184, 115537. doi: https://doi.org/10.1016/j.eswa.2021.115537.
14. **Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A., Rojas-Hernández, R. (2024).** Stock Market Prediction with Artificial Intelligence Techniques in Recession Times. In: **Jabbar, M.A., Tiwari, S., Ortiz-Rodríguez, F., Groppe, S., Bano Rehman, T.** (eds) Applied Machine Learning and Data Analytics. AMLDA 2023. Communications in Computer and Information Science, vol 2047. Springer, Cham. doi: https://doi.org/10.1007/978-3-031-55486-5_18.
15. **Wang, H., Dai, B., Li, X., Yu, N., Wang, J. (2023).** A Novel Hybrid Model of CNN-SA-NGU for Silver Closing Price Prediction. Processes, 11(3), 862. doi: https://doi.org/10.3390/ pr11030862.
16. **Valova, I., Gueorguieva, N., Aayushi, T., Nikitha, P., Mohamed, H. (2023).** Hybrid Deep Learning Architectures for Stock Market Prediction. Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science, pp. 1–8. doi: https://doi.org/10.11159/cist23.121.
17. **Liu, P., Wang, Z., Liu, D., Wang, J., Wang, T. (2023).** A CNN-STLSTM-AM model for forecasting usd/rmb exchange rate. Journal of Engineering Research, 11(2), 100079. doi: https://doi.org/10.1016/j.jer.2023.100079.
18. **John, A., Latha, T. (2023).** Stock market prediction based on deep hybrid RNN model and sentiment analysis. Automatika: Journal for Control, Measurement, Electronics, Computing and Communications, 64(4), 981–995. doi: https://doi.org/10.1080/00051144.2023.2217602.
19. **Jia, Y., Anaissi, A., Suleiman, B. (2024).** ResNLS: An improved model for stock price forecasting. Computational Intelligence, 40(1), e12608. doi: https://doi.org/10.1111/coin.12608.
20. **Widiputra, H., Mailangkay, A., Gautama, E. (2021).** Multivariate CNN- LSTM Model for Multiple Parallel Financial Time- Series Prediction. Complexity, 2021(1), 9903518. doi: https://doi.org/10.1155/2021/9903518.
21. **Wu, L., Kong, C., Hao, X., Chen, W. (2020).** A Short-Term Load Forecasting Method Based on GRU-CNN Hybrid Neural Network Model. Mathematical Problems Engineering, 1–10. doi: https://doi.org/10.1155/2020/1428104.
22. **Li, Y., Dai, W. (2020).** Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. The Journal of Engineering, 2020(13), 344–347. doi: https://doi.org/10.1049/joe.2019.1203.
23. **Guo, X., Zhao, Q., Zheng, D., Ning, Y., Gao, Y. (2020).** A short-term load forecasting model of multi-scale CNN-LSTM hybrid neural network considering the real-time electricity price. Energy Reports, 6, 1046–1053. doi: https://doi.org/10.1016/j.egyr.2020.11.078.
24. **Hao, Y., Gao, Q. (2020).** Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning. Applied Sciences, 10(11), 3961. doi: https://doi.org/10.3390 /app10113961.
25. **La O Reynier, L., González, R., Catania, C. A. (2024).** Generación de nombres de dominios mediante redes neuronales híbridas CNN-LSTM. https://sedici.unlp.edu.ar/handle/10915/165065
26. **Sonani, M. S., Badii, A., Moin, A. (2025).** Stock price prediction using a hybrid LSTM-GNN model: Integrating time-series and graph-based analysis. arXiv. https://arxiv.org/abs/2502.15813.

27. **Shen, Y., Feng, D., Chen, L. (2020).** Stock price forecasting based on EEMD and improved BP neural network. Applied Intelligence, 50(6), 1733–1747. doi: https://doi.org/10.1007/s10489-019-01608-y.
28. **Alsubaihin, A. A. (2025).** GRU-enhanced attention mechanism for LSTM in hybrid CNN-LSTM models for stock prediction. [Unedited manuscript].
29. **He, Q., Chen, F., Jiang, Z. (2021).** A hybrid CNN-GRU model with attention mechanism for stock price prediction. Expert Systems with Applications, 186, 115778. doi: https://doi.org/10.1016/j.eswa.2021.115778.
30. **Alameer, A. Z. (2025).** LSTM–Transformer-based robust hybrid deep learning model for financial time series forecasting. Journal of Risk and Financial Management, 18(1), 7. doi: https://doi.org/10.3390/jrfm18010007.
31. **Ahmad, Z., Bao, S., Chen, M. (2024).** DeepONet-Inspired Architecture for Efficient Financial Time Series Prediction. Mathematics, 12(24), 3950. doi: https://doi.org/10.3390/math12243950.
32. **Tu, T. (2025).** Bridging Short- and Long-Term Dependencies: A CNN-Transformer Hybrid for Financial Time Series Forecasting. arXiv.org. https://arxiv.org/abs/2504.19309v1.
33. **Mahdi, E., Martin-Barreiro, C., Cabezas, X. (2025).** A Novel Hybrid Approach Using an Attention-Based Transformer + GRU Model for Predicting Cryptocurrency Prices. MDPI. doi: https://doi.org/10.3390/math13091484.
34. **Caetano, R., Oliveira, J. M., Ramos, P. (2025).** Transformer-Based Models for Probabilistic Time Series Forecasting with Explanatory Variables. Mathematics, 13(5), 814. doi: https://doi.org/10.3390/math13050814.
35. **Kasture, S. S., Shirsath, S. S. (2024).** Enhancing stock market prediction: A hybrid RNN-LSTM framework with sentiment analysis. Indian Journal of Science and Technology, 17(18), 1880–1888. doi: https://doi.org/10.17485/IJST/v17i18.2239.
36. **Putra, R. S., Suteja, J. (2024).** Stock price prediction using hybrid deep learning model with technical indicators. Engineering, Technology & Applied Science Research, 14(3), 11103–11108. doi: https://doi.org/10.48084 /etasr.6567.
37. **Yang, S., Wang, X. (2023).** Financial time series forecasting with the deep learning ensemble model. Journal of Risk and Financial Management, 16(4), 201. doi: https://doi.org/10.3390/jrfm16040201.
38. **Yang, Y., Hu, Y. (2024).** CNN-CBAM-LSTM: Enhancing stock return prediction through long and short information mining in stock prediction. Journal of Risk and Financial Management, 17(12), 601. doi: https://doi.org/10.3390/jrfm17120601.
39. **Alomari, A. H., Alwidian, J. I., Alkhayyat, A., Alomari, A. (2023).** A hybrid CNN-BiLSTM model with multi-feature and sentiment fusion for stock trend prediction. Journal of Risk and Financial Management, 16(2), 94. doi: https://doi.org/10.3390/jrfm16020094.
40. **Ge, Q. (2025).** Benchmarking hybrid ANN-LSTM and physics-informed neural networks for forecasting stock market prices. International Journal of Computer Applications, 186(74), 1–6. doi: https://doi.org/10.5120/ijca2025927113.
41. **Zhang, F., Wang, Y., Li, H., Liu, Q., Zhao, J., Xu, X. (2023).** Deep learning models for price forecasting of financial time series: A review of recent advancements. Expert Systems with Applications, 220, 119615. doi: https://doi.org/10.1016/j.eswa.2023.119615.

Article received on 28/02/2025; accepted on 01/10/2025.
**Corresponding author is Christopher G. Linares.*