# Forecasting LEK/EUR Exchange Rate: A Comparative Deep Learning Study

Blerina Vika[1] and Denada Çollaku[2]

[1] University of Tirana, Faculty of Economy, Department of Statistics and Applied Informatics, Arben Broci, Tirana, Albania

[2] University of Tirana, Faculty of Natural Sciences, Department of Informatics, Bulevardi Zogu I, 25/1, Tirana, Albania

**Abstract**

This study contributes to the deep learning literature by investigating the applicability of different DL models to forecasting monthly LEK/EUR exchange rate. To demonstrate the effectiveness for exchange rate forecasting of this models we examine the performance of several deep learning techniques (DFNN, LSTM and 1D-CNN). For each architecture, we have used different configuration and diverse techniques to avoid overfitting of the models. The accuracy evaluation of each model was based on the out-of-sample prediction for different horizon (3, 6 and 12 months ahead), by analyzing the model estimation for pre- and post- Covid pandemic. The comparison results show that the LSTM model with two hidden layers stands out as the best prediction model in the run-up to forecast three and six months ahead, followed by the three-layered 1DCNN model. However, they change places in the race for the 12 months horizon as the 1DCNN-3L becomes the first-best predicting model while leaving the LSTM-2L rank in second. These results demonstrate the potential of deep learning techniques, and also, they emphasize the importance of well configuring, implementing and selecting the different topologies.

**Keywords**

Deep Learning, Time Series, Forecasting

CEUR-WS.org/Vol-3402/paper06.pdf

## 1. Introduction

Deep learning is considered a powerful tool for time series forecasting. The DL techniques are able to automatically learn linear and nonlinear relationships, extract features, handle large dataset and to capture temporal dependencies from sequential data in present of noise and missing values. However, it is important to mention that deep learning models can be computationally expensive and require large amounts of data to train effectively. Additionally, selecting the appropriate architecture, hyperparameters, and training procedure can be challenging and requires careful consideration.

An exchange rate is the rate at which one currency of a country can be exchanged with the currency of another country. It is an important economic variable because most economic decisions rely on it. Thus, for easier and better decision making, it is important to know how exchange rate will change in the future. Forecasting exchange rate is a challenging task due to the fact that currency exchange rates are influenced by many political and economic

Proceedings of RTA-CSIT 2023, April 26-27, 2023 Tirana, Albania
EMAIL: blerina.vika@unitir.edu.al (B. Vika); denada.xhaja@fshn.edu.al (D.Collaku);
© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

factors, such as interest rate, inflation, political stability, government intervention, etc. Government agencies, financial institutions and economists are paying attention to the use of financial models to analyze the parameters of dynamic time series in order to predict exchange rate [1].

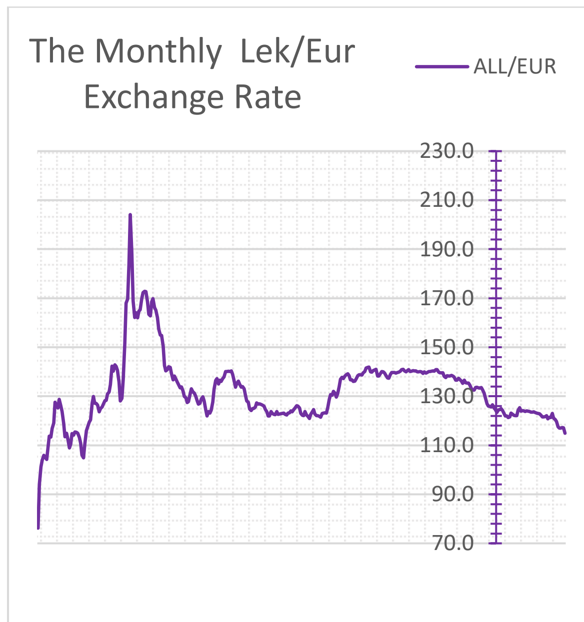
There are several methods used to forecast the exchange rate. One of them is regression analysis that tries to identify the relationship of currency exchange rates with various economic and political factors. This method is difficult to implement due to the fact that data regarding economic factors that influence exchange rate are not immediately available. Another method is time series analysis that attempts to make predictions based on historical currency exchange rate data. The time series models used for currency exchange predictions are moving averages, autoregressive integrated moving average (ARIMA) models, and exponential smoothing [1] . However, these models show problems. Moving averages and ARIMA do not work well when the time series being forecasted has a seasonality component. ARIMA models also require stationary time series and converting a non-stationary time series into a stationary one may remove some of the interesting dynamics of the time series. While the problem with exponential smoothing is that it may not capture the complexity of the time series being forecasted when there are irregular fluctuations. Recently academia and industry are demonstrating an increased interest in applying machine learning to economic and financial data [2].

This paper study the predictive performance of different deep learning models for various numbers of lags and horizon. We have selected three different forecasting univariate models, such as LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network) and DFNN (Deep Feed-Forward Neural Network) for out-of-sample forecast analysis.

## 2. Data and Methodology

In this paper, we analyze the historical data on the monthly LEK/EUR exchange rate for the period from January 1992 to December 2022 (fig. 1). The dataset consists of a total 372 data points, and is divided in three sets: the training set (from January 1992 to December 2017), the validation set (which consists of 12 observation data) and for the out-of-sample analysis it was used the training set (from January 2019 to December 2022). In order to avoid overfitting during the model building we use two datasets: the training and validation set.

**Figure 1.** The LEK/EUR time series (Jan1992-Dec2022)




These prediction experiments of the exchange rate were conducted with the help of PYTHON, TensorFlow packages. We have selected three different forecasting univariate models, such as LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network) and DFNN (Deep Feed-Forward Neural Network) for out-of-sample forecast analysis. Our main interest is to study the predictive performance of different deep learning models for various numbers of lags and horizon. The delayed data used as nodes value for the input layer on each model are twelve and eighteen and the predictive horizon is three, six and twelve months ahead. The predictive DL models were evaluated based on their performance on the test dataset using the root mean square error (RMSE) and mean absolute error (MAE).

By using the MAE and RMSE criteria simultaneously for the out-of-sample forecasting estimation, we can find the fluctuations in errors [2]. The RMSE can be greater than or equal to MAE. A very large difference between MAE and RMSE implies a large fluctuation in the error of the time series.2

We study the models' performance for the all out-of-sample periods. Also, for a deep analysis in our study we compare for each model configuration the model performance based on pre-Covid and post-Covid lockdown period

## 3. Models Description

Deep learning is considered a powerful tool for time series forecasting. The DL techniques are able to automatically learn linear and nonlinear relationships, extract features, handle large dataset and to capture temporal dependencies from sequential data in present of noise and missing values. However, it is important to mention that deep learning models can be computationally expensive and require large amounts of data to train effectively. Additionally, selecting the appropriate architecture, hyperparameters, and training procedure can be challenging and requires careful consideration.

In our work we have used three different types of DL architectures to study the Lek/Eur exchange rate forecasting: the Convolution Neural network (CNN), Deep Feed-Forword Neural Network (DFNN) and Long-Short Term Memory (LSTM). These three architectures are predominantly suited for time-series forecasting [2] [3].

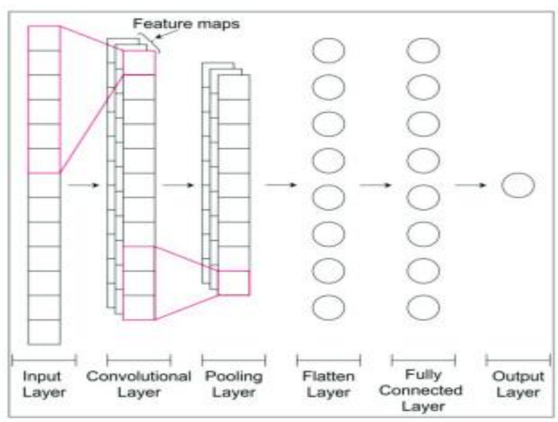
### 3.1. Convolutional Neural Network (1D CNN) model

The CNN model type is part of the artificial neural networks family and is widely used in image processing. The earliest presentation is LeNet and it was proposed by LeCun [4] et al. in 1998. The CNNs are designed to handle image input data efficiency but they are not limited. In order to use the CNN for sequence prediction the model needs to be a one- (1D) convolutional. CNN model is also called a deep learning model, due to the extra layers they have included in their structure called 'convolutional' block.

Use of the convolution block has some advantages, such as [1]: Each of the units operates with a vector of smaller length in order to make the estimation for the full size of the parameters, as was the case with DFNN. And also, the evaluation of the convolution parameters captures the time properties of the sequence data.

The typical architecture of 1D CNN consists of one-dimensional convolution layers, pooling layers, dropout layers and activation functions for manipulating the one-dimensional input data vector, figure 1.

**Figure 1**. One-dimensional convolutional neural network architecture [5]




The first layer that follows the input layer is the convolution layer. This type of layer contains filters (or also known as kernel) that are used to calculate the dot product of the weights and the input data by moving the filters along the input data. The role of this layer is to detect features in the input vector. The number and size of the kernels are crucial for adequately capturing the relevant features from the input data [5].

Each convolution layer, in the 1D CNN architecture, is associated with a pooling layer. The feature map output produced by convolution layer has a drawback, they are related with the precise position of the input sequence in the data vector. Pooling layers provide an approach to address this problem which is called down sampling. Two common pooling operations are *average* and *max* pooling that summarize the average occurrence of a feature and the most activated occurrence of a feature correspondingly.

The pairing of a convolutional layer with a pooling layer performs smoothing of the sequence data. Because they are part of the same function that outputs predictions, by optimizing the neural network loss, one optimizes smoothing parameters directly to perform well on a prediction task [1]. For time-series forecasting it needs to be mentioned that the smoothed vector, which is the output of the 1-dimensional convolutional and pool layer, was handled by the RELU layer in order to apply non-linearity to it. The following layers then use this smoothed sequence data and deal with the crucial part of the time series forecasting problem.

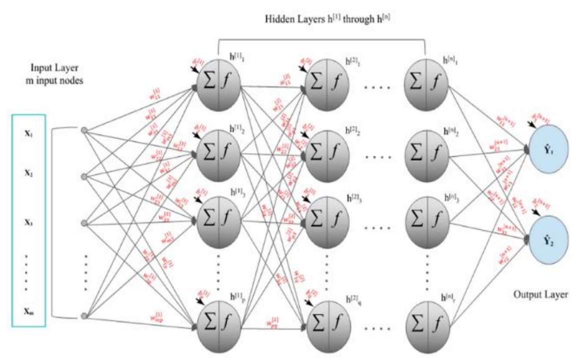
All these transformations for the initial time series, in the new data it will be much easier to identify the appropriate information from the rest of the CNN layers.

Generally, in time series analysis, it is desired to apply different types of smoothing techniques prior to analysis. In traditional forecasting, a moving average is commonly used to smooth the time series, and then on the result sequence data is applied a forecasting technique. The convolutions layers try to mimic this process and add more value because they perform a weighted smoothing of the time series by establishing automatically the ‘good’ parameters for smoothing. [6]

### 3.2. Deep Feed-Forward Neural Network (DFNN) model

The DFNN is a type of multilayer perceptron (MLP), composed of a sequence of layers, with a connectivity that flows in one direction from the input layer to the output layer. The layers are fully connected and do not form loops as all the information is processed in a forward way from the input layer to the output layer. The structure of DFNN consists of three different layers: the input layer, the hidden layers and the output layer. Each layer has a number of interconnected processing units called nodes. In this model, the processing units of a layer can be connected only with the nodes in the adjacent layer in forward direction.

**Figure 2**. General architecture for the deep feed-forward neural network [7].




The elements that need to be defined for the DFNN architecture are the quantity of hidden layers, quantity of neurons in the hidden layers, activation function in each layer and learning process to obtain the connecting weights. The optimal architecture of the DFNN model was generally determined through a trial-and-error process, which is an exponential combinatorial problem and a tedious task. [8]

The input layer contains a defined number of past values used to predict the next value(s) (in case of multistep ahead forecasting) of the univariate variable. The hidden layers take the signal sent from the previous layer and compute the inner product between the node value in the previous layer and the weight respectively and add up the value of the bias connected to this layer. The result value will be transformed by a function called the activation function, which is another factor that helps improve the accuracy of the information store. [9] The output of the activation function continues as the input for the node(s) of the next layer.

The learning process for the DFNN is supervised learning in which the weights, during the training phase, are adjusted in order to reduce the difference between the real output value and the model output. The general algorithm used for weights adjustment is the backpropagation algorithm. So, reaching the more effective model for time series forecasting, is not enough to define the DFNN architecture. The learning rule applied throughout the training phase needs to guarantee improvement on the out-of-sample forecasting performance. The backpropagation algorithm, used in the type of architectures, comes across with two problems: overfitting and the vanishing gradient.

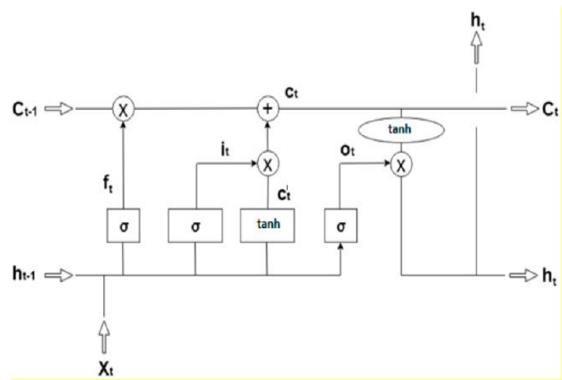
Overfitting occurs when the configured DFNN model fits exactly against its in-sample data used in the training process and performs poorly on out-of-sample forecasting. To avoid it, most research suggests the use of dropouts, in which on every epoch of model training only a percentage of nodes are used and the selection is random. The vanishing gradient is another problem that can be faced during the training phase; it happens when the backpropagation of the output error fails to reach the nodes in the input layer, thus the weights cannot be updated. Therefore, adding more hidden layers cannot improve the forecasting performance of the model.

### 3.3. Long Short-Term Memory (LSTM) model

LSTM is another architecture that can be used to model univariate time series forecasting problems. This architecture is a type of recurrent neural network based on LSTM cells in which the temporal dynamics of the input are modeled by means of recursive connections. [1] The LSTM are used as a solution for short term memory learning and are one of the most successful techniques that address vanishing gradients.

The unit in the LSTM model is a LSTM cell (figure 3) which contains an internal state ensuring that the gradient can pass across many time steps without vanishing or exploding. This connection type is characterized by the fact that the output of a unit, after a delay, can be the input of the same unit, along with the external inputs or the inputs of the lower layers. [1] The important elements on a memory cell are: a cell state ($C_{t-1}$ and $C_t$), hidden state ($H_{t-1}$ and $H_t$) and three gates which contribute them with the power to selectively learn, unlearn or retain information from each of the units. [10].

**Figure 3**. General architecture for the LSTM cell [11].



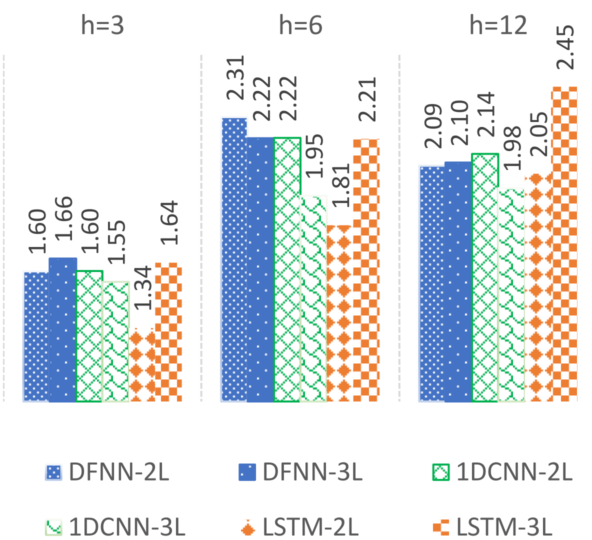

The cell state in the LSTM model helps the information to flow through the units without being altered by allowing only a few linear interactions. Each unit has an input, output and a forget gate which can add or remove the information to the cell state. The input gate is where the information enters in the cell and is used to learn new information from it; the forget gate determine whether the information arrived from the previous timestamp is to be remembered or is irrelevant and can be forgotten, and the third, throw the output gate the cell transmit the updated information from the current timestamp to the next timestamp. For the memory cell as input at time $t$ are the current external input vector or the inputs of the previous layers and also the output of the current memory cell. For time series forecasting problems the memory was implemented in such a way that the input units of the LSTM model perceived information from a set of past samples including a set of past delayed outputs.

The LSTM network consists of the delays and the hidden layers' sizes obtained from the time-series data by applying training data. [9]

## 4. Forecasting Results

Although the number of tested inputs in this article is chosen rather arbitrarily, increasing the number of inputs from 12 to 18 lags reduces the forecast errors and provides clear evidence on the best prediction model for each forecast horizon and in accordance with all forecast evaluation metrics, such as RMSE and MAPE (please see Table in the Appendix). Moreover, the selected best networks for the whole out-of-sample analysis perform similarly even in the post-pandemic period, particularly for horizons up to six months. For that reason, the following analysis on the performance of different types and structures of neural networks will pertain to the evaluation of models with 18 lags.

Graph 1. RMSE for the Whole Out-of-Sample Forecast Evaluation, 2019M01:2022M12

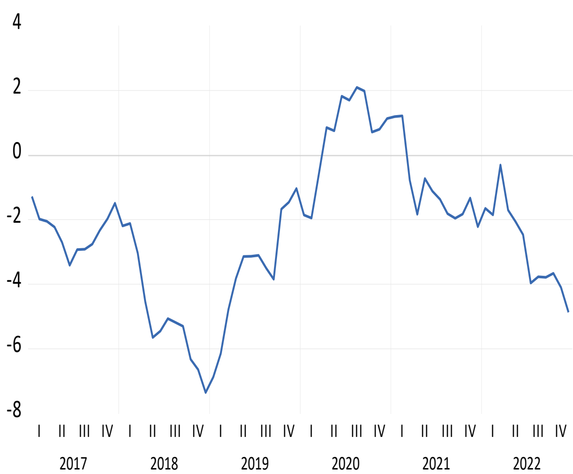



The forecast evaluation is based on the root mean square error (RMSE) metric, which is a quadratic scoring rule that measures average model prediction error. As such, lower RMSE values are an indication of better forecast ability. Graph 1 compares the performance of DFNN, DCNN and LSTM networks in predicting the lek-euro exchange rate during the whole out-of-sample period, starting from January 2019 to December 2022. It appears that none of the individual networks is shown to systematically provide superior forecasts for all of the tested horizons. The LSTM model with two hidden layers (LSTM-2L) stands out as the best prediction model in the run-up to forecast three and six months ahead, followed by the three-layered 1DCNN model (1DCNN-3L). However, they change places in the race for the 12 months horizon as the 1DCNN-3L becomes the first-best predicting model while leaving the LSTM-2L rank in second, even though the difference between their RMSEs does not seem to be significantly wide (1.98 vs 2.05, respectively).

If classed together, a certain network group appears to perform better at certain forecasting horizons, but none of them stands out as the best prediction technique at every selected horizon, namely the 3, 6 and 12 months ahead. Conjointly, the couple of LSTM models with two and three hidden layers performs on average better at horizons 3 and 6, but they rank last as a pair at the one-year horizon. Also, the LSTM models show greater RMSE difference between each other if compared to the other coupled models in the DFNN and 1DCNN methods, suggesting the structure of LSTM models is more sensitive to the number of hidden layers. Finally, prediction errors of the LSTM models inflate in line with common intuition as we forecast for longer periods ahead, unlike most of the DFNN and 1DCNN models that display lower errors at the 12 months horizon than at 6 steps ahead.

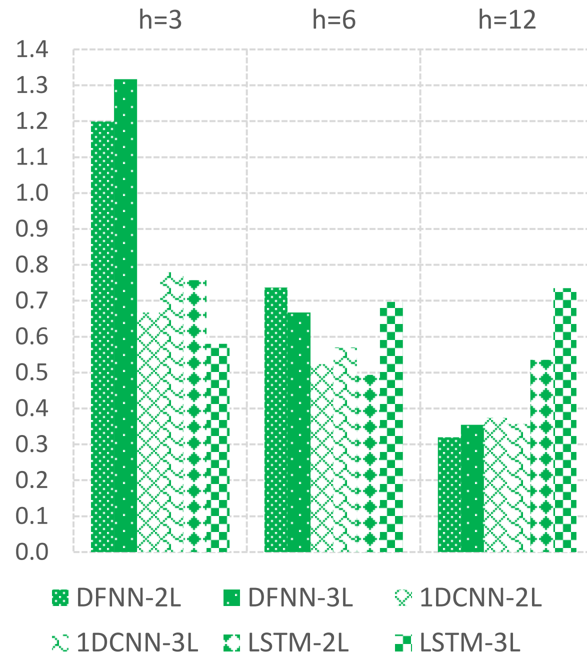
The forecast evaluation period (together with the validation sample) is characterized by an appreciation trend of the domestic currency. However, the strengthening of lek against euro has been quite uneven (please see Graph 2). In annual terms, the Albanian currency was enjoying an appreciation rate of 3.4% on average from January 2017 to March 2020, which gained momentum in 2018. The economic lockdown that aimed to contain the Covid-19 pandemic disrupted external trade of goods and services, including the tourism sector, thus leading to a year-on-year lek depreciation for the subsequent eleven months. Anyhow, the appreciation rate resumed again thereafter, averaging 2.2% for the period up to December 2022.

Graph 2. Year-on-year % change of the lek/eur exchange rate



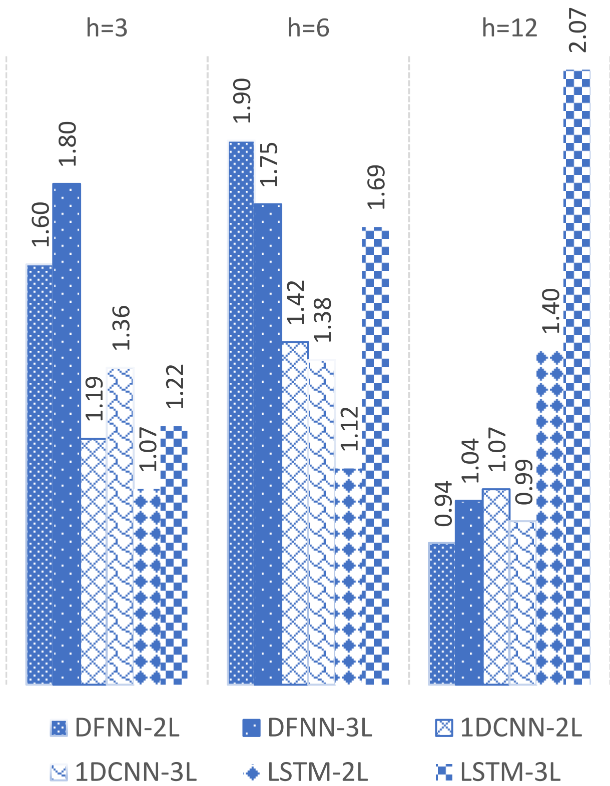

We are interested to analyze whether the neural networks ranking above holds for the entire period, or has changed in the post-Covid lockdown months. Graph 3 displays the root mean square errors after the pandemic lockdown months relative to the pre-Covid restriction period. A ratio of equal to, or greater than 1 indicates a deterioration in the forecast performance. Obviously, the forecast ability has improved by a large margin at every prediction horizon after July 2021. With the exception of the noticeable underperformance of the DFNN models at 3-steps ahead, the forecast gains of the others range between 32 and 78 percent, or an average ratio of 57% of the pre-Covid lockdown period. The improvements are most noteworthy for the DFNN and 1DCNN models, whose magnitudes of forecast errors are halved at 12-months forecast horizon.

Graph 3. RMSEs for the post-Covid relative to the pre-Covid forecast evaluation
(excluding lockdown months)




The widening differences in each model performance between sample periods have been reflected in the reconfiguration of gaps among the pairs as well as the changing of model positions at certain horizons. A quick view at Graph 4 (in comparison to Graph 1) reveals that the two-layered LSTM-2L model has confirmed its place as top-performer at horizons of 3 and 6 steps, but is ranked among the worst performers at the 12-months horizon. On the other hand, the simplest DFNN-2L model has earned itself the top position in the post-Covid lockdown period for its ability to provide the best predictions at 12 steps ahead, albeit being among worst performers at shorter horizons.

Graph 4. RMSE for the Post-Covid Lockdown Period, 2020M07:2022M12




To summarize, the performance of our selected neural network methods and their structures varies over time and depends crucially on the selected parameters, such as the number of lags and hidden layers. The long short-term memory (LSTM) network with two hidden layers could be preferred as the best choice in predicting the lek-euro exchange rate up to six months ahead. However, the model seems to lose its appeal at longer horizons, especially in the more recent period. The LSTM technique should, nevertheless, be applied with caution as increasing the number of hidden layers is shown to deteriorate and underperform considerably in comparison to other less complicated neural networks.

On the other hand, the simplest deep feedforward neural network (DFNN), which is the quintessential deep learning method, might be the least capable at short term forecast horizons such as 3 and 6 months, but it is shown very promising at predicting 12-months ahead since the economic lockdown.

## 5. Concluding Remarks

This paper investigates the applicability of different DL models to forecasting monthly LEK/EUR exchange rate. For this reason, we examine the performance of several deep learning techniques (DFNN, LSTM and 1D-CNN). For each architecture, we have used different configuration and diverse techniques to avoid overfitting of the models. The accuracy evaluation of each model was based on the out-of-sample prediction for different horizon (3, 6 and 12 months ahead), by analyzing the model estimation for pre- and post- Covid pandemic.

The comparison results show that the LSTM model with two hidden layers stands out as the best prediction model in the run-up to forecast three and six months ahead, followed by the three-layered 1DCNN model. However, they change places in the race for the 12 months horizon as the 1DCNN-3L becomes the first-best predicting model while leaving the LSTM-2L rank in second.

For the post-Covid lockdown period the two-layered LSTM-2L model has confirmed its place as top-performer at horizons of 3 and 6 steps, but is ranked among the worst performers at the 12-months horizon. On the other hand, the simplest DFNN-2L model has earned itself the top position in the post-Covid lockdown period for its ability to provide the best predictions at 12 steps ahead, albeit being among worst performers at shorter horizons. These results demonstrate the potential of deep learning techniques, and also, they emphasize the importance of well configuring, implementing and selecting the different topologies.

## 6. References


[1] Claveria O., Monte E., Soric P., & Torra S., "“An application of deep learning for exchange rate forecasting”," *University of Barcelona, Regional Quantitative Analysis Group,* 2022,revised Jan 2022..

[2] Bandara, K., Bergmeir, C., & Smyl, S, "Forecasting across time series databases using ecurrent neural networks on groups of similar series: A clustering approach.," *Expert Systems with Applications,* vol. 140, no. 112896, 2020.

[3] Lara-Benítez, Pedro et al., "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems,* vol. 31, no. 2130001, 2021.

[4] Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 11, no. 86, p. 2278–2324, 1998.

[5] Guessoum, S.; Belda, S.;Ferrandiz, J.M.; Modiri, S.; Raut, S.;Dhar, S.; Heinkelmann, R.; Schuh, H, "The Short-Term Prediction of Length," *Sensors,* vol. 22, no. 9517., 2022.

[6] Alex, "Boostedml (Articles on Statistics and Machine Learning)," 14 04 2020. [Online]. Available: https://boostedml.com/2020/04/1-d-convolutional-neural-networks-for-time-series-basic-intuition.html. [Accessed 31 01 2023].

[7] Jiang, Xia, & Chuhan Xu., "Deep Learning and Machine Learning with Grid Search to Predict Later Occurrence of Breast Cancer Metastasis Using Clinical Data.," *Journal of Clinical Medicine,* vol. 11, no. 19, p. 5772, 2022.

[8] X.J. Luo, Lukumon O. Oyedele, Anuoluwapo O. Ajayi, Olugbenga O. Akinade, Juan Manuel Davila Delgado, Hakeem A. Owolabi, Ashraf Ahmed,, "Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings," *Energy and AI,* vol. 2, no. 100015, 2020.

[9] Tuga Mauritsius, Riyanto Jayadi, Samsul Arifin et al., "ultivariate time series prediction of Covid-19 daily new cases in Indonesia based on Deep Learning: Unlocking the role of effective reproduction number," 15 December 2022.

[10] "Tutorialspoint," [Online]. Available: https://www.tutorialspoint.com/time_series/time_series_lstm_model.htm. [Accessed 5 02 2023].

[11] Poornima, S., and M. Pushpalatha, "Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units," *Atmosphere,* vol. 10, no. 11, p. 668, 2019.

## Appendix

Table 1. Forecast Evaluation of Univariate Neural Networks of the LEK-EUR Exchange Rate

| Evaluation metrics: | RMSE | | | MAE | | | MAPE | | |
|---|---|---|---|---|---|---|---|---|---|
| Forecast horizons: | 3 | 6 | 12 | 3 | 6 | 12 | 3 | 6 | 12 |
| Whole out-of-sample forecast evaluation period (Jan-2019 to Dec-2022) | | | | | | | | | |
| *Neural Networks = 12 inputs* | | | | | | | | | |
| DFNN (2 layers) | 1.35 | 1.93 | 1.91 | 1.08 | 1.50 | 1.37 | 0.89 | 1.22 | 1.11 |
| DFNN (3 layers) | 1.45 | 1.82 | 1.92 | 1.20 | 1.40 | 1.51 | 0.99 | 1.14 | 1.23 |
| 1DCNN (2 layers) | 1.44 | 2.17 | 1.96 | 1.11 | 1.71 | 1.42 | 0.91 | 1.39 | 1.15 |
| 1DCNN (3 layers) | 1.76 | 1.98 | 1.78 | 1.39 | 1.52 | 1.32 | 1.14 | 1.24 | 1.07 |
| LSTM (2 layers) | 1.60 | 1.84 | 2.10 | 1.16 | 1.42 | 1.52 | 0.96 | 1.16 | 1.23 |
| LSTM (3 layers) | 1.44 | 2.24 | 2.28 | 1.23 | 1.73 | 1.70 | 1.01 | 1.41 | 1.38 |
| *Neural Networks = 18 inputs* | | | | | | | | | |
| DFNN (2 layers) | 1.60 | 2.31 | 2.09 | 1.32 | 1.90 | 1.47 | 1.08 | 1.54 | 1.19 |
| DFNN (3 layers) | 1.66 | 2.22 | 2.10 | 1.40 | 1.78 | 1.50 | 1.15 | 1.45 | 1.22 |
| 1DCNN (2 layers) | 1.60 | 2.22 | 2.14 | 1.25 | 1.72 | 1.57 | 1.02 | 1.40 | 1.27 |
| 1DCNN (3 layers) | 1.55 | 1.95 | 1.98 | 1.21 | 1.49 | 1.41 | 0.99 | 1.22 | 1.15 |
| LSTM (2 layers) | 1.34 | 1.81 | 2.05 | 1.03 | 1.34 | 1.59 | 0.85 | 1.09 | 1.29 |
| LSTM (3 layers) | 1.64 | 2.21 | 2.45 | 1.26 | 1.72 | 1.95 | 1.03 | 1.40 | 1.59 |
| Post-Covid lockdown period (July, 2020 to December, 2022) | | | | | | | | | |
| *Neural Networks = 12 inputs* | | | | | | | | | |
| DFNN (2 layers) | 1.22 | 1.33 | 0.96 | 0.93 | 1.09 | 0.75 | 0.77 | 0.90 | 0.62 |
| DFNN (3 layers) | 1.47 | 1.36 | 1.34 | 1.18 | 1.07 | 1.07 | 0.98 | 0.88 | 0.88 |
| 1DCNN (2 layers) | 1.10 | 1.51 | 0.97 | 0.86 | 1.29 | 0.75 | 0.71 | 1.06 | 0.62 |
| 1DCNN (3 layers) | 1.59 | 1.45 | 0.90 | 1.25 | 1.13 | 0.72 | 1.04 | 0.93 | 0.59 |
| LSTM (2 layers) | 1.75 | 1.34 | 0.94 | 1.22 | 1.11 | 0.74 | 1.02 | 0.91 | 0.61 |
| LSTM (3 layers) | 1.45 | 1.51 | 1.52 | 1.26 | 1.27 | 1.08 | 1.03 | 1.04 | 0.89 |
| *Neural Networks = 18 inputs* | | | | | | | | | |
| DFNN (2 layers) | 1.60 | 1.90 | 0.94 | 1.32 | 1.64 | 0.74 | 1.09 | 1.34 | 0.61 |
| DFNN (3 layers) | 1.80 | 1.75 | 1.04 | 1.51 | 1.46 | 0.85 | 1.24 | 1.19 | 0.70 |
| 1DCNN (2 layers) | 1.19 | 1.42 | 1.07 | 0.96 | 1.18 | 0.84 | 0.79 | 0.97 | 0.69 |
| 1DCNN (3 layers) | 1.36 | 1.38 | 0.99 | 1.04 | 1.13 | 0.80 | 0.86 | 0.93 | 0.66 |
| LSTM (2 layers) | 1.07 | 1.12 | 1.40 | 0.85 | 0.88 | 1.14 | 0.70 | 0.72 | 0.94 |
| LSTM (3 layers) | 1.22 | 1.69 | 2.07 | 0.98 | 1.33 | 1.64 | 0.81 | 1.09 | 1.35 |

*Note: RMSE = root mean square error; MAE = mean absolute error; MAPE = mean absolute percent error.*