Özyeğin University
CS454 Introduction to Machine Learning and Artificial Neural Networks Project

Multivariate Currency Exchange Prediction Using LSTM, SVR, and MLP

Furkan Cantürk
BSc. Industrial Engineering

Yuşa Yalçın
BSc. Computer Science

01.11.2020

***Abstract* - In today's global economy, accuracy in predicting the exchange rate or at least predicting the trend correctly is of crucial importance for any future investment. In this report, we developed and investigated three models using Support Vector Regression (SVR), Multi-Layer Perceptron (MLP) and Long-Short Term Memory (LSTM) to predict USD/TL, EUR/USD, and USD/JPY rates. These 3 models are compared with 3 datasets of the above exchange rates whose features are past currency exchange rates, inflation rates and interest rates in corresponding countries.**

I. Introduction

Exchange rates tell how valuable a currency is. The economic success of a country can be determined with the exchange rate. Exchange rates are the rate at which one local currency is converted to another [2]. Think of it as the price taken to buy that currency. Having a weak currency has mostly disadvantages for countries. Weaker currency makes exports cheaper and imports more expensive. If the currency is weak, investors will not hold it. So, currency may lose more value because of that. In addition, weak currency affects people who travel abroad. Since currency is weak, everything is more expensive. On the other hand, a strong currency has advantages. Exports cost less and travelling abroad is cheaper.
Foreign exchange traders decide the exchange rate for most currencies. In addition, interest rates and inflation rates are also affecting the currencies. In our project, we predicted future exchange rates. To predict future exchange rate we used: interest rates, inflation rates and exchange rates. High interest rates increase the value of a currency because investors prefer to invest money in high interest rate currencies. If the interest rate is low, this will cause a decrease in the currency and also in the exchange rates. On the other hand, this is the opposite for the inflation rate. If the inflation rate decreases, this causes the exchange rate to increase but if it increases, it causes to decrease.

The historical data is the most important player in the forecasting process. The purpose of this project is to investigate and compare three prediction techniques with three different datasets. Three prediction techniques that we used are Support Vector Regression (SVR), Multi-Layer Perceptron (MLP) and Long-Short Term Memory (LSTM). We used the datasets of Turkey, Japan and European Union central banks. Interest and inflation rates of the central banks and USD/TL, EUR/USD, and USD/JPY past exchange rates are used to make future predictions for the given exchange rates.

II. Learning Techniques

Support Vector Regression (SVR)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in different side.
SVR is the regression version of SVM.

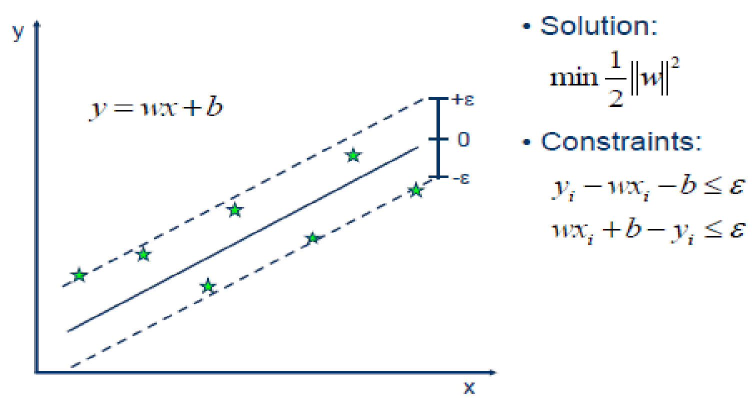

Figure 1. SVR[1]

Coming to the major part of the SVM for which it is most famous, the kernel trick. The kernel is a way of computing the dot product of two vectors x and y in some (very high dimensional) feature space, which is why kernel functions are sometimes called "generalized dot product. Applying kernel trick means just to replace dot product of two vectors by the kernel function. [6]

Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words, C behaves as a regularization parameter in the SVM.

Multi-Layer Perceptron (MLP)

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). There can more than one linear layer. Figure 2 is a three-layer network. First layer is input, last layer is output and middle ones are hidden layers. The number of hidden layers can be increased according to the task.

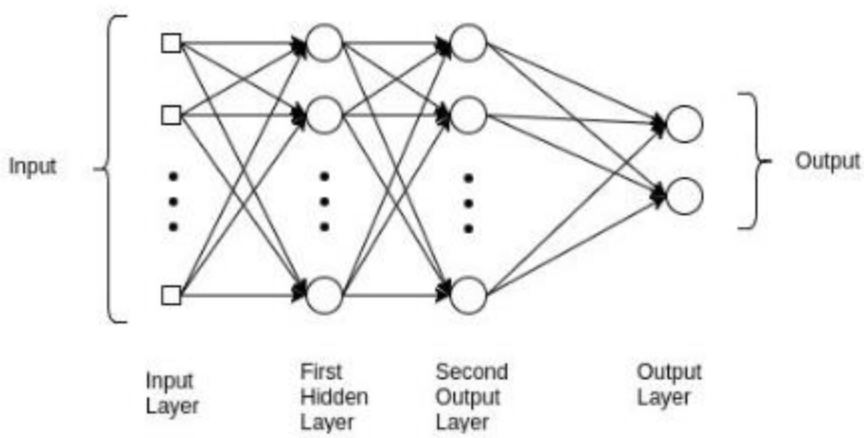

Figure 2. MLP Representation [3]

To train the model, we just pass the input to model and multiply it with weights which are initialized by random values. Then, bias is added to each layer and find the calculated output of the model.

When we pass the data sample, we get some output from the model called predicted output, and we have a label that contains the real output or the expected output. Based on these, we calculate the loss that we have to backpropagate using Backpropagation algorithm.

After calculating the loss, we backpropagate the loss and update the model's weights using a gradient. In this step, the weights will be adjusted according to the gradient flow in this direction

```
Model: "MLP"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_48 (Dense)             (6, 24, 100)              400
_________________________________________________________________
input_layer_dropout (Dropout (6, 24, 100)              0
_________________________________________________________________
dense_49 (Dense)             (6, 24, 20)               2020
_________________________________________________________________
flatten_11 (Flatten)         (6, 480)                  0
_________________________________________________________________
dense_50 (Dense)             (6, 1)                    481
=================================================================
Total params: 2,901
Trainable params: 2,901
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. Our MLP Model Summary

We used a dropout layer which eliminates 25% of existing units' weights.

Long-Short Term Memory

LSTM is an artificial recurrent neural network (RNN) architecture which is used in deep learning [7]. LSTM is capable of learning long-term dependencies. With LSTM, the information flows through a mechanism known as cell states. By this way, LSTM can selectively remember or forget things. The information at a particular cell state has three different dependencies.

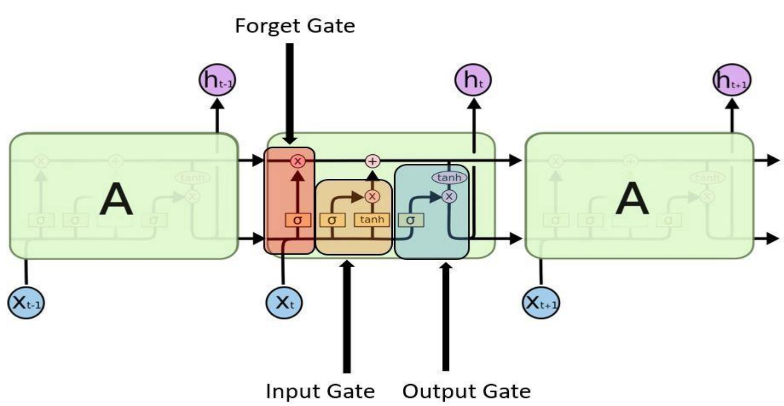

Figure 4. LSTM Unit Representation [5]

Information is kept by the cells and the memory manipulations are done by the gates. There are three gates:

Forget Gate: The information that no longer is used in the cell state is removed with the forget gate. Two inputs x_t (input at the particular time) and h_t-1 (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for the output 1, the information is kept for future.

Input gate: Addition of useful information to the cell state is done by input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered using inputs *h_t-1* and *x_t*. Then, a vector is created using *tanh* function that gives output from -1 to +1, which contains all the possible values from h_t-1 and *x_t*. At the end, the values of the vector and the regulated values are multiplied to obtain information.

Output gate: The task of extracting information from the current cell state to be presented as an output is done by output gate. A vector is generated by applying tanh function on the cell. Then, information is regulated using the sigmoid function and filter the values to be remembered using inputs *h_t-1* and *x_t*. At the end, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell. [4]

```
Model: "RNN"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_layer (LSTM)           (6, 100)                  41600
_________________________________________________________________
input_layer_dropout (Dropout (6, 100)                  0
_________________________________________________________________
leaky_re_lu_11 (LeakyReLU)   (6, 100)                  0
_________________________________________________________________
dense_47 (Dense)             (6, 1)                    101
=================================================================
Total params: 41,701
Trainable params: 41,701
Non-trainable params: 0
_________________________________________________________________
```

Figure 5. Our LSTM Model Summary

We used a dropout layer which ignores 25% of the units' weights to prevent overfitting. Also, it is observed that LeakyReLU function, which allows a small gradient when the unit is not active, make the model more accurate than ReLU. So, we preferred it as the activation function.

III. About Data

We used datasets which are gathered from the central banks of Turkey, Japan, and European Central Bank. We set the frequency for each feature as months because yearly inflation and yearly interest rates are announced usually monthly. It might be a considerable further study of using daily past exchange rates, and monthly interest and inflation rates.

We preprocessed data sets to get a common date range and frequency among features. Data is scaled between 0 and 1 according to minimum and maximum values within features.

Training datasets are formed by the first 0.75 split of all data.

Table 1. EUR/USD Statistics: 192 instances 2003-01-01 to 2019-01-20

	Min	Max	Range
Exchange	1.045	1.58	0.535
Inflation	-0.6	4.1	4.7
Interest	1.79	5.59	3.8

Table 2. USD/TL Statistics: 162 instances from 2005-01-01 to 2018-08-01

	Min	Max	Range
Exchange	1.15	4.99	3.84
Inflation	3.99	17.9	13.91
Interest	11.81	33.82	22.01

Table 3. USD/JPY Statistics: 184 instances from 2002-01-01 to 2017-04-01

	Min	Max	Range
Exchange	76.15	133.45	57.3
Inflation	-2.5	3.7	6.2
Interest	0.1	0.75	0.65

IV. Computational Results

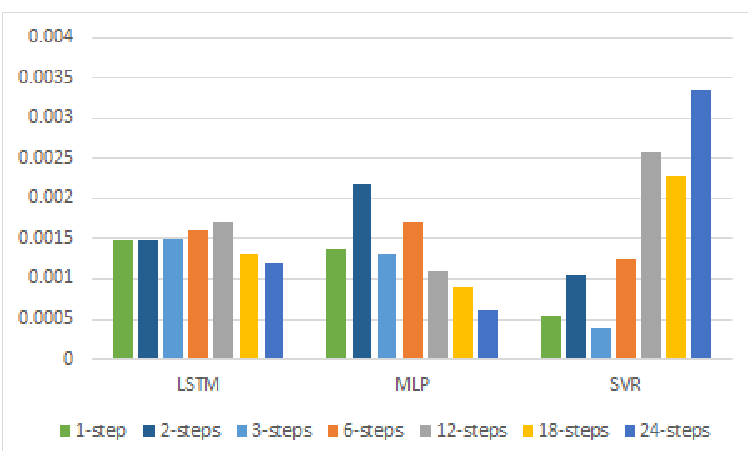

Figure 6. Training MSEs for EUR/USD

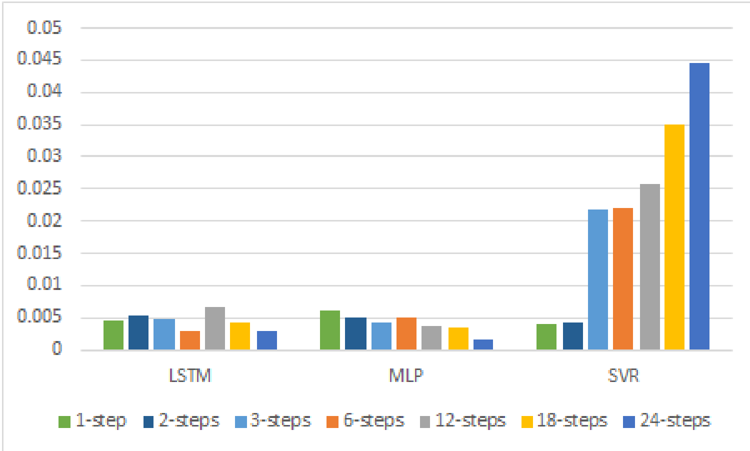

Figure 7. Training MSEs for USD/TL

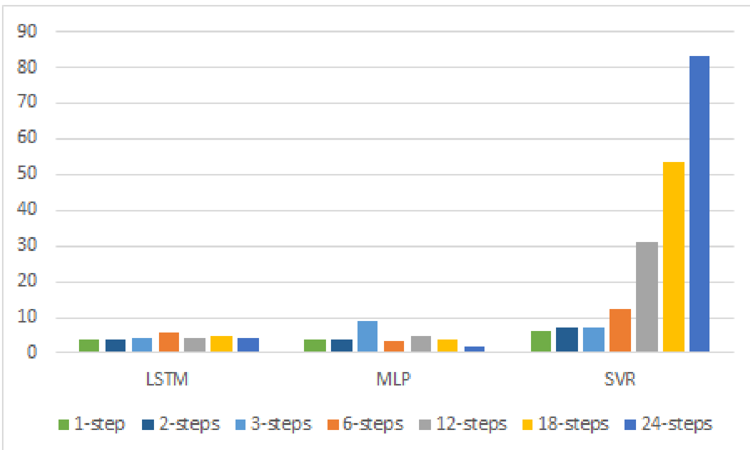

Figure 8. Training MSEs for USD/JPY

LSTM, and MLP do not show a remarkable change when time steps become larger whereas SVR performs poorly with high time steps. It shows that LSTM and MLP eliminate the effect of old features during the training, and SVR failed with high cardinality as expected.

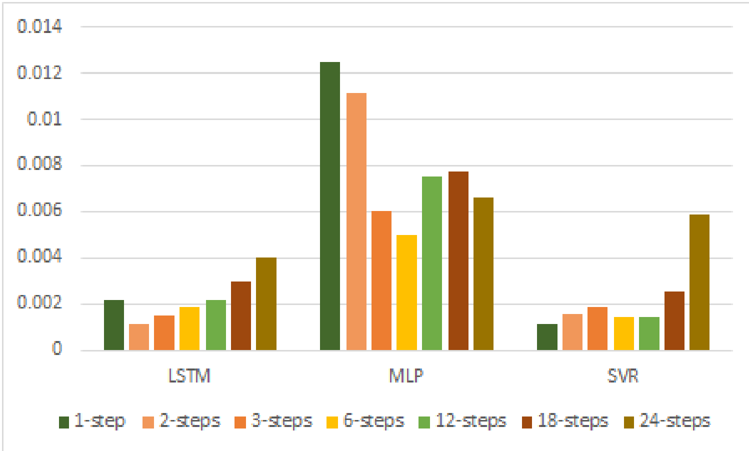

Figure 9. Test MSEs for EUR/USD

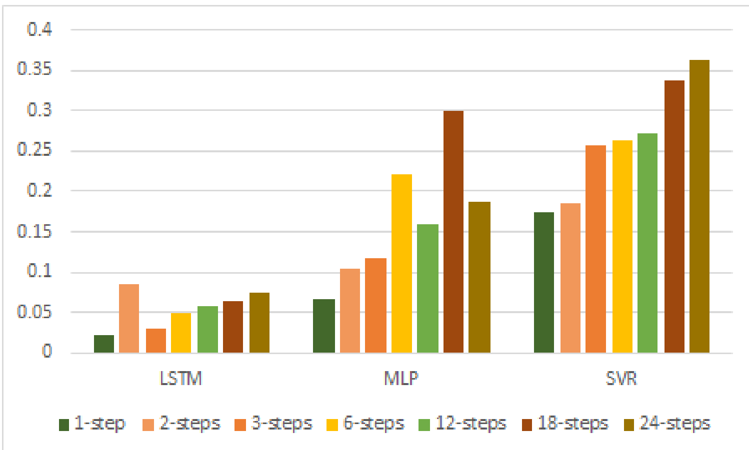

Figure 10. Test MSEs for USD/TL

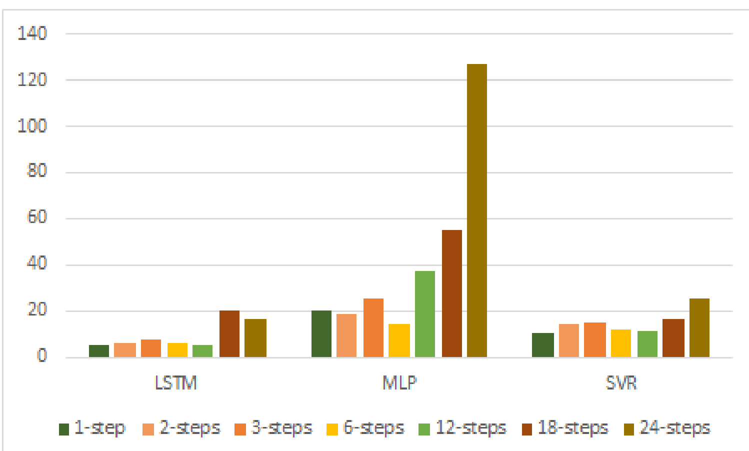

Figure 11. Test MSEs for USD/JPY

According to the test results, LSTM is a more regularized model for each currency exchange prediction whereas MLP seems overfitted. Besides, the test results of SVR for EUR/USD and

USD/JPY predictions shows that it does not perform badly as much as in the training when it is compared to the other two models.

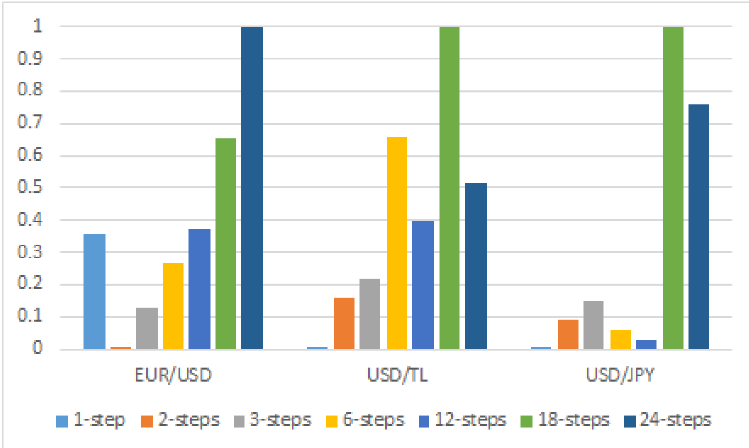

Figure 12. Min-Max Scaled MSEs of LSTM

Since each currency exchange have different value ranges, MSEs are scaled between 0 and 1 within each dataset results.

For EUR/USD prediction, LSTM performs best with 2 time steps. Since range for EUR/USD is so narrow, prediction becomes more sensitive. So, considering only last month levels of exchange, inflation, and interest may be deceptive.

For USD/TL prediction, 1-step LSTM becomes a successful predictor. It shows us that USD/TL is not a steady currency exchange. In other words, the most effective rates are last month's.

For USD/JPY prediction, LSTM performs well with 1 to 12 time steps. It shows that LSTM can success to penalize weights of the older time input units when it is trained with USD/JPY data set.

	Min	Max	Value Range	Best MSE	MSE/Range	MSE/Median
EUR/USD	1.045	1.58	0.535	0.001135	0.21%	0.09%
USD/TL	1.15	4.99	3.84	0.066826	1.74%	2.18%
USD/JPY	76.15	133.45	57.3	4.9492	8.64%	4.72%

Figure 13. MSE-Value Range Ratios of 3 Currency Exchanges

To evaluate which currency exchange prediction is done better by LSTM model, the lowest MSE and range value of exchange currency (MSE/Range), and the lowest MSE and median exchange currency value (MSE/Median) proportions are observed. LSTM performs best on EUR/USD prediction. It is obvious that the reason is EUR/USD is the most stable exchange.

V. Conclusion

In this project, we investigated three learning models to predict three exchange rates which are USD/TL, EUR/USD, and USD/JPY. To make predictions, past exchange rates, inflation rates and interest rates are used. LSTM, MLP and SVR models are trained with 1, 3, 6, 12, 18, 24 time steps for each currency exchange.

Large time windows like 18 and 24 months does not give healthy information to predict next month currency. LSTM performs well to predict next month level for each type of exchange

	EUR/USD	USD/TL	USD/JPY
LSTM	2	1	1
MLP	6	1	6
SVR	1	1	1

Figure 14. Time steps of the models for the best MSE

REFERENCES

[1]Support Vector Machine - Regression(SVR). (n.d). Available at https://www.saedsayad.com/support_vector_machine_reg.htm

[2] 5 Key Factors That Affect Currency Exchange Rates. (n.d). Available at https://treasuryvault.com/articles/5-key-factors-that-affect-currency-exchange-rates/

[3] N. Kain (2018). Understanding of Multilayer Perceptron. (n.d). Available at https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f

[4] Deep Learning | Introduction to Long Short Term Memory. (n.d). Available at https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[5] A. Mittal, (2019). Understanding RNN and LSTM. Available at https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[6] A. Yadav (2018). Support Vector Machines (SVM). Available at https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589

[7] Long short-term memory (2015). Available at https://en.wikipedia.org/wiki/Long_short-term_memory