

Forex Forecasting*

*Note: All authors contributed equally to this project

Eustathios Kotsis
StudentID: 7115152200008
efstathkot@di.uoa.gr

Michael Darmanis
StudentID: 7115152200004
mdarm@di.uoa.gr

Vasilios Venieris
StudentID: 7115152200017
vvenieris@di.uoa.gr

Abstract—The foreign exchange (forex) market, a complex and volatile entity, has been a subject of extensive research, especially in the domain of predicting currency movements. With only 2% of retail traders successfully predicting currency movement, the challenge is evident. This study implements two variants of machine learning and its statistical models in forex market forecasting. Through a hybrid implementation of RNNs, CNNs, and Exponential Smoothing, we observed that the purely statistical methods often outperformed the hybrid models, especially when sufficient, and good quality data was not available.

Index Terms—time series analysis, forex forecasting, exponential smoothing, vector autoregression, convolutional neural networks, recurrent neural networks, hybrid methods

I. INTRODUCTION

The forex market stands as the world's largest financial market, with a daily volume surpassing \$6.6 trillion. Its non-centralised nature, operating 24 hours a day, sets it apart from other financial markets. The inherent high volatility, nonlinearity, and irregularity make the forex market one of the most intricate to navigate [8]. Traders in this market deal with currency pairs, comparing the value of one currency to another. Over the years, two primary techniques, fundamental and technical analysis, have been employed for forecasting. However, with the advent of machine learning, a surge in research activity has been observed, aiming to take advantage of these advanced models for more accurate forex-market predictions.

II. PREVIOUS WORK

The realm of forex-market-forecasting, using machine learning, has witnessed significant contributions. Fletcher [9] emphasised the efficacy of discriminative techniques like Support Vector Machines (SVM), Relevance Vector Machines (RVM), and Neural Networks when integrated with advanced exogenous financial information. Panda et al. [12] conducted an SLR on Exchange Rate Prediction, introducing novel models like Artificial Neural Network (ANN) and Support Vector Regression (SVR). Their findings highlighted the potential of these models in predicting exchange rate projections.

Islam et al. [10] explored the recent advancements in FOREX currency prediction using machine-learning algorithms. Their research, spanning from 2017 to 2019, revealing a growing academic interest in neural network models and

optimisation methodologies. Deep learning algorithms, especially the gated recurrent unit (GRU) and long short-term memory (LSTM), emerged as promising contenders in time series prediction.

Ryll & Seidens [11] undertook a comprehensive review of over 150 publications on machine learning in financial market forecasting. Their analysis suggested that machine-learning algorithms, particularly recurrent neural networks, outperformed traditional stochastic methods in financial market forecasting.

Based on the above insights, our approach focused on using the strengths of Exponential Smoothing [13], a purely statistical model, in conjunction with the capabilities of RNNs and CNNs, in order to develop hybrid models suitable for forex-forecasting. A Vector Autoregression model was also employed to benchmark our data.

III. OUR APPROACH

Our initial approach to forex-market-forecasting lied on neural models. The aim was to supplement them with statistical methods. However, the results from this combination were mediocre at best. We then reconsidered our strategy. Instead of primarily relying on a neural model, we turned to the Holts-Winter exponential smoothing technique [5], a well-regarded statistical forecasting method. Using this as a foundation, we then looked into enhancing its predictive power by incorporating Recurrent Neural Networks (RNNs).

The second approach actually made a lot of sense. While the Holts-Winters model has its merits, it inherently assumes that the underlying trend in the data is linear [5]. Forex markets are, by their very nature, dynamical systems whose trends often exhibit non-linear characteristics [8]. Thus, incorporating an RNN architecture for the level seemed like the way to go.

In the following part of the report, we will briefly summarise how these two hybrid models work, how we assessed their performance, the experiments performed and the conclusions drawn thereof.

IV. HYBRID MODELS

A. Smoothed Convolutional Neural Networks (S-CNN)

As mentioned earlier, the core idea here was to enhance the CNN's ability of forecasting by using a purely statistical method. For that purpose we used a simple exponential smoothing technique, which enhances forecasting accuracy [6]

by calculating an average of previous values. The simplest form of exponential smoothing is given by Equation 1:

$$s_t = \alpha x_t + (1 - \alpha)s_{t-1} = s_{t-1} + \alpha(x_t - s_{t-1}). \quad (1)$$

Note that the significance of past observations diminishes exponentially as one moves further into the historical data. Wibawa et al. [6] method particularly beneficial for erratic and fluctuating datasets.

Initially, the data undergoes "exponential smoothing," emphasising recent observations, while gradually reducing the weight of older data, effectively reducing noise and capturing the primary trend.

Post-smoothing, individual CNNs are trained for every single time series, allowing the model to thoroughly learn each serie's distinct characteristics. These CNNs are not only designed to predict the next value, but also to forecast multiple future points simultaneously.

B. Exponential Smoothing Recurrent Neural Networks (ES-RNN)

This approach takes from the ES-RNN algorithm, winner of the M4-competition [1], discussed in detail in a blog post by Slawek [7]. The algorithm is divided into two distinct layers: a pre-processing layer that uses exponential smoothing and a LSTM layer that updates the per-series parameters of the Holts-Winter model.

In the pre-processing layer, Holts-Winter exponential smoothing [5] with multiplicative seasonality and trend is applied via

$$l_t = \alpha\left(\frac{y_t}{s_{t-m}}\right) + (1 - \alpha)l_{t-1}b_{t-1} \quad (2)$$

$$b_t = \beta\left(\frac{l_t}{l_{t-1}}\right) + (1 - \beta)b_{t-1} \quad (3)$$

$$s_t = \gamma\frac{y_t}{l_{t-1}b_{t-1}} + (1 - \gamma)s_{t-m} \quad (4)$$

$$\hat{y}_{t+h} = l_t b_t^h s_{t-m+h_m^+} \quad (5)$$

where l is a state variable for level, b is a state variable for trend and s is a multiplicative seasonality coefficient. α, β and γ are smoothing coefficients between zero and one. An h step forecast is given by Equation 5.

Since the model no longer considers local linear trend¹, Equation 3 is replaced by an RNN from the model as follows:

$$\hat{y}_{t+1\dots t+h} = RNN(X_t) * l_t * s_{t+1\dots t+h} \quad (6)$$

$$x_i = \frac{y_i}{l_t s_i} \quad (7)$$

X_t is a vector of normalised², de-seasonalised features of which a scalar component x_t is calculated via Equation 7.

It is important to note that the RNN and the classical Holts-Winters parameters detailed are jointly trained. The power of the ES-RNN [4] lies in the co-training of both the per-time series Holts-Winters parameters and the general RNN parameters.

¹remember, that is one of the caveats of the Holts-Winters model

²normalised using the level

So, to sum up, the series passes through a pre-training layer, gets its seasonality and level extracted, becomes normalised and smoothed by them, and then gets fed to a RNN in order to capture its trend. The output of the RNN is then used in Equation 6 in order to reconstruct the time-series-projection in a multiplicative³ manner.

V. PERFORMANCE MEASURES FOR POINT FORECASTS

There are many measures available in the forecasting literature for evaluating the performances of forecasting methods [3]. We decided to use two popular ones, namely, the symmetric mean absolute percentage error (sMAPE⁴; Makridakis, 1993 [2]) and the mean absolute error (MAE; Hyndman & Koehler, 2006 [3]). We did that in the hope of not overcomplicating things, while, at the same time, trying to maintain some level of objectivity.

VI. DATA DESCRIPTION

The dataset, sourced from the European Central Bank [14], encompasses historical exchange rates of various currencies against the Euro, typically released around 16:00 CET, on a daily basis. These rates reflect the official currencies of non-euro area Member States of the European Union and major world currencies active in spot FX markets. A preview can be seen on Table I. The data spans from 1999 to 2023, illustrating

TABLE I
ECB FOREX DATASET

Date	USD	JPY	BGN	...	THB	ZAR
2023-09-15	1.0658	157.50	1.9558	...	38.145	20.2968
2023-09-14	1.0730	158.13	1.9558	...	38.387	20.3109
2023-09-13	1.0733	158.28	1.9558	...	38.397	20.3300
...
1999-01-04	1.1789	133.73	NaN	...	NaN	6.9358

fluctuations of currencies like USD, JPY, BGN, THB, and ZAR against the Euro.

VII. EXPERIMENTS

For both approaches (and the benchmark), a common pre-processing step was performed; to equalise the serie's lengths. The series in the ECB dataset are, as shown in Table I of variable length because of NaN values. So to simplify the vectorisation implementation, we equalised all series to a fixed length by frequency and disregard all series beginning (or ending) with (trailing) NaN values.

For the above reasons several currencies were excluded from our analysis. The list of discarded currencies is presented in the Table II.

Also, in order to deal with various of frequencies, we resampled the ECB dataset into weekly, monthly, quarterly and yearly values.

³this can also be done in an additive manner, our implementation contains both options

⁴this was actually part of the M4-Competition's performance metric known as the overall weighted average (OWA)

TABLE II
LIST OF EXCLUDED CURRENCIES

BGN	CYP	EEK	LTL	LVL	MTL
ROL	RON	SIT	SKK	ISK	HRK
RUB	TRL	TRY	BRL	CNY	IDR
ILS	INR	MXN	MYR	PHP	THB
Unnamed: 42					

As for the testing, we followed the prediction rules as dictated by the M4 Competitio [1]. A summary of the forecasting horizons, for the different types of time series frequencies, is seen in Table III.

TABLE III
FORECASTING HORIZONS IN M4 COMPETITION [1]

Frequency	Forecasting Horizon
Yearly	6
Quarterly	8
Monthly	18
Weekly	13
Daily	14
Hourly	48

A. S-CNN

Data normalization played a crucial role in addressing the dynamic and nonlinear nature of most time-series. The primary objective of data normalization was to ensure data quality before feeding it into any model, as it significantly impacts the model’s performance.

1) *Architecture*: The raw input data was treated as a 1D array, which was then read and filtered by the CNN model. The model was experimented with varying numbers of lucas hidden layers, specifically 3, 11, 47, and 76. For each combination of frequency, training length, and currency, the model was trained for 200 epochs. A specific architecture of CNN, known as WaveNet [6], was finally chosen for its superior performance in forecasting financial time-series.

2) *Training*: We implemented the model for handling one-dimensional time series, and, regardless of its frequency or currency, normalised and the smoothed using the simple exponential smoothing.

The output from each series served as an input into the convolutional model. The objective was to train this model to forecast a horizon of ‘K’ steps according to Table III.

A balance was also sought in selecting the optimal series length for training. A very short series length might miss out on identifying potential trends or seasonality, while an excessively long one might overemphasise past events or mistakenly identify non-existent periodic patterns. To strike a balance, different series lengths were used for each frequency.

B. ES-RNN

1) *Architecture*: Our ES-RNN model contained two methodologies: the Holts-Winters time series forecasting technique and a Recurrent Neural Network (RNN). The Holt-

Winters component captured the time series’ level and seasonality, while omitting trends. Conversely, the RNN, structured as a Gated Recurrent Unit (GRU) using 8 hidden-state-features, processes the Holt-Winters’ deseasonalised and de-leveled output. The RNN’s predictions were then channeled through a linear layer to yield the final forecasts, which were multiplied with the previously extracted seasonality and level in order to make the predictions.

2) *Training*: The ES-RNN model underwent iterative training across multiple epochs and using a validation set in batches. In order to avoid overfitting, we implemented an early stopping criterion, of 2 steps patience and 0.01 threshold. Holts-Winters coefficients for level (α) and seasonality (γ) were trained, while the L1-loss was used as the loss function because of its similarity with the sMAPE; both of them are mean absolute differences between forecasted and actual values, though differently normalised⁵.

To enhance the robustness, and to simulate a variety of historical contexts, we adopted a random segment-based training approach. Rather than consistently using a fixed window of historical data for predictions, we extracted random sub-segments of the time series during training. Each epoch introduced a series of random starting points within the available time series data. This created a segment which served as the recent history for model training, with the first part being the input (input window) and the subsequent prediction points being the target (output window). Both the input and output windows were, heuristically, taken equal to the horizon number. By constantly altering the perspective of the model, we aimed to achieve a more generalised understanding of the dataset’s underlying patterns.

Training was performed in all frequencies, using both validation and testing, the suggested horizon from Table III.

VIII. RESULTS

The results of both hybrid models and the benchmark’s, for daily and weekly frequencies, are presented in Table IV. This table is representative of the overall performance the models had considering all used urrencies.

The V-AR model appears to be the best performer when considering both sMAPE and MSE. The ES-RNN, while having low sMAPE values, has significantly high MSE values, indicating potential issues. The S-CNN has the highest sMAPE values, suggesting it is the least accurate among the three.

The results of the S-CNN were Indeed not impressive. Although some long term patters were captured with, mainly for relatively stable currencies and for big frequencies like quarterly and monthly, the model seems unable to handle non-stationary data. Also the architecture might have not been well-suited for the underlying patters of the ECB dataset, or it might had even overfitted. It is also a well-known fact that high levels of noise in the data, greatly affect a CNNs ability to discern patterns.

⁵although again a bit differently (divided by level), so the consideration was that they are similar enough

TABLE IV
AVERAGED PERFORMANCE ESTIMATORS FOR DAILY AND WEEKLY FREQUENCIES

	Daily		Weekly	
	<i>sMAPE</i> (%)	<i>MSE</i>	<i>sMAPE</i> (%)	<i>MSE</i>
S-CNN	59.70	0.006	81.45	0.010
ES-RNN	13.72	560.580	7.96	323.021
V-AR	0.83	0.001	2.24	0.000

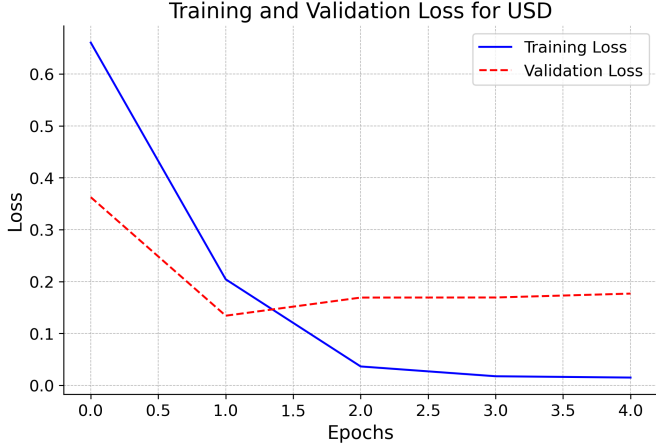


Fig. 1. Loss during ES-RNN training, for the USD, for a Weekly frequency.

The choice of choosing the L1 as the training loss function seems to have backfired. On a qualitative assessment, the system tended to have a positive bias (see Figure 1).

The above results emphasise the importance of having ample data when deploying hybrid algorithms, and suggests that traditional statistical methods can still hold their ground in such scenarios.

IX. CONCLUSIONS

In this project we tried blending neural-network techniques with traditional statistical models for forex market forecasting. By using RNNs, CNNs, and Exponential Smoothing together, we hoped to improve predictions. However, our results showed that the V-AR outperformed both of the latter. The training loss function for the ES-RNN might not have been the best choice, as our system consistently leaned towards a positive bias, while the S-CNN did not seem able to handle non-stationary, and short data. In all cases, the old-school statistical methods worked better than the newer hybrid ones did. This tells us that having enough data is crucial, and sometimes, the simpler methods still work best.

ACKNOWLEDGMENTS

This report was typeset using \LaTeX , originally developed by Leslie Lamport and based on Donald Knuth's \TeX . A template that can be used to format documents with this look and feel has been released under the [IEEE LICENSE](https://www.ieee.org/publications_standards/publications/rights/index.html), and can

be found online at <https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhncsqn>.

REFERENCES

- [1] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: 100,000 time series and 61 forecasting methods," *Int. J. Forecasting*, vol. 36, no. 1, pp. 54–74, 2020. <https://doi.org/10.1016/j.ijforecast.2019.04.014>
- [2] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *Int. J. Forecasting*, vol. 9, no. 4, pp. 527–529, 1993. [https://doi.org/10.1016/0169-2070\(93\)90079-3](https://doi.org/10.1016/0169-2070(93)90079-3)
- [3] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
- [4] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *Int. J. Forecasting*, vol. 36, no. 1, pp. 75–85, 2020. <https://doi.org/10.1016/j.ijforecast.2019.03.017>
- [5] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management science*, vol. 6, no. 3, pp. 324–342, 1960. <https://doi.org/10.1287/mnsc.6.3.324>
- [6] A.P. Wibawa, A.B.P. Utama, H. Elmunsyah, et al., "Time-series analysis with smoothed Convolutional Neural Network," *J Big Data*, vol. 9, pp. 44, 2022. <https://doi.org/10.1186/s40537-022-00599-y>
- [7] Slawek Smyl, Jai Ranganathan, Andrea Pasqua, "M4 Forecasting Competition: Introducing a New Hybrid ES-RNN Model." Accessed 15 Sep 2023. <https://www.uber.com/en-GR/blog/m4-forecasting-competition/>
- [8] M. Ayitey Junior, P. Appiahene, O. Appiah, et al., "Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis," *J Big Data*, vol. 10, pp. 9, 2023. <https://doi.org/10.1186/s40537-022-00676-2>
- [9] TSB. Fletcher, "Machine learning for financial market prediction," 2012, p. 207. <http://discovery.ucl.ac.uk/1338146/>. Accessed 15 Sep 2023.
- [10] M.S. Islam, E. Hossain, A. Rahman, M.S. Hossain, and K. Andersson, "A review on recent advancements in FOREX currency prediction," *Algorithms*, vol. 13, no. 8, pp. 1–23, 2020. <https://doi.org/10.3390/A13080186>
- [11] L. Ryll and S. Seidens, "Evaluating the performance of machine learning algorithms in financial market forecasting: a comprehensive survey," 2019. <http://arxiv.org/abs/1906.07786>.
- [12] M.M. Panda, S.N. Panda, and P.K. Pattnaik, "Exchange rate prediction using ANN and deep learning methodologies: a systematic review," in *Indo-Taiwan 2nd International Conference on Computing, Analytics and Networks, Indo-Taiwan ICAN 2020—Proceedings*, 2020, pp. 86–90. <https://doi.org/10.1109/Indo-TaiwanICAN48429.2020.9181351>.
- [13] R. G. Brown, "Exponential Smoothing for Predicting Demand," Cambridge, Massachusetts: Arthur D. Little Inc., p. 15, 1956. <https://www.industrydocuments.ucsf.edu/tobacco/docs/#id=jzlc0130>
- [14] European Central Bank, "Euro foreign exchange reference rates," 2023. https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html. Accessed 15 Sep 2023.