Faculty of
Computer Science and Financial engineering

Valentin Parisot

Forecasting the CHF/USD exchange rate with machine learning methods

Supervisors:
Prof. Luisa Lambertini
Prof. James Larus
Scientist. Stylianos Tsiaras

Laboratory:
Chair of international Finance (CFI)

Academic Year 2022-2023

Contents

Chapter 1

Introduction

Forecasting a currency is a process of making predictions about the future exchange rate against others currencies based on trend analysis and past and present data.So essentially data is collected, studied and the analysis is done to forecast future scenarios that are likely to occur. We are interested in CHF forecasting against major currencies, we will focus on CHF/USD. Since Meese and Rogoff (1983a,b, 1988), it has been known that exchange rates are complicate to predict using economic models; in particular, a simple model such as the random walk is frequently found to generate better exchange rate forecasts than economic models. That's why we use ML and ANN, to prove that we can beat the the random walk model. To achieve that firstly, we are going to use Neural Networks to forecast daily predictions of exchange rate and compare them to a random walk. Secondly, use neural networks with monthly data to compare it to already existing ML results from "Fundamentals and exchange rate forecastability with simple machine learning methods" by Christophe Amat, Tomasz Michalski and Gilles Stolt.

1.1 Problem statement

The development of methods applied to financial forecasting is growing. This is the case with ANNs and machine learning, these methods increasingly apply to areas such as finance, they are useful and needed for innovation and development. ANNs certainly seem to have the potential to distinguish unknown and hidden patterns in data which can be very effective for exchange rates prediction. The aim of this paper is firstly to study different methods to forecast CHF/USD exchange rate from classical ML methods to very specific neural networks over daily and monthly predictions. Secondly, within these methods how the accuracy of the forecasted prices are affected for a neural network depending on the number of neurons and the distribution of the training dataset.

1.2 Purpose

We will see whether ANNs, ML models or similar models can be used to forecast CHF/USD exchange rate movements. Secondly, we will discuss how two significant factors; the number of neurons and the training data distribution, can affect the network's accuracy. These two factors were chosen because of their presence in all types of ANNs making them fundamental and important.

Chapter 2

Datasets

The dataset used in this project is a time series. A time series is univariate or multivariate quantitative data collected over some period of time .There are different techniques and approaches for time series forecasting. The quantitative technique refers to forecasting based on historically and statistically analysed data. This applies to exchange rate data and how the values of the exchange rate change over time in a non-linear way. The historical data can then be used to investigate the variable of interest and forecast its future values, which in this case are the CHF/USD exchange rate prices.

Enough data is required to investigate the different distributions of the training dataset. Therefore, the collected data for the study consisted of 18 year (1999-2017) end-of-month (EOM) data from the International Monetary Fund (IMF), real-time data for fundamentals.

2.1 Data processing

Before data is used by a network, it can go through several transformations in order to adjust the inputs to a given model. The quality of the input data is directly connected to the success of the methods, because different methods can handle different samples of data. Therefore, it is recommended to take advantage of certain data features in order to find out which pre-processing transformation works best.

Normalization : The volatility of the forex market is largely responsible for the erratic character of prices. As a result of the high price changes disturbing the machine learning's and neural network's learning process, this could make learning more challenging. All input data is normalized within the same boundaries as the activation function's to prevent inconsistencies. The range of the activation function determines the method of normalization, therefore the min-max normalization function below is one way to normalize prices between -1 and 1.

$$X_{scaled,i} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{2.1}$$

Here X_i denotes the i'th input of the vector, the minimum and maximum values of the vector are denoted X_{min} and X_{max} respectively.

Dividing Data :

For training a machine learning algorithm or a neutral network one has to divide the available dataset into two different subsets. These two subsets are called training and testing sets. The training set is used for training the algorithm, which means updating the weights and biases depending on the deviation between the target value and actual output. The testing set is a data set independent of the training set, this set is composed of the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. We will see later in the paper how divide the data set and the effects of this split.

Chapter 3

Methods

We can use classic methods as conventional rolling or recursive OLS regression, but our goal is to achieve forecasting with machine learning methods (ML) and see whether ML methods perform better than traditional OLS. We will then discuss of two methods stemming from the field of sequential learning. Those methods are general forecasting methods that were not specifically designed for exchange rate forecasting, we can use them to solve others various problems such as air quality and electricity consumption forecasting; see Cesa-Bianchi and Lugosi, (2006); Mauricette et al., (2009); Stoltz, (2010). Finally we will explore two different method which arise from Machine Learning.

3.1 Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. Time series forecasting is an important area of machine learning, we used two ML methods to forecast the CHF/USD exchange rate. Both methods that we are going to discuss are from "Fundamentals and exchange rate forecastability with simple machine learning methods" by Christophe Amat, Tomasz Michalski and Gilles Stolt. They are both used with discount factors, in modelling it represents a decimal number multiplied by a cash flow value to discount it back to its present value. For us the discount factor will always be represented by γ which is an hyperparameter. Also we will use $\beta_{j,t}$ as our slope coefficients/weights based on information available up to time t and j is just representing the fundamental considered at time t. The fundamentals used here stem from the simple exchange rate models of the 1970s and Taylor-rule based models. Every methods and formula that we will discuss are based on the general forecasting equation for exchange rate for a currency pair which follow the formula below (cf Figure 1) with the following parameters:

- s_t is the logarithm of the exchange rate at time t.
- α_t are the intercept up to time t.
- $\beta_{j,t}$ are the slope coefficients/weights up to time t.
- $f_{j,t}$ are the fundamentals (Inflation and interest rates) at time t.

The general forecasting equation is given below:

$$\hat{s}_{t+1} - s_t = \alpha_t + \sum_{j=1}^{N} \beta_{j,t} f_{j,t} \tag{3.1}$$

Our goal is then to find $\beta_{j,t}$ with ML methods.

3.1.1 Exponentially weighted average strategy

The first method is the exponentially weighted average strategy with discount factors (EWA). This method was introduced in the early 90s by Vovk(1990) and Littlestone and Warmuth(1994). An exponentially weighted average is a measure used to model or describe a time series in finance this is useful for technical analysis and volatility modelling. In machine learning an exponential moving average is a type of moving average which applies more weight to the most recent data points, in other words this method is like giving more importance to the last experience or memories than to older one. In addition, a moving average in ML is a statistical tool to determine the direction of a trend. This forecasting methods involves different parameters, a sequence of η_t positive numbers which are the learning rates, γ which is the discount factor and $\kappa \geq 0$ which is the discount power, they are all hyperparameters. Then we need to picks weights for $\beta_{j,t}$ the rules is given by the equation below.

$$\beta_{j,t} = \frac{1}{Z_t} exp \left(-\eta_t \sum_{\tau=1}^{t} \left(1 + \frac{\gamma}{(t+1-\tau)^{\kappa}} \right) (s_\tau - s_{\tau-1} - f_{j,\tau-1})^2 \right) \tag{3.2}$$

Z_t is a normalization factor.

The weights obtained create a sub-convex weighted vector then all j,t are non-negative and sum up to something smaller or equal to 1. This method has been studied and developed by principally Cesa-Bianchi et al. (1997), Cesa-Bianchi (1999), Auer et al. (2002).

3.1.2 Sequential ridge regression

The second method is the sequential ridge regression with discount factors (SR). This method of ridge regression was introduced by Hoerl and Kennard(1970) in a stochastic setting. SR method is an evolution in the ML field of the basic OLS regressions, an OLS regression is the method used to find the simple linear regression of a set of data, the problem with the OLS regression is that it tends to overfit past data. The goal is to prevent this, then we add a regularization term to the squared error to help control the range of components $\beta_{j,t}$. In general ridge regression solves the problem of overfitting, as just regular squared regression fails to recognize the less important features and uses all of them, leading to overfitting as discussed above and this is one of the main problem of OLS regression. Ridge regression adds a slight bias/regularization term, to fit the model according to the true values of the

data. This bias or regularization term will be λ for us, which is an hyperparameter.
The SR method is very close to OLS regression, the OLS regression is just a special case when λ is equal to 0. This method involves two parameters γ which is the discount factor and $\lambda \geq 0$ which is the regularization. Then we need to picks weights for $\beta_{j,t}$ the rules is given by the equation below. What follows relies on recent new analyses of ridge regression in the machine learning community, see the papers by Vovk (2001) and Azoury and Warmuth (2001), as well as the survey in the book by Cesa-Bianchi and Lugosi (2006).

3.2 Deep learning and Artificial Neural Networks(ANNs)

Deep learning, also known as hierarchical learning, is a subset of machine learning that can imitate or copy the computing capabilities of the human brain and create patterns similar to those used by the brain for making decisions. Deep Learning is associated with the transformation and extraction of features that attempt to establish a relationship between stimuli and associated neural responses present in the brain, whereas Neural Networks use neurons to transmit data in the form of input to get output with the help of the various connections.

3.2.1 Neural networks

Networks are used in many different situations and can be designed for various purposes. In general, a network can be compared to a graph which consists of units called nodes and connections between these nodes which are edges. The nodes can be seen as computational units, they take inputs then process it in order to produce an output and this output can be spread to other nodes through connections edges. The connections edges are for the information flow, meaning the input and the output, and can be directed.

The type of network we are interested in is an artificial neural network which consists of artificial neurons that can be connected in different ways in order to obtain desired complexity and functionality. Each nodes is now a neuron and can perform mathematical operations such as integration with a given input.

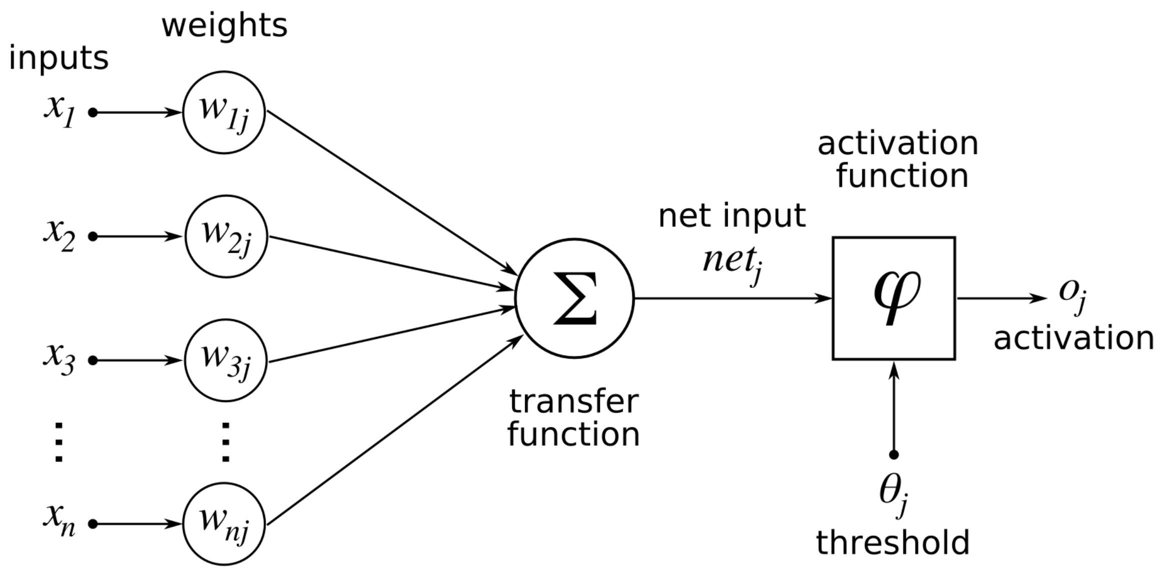

Figure 3.1: The basic functionality of a simple artificial neuron.

An artificial neuron can be supplied with a variety of inputs, as shown in Figure 2.1. The influence of the overall input on the neuron will vary depending on the weights' value, as will how the transfer function and output are computed. Higher weight values essentially increase the strength and influence of the related inputs. This is due to the fact that each input is multiplied by its appropriate weight, which will alter the output of the neurons. In order to create the net input to the neuron, the transfer function sums the weighted inputs, as in

$$n_i = b + \sum_{k=1}^{m} w_{k,i} y_k \tag{3.3}$$

Where n_i denote the actual neuron, y_k is an input where k goes from 1 to m there are m links, $w_{k,i}$ is a weighted value where k goes from 1 to m and b is the bias. Weights enable an artificial neural network to adjust the strength of connections between neurons, bias can be used to make adjustments within neurons, it represents the connection between units. If the weight from node 1 to node 2 has greater magnitude, it means that neuron 1 has greater influence over neuron 2. The purpose of the bias term is to shift the position of the activation function left or right to delay or accelerate the activation of a node. Further on, the final output of a neuron is given by the activation function applied to n_i. There are a lot of activation functions, we have chosen the rectified linear activation function it will be explained in the next section.

$$Relu(x) = max(0, x) \tag{3.4}$$

$$Output_i = Relu(n_i) \tag{3.5}$$

The puropose of a neural netowrk is to put multiple neuron as described above together and get a topology for the network. The only condition for such a network works is that the information must flow in only one direction, from input to output, with no connections such as cycles or loops. The chosen topology is the basic one, it consists of layers of interconnected neurons where each neuron in the input layer is connected to all neurons in the next layer, and each neuron in this layer is connected to all neurons in the next layer and so on until finally they are connected to the final. All layers between the input and output layers are called the hidden layers, simply meaning that they are neither input nor output neurons . The figure below illustrates a neural network.

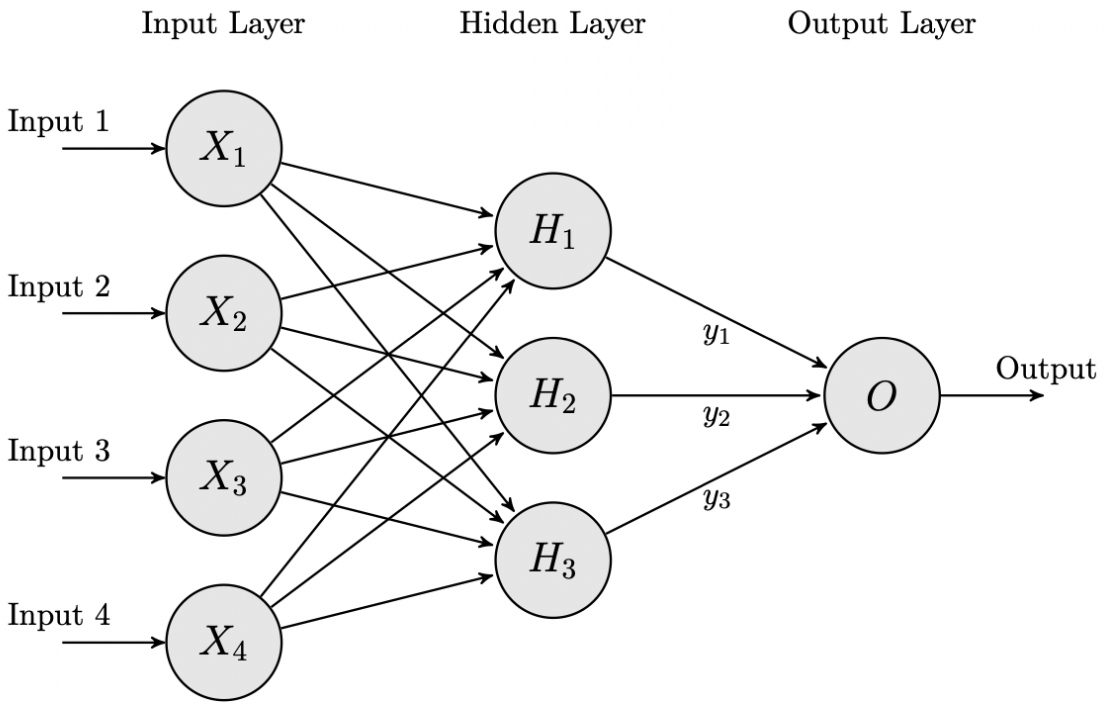

Figure 3.2: Topology of a neutral netowrk.

Neurons $X1$ to $X4$, the network's inputs, make up the input layer. The hidden layer's neurons, designated $H1$ to $H3$ in figure 3.2, process input according to the ideas in Equations 3.1 and 3.3 to determine this layer's outputs, which are shown in Equation 2.4.

$$y_i = ReLu(\sum_{k=1}^{4} w_{k,i}x_k + b_i) \quad \forall i \in (1,2,3,4) \tag{3.6}$$

The output of the network is know computable as :

$$Output = o_i = ReLu(\sum_{k=1}^{3} v_{k,i}y_k + b_O) \quad \forall i \in (1,2,3) \tag{3.7}$$

3.2.2 Learning

Weights are optimized such that the NN outcomes are as close as possible to the true data. The method by which the optimized weight values are attained is called learning. The purpose in learning is to teach the network, which at the beginning has random weights, how to produce the output when the corresponding input is presented. Once the learning is finished the trained neural network, with the updated optimal weights, should be able to produce the output within desired accuracy corresponding to an input pattern. The learning used in the paper is Supervised learning with error correction means that given a vector of inputs to the network, it will produce the respective outputs and compare it to the correct answers at hand and minimize the error. The error is calculated using the mean square error function, where o_i is the i-th predicted values and t_i the i-th observed values.

$$MSE = (t_i - o_i)^2 \tag{3.8}$$

The backpropagation algorithm is mainly used with ANNs since data is only fed forward in an NN, errors are propagated backwards while adjusting the weights and biases according to the error. When the appropriate inputs are provided, the backpropagation algorithm seeks to produce the target as an output. The error depends on the weights and biases in the neurons because it is the difference between the target and output. As a result, the biases and weights are modified to reduce inaccuracy. The decision to alter the weights and biases is made based on how they affect the error, the method used to accomplish this is outside of this study.

3.2.3 Simple ANN and Long short term Memory LSTM

For the project we choose two architectures. The first one is the Simple ANN network, composed of 1 hidden layer itself composed of neurons, the number of neurons is given in the next section. There is one input, the price of the CHF/USD exchange rate and one output the predicted price.

LSTM networks are a type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems. LSTM was designed by Hochreiter Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. LSTM can by default retain the information for a long period of time, those networks are well-suited to making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. It is used for processing, predicting, and classifying on the basis of time-series data.

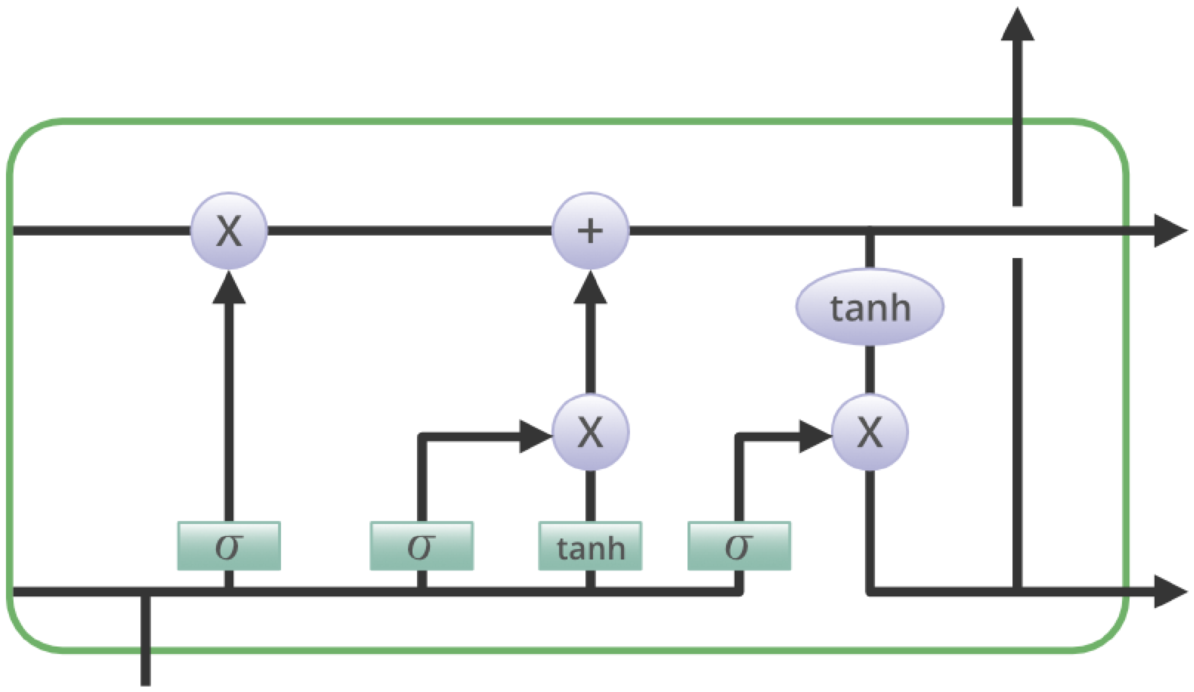

Figure 3.3: LSTM chain structure.

Information is retained by the memory blocks called cells and the memory manipulations are done by the gates. There are three gates, firstly the Forget Gate which decides how much of the previous data will be forgotten and how much of the previous data will be used in next steps. Two inputs X_t (input at the particular time) and h_{t-1} (previous cell output) are fed to the gate. 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 output 1, the information is retained for future use.

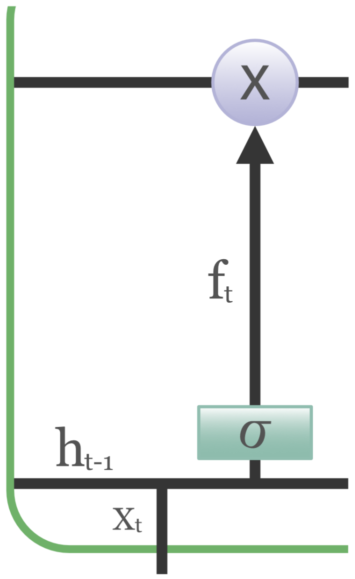

Figure 3.4: Forget Gate structure.

Secondly, the Input Gate, which decides what relevant information can be added from the current step. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_{t-1} and X_t. Then, a vector is created using tanh function that gives an output from -1 to +1.

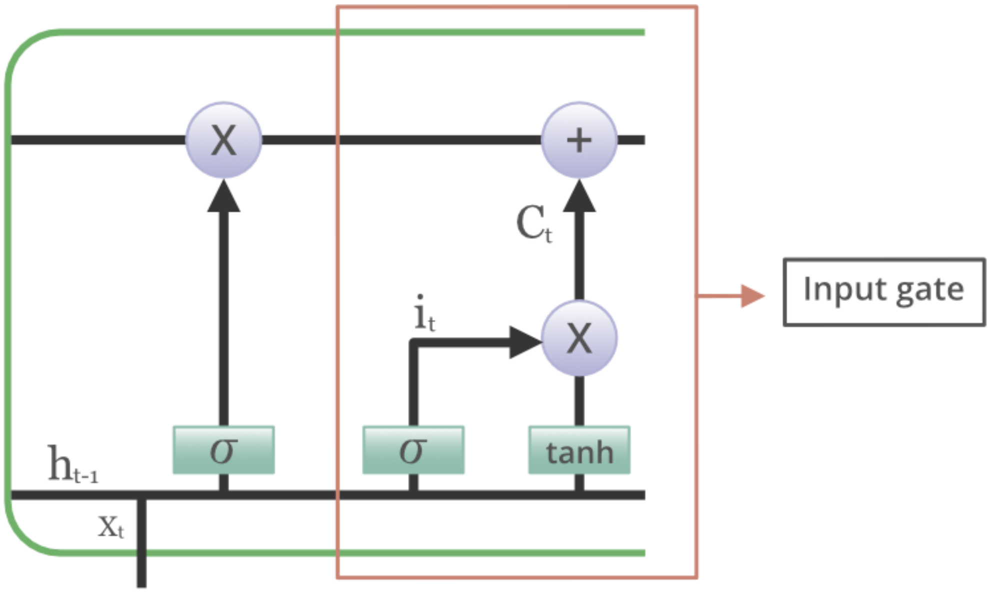

Figure 3.5: Input Gate structure.

Finally, the Output Gate which extract useful information from the current cell state to be presented as output. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs h_{t-1} and X_t. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

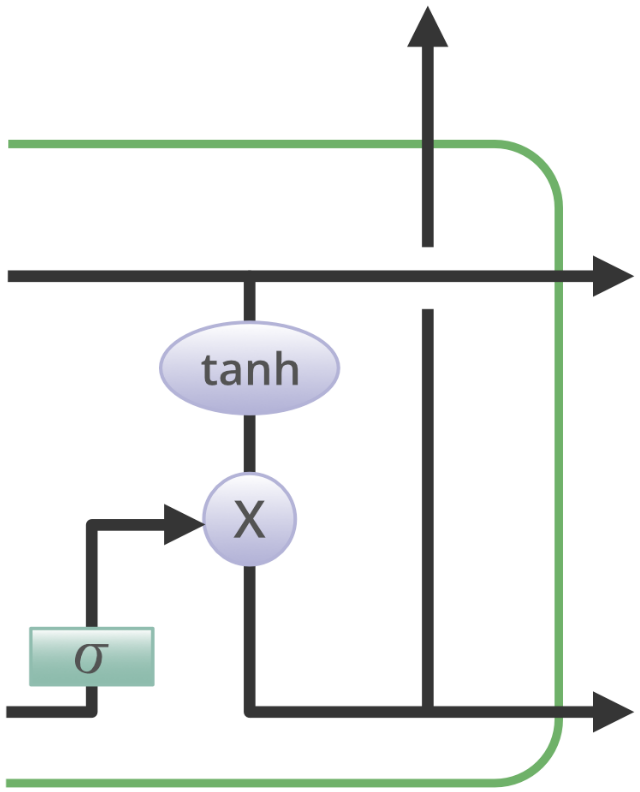

Figure 3.6: Output Gate structure.

Chapter 4

Results

In this module there is 2 main sections, the first one focuses on daily forecasting while the second on monthly forecasting. There are tree types of tests executed in this project. Firstly we examine how the number of neurons in the ANN and LSTM can make to a possible change in the predicted prices. The second test's purpose is to study how the distribution of the training dataset impacts the same predictions for the LSTM and ANN. Finally we will compare which model over ML and NN is the best for predictions over the CHF/USD exchange rate. As stated in the introduction we compare all of our tests to the Random walk because for years it has been proven to be a better predictor of exchange rate than any economic model, we will prove that the Random walk is weak against ML and NN methods.

4.1 Daily

We perform forecasting with daily data only with the ANN and LSTM. The number of neurons and the training dataset distribution are independent so we can optimize both of them independently. The default PYTORCH values for the number of neurons and the training dataset distribution is used. These are ten neurons in the hidden layer and the distribution is 70% for training and 30% for testing. To find in each case the best combination we compute the MSE between the model prediction and the actual exchange rate.

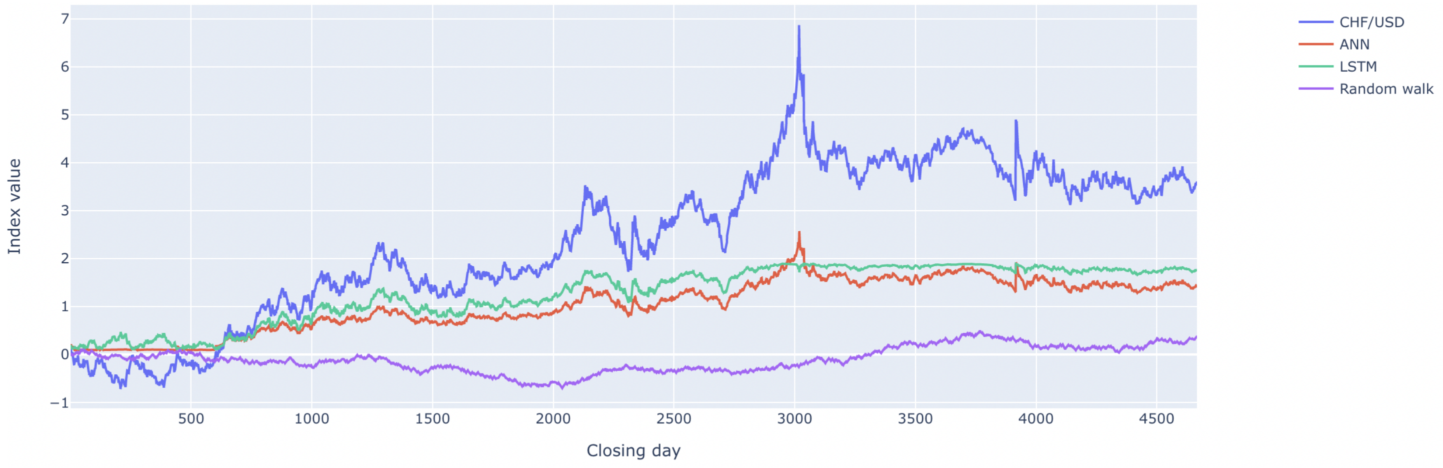

Figure 4.4: Daily predictions with 10% Training and 90% Testing.

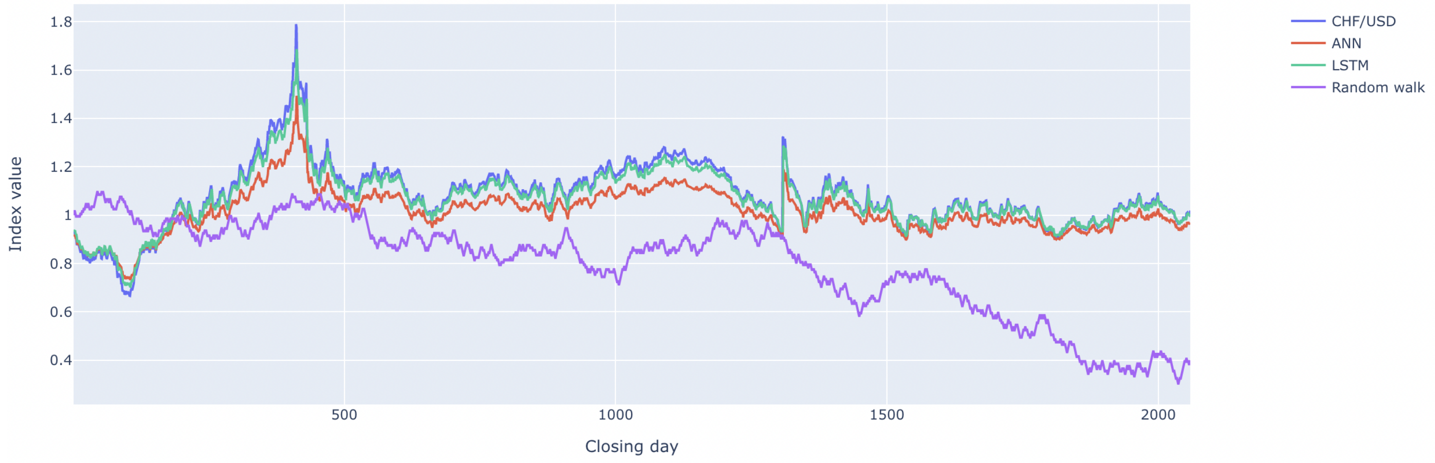

Figure 4.5: Daily predictions with 60% Training and 40% Testing.

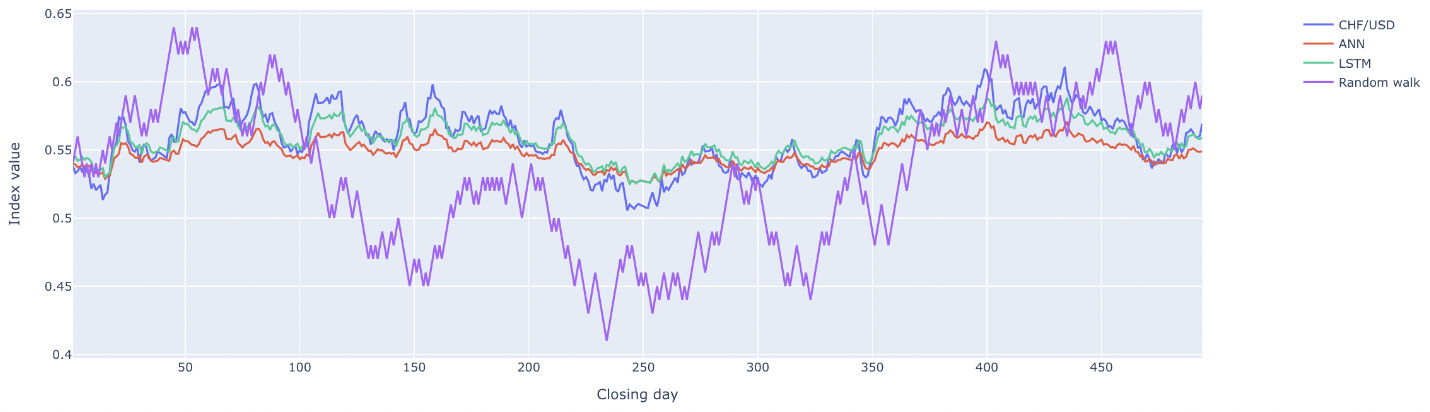

Figure 4.6: Daily predictions with 90% Training and 10% Testing.

4.1.1 Number of neurons

In a NN there are different hyperparameters, we kept them all as non-variable parameter except the number of neurons.

Number of neurons	MSE of ANN	MSE of LSTM	MSE of Random walk
2	3,5	2,8	10, 29
5	4,63	2,06	10, 29
10	3,22	1,87	10,29
25	2,60	1,98	10,29
50	2,85	2,20	10,29
100	3,44	2,62	10,29
200	3,90	2,70	10,29
500	4,33	4,09	10,29

Table 4.1: MSE depending of the number of neurons.

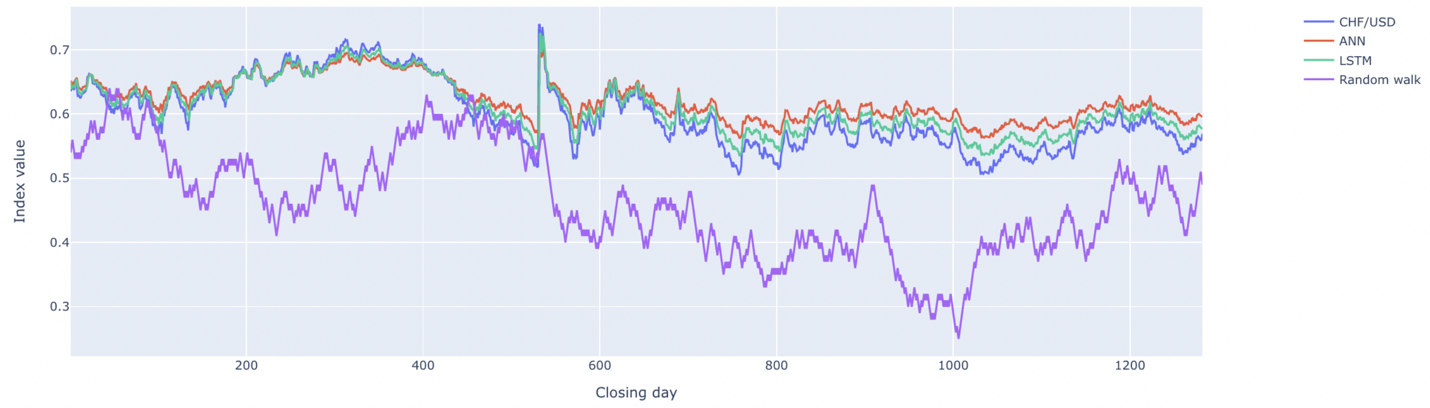

Figure 4.1: Daily predictions 10 neurons.

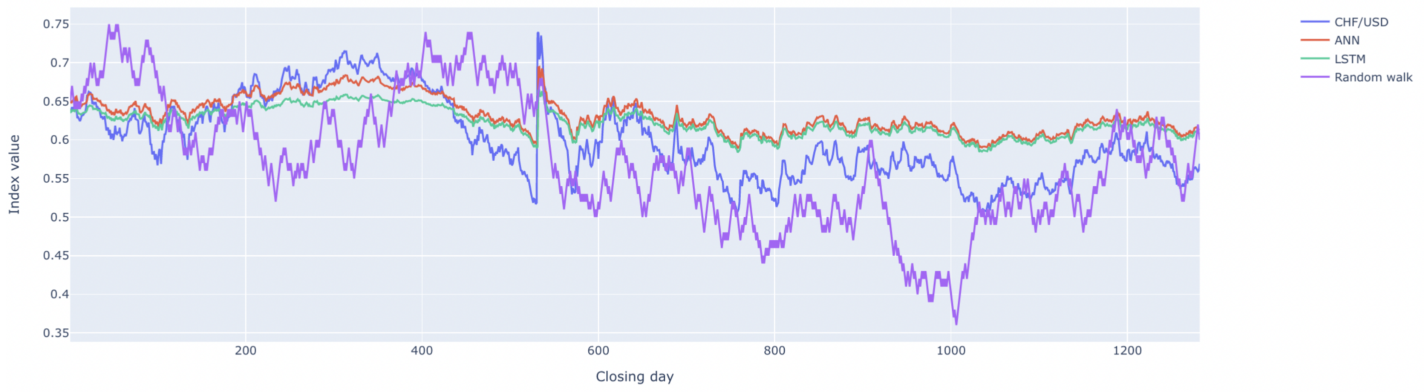

Figure 4.2: Daily predictions 500 neurons.

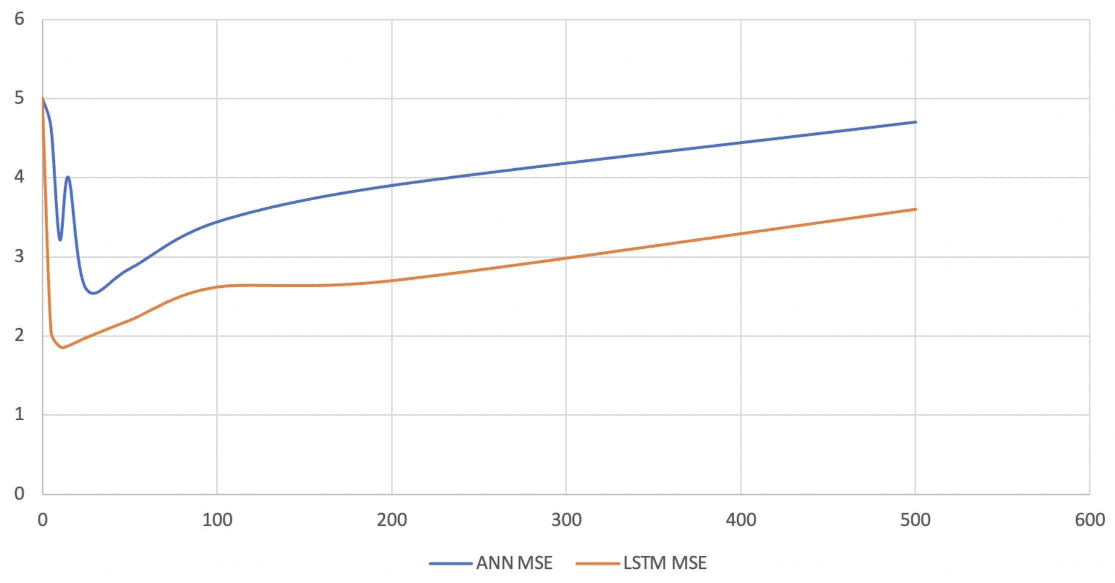

Figure 4.3: Evolution of the MSE depending on the number of neurons.

The minimal MSE for daily forecasting is given by the LSTM with a MSE of 1,87 with 10 neurons.

4.1.2 Training dataset distribution

Given a set of data we need to separate this set into two set, the training set and the testing set. The distribution of the training set may impact the result of the forecasting. We are using the default number of neurons.

Distribution (Train,Test)	MSE of ANN	MSE of LSTM	MSE of Random walk
(10%,90%)	172,03	151,96	297, 56
(20%,80%)	103,50	60,58	154, 55
(30%,70%)	30,60	18,98	111, 11
(40%,60%)	14,85	10,20	91,40
(50%,50%)	8,44	6,11	81, 31
(60%,40%)	3,92	3,02	27,80
(70%,30%)	2,35	2,02	173, 11
(80%,20%)	2,12	1,81	31, 52
(90%,10%)	2,22	1,81	29, 82

Table 4.2: Training dataset distribution.

4.1.3 Final models

For daily forecasting, as seen in the previous sections, the best models is the LSTM with 10 neurons and a training dataset distribution of (80%,20%).

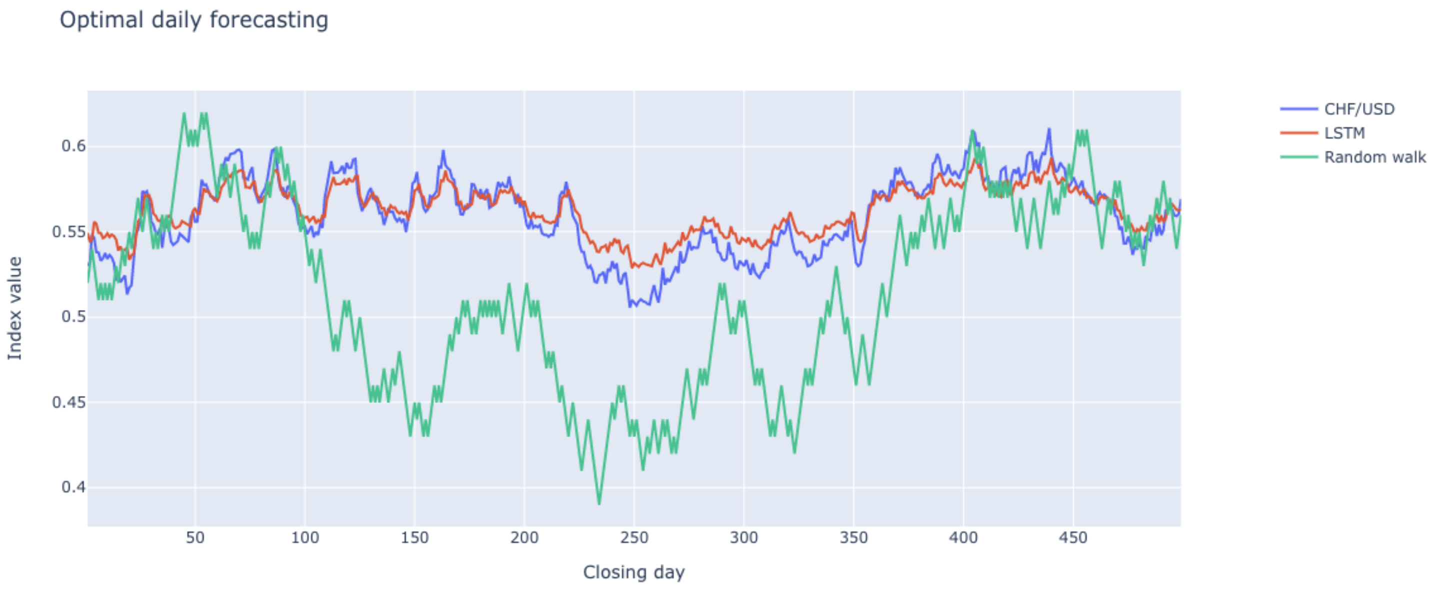

Figure 4.7: Optimal daily predictions for neural network.

The LSTM used is very accurate over daily predictions it has a MSE of 1,7 which is the best MSE achieved in this paper for a model. The final architecture of this neural network is composed of only 10 neurons and run in 131 seconds under python. It would be interesting to know if our model can be applied to Foreign Exchange Market, the model can be apply if the average difference between the predicted price and the actual price is lower than the average actual fluctuation. We will discuss about that further in the next section. The following graphic show the difference between the predicted price and the actual price, which is helpful to know if the model is pertinent for Foreign Exchange Market.

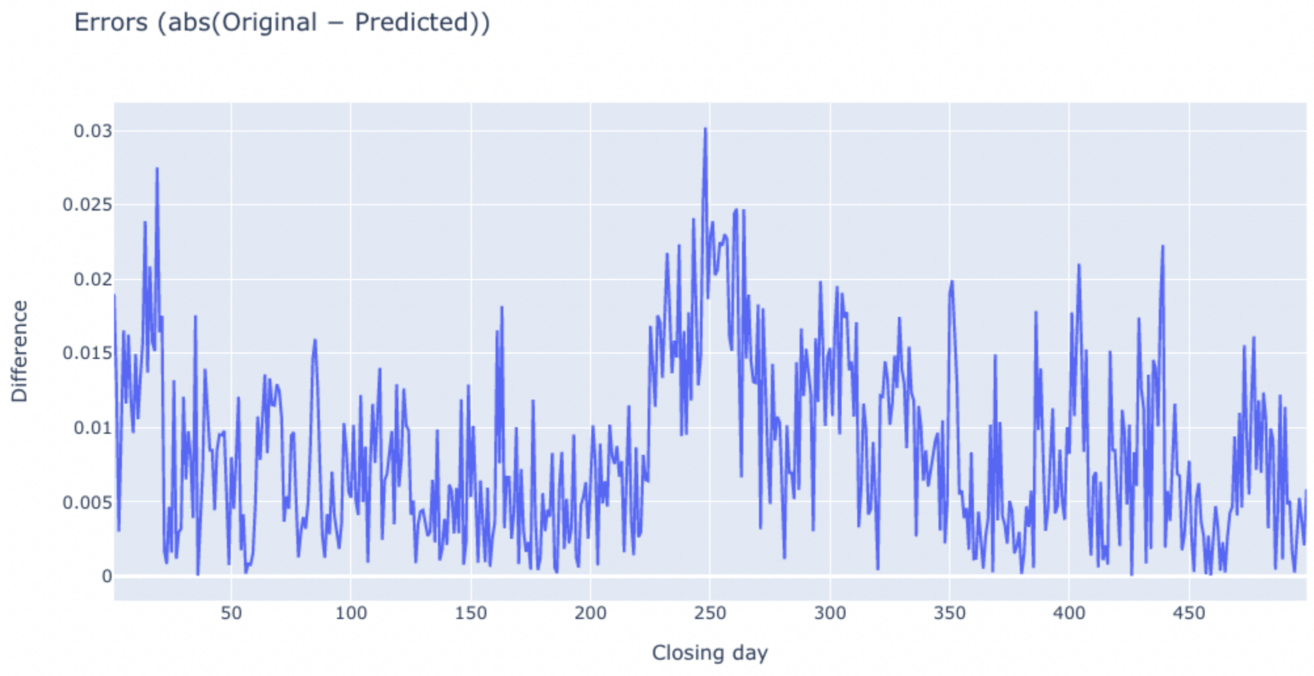

Figure 4.8: Price difference between actual and predicted exchange rate.

4.2 Monthly

We performd monthly forecasting with the ANN, LSTM and ML models. The number of neurons and the training dataset distribution used for LSTM and ANN are the one computed above (optimal one). To find the best model over monthly forecasting we have forecasted the echange rate each case the best combination we compute the MSE between the model prediction and the actual exchange rate. We do monthly forecasting because the results of the ML models done in "Fundamentals and exchange rate forecastability with simple machine learning methods" by Christophe Amat, Tomasz Michalski and Gilles Stoltz are in monthly frequency and we want to compare them to our NN results fed with monhtly exchange rates.

4.2.1 ML results

For the Machine learning methods we used the tests and source code of Christophe Amat, Tomasz Michalski and Gilles Stoltz from their paper Fundamentals and exchange rate forecastability with simple machine learning methods(2018). Thanks to this we got the MSE and some statistics tests, for the Exponentially weighted average strategy (EWA) and the Sequential ridge regression (SR), for 1-month ahead forecasts over our data.

Model/Method	Mean squared error	Theil ratio	Proportion of changes predicted
EWA	3,38	0,9975	53,8%
SR	3,35	1,002	54,1%

4.2.2 ML vs ANN

To find the best model for monthly prediction, we train the four models over the same data period and test them over the months of 2016 and 2017.

Model/Method	MSE 2016	MSE 2017	Average proportion of changes predicted
EWA	3,33	3,27	54,8%
SR	3,28	3,20	56,1%
ANN	2,99	2,79	65,3%
LSTM	2,89	2,71	67,8%

Chapter 5

Discussion

The results in Section 4, the fig 4.8 show the difference from the actual data and it reveals that the predictions have a difference of at most 0.03 CHF for our optimal neural network, compared to the actual prices. However, this only applies to one peak and it is observed that most errors are below the 0.015 CHF difference. Considering the exchange rate around 1CHF, this is not a huge difference, but of course it is still notable. Since the actual prices usually do not fluctuate more than 0.01 CHF in a single day, except for some cases, this error is about three times the size of the actual fluctuation. The practical application of this on the CHF/USD exchange rate is risky. This will be discussed further in this Section.

5.1 Changing the Number of Neurons

A comparison of the effect of the number of neurons on prediction is shown in Table 4.1 and Figures 4.1, 4.2, and 4.3. These results clearly show that increasing the number of neurons does not help prediction, it only significantly increases complexity and execution time. This reinforces the fact that network complexity increases with the number of nodes. In this case, the problem applied to the network is not complex enough to require many nodes. As seen in the results, using too many neurons is an example of over-fitting the data and the network struggles to generalize patterns to future predictions. From Figure 4.3, it is clear that beyond the optimal number of neurons of 10, the MSE increases with more neurons.

5.2 Changing the Training Dataset Distribution

Comparing how the distribution of the training data have an effect on the predictions brings us to Section 4.1.2. Simply searching at table 4.2, the much less correct predictions are the configurations with low training set. It is hard to discover a relation in how the real distribution of the training set influences the predictions, however it's clear that there must be enough data for the training subset or the network will now no longer be capable of adapt and generalize to the hassle as correct as possible. But it's important to see that the more you have data in the training set longer and more complex is the training. The MSE of both neural network decrease a lot from a training set of 10% to 60% but after that the decline is slow ans smooth but the training keep getting complex, so it's not worth to have a distribution of (90%,10%) because the complexity grows exponentially but the MSE only decrease of 0.1 point in average. The testing data is not used for training the network and thus has no effect on the networks performance, which can be seen in the results as well.

5.3 Method Discussion

Since there are many different models and structures of ANNs there are many other possibilities that can be used and tested for exchange rate forecasting. Trying other models might give more insight into the problem at hand and manage to find more accurate and suitable systems for exchange rate forecasting. The data used for training and testing could also benefit from being varied and could therefore generate other results because of more complex patterns or more volatile price development. Among our methods, neural network is clearly the more accurate one. For daily forecasting we found out an architecure of an LSTM which perform well, as discuss above using this LSTM in reality is possible because the average price difference between actual and predicted is 0.093 CHF which is lower than the average fluctuation of the exchange rate which is 0.01 CHF. But for all other methods this is not practically useful since they span over periods longer than a day or are not accurate enough. The method has nonetheless shown potential given the few input variables on the specific problem.

5.4 Conclusion and possible improvements

There is potential for one day exchange rate prediction with ANNs. Even if this is not practically applicable, except for the LSTM optimal, it could instead be used as an indicator to if the exchange rate is rising or decreasing. Adapting the network to fit the problem at hand is very important, the number of neurons chosen for the ANN should be decided by the complexity of the problem. The same thing can be said about the training data set distribution. It is important to have enough data for training the network and allowing it to generalize itself to it. In this paper we found a great neural network with one input, the price which can be used in reality for one day CHF/USD exchange rate forecasting. Predicting only one day ahead might arguably not be very useful, as stated earlier and the dimensionality of the network inputs, meaning the number of variables, required to predict CHF/USD exchange rate is low (only one in this paper) and in reality there would be more factors that affect an exchange rate. One could improve our network by adding more inputs and more outputs and extend the possible application for daily predictions to weekly or monthly predictions.