Time-series Forecasting: Predicting Google Trend using Long Short Term Memory Neural Network

Salta Talgat  
MSc Computing and Information Systems

|  |
| --- |
| *Abstract*—Forecasting time series has always been a huge chal­lenge. Mainly, because of seasonality and trend, it is difficult to achieve a reliable forecasting performance. Developing forecasting models has always been a resource consuming and costly process, especially when it comes to data collection. The use of publicly available Internet data such as Google trends can be efficient, but this is nowcasting rather than forecasting where a longer leading lag is required. This project presents the design, training, and testing of a long-short-term-memory neural network for google trend series forecasting. Five years of google trend data was used for training and testing the model. The single-layer LSTM showed similar results as more complex stacked LSTM networks and can be compatible with RNN and GRU. One of the examples where the proposed model can be used is web applications that utilize customer input series and Google keywords and output forward forecasts. |

Keywords — time series forecasting; machine learning; artificial neural networks; LSTM networks forecast; trend component; google trend

# Introduction

Time series forecasting has often been identified as a very important aspect in medicine, weather, biology, finance and economics forecasting [1, 7]. However, real-world data in most cases have trend, cyclical or seasonal components. A trend component can be characterized as long-term upward or downward in the data which can be nonlinear. At present, the development of algorithms that can forecast a time series with a trend component is one of the active research areas in deep learning domain [3]. The most frequently used neural networks for time series forecasting are the recurrent neural networks with long short term memory (LSTM) [2]. These algorithms have certain advantages when it comes to time series trend component, cycles and seasonality. LSTM not only learns nonlinear associations but also remembers long and short trends within cell states and gates [2].

Most recent time series forecasting studies have indicated that search engine query data can be readily incorporated into forecasting models, and one of the most used is Google Trend (GT). The GT is becoming one of the widely used leading predictor in forecasting studies nowadays [10]. It is a free service and returns the normalized search volume for given terms within a time window and geography. In the majority of studies GT keyword is used as predictor variable whose observations are used to predict the value of the target variable, but limited research and experience existed with regard to the GT forecasting itself. A forecasting model can give accurate results if the future values of explanatory variables are known or forecasted. In order to fill this research gap, the aim of this paper is to propose LSTM neural network suitable for google search volume series forecasting.

# PROBLEM STATEMENT

The purpose of this work is to forecast GT using LSTM neural network. There are several objects which were faintly studied in this area. In this line, there are several objectives that this work attempts to examine.

The first one is related to trend variations: they are not often constant over time and their decomposition is not efficient with classical liner models [1]. This work attempts to investigate whether LSTM actually struggles to capture the trend on its own or whether the detrending procedure is required before the training. Therefore, training is conducted with raw and differenced data.

The next objective of this work is concerned with hidden layers. There is no single approach for defining their number thresholds, it depends on the individual problem. Simple networks due to small number of estimated parameters tend to be less accurate compared to complex ones with several layers. At the same time networks, which are too large, are difficult to train and always run the risk of data overfitting. This issue is examined in this work by conducting experiments on different number of layers ranging from one to four.

One of the most discussed questions in the literature relates to the number of neurons in network layers. The comparative studies in terms a layer size show that 20 neurons can be optimal within 100 epoch [11]. However, most experimental studies argue that it is specific for each input size, data nature and model, and should be tested for different number of neurons. Therefore, this work considers objective testing forecasting accuracy on different numbers of neurons by taking into account the input size.

This study also attempts to test multivariate types of LSTM, where the synonym search words were supplied into the model as predictor variables. The main idea behind that, is that the input synonym observations contain useful information that allow us to control the target variable in forecasting.

According to studies neural networks cannot capture trend variations effectively. One of the most influential factors is the length of the input sequences. If the input is small then the model will capture the short trends and with longer trends, the model performance can be low. Thus this study experiments with different numbers of window size.

The impact of the input size on forecasting performance of time series with a trend using LSTM neural networks has scarcely been explored. If the forecasting of time series with a trend was done with an input size of three months but there are long trend lasting six months would expect the model performance to be low. In order to capture both long and short trends, this study experiment with several input size.

Despite a solid research base in comparing time series forecasting models, it is evident from literature review that a gap exists in the comparison of performances of the different models related to GT forecasting using LSTM neural networks. This study compares different recurrent type of networks such as RNN, LSTM and GRU.

# CONTRIBUTION

Internet data, especially search queries are becoming important observations used in forecasting in many areas. One of the most widely used is GT data. The fast speed of collection and its free availability make them a good predictor. Despite the growing number of studies using GT in predictive models, there is limited research on forecasting GT itself. Therefore, this work contributes to studies that forecast GT with machine learning algorithms, mainly with LSTM versions of recurrent neural networks.

Time series forecasting is one of the toughest problems, this is mainly because of trend component. However, limited experimental studies focus on trend component capturing in time series forecasting using LSTM. Forecasting in this study is performed with raw and differenced series, where differencing account for stochastic trend. Forecasting accuracy on first differenced and second differenced series differed only slightly from raw series. This evidence contributes to studies claiming that neural networks can forecast time series without subtracting trend from the series.

There is not any best single approach for hyper parameters selection of the LSTM neural network for time series forecasting. In this work, different numbers of parameters such as layers and neurons number were experimented with. Thus, this work contributes to studies that support the idea that complex neural networks are longer to train and the performance does not differ significantly from those of Single layer neural networks.

According to studies the forecasting of model performance depends on network layer size [11]. A Literature review conducted in this study shows that there are no specific approaches on number of neurons in the LSTM neural networks for time series tasks. This study used different number of neurons during the experiment. Therefore, experiment results contribute to studies that experimentally defined the optimal range of neurons number for GT forecasting tasks and guided future researchers in this field.

Multilayer LSTM neural networks can extract more parameters compared to Single-layer. The ability to learn specific pattern depends on the number of layers, but the addition of layers in the model requires more time for learning and computational resources. This study compares models with different number of layers. Thus the results from the experiments contribute to future studies focusing on the difference between Single Layer and Multi-layer LSTM neural networks.

Trend components of time series can be short or long term. If window size configuration is small than there is possibility that longer trends cannot be captured. The majority of studies in time series forecasting using LSTM neural networks experimented with small and fixed window sizes and the reason behind this related to the cost of computations and the longer training time. Different window sizes were tested during the experiment. Thus the results contribute to studies on LSTM neural network window size selection and effect.

Despite much research on forward forecasting models that include relevant explanatory variables, there is a gap in studies that uses explanatory variables for GT forecasting. In this study, the experiments conducted provide evidence that the forecasting performance decreases when a set of synonym words is used as an explanatory variable in forecasting within the experimented single-layer LSTM neural network. For multivariate time series need to build complex multi-layer LSTM.

# BACKGROUND

## Reccurent neural networks (RNNs)

# Recurrent neural networks developed for modeling time series. Comparing to standard multilayer perceptron their hidden states are designed with feed-back connection (Figure 1). For each time step t, the RNN (A) takes an input value, output a hidden value which will be fed into next step t + 1. This distinction allows us to learn temporal associations related to time delay. By unfolding the RNN in time, each time step can be seen as a unit cell of the RNN architecture. Meanwhile, the main disadvantage of RNNs such as exploding and vanishing gradients make them weak in capturing the long-term time dependencies [5, 8].

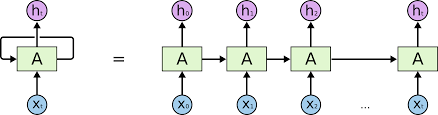


Figure 1. Standard RNN

Training the neural network is an iterative process that maps the input with the output. If at the first iteration weights assigned randomly, then the next steps are updated using loss function (Eq. 1) of ground truth and approximated function of observation. The output through a loss function is scaled by a learning rate and optimized until a minimum is reached (Eq. 2).

(1)

(2)

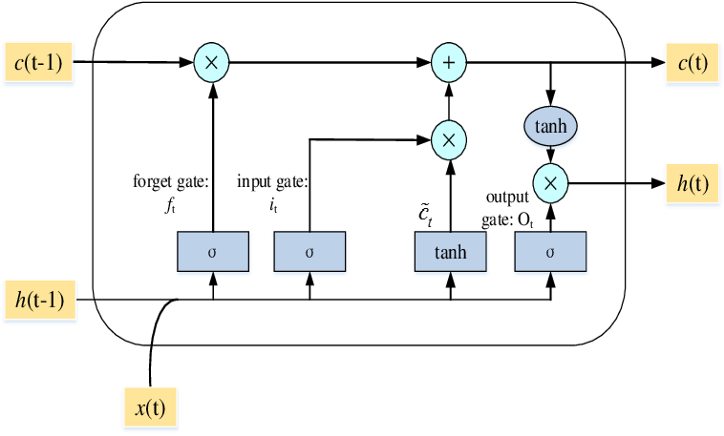
The RNN is based on the concept of error backpropagation and gradient descent. The partial derivative of the cost function is relative to the estimated weight updates model parameters to minimize errors. Randomly assigned weights are usually close to zero and their multiplication led to a small gradient and may vanish. In opposite to this when the derivative is big then the gradient explodes. The gradient vanishing and gradient exploding issues may be eliminated by applying techniques such as gradient clipping or using LSTM neural networks. If gradient clipping is based on manually defined thresholds that control the gradient, then the LSTM neural network uses the concept of gates (see section IV B).

Neural network architecture consist of several hyper parameters that configure the model: number of hidden layers, number of neurons, learning rate, activation function and optimizer settings. These parameters can be tuned manually or in an automated way. Among the most used optimization algorithms are Grid search, Random search and Bayesian optimization. If Grid search builds possible combinations of hyperparameters from specified values, then Random search builds random combinations from the statistical distribution of values. These methods utilize solid computational resources and processing time. In terms of speed, presiding Bayesian optimization is faster, because it uses results from the previous iteration to select next value combination [6**Error! Reference source not found.**].

The model forecasting performance is highly dependent on optimizers, they play a very important role and the Gradient descent technique is one of the widely applied methods. This method updates the weights to minimize the loss or cost function [5]. Gradient descent algorithms multiply the gradient by a learning rate. The small learning rate required multiple steps to find the minimum and the maximum may result in a divergence. There are several other variants of optimizers such as Stochastic Gradient Descent (SGD), Root Mean Square propagation (RMSProp), and Adam. If SGD proceeding a single update at a time and RMSprop adjusts the learning rate automatically for increasing variation then the Adam update learning rates for individual parameters based on fist difference. The Adam optimization algorithm is one of the most used in time series forecasting problems.

## Long short term memory networks (LSTMs)

Long short-term memory networks were developed to solve the issue of long-term memory in 1997 by Hochreiter and Schmidhuber [12]. In addition to the hidden state, this network also incorporates the cell state of previous information functioning as a long term memory [4, 8, 12].



σ

σ

σ

Figure 2 . LSTM cell

Sigmoid function δ convert values to nonlinear form between 0 and 1. This ensures that only important information is saved. If W is the recurrent connection between the previous hidden layer and current hidden layer then U is the weight matrix that connects the inputs to the hidden layer. Hidden state (Eq. 6) is responsible for the short-term memory and the cell state (Eq. 4) for the long-term memory. Three different gates are used to derive the cell state and the hidden state [5]. The diagram of an LSTM cell can be seen in Figure 2. If the forget gate (Eq. 4) decides what information from the new input and the hidden state should be forgotten, then the input gate (Eq. 3) decides what new information must be passed to the cell. These gates allow LSTM networks keep long trend information [3, 4].

(3)

(4)

(5)

(6)

(7)

(8)

The standard LSTM neural network has sigmoid (Eq. 9) and hyperbolic tangent functions (Eq. 10).

(9)

(10)

The Sigmoid function ranges between 0 to 1, and the tangent between -1 and 1. There are limited studies on the use of other activation functions developed for the neural networks. The value of the activation function determines the decision borders and the total input and output signal strength of the node. The activation functions can also affect the complexity and performance of the networks and the convergence of the algorithms [5]. Careful selection of activation functions has a large impact on the network performance.

There is no single approach which defines how many layers or nodes to select, depending on the certain data types and problems. Among the most widely used methods are grid search, random search methods. Although they can decrease forecasting errors, the calculation time increases. According to recent studies some analytical conclusions have been derived that the number of hidden neurons should be between the size of the input layer and the size of the output layer or more precisely less than twice the size of the input layer. Therefore, this study tests different hidden layer and node numbers to define optimal one [4**Error! Reference source not found.**].

Another interesting type of LSTM is bi-directional LSTM in which the network trains two models and then combine them. If the first model is an ordered input, then the second model is its inverse input in separate hidden layers. In time series problems Bi-directional LSTM is mainly applied for filing missing values in observations and showed goof model performance compared to LSTM but required more than twice the time for training.

## Gated Recurrent Unit (GRU)

The GRU is one of the variants of the RNN network and an alternative version of LSTM proposed by Cho et al. in 2014. If LSTM networks have separate input and output gates then in GRU they are combined, thus reducing the number of parameters [7].

(11)

(12)

(13)

(14)

When the reset gate transfers important information from the previous hidden state and new input to a next state, then the update gate is responsible for transferring information from the previous hidden state and reset gate to the next hidden. While LSTM stores its longer-term dependencies in the cell state and short-term memory in the hidden state, the GRU stores both in a single hidden state. GRU uses less memory and is faster than LSTM, however, LSTM is more accurate when using datasets with longer sequences [7]. The literature review on the comparison between LSTM and GRU advantages regarding time series forecasting showed that the there is no achieved consensus among researchers agreeing on distinct advantages one of them. For instance, if in [16] GRU forecasted better, then in [6] argued that GRU and LSTM accuracy are at same level. However, taking into account the difference in the architecture of these networks, it is still difficult to select for time series forecasting. Most reported experiments mainly refer to problems other than time series forecasting.

## *Quantum-inspired LSTM*

## The theory of quantum computations is an important and rapidly developing area of deep learning theory in our days. One of the most widely used quantum algorithms that incorporated into classical deep learning methods in classical computers are variation quantum algorithms (VQA) [13]. These algorithms showed good results on optimization and classification problems [14]. In the quantum field there are two concepts,superposition and entanglement, which depend on encoding methods. In order to map classical data into quantum states encoding procedures should be applied. One of the most widely used is amplitude encoding, where the input time series? vector normalized in the range [0 ; ] and applied to Bloch sphere aplitudes states ilustrated in Fig.1. The north pole and the south are the basis states |0⟩ and |1⟩ and by rotation in this unit presents possible state combinations.

## 

Figure 3. Geometrical representation of Bloch sphere

## The entalgment can be achived by appying controled gates for each qubit states. Amplitude of each single qubit after entanglement through gates can be optimized by classical optimization methods used in neural networks training. There are limited studies on time serires forecasting with quantum enhanced LSTM [15]. However, more research is required on the problem of learning time series in the quantum field.

## Related work: Time series forecasting

Time series forecasting is one of the main research tasks across many disciplines. The reason behind this interest relates to future values that they can produce based on historical observations (Eq. 13).

(15)

Variations that are not explained by the observations are included in the error term. In the relevant literature, there have been numerous successful applications in different fields, such as finance and medicine. Among the most popular linear methods in time-series forecasting are auto-regressive integrated moving average, partial least squares, lasso regression. Univariate time series forecasting most often uses methods that are auto-regressive models that detect the previous signals. This models advanced forms are ARMA and ARIMA, which combine past signal with moving average and differencing technics. The specification of the level of differencing allows capturing complex patterns. However, it has also been found that many real time series seem to follow non-linear behavior and the liner approach is insufficient to represent their dynamics. Thus in the most relevant literature have been presented a wide range of models based on neural networks [2, 4, 5]. Neural networks are efficient in forecasting time series, especially with LSTM and GRU variants of recurrent networks. But still there are a gaps in the literature and with the most prominent gap raises questions on the capturing trend, seasonality and cyclical components. The trend component of time series is one of the important researching aspects in forecasting and there have been mixed opinions among researchers. Real world time series encapsulate several components translating seasonal, cycles or trend features of observed processes. A trend component can be long or short term, in classical forecasting methods this component is removed by applying detrending methods. There are several methods such as moving average filters, differencing at level and order. The literature review on effects of detrending application in forecasting shows that there are limited experimental studies with solid theoretical ground on such important questions in forecasting. If seasonal and cyclic fluctuations in the majority of real world data are short then the trend component can be short and long term. If early works in this area state that neural networks can efficiently capture a trend then more recent experiments state the opposite and suggest separating the trend before the network training [4]. This work alone with proposing the LSTM neural network also testing whether LSTM can forecast time series with a trend without applying any detrending techniques. Therefore, experiments are conducted with and without removing the trend component.

Many studies in time series forecasting including in the model explanatory variables and transform from univariate model into mixed multivariate version. Explanatory variables consist information that can control target variable (Eq. 14).

(16)

In practice it is difficult to define explanatory variables and the main barrier is concerned with resources and data available. This acquires even more importance in view of the need of explanatory variables the value of which are known or forecasted.

## Google trend as a leading index

The Google trend is the collected search queries of Google users. This data is normalized to 100 according to the volume within a given time range. Due to the speed at which this data can be collected, they can be used as a leading indicator for many forecasting and nowcasting models [9].

If leading indicators decrease, then after some time steps over variables will also decrease. This warning signal is one of the most important in forward forecasting and widely used in many fields. The identification leading index is one of the most difficult aspects of time series forecasting. There are several requirements that are assigned to the leading indicator, and one of the most important is degree of lead-lag. In order to improve the forecast accuracy, most modern forecasting models contain google search series as an input predictor. For example, GT can improve the prediction of future sales, tourism demand, unemployment, and the stock price [10]. The majority of studies utilize search engine query data as a leading indicator that contains predictive information at least as many periods in advance. According to the literature review, GT encapsulates information that signal future directions and may be used as powerful leading indexes. But limited research focuses on forecasting this leading index itself. The forward forecasting of GT allow them to be used in long term forecasting models and not only in nowcasting. Therefore, this study attempts to forecast GT using a LSTM neural network that is capable of capturing google trend variations and giving an accurate forward forecast result.

# Methods

## Datasets

Google Trends data have been shown to improve forecasting for stock prices, sales and demand for certain categories and goods [10]. This study experiments with forecasting GT future values and uses five years weekly history series as an input data. GT differentiate between keywords, top three words illustrated in Fig. 3 and their two synonyms were selected related to cryptocurrency, stock price, and Covid-19 symptoms subjects. The keyword and their synonyms are summarized in Table 1.

Table 1. List of keyword for GT time series

|  |  |
| --- | --- |
| Keyword | Set of synonym keyword |
| bitcoin | cryptocurrency |
| blockchain+bitcoin |
| Covid-19 | cough+covid |
| symptoms+covid |
| Stock price | stock+index |
| futures+stock |

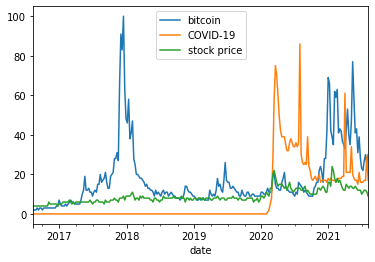


Figure 4 GT time series on selected keywords

## Pre-processing

Min-Max Normalization is applied to the data. The training and validation datasets are separated and we reserve a part from the end for validation with a length equal to the forecast horizon [**Error! Reference source not found.**]. The differencing on the first (7) and second order (8) was applied to remove the trend.

(17)

(18)

## Model Architecture

In this study, single layer and multi LSTM neural networks are tested for google search index forecasting. The hyper parameters of the model are defined manually based on the extensive literature review. The proposed model is mainly designed for forecasting web applications based on google trend and user input series. So the model conversions should be fast in same have a good forecasting performance.

The hyper parameters of the model and their prior distributions are summarized in Table 2.

Table 2 List of parameters and their corresponding range of values used in the grid search.

|  |  |
| --- | --- |
| Hyperparameter | Considered values/functions |
| Number of Hidden | {1,2, 3,4} |
| Number of Neurons | { 4, 8, 20,40} |
| Batch Size | {1} |
| Optimizer | {ADAM } |
| Activation Function | {tanh, , sigmoid} |
| Learning Rate | {0.001} |
| Number of Epochs | {100} |

Single layer networks and small number of neurons can produce small number of parameters that can not sufficiently capture variations of time series. In same time several layers stacked to each other can overestimate data. Therefore, different number of hidden layers are tested. This issue is examined in this work by conducting experiments on different number of layers ranging from 1 to 4. The number of hidden units in each hidden layer 4, 8, 20, 40, are tested. The network is trained for 100 epochs and a batch size of 1 is used.

## Training

#### In this study, the sliding window approach was used and window size was fixed for 12 and 24 weeks with an overlap of 1 week’s information, so the prediction was made for 1 week in future. The model received just over three and six months of data as input and forecasted up to 1 week into the future.

## Evaluation

To quantitatively assess the overall performance of the model, a Root Mean Square Error (RMSE) is used to estimate the prediction accuracy as in (Eq. 9). RMSE is a scale dependent metric which quantifies the root difference between the forecasted values and the actual values of the quantity being predicted by computing the average sum of squared errors. The average RMSE of the top five google time series data are used for comparison of the proposed models.

(19)

# Results

The average RMSE of single-layer LSTM model tested on the top three google trend searches. Differenced series on second level shows 25% more errors compared to first level differencing and primary data performances. Therefore, further experiments in this work have been conducted with first differenced and raw series.

Fig. 1. Test RMSE for selected google search series for Single-Layer LSTM and neuron 4

Single-layer LSTM have been used to test the effect of neuron number on forecast accuracy. The RMSE on neuron size equal to 4, 8, 16, 32, 60 and window size equal to 12, 24 starts to increase when the model is calibrated to 16 neurons and 24 window size. This in line with studies arguing that the size of neurons depends on input numbers. Therefore, further experiments in this work have been conducted with models fixed with 4 neurons and window size 12 and 24. Multivariate model with two additional synonyms as a predictor variable showed comparable forecasting results compare to univariate models. This in line with studies devoting that input variables contain information allows better predict the target variable but need more hidden layer to extract complex information [9, 7].

TABLE 3. Test results of RMSE[[1]](#endnote-1) for Single-Layer LSTM with window size 12 (24)

|  |  |  |  |
| --- | --- | --- | --- |
| Hidden size (neurons) | Differenced | Raw | Multivariate |
| 4 | 0.154 (0.157) | 0.126 (0.122) | 11.520 (11.923) |
| 8 | 0.164 (0.156) | 0.115 (0.122) | 11.129 (14.738) |
| 16 | 0.162 (0.157) | 0.116 (0.156) | 10.468 (10.422) |
| 32 | 0.156 (0.157) | 0.110 (0.110) | 10.321 (10.375) |
| 60 | 0.155 (0.154) | 0.107 (0.110) | 10.011 (10.645) |

*Bitcoin keyword*

TABLE 3 shows RMSE that the tested models for GT time series forecasting are reliable and that single-Layer LSTM is compatible with more complex multivariate LSTM neural networks. Single-layer LSTM can achieve better forecasting performance on single time series compare to multivariate time series, and processing time is less.

TABLE 4. Test average RMSE[[2]](#footnote-1) for single layer models with window size 12 (24)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Differenced | Raw | Multivariate |
| models | RMSE | RMSE | RMSE |
| LSTM | 0.154 (0.157) | 0.110 (0.136) | 11.520 (11.923) |
| RNN | 0.155 (0.144) | 0.114 (0.108) | 9.316 (9.294) |
| GRU | 0.150 (0.143) | 0.105 (0.115) | 14.102 (13.619) |

*4 neurons*

TABLE 4 compares models for google trend time series forecasting, Single-Layer LSTM is compatible with RNN and GRU neural networks. The prediction performance comparison shows that Single-layer LSTM has about 5% and 2% fewer errors then RNN and GRU. The processing time for LSTM is reliable and at the same level as RNN but longer relative to GRU.

# Conclusion

This study presents the design, training, and testing of LSTM neural network for GT series forecasting. Five years of GT data were used for training and testing the model. The single-layer LSTM achieved comparable forecasting performance and showed similar results as more complex stacked LSTM networks and can be compatible with RNN and GRU.

The trend component of time series is one of the most challenging issues across many forecasting tasks. This study examines the capability of LSTM neural networks in modeling a real time series with a trend. Experiments are conducted on GT time series. The forecasting accuracy with raw series showed about the same accuracy compared to first and second differenced GT observations. These results are in line with studies that confirming the neural network’s ability to capture a trend and do not require additional detrending procedures before training.

The comparative studies in terms of layer size show that the neuron’s number may also affect the forecasting performance. [11]. Therefore all our tests use a different layer size. The model forecasting accuracy of Single-layer LSTM proportionally increased for about 0.8 % with each new added neuron to the model. But this gradual increase stops when the number of added neurons reached 60, then starts decreasing rapidly; the model performance dropped by about 7%. This is in line with studies arguing that the size of neurons influences the forecasting performance but only within a certain optimal number range. By taking into account the model should be fast for the future deployment in forecasting web application, this study suggests optimal range is between 4-8 neurons.

Single–layer LSTM showed 2% less accuracy but used less training time, which is compatible result especially if the speed of learning time is important. This finding in studies claims that single layer LSTM can achieve roughly the same level of forecasting accuracy as multilayer LSTM but required shorter training time.

Multivariate single-layer LSTM showed lesser forecasting accuracy compare to univariate model. This is in line with studies suggesting to use additional explanatory variables instead of treating single series [2], but only with multi-layer complex networks that able properly extract features compare to single layer networks.

Different window sizes tested in this study showed better performance with 12 and 24 window size. The 12 week window size can only capture short trends, but for longer trends the experiments demonstrated better forecasting performance within the window of 24 months. This is in line with theory arguing for greater window size for capturing long nonlinear trends.

The comparison of Single-Layer LSTM with RNN and GRU neural networks showed the same level of fitting degree and accuracy of the prediction results. The proposed LSTM model can be deployed in applications for forecasting GT search queries.

# Limitations and further work

However, there are some limitations to consider, this study uses a differencing method for detrending and then time series is modelled using the LSTM neural network. This imposes a number of constraints and difficulties relating to network training. If nonlinearities are linked to the level, then the neural network is less capable of capturing the trend. Therefore, future experiments can employ other detrending methods. This study used standard LSTM neural network with sigmoid function as the gating function. This function outputs a value between 0 and 1 which may eliminate information concerning the trend component. However, more research is required on the activation of function types and their influence on time series with a trend.

The next aspect that is not explored in this work but can be further researched relates to optimizer, learning rate and batch normalization resulting faster conversions of the model.

In this work a single-layer LSTM network showed the same level of accuracy as complex stacked LSTM networks, but still further research would have to be done to confirm these findings. Specifically, the performance of single-layer LSTM networks should be compared to that of larger LSTM networks in various numbers of nodes to see if they are indeed superior for google trend forecasting.

There is an active field of research on quantum enhanced LSTM. In this study, the quantum vertion of LSTM used only for comparision to classical LSTM neural netwoks, but further research is needed in this direction.

##### Acknowledgment

##### Thanks are due to my supervisor Dr. Fabrizio Smeraldi and co-adviser Dr. Luk Arnaut for their advice and for review of this work.

##### References

1. Hylleberg, S., 1992. General introduction. In: Hylleberg, S. (Ed.), Modelling Seasonality. Oxford University Press, Oxford, pp. 3–14.
2. Farway, J., Chatfield, C., 1995. Time series forecasting with neural networks: A comparative study using the airline data. Applied Statistics 47, 231–250.
3. Claveria, O., Torra, S., Jan. 2014. Forecasting tourism demand to catalonia: Neural networks vs. time series models. Econ. Model. 36, 220-228.
4. Zhang, G. P., Qi, M., Jan. 2005. Neural network forecasting for seasonal and trend time series. Eur. J. Oper. Res. 160 (2), 501-514.
5. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: S.C. Kremer, J.F. Kolen (eds.) A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press (2001)
6. Gre, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10), 2222-2232 (2017)
7. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270(2), 654-669 (2018)
8. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166.
9. Challet, D., & Ayed, A. B. H. (2013). Predicting financial markets with Google Trends and not so random keywords. arXiv preprint arXiv:1307.4643.
10. Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. Economic Record, 88, 2-9.
11. Robin Devooght and Hugues Bersini. Collaborative Filtering with Recurrent Neural Networks. 2016. arXiv: 1608.07400 [cs.IR]
12. Hochreiter, Sepp and Schmidhuber, Jurgen. Long short- ¨ term memory. Neural computation, 9(8):1735–1780, 1997
13. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, “Quantum circuit learning,” Physical Review A, vol. 98, no. 3, p. 032309, 2018.
14. Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, “The expressive power of parameterized quantum circuits,” arXiv preprint arXiv:1810.11922, 2018.
15. Quantum Long Short-Term Memory , SYC Chen, S Yoo, YLL Fang arXiv preprint arXiv:2009.01783
16. Yamak, P.T., Yujian, L., & Gadosey, P.K. (2019). A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. *ACAI 2019*.

**MSc Project - Reflective Essay**

|  |  |
| --- | --- |
| **Project Title:** | Time-series Forecasting: Predicting Google Trend using Long Short Term Memory Neural Network |
| **Student Name:** | Salta Talgat |
| **Student Number:** | 200819617 |
| **Supervisor Name:** | Dr. Fabrizio Smeraldi |
| **Programme of Study:** | MSc Computing and Information Systems |

Project on GitHub: <https://github.com/salta-ak/Google-trend-forecast-with-LSTM>

# INTRODUCTION

Most recent time series forecasting studies have indicated that search engine query data can be readily incorporated into forecasting models, and the most frequently used is the Google Trend (GT). The GT is becoming one of the widely used leading predictor in forecasting studies nowadays [10]. It is a free service which returns the normalized search volume for given terms within a time window and geography. In the majority of studies GT keyword is used as a predictor variable whose observations can be to predict the value of the target variable, but limited research and experience exist with regard to the GT forecasting itself. One of the most important concerns in forecasting is that it is required to know or forecast the future values of predictors in order to use them in the model. This study attempted to forecast GT using LSTM neural network and contribute to existing research gap. An experimented was conducted with five years weekly GT time series as an input data, top three words and their two synonyms were selected related to cryptocurrency, stock price, and Covid-19 symptoms subjects. Based on extensive and critical literature review of time series forecasting models, the LSTM neural networks was selected as a forecasting model. At present, the development of algorithms that can forecast a time series with a trend component is one of the active research areas in deep learning domain [3]. The most frequently used neural network for time series forecasting are the recurrent neural networks with long short term memory (LSTM) [2]. These algorithms have certain advantages when it comes to time series trend component, cycles and seasonality. LSTM not only learns nonlinear associations but also remembers long and short trends within cell states and gates [2]. There is shortage of studies that utilise LSTM neural network for GT forecasting. Thus, study findings contribute to existing research gap on LSTM use for GT forecasting. This study presents the design, training, and testing of LSTM neural network for GT series forecasting. Five years GT data were used for training and testing the model. The single-layer LSTM achieved good performance results and showed similar results as a more complex stacked LSTM networks and can be compatible with RNN and GRU.

# STRENGTHS

This study’s experimental findings concern the approach of selecting optimal number of layers and nodes to enhance forecasting performance and could contribute to future studies designing and proposing LSTM neural networks for GT forecasting. The neural network architecture consists of several hyper parameters that configure the model: number of hidden layers, number of neurons, learning rate, and activation function and optimizer settings. There is no single approach on defining how many layers or nodes to select, depending on the certain data type and problem. Among the most widely used methods are grid search, random search methods, Bayesian optimization. Although they can decrease forecasting errors and can better determine model parameters, the calculation time increases. According to recent studies, some analytical conclusions have been derived that the number of hidden neurons should be between the size of the input layer and the size of the output layer or more precisely less than twice the size of the input layer. Therefore, this study tests different hidden layers and node numbers to define the optimal one.

The comparative studies in terms of layer size show that the neurons number may also affect the forecasting performance. [11]. However, most experimental studies argue that it is specific for each input size, data nature and model, and should be tested for different number of neurons. Therefore, this work considers objective testing forecasting accuracy on different numbers of neurons by taking into account the input size. Therefore all our tests use a different layer size. The number of hidden units in each hidden layer 4, 8, 16, 32, 60 are tested. The network is trained for 100 epochs and a batch size of 1 is used. The model forecasting accuracy of Single-layer LSTM proportionally increased for about 0.8% with each new neuron added to the model, but after 20 neurons accuracy starts to decrease rapidly. This result is in line with studies arguing that the size of the neurons influences the forecasting performance but only within a certain optimal number range. By taking into account that the model should be fast for the future deployment in forecasting web application, this study suggested optimal range is between 8-20 neurons.

Multilayer LSTM neural networks can extract more parameters compared to Single-layer. The ability to learn specific patterns depends on the number of layers, but with more layer training may be more difficult and required more resources. This issue is examined in this work by conducting experiments on different number of layers ranging from 1 to 4. Single–layer LSTM showed 2% less accuracy but used less training time, which is a compatible result especially if the speed of learning time is important. This finding results in studies claiming that single layer LSTM can achieve nearly about the same level forecasting accuracy as multilayer LSTM but required shorter training time.

Many studies in time series forecasting including in the model explanatory variables and transformed from univariate model into mixed multivariate version. Explanatory variables consist information that can control target variable. Despite much research on upward forecasting models that include relevant explanatory variables, there is a gap in studies that uses explanatory variables for GT forecasting. In this study, the experiments conducted provide evidence that the forecasting performance increases by 15% when a set of synonym words is used as an explanatory variable in forecasting within the proposed LSTM neural network. This is in line with studies suggesting the use of additional explanatory variables instead of treating a single series [2],but only with multi-layer complex networks that able properly extract features compare to single layer networks.

The trend component of time series can be short or long term. If window size configuration is small than there is possibility that longer trend cannot be captured. Different window sizes were tested during the experiment. Different window sizes tested in this study showed a better performance with 12 and 24 window sizes. The 12 week window size captures only short trends, but for longer trends this study’s experiments showed better forecasting performance with a window size of 24 months. This is in line with theory arguing for greater window size for capturing long nonlinear trends.

The proposed LSTM model comparison with RNN and GRU models contribute to other comparative studies of different neural networks and future research on recurrent neural networks (RNN). In time series forecasting, the main problem is accurately capturing the trend component that consists of the real-world time series. This aspect is at the forefront of theoretical and applied research. Among the most popular linear methods in time-series forecasting are auto-regressive integrated moving average, partial least squares, lasso regression. For univariate time series forecasting most often used methods are auto-regressive models and autoregressive integrating moving average methods (ARIMA). However, it has also been found that many real time series have trends and the liner approach is insufficient to represent their dynamics [2, 4, 5]. Neural networks are efficient in forecasting time series with a nonlinear trend, especially with LSTM and GRU variant of recurrent networks. A recurrent neural network developed for modeling time series. Comparing to standard multilayer perceptron its hidden states are designed with feed-back connection and allows us to learn temporal associations related to time delay. Meanwhile, the main disadvantage of RNNs such as exploding and vanishing gradients make them weak in capturing the long-term time dependencies [4, 8, 12]. Long short-term memory networks were developed to solve the issue of long-term memory in 1997 by Hochreiter and Schmidhuber [12]. In addition to the hidden state, this network also incorporates the cell state of previous information functioning as a long term memory. The GRU is one of the variants of the RNN network and an alternative version of LSTM proposed by Cho et al. in 2014. If LSTM networks have separate input and output gates then in GRU they are combined, thus reducing the number of parameters. While LSTM stores its longer-term dependencies in the cell state and short-term memory in the hidden state, the GRU stores both in a single hidden state. GRU uses less memory and is faster than LSTM, however, LSTM is more accurate when using datasets with longer sequences. This work presented LSTM model performance compared with other neural network variants such as RNN, GRU. According to results Single-Layer LSTM is compatible with RNN and GRU neural networks. The prediction performance comparison shows that Single-layer LSTM has about 5% and 2% fewer errors then RNN and GRU. The processing time for LSTM is reliable and at the same level as RNN but longer relative to GRU.

This study contributes to the debate regarding trend component of time series. Time series forecasting has often been identified as a very important aspect in medicine, weather, biology, finance and economics forecasting [1,7]. However, real-world data in most cases have trend, cyclical or seasonal components. A trend component can be characterized as long-term upward or downward in the data which can be nonlinear. At present, the development of algorithms that can forecast a time series with a trend component is one of the active research areas in deep learning domain, and has been dominated by two opposing perspectives [3]. Recent studies demonstrate that the neural networks can learn nonlinear time series trend and hence have low forecasting capability if used raw data without removing a trend component. In same time exist studies that claiming that detrending do not increase forecasting performance and neural networks can capture a trend component. But still there is a gap in research accessing the influence of detrending time series before training on forecasting performance using LSTM. In order to attempt to fill this research gap, experiments in this study are conducted with and without removing the trend component. Differenced series on second level shows 25% more errors compared to first level differencing and primary data performances.

This study proposed GT forecasting model using LSTM neural network and highlighted findings can be can be used by forecasters. Thus this study demonstrates that the LSTM neural networks can be efficient for recognizing the behaviour of nonlinear or dynamic time series which has trend components.

# WEAKNESSES

However, there are some limitations to consider. This study uses differencing method for detrending and then the time series is modelled using LSTM neural network. This imposes a number of constraints and difficulties relating to network training. If nonlinearities are linked to the level, then the neural network is less capable of capturing the trend. Therefore, future experiments can employ other detrending methods. This study used a standard LSTM neural network with a sigmoid function as the gating function. This function outputs a value between 0 and 1, and more research is required on the activation of function types utilized in LSTM and its influence on trend component capturing.

The next aspect that is not explored in this work that can be further researched related to faster conversions aspects. The growing use of machine learning based forecasting web applications required fast learning models. In this line, the efficient optimizers play a very important role. In this work the Stochastic Gradient Descent (SGD) was used in proposed LSTM. However, to better ensure the forecasting ability of LSTM for GT time series, further studies and discussions on other variants of optimizers are necessary. There are several other variants of optimizers, but in terms of maximally capturing a trend variations the Root Mean Square propagation (RMSProp) and Quantum natural Gradient optimizer can be tested.

In this work single-layer LSTM network showed the same level of accuracy as complex stacked LSTM networks, but still further research would need to be undertaken to confirm these findings. Specifically, the performance of single-layer LSTM networks should be compared to that of larger LSTM networks in various numbers of nodes to see if they are indeed superior for google trend forecasting.

There is an active field of research on quantum enhanced LSTM. In this study the quantum version of LSTM used only for comparison to classical LSTM neural network, but further research needs to move in this direction.

# Presentation of possibilities for future work

If I had more time on this project, I would want to focus on increasing the speed of the model conversion by developing the model into a quantum-hybrid LSTM. The neural network training is an iterative process that required time. The application of quantum approach may speed up the training process. The theory of quantum computations is an important and rapidly developing area of deep learning theory nowadays. One of the most widely used quantum algorithms that incorporated into classical deep learning methods in classical computers are variation quantum algorithms (VQA) [13]. This algorithms showed good results on optimization and classification problems [14]. Experiment results [23] demonstrate that the variation quantum circuits can be used for time series forecasting. However, despite the many studies, more research on the problem of learning time series in quantum field is still required.

Finally, if provided with more time, I would like to build a machine learning based GT forecasting web application with user input of keywords. Most small and medium enterprises cannot afford the cost of forecasting software and relay on internet data such as GT in their sales, inventory or demand planning.

COMPARISION BETWEEN THE THEORETICAL AND PRACTICAL WORK

This study attempts to ensure better integration between the theoretical and the practical work. In practice, forecasting time series has always been a huge chal­lenge. Mainly, because of seasonality and trend, it is difficult to achieve a reliable forecasting performance. Developing forecasting models has always been a resource consuming and costly process, especially when it comes to data collection. The use of publicly available Internet data such as Google trends can be efficient, but this is nowcasting rather than forecasting where a longer leading lag is required. This project presents the design, training, and testing of a long-short-term-memory neural network for google trend series forecasting. Five years of google trend data was used for training and testing the model. One of the examples where the proposed model can be used is web applications that utilize customer input series and Google keywords and output forward forecasts.

# LEGAL, SOCIAL ETHICAL ISSUES AND SUSTAINABILITY

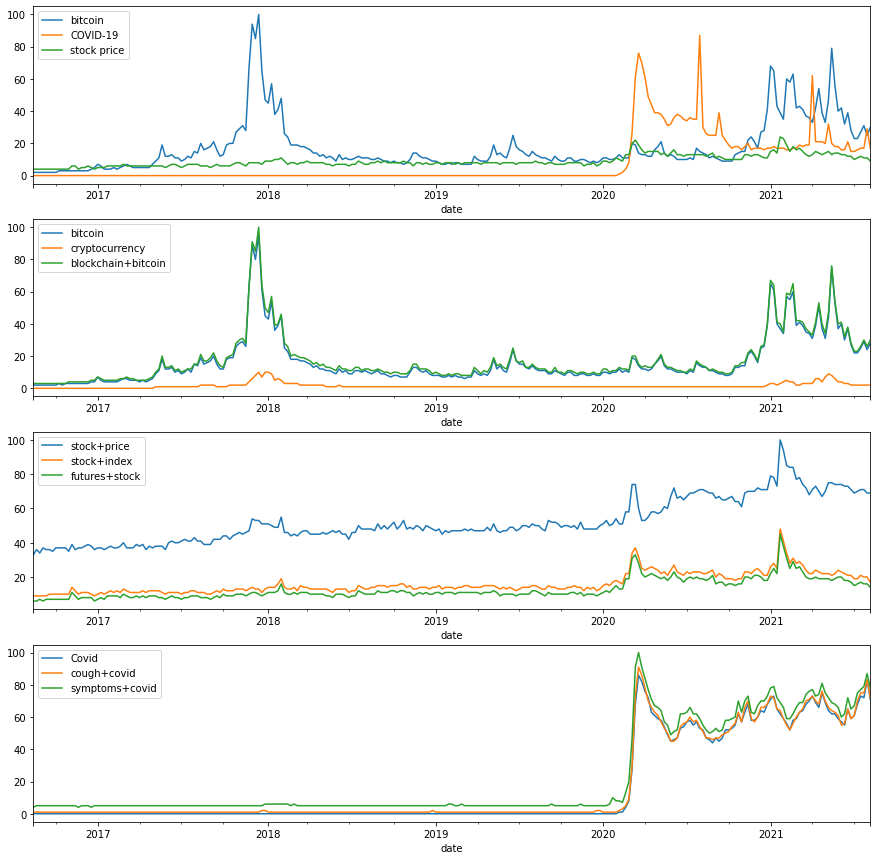
This study forecasts GT using LSTM neural network. Concerning the legal, social and ethical issues, there is no potential barriers to forecast GT. Privacy concerns of GT is managed appropriately to ensure that data linkage is done in accordance with legislative requirements. GT encapsulate data from search engines, that log all search terms in a database along with the date and time of search, browser and operating system, IP address of user, the Google cookies, and the URL that shows the search engine and search query. GT data publicly available for no charge, search queries are anonymized and aggregated.

# REFERENCES

1. Hylleberg, S., 1992. General introduction. In: Hylleberg, S. (Ed.), Modelling Seasonality. Oxford University Press, Oxford, pp. 3–14.
2. Farway, J., Chatfield, C., 1995. Time series forecasting with neural networks: A comparative study using the airline data. Applied Statistics 47, 231–250.
3. Claveria, O., Torra, S., Jan. 2014. Forecasting tourism demand to catalonia: Neural networks vs. time series models. Econ. Model. 36, 220-228.
4. Zhang, G. P., Qi, M., Jan. 2005. Neural network forecasting for seasonal and trend time series. Eur. J. Oper. Res. 160 (2), 501-514.
5. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: S.C. Kremer, J.F. Kolen (eds.) A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press (2001)
6. Gre, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10), 2222-2232 (2017)
7. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270(2), 654-669 (2018)
8. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166.
9. Challet, D., & Ayed, A. B. H. (2013). Predicting financial markets with Google Trends and not so random keywords. arXiv preprint arXiv:1307.4643.
10. Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. Economic Record, 88, 2-9.
11. Robin Devooght and Hugues Bersini. Collaborative Filtering with Recurrent Neural Networks. 2016. arXiv: 1608.07400 [cs.IR]
12. Hochreiter, Sepp and Schmidhuber, Jurgen. Long short- ¨ term memory. Neural computation, 9(8):1735–1780, 1997
13. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, “Quantum circuit learning,” Physical Review A, vol. 98, no. 3, p. 032309, 2018.
14. Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, “The expressive power of parameterized quantum circuits,” arXiv preprint arXiv:1810.11922, 2018.
15. K. Fujii and K. Nakajima, Harnessing disorderedensemble quantum dynamics for machine learning, Phys. Rev. Applied 8, 024030 (2017).
16. Yamak, P.T., Yujian, L., & Gadosey, P.K. (2019). A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. ACAI 2019.

Appendix 1. Google trend on selected keyword

<https://github.com/salta-ak/Google-trend-forecast-with-LSTM/blob/main/all_diss.png>



1. [↑](#endnote-ref-1)
2. [↑](#footnote-ref-1)