Nowcasting Gross Domestic Product with Long Short-Term Memory Networks

Kamran Nejad-Sattary
Course Project - Financial Econometrics FIN-407, EPF Lausanne, Switzerland
20 June 2021

Abstract—**Gross Domestic Product (GDP) Nowcasting remains a valuable yet demanding practice. This paper sets out to examine how with aid of strong long short-term networks, capable of capturing memory by processing sequences of data, such a strenuous task can be affronted. In a first part, attention is afforded towards devising a pre-processing pipeline to enable the modelling of monthly level GDP growth. Then two separate models are constructed, from financial and macroeconomic data, with the intention of contrasting predictive power of the differing input data. Model selection is performed via iterations over an extensive grid search, incorporating as hyperparameters the pre-processing imputation methods, as well as network architecture. Findings yield that additional complexity is not leveraged by the models, and that the choice of the loss function is crucial during training, as the lack of a harsh penalty provided by the Mean Absolute Error leads to a futile model, disregarding economic shocks in favour of predicting a flat growth rate. While the best performing Mean Squared Error model does attempt to identify such shocks, it struggles to quantify the magnitude of these black swan events. Further, inconsistent releases across time of macroeconomic indicators point towards a simpler implementation over longer time horizons of financial-driven networks.**

I. Introduction

Owing to the recent proliferation of Machine Learning — in particular Neural Networks — across sciences, the impending interrogation of whether the translation of these computational self-learning techniques towards financial econometrics enables and leverages additional performance, remains partial. Indeed, in their specific applications, a variety of research, e.g. [1], [2], and [3] have already demonstrated the ability of Neural Networks to leverage the use of macroeconomic, financial data, and even financial news in favour of outperforming existing techniques.

One such econometric challenge pertains to the nowcasting of the Gross Domestic Product (GDP), the practice of predicting the present Gross Domestic Product, which due to its lengthy aggregation procedure, accompanies a delayed release, and is even subject to revisions. Allowing for the realtime estimation of the GDP allows actors, be it the federal reserves or the government to also act in real-time to changes in the economy, by adjusting and reassessing for example interest rates, or foreseeing unemployment to supplement social aid. As such, should nowcasting become reliable, the lag resulting from the counting procedure of GDP may be overcome. While this on the one hand may favour and enable timely policy-making, the potential harm caused from significant policy adjustments founded on erroneous estimates must also be weighted into the picture, only reemphasizing the crucial importance of a refined estimation procedure.

While consensus emerges over the state of the art of Machine Learning in alternative application fields, little consensus has risen over the praxis of nowcasting Gross Domestic Product, as evidenced by the diverging employed methods even amongst Federal Reserves, see [4] and [5], justifying further investigation into this domain.

This study aims to contribute to this discussion, by presenting its own solution and findings, utilizing the fabled long short-term memory architecture, which despite merely residing in its infancy for nowcasting economic variables, has shown promise over existing Dynamic Factor Models [3], [6]. Attention is first afforded towards the chosen input data and accompanying pre-processing techniques, prior to depicting the modelling and hyperparameter selection procedure, with the objective of nowcasting monthly US GDP returns. Finally, scrutiny is provided towards model tuning and results, and accompanied by both critical discussion of shortcomings of the proposed methodology and potential further improvements.

II. Data Methodology

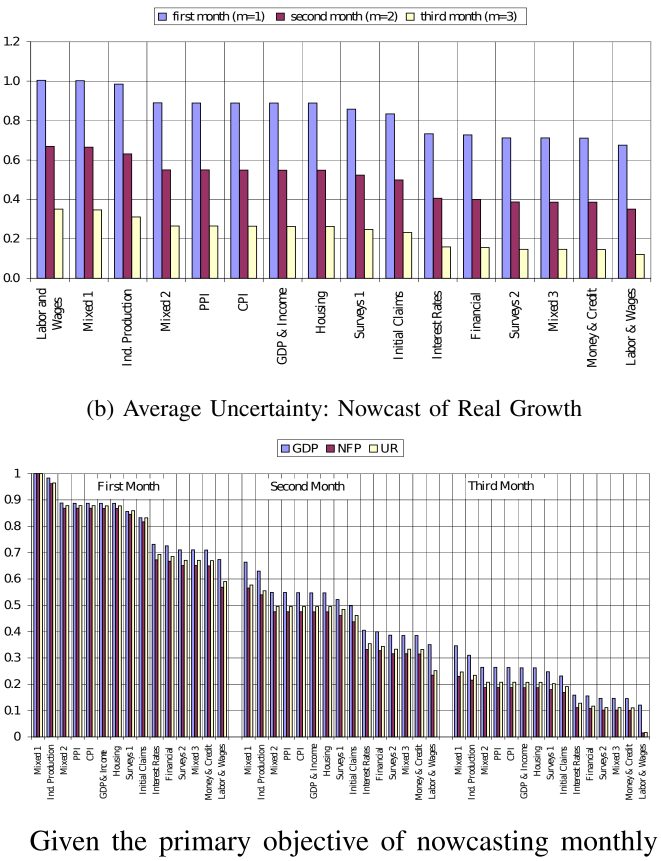
A. Data Selection

A crucial first step into devising a full fledged pipeline for the nowcasting of GDP concerns the raw data input, which will itself later embody the trained model. While a large portion of macroeconomic data is accompanied by the disadvantage of being difficult to compute, and requiring of a lengthy measuring procedure, leading to data only being available after the fact, financial data is readily available and effortlessly accessible. Furthermore, financial markets allow for a temporal granularity by the second which remains unreciprocated by macroeconomic indicators, many of which are at best obtainable quarterly.

In fact, Giannone et al. [7], one of the first studies to formalize the nowcasting problem into a comprehensive framework, stresses the importance of a temporally granular data set, as quarterly releases of data result in a substantial loss of information, again advocating for the financial figures. Figure 1 illustrates their preliminary findings, supported by their construction of both a news and uncertainty measure. Specifically and in contrast to the quarterly level, intra-month information contains additional pertinent detail, as monthly data releases lead to estimate updates in 1a, and uncertainty decreases uniformly throughout the quarter, as portrayed in 1b. The multitude of benefits financial time series present, justifies their inclusion into the modelling pipeline of this study, however for comparison purposes a macroeconomic-driven model is constructed too.

Figure 1: The Importance of Temporal Granularity [7]

(a) Average Size of News: Nowcasts of Real Growth



(b) Average Uncertainty: Nowcast of Real Growth

Given the primary objective of nowcasting monthly US GDP to identify large economic shocks, particular emphasis was placed upon incorporating a wide span of shocks when assembling the data set. Therefore time series were assembled over the horizon of the last thirty years, between the years 1990 up until the end of 2019. Specifically, for financial series, the CRSP database [8], facilitated the access of daily stock returns, whereby the top 20 stocks, determined by market-cap at end of 2019, were retained, and combined with the `TFZ_MTH_RF` serie representing the monthly risk-free rate, as well as the two and ten year treasury yields. Major Market Indices, the `SP500`, `NASDAQ`, `NIKKEI`, `HSI`, and the `DAX`, were accessible via Yahoo Finance [9], from which monthly returns were computed utilizing the monthly adjusted close prices.

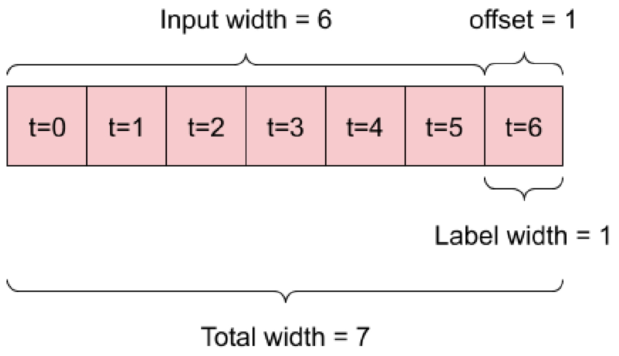
On the other hand, ease-of-access was unattainable for macroeconomic variables. In particular, accessing series fully extending over the decided thirty year horizon, was troublesome. Furthermore, a variety of macroeconomic variables were only recorded from the start of the millennia, while others' historical values only became available at this point. Indeed, the question of whether it would be sensible to nowcast Gross Domestic Product in 1990 with knowledge attained in the year 2000, remains crucial, and this study did not contribute to this potential information asymmetry between a present and future nowcasting agent. As such, solely figures available at the time were included into the nowcast of current GDP, leading to the selection of only five macroeconomic variables. These included the seasonally adjusted real GDP, exports and imports of goods and services, unemployment rates, as well as unadjusted total public debt. For comparison purposes, and despite inconsistent release dates of macroeconomic indicators over longer time horizons, this same time horizon from 1990 to 2019 was maintained.

B. Pre-Processing

Given the numerous collected time series, the treatment of singular time series remained infeasible, emphasizing the importance of creating a unified pre-processing procedure for all series. First, series were transformed to growth-rates, serving the function of normalization. Secondly, should there be missing observations, or should series only be provided on the quarterly-level, imputation methods were required to fill these gaps, in order to preserve the entire thirty year horizon and monthly-level granularity of the financial time series. Both linear interpolation, and a Seasonal Auto-Regressive Integrated Moving Average (SARIMA(p,d,q)) model were employed and compared, constituting an additional tuning hyperparameter. Selection of the optimal p and q was made according to the lowest Akaike Information Criterion amongst all combinations $p, q \in \{1, ..., 6\}$, while no additional integration was imposed. Indeed, the assumption here of stationarity employed by the model, although perhaps aided by the growth-rate transformations, remains presumptuous. Nonetheless, this serves a simplifying modelling assumption. Future elaboration of a more complex SARIMA-GARCH model may help tackle data embedded heteroskedasticity.

Lastly, windowing of the time series was performed, in order to provide directly into the long short-term memory network its desired sequence structure. Explicitly, this procedure consisted of constructing all possible observation pairs of $x, y = (x_{t-w-1}, ..., x_{t-1}), (y_t)$ where x represents the matrix of column vectors of the feature variables known from time $t - w - 1$ to $t - 1$, and y is the Gross Domestic Product at time t, with w being the selected window size, constituting an additional tuning parameter. Figure 2 illustrates this maneuver given a window size w of 6.

Figure 2: Windowing Procedure for $w = 6$



III. Model Selection

A. Model Architecture

Persistent direction towards the outperformance by long short-term memory (LSTM) architectures of traditional time serie forecasting techniques, e.g. [6], [10], hint and justify further investigation into these methods. The LSTM architecture, originally coined as such by Hochreiter and Schmidhuber [11], enables this form of recurrent neural network (RNN) to simulate a memory component via the addition of a memory cell alongside traditional hidden node outputs. This longer-term memory cell constitutes a powerful mechanism to overcome and upgrade the standard recurrent network, which is unable to deal with long minimal-time lags between relevant signals [12]. Specifically, via a modified RNN coupled with a memory cell, gates tamper with the long-term memory cell and the shorter term traditional outputs, in order to make use of a two component system to preserve longer horizon information for later hidden nodes usage. While traditionally LSTM layers were employed alone, as it was not perceived useful to stack such layers vertically, this was reexamined in both the fields of Natural Language Processing and Speech Recognition, where the incorporation of vertical stacking led to performance increases [13], [14], resulting in the addition of vertical LSTM stacking to the list of hyperparameters to tune for.

Acknowledging the vast array of hyperparameter options presenting themselves in LSTM architectures, be it this vertical stacking of hidden layers, the incorporation of dropout layers to reduce overfitting, or the very choice of the number of hidden neurons in each layer, a sturdy model selection pipeline is conceived. The assembled data set is first split into a training, validation and test set, each accounting for seventy, twenty and ten percent of the data set respectively. In this approach, a training set is employed and will embody the trained model. The validation set gains its utility from its ability to simulate an out of sample data set, which serve to identify best performing hyperparameters according to a chosen performance metric. Finally, true out of sample performance, and to detect and avoid overfitting towards the validation set, is obtained from the test set.

B. Hyperparameter Tuning

In this light, a large grid of all hyperparameter combinations is gathered, and extensively iterated throughout. Random search instead of extensive grid search may lead to a near-optimal solution and reduce computational complexity by reducing the number of necessary grid-combinations [15], though due to rapid convergence when coupling the already small sample size with early stopping criterions, was deemed an unnecessary shortcut in this pipeline. A vital step into devising this grid was the cautious selection of ranges for each tuning parameter. Indeed, should the performance criterion suggest that the parameter value at the edge of the specified range is best, it would be questionable to not test moving even further into that direction.

Table I contains the list of all tuning parameters tweaked, with their associated tested values. All pre-processing variations, including the imputation methods for filling values, the window-sizes fed into the model, and the incorporation of Principal Component Analysis (PCA) by themselves introduce a vast and diverse grid. The addition of layer stacking, various hidden neuron specifications and intermediate dropout layers at each vertical layer, enable the model to flexibly adapt to the different pre-processing techniques, and verify whether combining with these model variations increases performance. As for the hidden LSTM layers, these are identical, and each use the standard hyperbolic tangent activation function, with sigmoidal recurrent activations and a final linear layer due to the regression nature of the task at hand. Both the Mean Squared Error and Mean Absolute Error losses were put to use

Table I: Grid Hyperparameters

Imputation	Linear, SARIMA
Window	6, 8, 12, 16 months
PCA	None, Capturing 90% Variance
Layers	1, 2, 3
Neurons	25, 50, 75
Dropout	None, $p = 0.2$ and 0.5
Loss	Mean Squared Error, Mean Absolute Error

on training and performance evaluation, due to their different approaches in penalizing outliers. Specifically, intuition would suggest that should one want to attribute most importance to identifying large economic shocks, the harsher penalty from the squaring in the Mean Squared Error may force the model to not disregard these black swan events, and favour predicting a near constant growth rate.

Finally, model training was performed by minimizing the selected loss function over all combinations of the defined grid. The selected optimizer was ADAM due to its efficiency and robustness, particularly its ability to compute individual parameter-specific adaptive learning rates, with the incorporation of the traditional learning rate under the form of decay rate, causing it to be invariant to gradient rescaling [16]. An arbitrarily large number of epochs (300 with each 1000 iterations) was selected and coupled with an early stopping criterion which upon first sight of the validation loss worsening, performed two additional epochs of 1000 iterations. This ensured that each model combination, was provided appropriate training time which may well vary per model complexity or per random weight initiation. Records of the model performance of both the final validation error and test error were stored, alongside the charts of accompanying predictions over the thirty year horizon to assist performance examination.

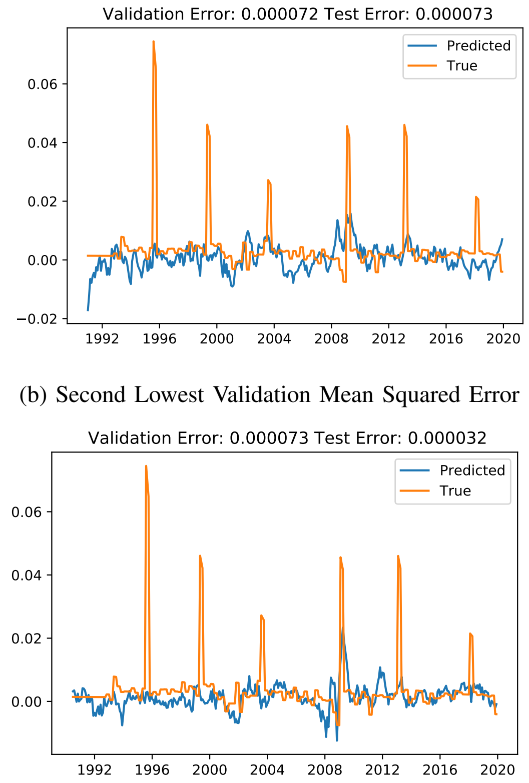
IV. Results & Discussion

In this section, training results are critically assessed. Particular emphasis is placed on comprehending why certain hyperparameter modifications lead to certain performances, in order to adjust the depth and breadth of initially composed grid search, and suggest further improvement opportunities.

A. Choice of the Loss Function

A primary struggle across the modelling procedure concerned the unwillingness of the network architecture to attempt to fit anomaly time periods. Indeed, rather than to attempt to fit rare economic shocks, and as byproduct identify shocks where they are none, or mischaracterize the extent of shocks, it remains facile during training to favour a less daring nowcasting fit. The introduction of both the Mean Squared Error (MSE) alongside the Mean Absolute Error (MAE) served the purpose of comparison in this regard. Figure 3 and 4 illustrate the best results of both the MSE and MAE, determined according to the respective validation losses.

Figure 3: Financial built LSTM Mean Squared Error Performance



(a) Lowest Validation Mean Squared Error

(b) Second Lowest Validation Mean Squared Error

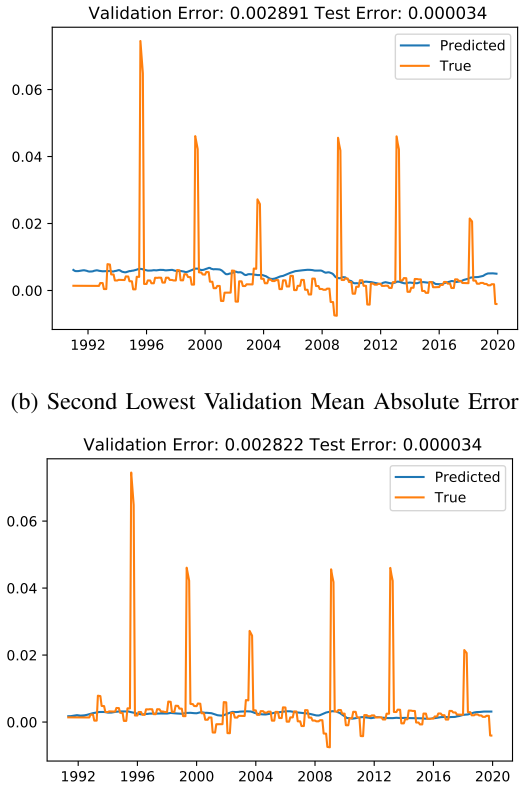
It is remarkable at first glance that while the MSE attempts to nowcast shocks, as is particularly noticable in both 3a and 3b around the notorious global financial crisis period, it rarely is daring enough to characterize or grasp the full extent of the shock. On the other hand, the MAE, illustrated in figure 4 selects a near constant growth rate, defeating the initial objective of identifying short-term economic shocks and rendering the trained model superficial, likely due to its lack of a harsher squared penalty term. Despite the struggles of both loss functions, the low test error and ability of the LSTM portrayed in 3b to accurately mimick smaller growth movements is considerable. While the figures illustrated only portray a specific combination of hyperparameters, these findings were consistent over the grid too.

B. Comparing the Financial and Macroeconomic built models

Up until this point, only the financial-driven models have been examined. In contrast to these models, the macroeconomic ones, of which the best performing one is illustrated in figure 5, appear particularly daring in their nowcasting, as observed by the monumental predicted downturn in growth in 1997. While indeed it is desirable for the chosen model to overcome the aforementioned inability to indicate the entire magnitude of shocks, the disadvantages of erroneous nowcasting upon which significant policy adjustments are founded remain costly too. Examination into the data input of this macro-built model, highlights the inconsistent release dates throughout the 1990s. In particular, the quarterly release of data is not upheld throughout longer horizons, thus tampering with the windowing procedure which no longer consistently represents an even lag (recall the lags are performed upon data available in a certain month, not the data values corresponding to that month) across observations in the dataset.

Figure 4: Financial built LSTM Mean Absolute Error Performance

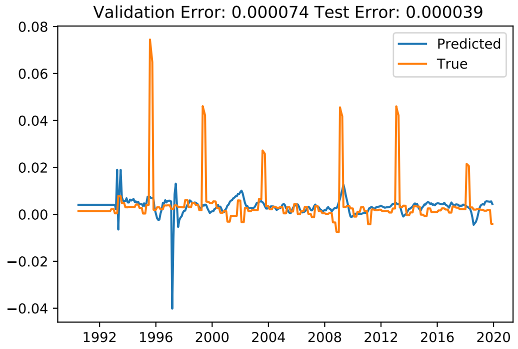
(a) Lowest Validation Mean Absolute Error



(b) Second Lowest Validation Mean Absolute Error

Furthermore, it appears that in contrast to the financial models depicted in figure 3, forecasts are smoothe, presenting little monthly-level granularity. Since both the best financial-driven and the best performing macroeconomic models nearly only vary in their input, as each of them display preference for similar tuning parameters, namely a shallow and less complex model architecture as is later examined, perhaps this smoothening results from the substantial use of imputation techniques such as linear interpolation (which was unanimously selected) to obtain monthly values from quarterly released macroeconomic data. Consequently, the incorporation of time granular nowcasting seems impaired. Should this granularity prove vital, further model improvements may result from increasingly frequent data points, i.e. daily level stock returns, suggesting that the full utility and benefits from financial data may not yet be exploited.

Figure 5: Best Performing Mean Squared Error trained model



C. *Examining Hyperparameter Selection*

While the previous two subsections did not afford much care towards clarifying which hyperparameters were preferentially paired with others for the sake of legibility, preferences over the aforementioned loss functions and input data types remained overall stable. This phenomenon was particularly helpful to not make sweeping generalities and presumptuous conclusions regarding specific parameter choices.

Table II includes for the sake of completeness the hyperparameters employed in creating each of the previous figures. The MSE shows inclination towards a minimal model with a single layer, a relatively small window size and the inclusion of dropout, while the best performing MAE networks tended towards a larger number of layers and larger window sizes. This phenomenon was replicated throughout the grid iterations, although differences were extremely minimal.

Figure 6 illustrates the validation and test errors over this search for all hyperparameter combinations using the Mean Squared Error loss, for varying window sizes, although varying the number of LSTM layers, the number of hidden neurons, or the Dropout yield analogous findings. Extremely minor decreases in performance are on average observable for increasing window sizes, while for the Mean Absolute Error, the opposite phenomenon occurs, the performance increases ever so slightly for larger window sizes. These decreases and increases remain so minor that should one decide on a final model, Occam's Razor would suggest to pick the simplest, given that the additional complexity is not truly leveraged. As such, across the board, it appears that a shallow architecture is favoured in this scenario, although the incorporation of additional predictors may distort this finding.

Table II: Grid Hyperparameters

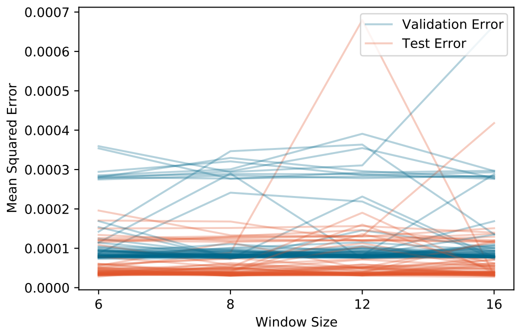
Figure	3a	3a	4a	4a	5
Imputation	Linear	Linear	Linear	Linear	Linear
Window	12	6	12	16	6
PCA	None	None	None	None	None
Layers	1	1	3	3	1
Neurons	50	50	50	50	25
Dropout	0.2	0.5	0.2	0.2	None
Loss	MSE	MSE	MAE	MAE	MSE

D. Limitations & Improvement Opportunities

Analogously to the finding that most these hyperparameters only tweaked the models to a very minor extent, Principal Component Analysis did not prove particularly beneficial. In fact, in the financial-driven model, the incorporation of PCA, with the first n components required to capture over 90% sample variance, resembled the smoothing effect observed in the less-granular macroeconomic models. Remarkably, to capture over 90% sample variance, merely the first principal component was needed, hinting that input data is redundant. Perhaps the inclusion of alternative financial and macroeconomic data, the construction of a model built upon both of these, or even including drastically different and less-intuitive factors could prove beneficial.

Indeed, the breakthrough study provided by the comprehensive framework of Giannone et al. in cooperation with the European Central Bank, assembled over two hundred macro-economic indicators alone, before even considering the inclusion of financial figures. Principal Component Analysis' ability to project the feature set into a lower dimensional space, allows the inclusion of a majority of this information into the modelling procedure without suffering the full extent of the curse of dimensionality and noise. While perhaps it remains difficult to access this quantity of data without cooperation of entities such as the European Central Bank, this at least suggests a further avenue for potentially improving model performance, as perhaps differing shocks are driven by differing phenoma, which remain unidentified in the data set devised in this study's pipeline, but even more interestingly, this decomposition into a singular factor accredits the latent factor theory.

Figure 6: Performance by Window Size



As for pre-processing improvements, the SARIMA imputed macroeconomic data led both, on average, and within comparisons amongst best performing models for each loss function, to a worse-performing model. The presumptuous assumption of homoskedasticity may lead to this SARIMA model being more harmful than beneficial in the pipeline. The replacement in later attempts with a GARCH-type model may aid in this regard.

Lastly, the addition of an increasing number of hyperparameters, which due to the expontential expansion in parameters would drastically extend the grid search procedure, could aid in shaping the model towards its desired objective. Including differing loss functions within the LSTM components, be it fully-linear layers, or even including additional weight to economic shocks via a harsher training loss function, are all worthwhile investigating. However, owing to the substantial attention which has been provided into devising an already robust pipeline for hyperparameter selection and tuning, beyond the additional computational complexity, this extension remains effortless.

V. Conclusion

All in all, this study addressed how to approach devising a full-fledged pipeline for the adoption of long short-term memory networks to nowcast monthly US Gross Domestic Product, both for financial and macroeconomic-driven models. In a first step, the incorporation and pre-processing of a variety of time series, with differing imputation and windowing methods, is tackled. Next, upon decision to create a long short-term memory network, capable of representing memory via the addition of a memory cell alongside the traditional RNN architecture, a powerful extensive grid of hyperparameters is gathered, and iterated throughout. The multitude of parameter combinations tested led to the best performing model, and demonstrated as crucial the selection of the training loss, the Mean Squared Error, and the pre-processing imputation via linear interpolation. The inclusion of LSTM layer stacking, or additional parameters providing complexity to the model, remained mostly unleveraged by the model, pointing towards a more simplistic, shallow architecture for the task at hand. As for the comparison of Financial and Macroeconomic data, the inconsistent data releases of macroeconomic indicators in the 1990s proved harmful for the training procedure, and confused the model. Lastly, Principal Component Analysis only requiring a single component to capture over 90% of data variance, highlighted potential of a highly redundant data set. The introduction of alternative factors, or the combination of both Macroeconomic and Financial variables, provide further opportunities for investigation and improvement.

Supplementary Materials

The methodology including all data collection, pre-processing, and computations steps are publicly accessible in the git repository https://github.com/KamranSattary/FinEcon21 or here.

References

[1] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[2] A. Nazemi, J. Jakubik, A. Geyer-Schulz, and F. J. Fabozzi, "Incorporating financial news for forecasting bitcoin prices based on long short-term memory networks," *Available at SSRN 3733398*, 2020.

[3] J. Loermann and B. Maas, "Nowcasting us gdp with artificial neural networks," 2019.

[4] S. Grover, K. L. Kliesen, and M. W. McCracken, "A macroeconomic news index for constructing nowcasts of us real gross domestic product growth," 2016.

[5] Federal Reserve Bank of New York, "Nowcasting report," https://www.newyorkfed.org/research/policy/nowcast/methodology.html, accessed: 30 May, 2021.

[6] D. Hopp, "Economic nowcasting with long short-term memory artificial neural networks (lstm)," 2021.

[7] D. Giannone, L. Reichlin, and D. H. Small, "Nowcasting gdp and inflation: the real-time informational content of macroeconomic data releases," 2006.

[8] Center for Research in Security Prices, "Crsp/compustat merged," http://wrds-web.wharton.upenn.edu/wrds/, accessed: 30 May, 2021.

[9] Yahoo, "Yahoo finance," https://finance.yahoo.com/, accessed: 15 May, 2021.

[10] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1394–1401.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] S. Hochreiter, and J. Schmidhuber, "Lstm can solve hard long time lag problems," *Advances in neural information processing systems*, pp. 473–479, 1997.

[13] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649.

[14] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[15] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.