Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 222 (2023) 488–498

Procedia Computer Science

www.elsevier.com/locate/procedia

International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023)

The Multi-Recurrent Neural Network for State-Of-The-Art Time-Series Processing

Oluwatamilore Orojo[a], Jonathan Tepper[a,b], T. M. McGinnity[a,c], Mufti Mahmud[a]

[a] *School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, United Kingdom*
[b] *Perceptronix Ltd, Avon Way, Derby DE65 5AE, United Kingdom*
[c] *Intelligent Systems Research Centre, University of Ulster, Magee Campus, Derry BT48 7JL, United Kingdom*

Abstract

Innovations in recurrent neural networks (RNNs) for time-series modelling has enabled effective prediction of sequence data prevalent in real-world dynamic processes. For example, problem domains such as speech and language modelling, weather prediction, financial forecasting and patient healthcare monitoring have all significantly benefited from developments in RNNs over the last 20 years. Today's vast availability of data has enabled models to learn from higher quality historical information, identify loopholes, and better understand and contextualise evolving information in real-time. This is arguably extremely important as access to fast reliable predictive information empowers decision makers to prepare for and effectively respond to future opportunities or crises. Also, the simpler the forecasting model, the more amenable it is to residing on multiple mobile platforms improving accessibility of predictive power. Given this, we comparatively assess the importance of the Multi-recurrent Neural Network (MRN) with a single hidden layer against current state-of-the-art models using a range of real-world problem domains from oil price prediction to predicting the spread and mortality rate of COVID-19. We find strong evidence that the simple and shallow MRN consistently offers superior performance over the Long Short-term Memory model (LSTM), the current state-of-the-art RNN and Support Vector Machines (SVM), a more traditional statistical approach. The MRN required much fewer adjustable parameters than the LSTM to learn the task and generalise competitively. This suggests that the simpler architecture offers significant value with respect to computational resources and effective predictive abilities for real-world applications which is particularly useful given the current ubiquitous shift towards the use of Internet of Things devices.

Keywords: Time-series modelling; Predictive analysis; Multi-Recurrent Network; MRN

E-mail address: jtepper@perceptronix.net

1877-0509
10.1016/j.procs.2023.08.187

1. Introduction

Recent innovations in machine Learning models have revolutionised how we perceive, interpret and predict various real-world phenomena. More specifically, the Recurrent Neural Network (RNN) class of models have been shown to be well-suited and possess the ability to drastically enhance predictive performance for time-series modelling over traditional statistical approaches such Vector Autoregressive Models and Kernel Regression methods. This is largely due to their sophisticated memory mechanism which provides a storage bank to better understand and conceptualise our world [1]. The embedded memory within RNNs enables them to capture temporal dependencies for the identification of interactions in spatio-temporal domain, thus informing behaviour dynamics [2]. These models are trained using the Back-propagation Through Time (BPTT) where the network 'unfolds' over a given number of time steps as parameters adapt temporally (given the interdependence between outputs over time). [3].

A number of RNN variants have been widely applied, for example; Simple Recurrent Networks (SRNs), Jordan networks, Echo-State Networks (ESNs) and current state-of-the-art, Long-Short Term Memory (LSTM) and have proven to offer superior performance. The SRN, a simple and well-known RNN variants, employs feedback loops within the memory to store historical state information relating to the preceding time step. Another simple RNN variant is the Nonlinear Autoregressive models with eXogenous input network (NARX) which accounts for historical temporal dependencies using lagged input and output variable values. Whilst the simplicity offered by the SRN and NARX models provides a useful level of computational efficacy, performance is compromised as simplification of the historical information leads to a loss of information explicitness and thus representation [4]. In addition, these models can not deal with an long histories [5, 4].

More complex RNN variants such as ESNs [6] and LSTMs [7] have been developed to deal with such limitations of simpler RNNs. ESNs possess a high dimensional hidden layer consisting of a dynamic reservoir of non-linear hidden neurons that are sparsely interconnected with fixed random weight strengths allowing for faster training. The weights between the dynamic reservoir and output units can be learned so that the ESN can reproduce specific temporal patterns [8, 9]. Whilst the ESN performed well on small toy synthetic data sets, it's applicability to large real-world problem domains has not yet been proven due to the computationally intractable reservoir sizes that would be required. Whereas, the LSTM employs a sophisticated gating mechanism that determines how much historical information is retained, forgotten and utilised by the network during learning and has been proven to learn complex real-world temporal problems. However, the gating mechanism is convoluted and increases model complexity making them difficult to train and easy to over-fit (easily latching onto noise rather than critical temporal dependencies) [10, 11, 12].

Ulbricht [5] presented the Multi-recurrent Neural Network (MRN), an RNN employing four levels of feedback allowing recurrent connections from: (i) the output layer back to dedicated context units in the input layer, as found in Jordan networks [13], (ii) the hidden layer back to the dedicated context units in the input layer, as found in SRNs [14], (iii) the external input nodes back to the dedicated context units in the input layer to form input memories, and (iv) from the context units within the input layer back to themselves, i.e., self-recurrent links. To date, very few researchers have applied and assessed the MRN across a broad array of realistic time-series forecasting tasks or sufficiently benchmarked them against more traditional statistical and machine learning approaches [15, 8, 16, 8, 17, 18]. This paper seeks to therefore provide a systematic assessment of the MRN's performance and suitability for time-series modelling for real-world applications.

The structure of the paper: Section 2 provides a very brief introduction to the MRN architecture and forecasting methodology adopted; Section 3 presents the data and pre-processing techniques; Section 4 details the experimental results; Section 5 presents an analysis of the experiments undertaken, and the conclusion is presented in Section 6.

2. Methodology

We implement the MRN as originally proposed by [5] with a single hidden layer as shown in Figure 1. A sliding window approach to time-series processing is adopted and as previously detailed in [17]. The sliding window mechanism allows for independent training sequences to be generated of length n where n is the number of inputs sequentially processed by the MRN before a prediction is stored at time t. As the memory is reset at the beginning of each sequence, the MRN acts as a finite memory model. A detailed mathematical description of the MRN can be found in [17], however, in this paper we also utilise *Input layer recurrency*. The *Input layer recurrency* enables infor-

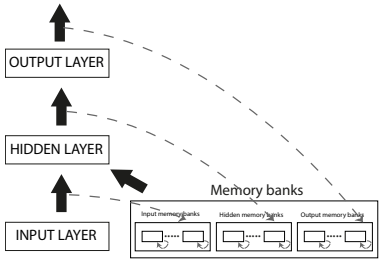


Fig. 1: MRN Architecture

mation from previous input layers to be fed back to the input layer during the forward pass and this is used to update the weights during the Back-propagation Through Time learning algorithm. An additional function to those defined in [17] is incorporated to implement an input memory bank given the input units at time $t-1$, I_{t-1} and the input memory at time $t-1$, M_{t-1_i}. Thus eq.3 in Section III.B in [17] is updated as including the following term $\sum W_{M_{ih}} M_{t_i}$ that where M_{t_i} is the input memory and $W_{M_{ih}}$ its weights to the hidden layer.

An important feature of the MRN is that a 'sluggish state space' is formed when the activation values being fed back from the input, hidden and output layers during each time step are combined, to varying degrees, with their respective context unit values in the input layer (as determined by the weighting values applied to their recurrent links). The influence of the previous context unit values is determined by the weighting values applied to the self-recurrent links. When the weights on the recurrent links are greater than those on the self-recurrent links more flexible memories are formed storing more recent information at the expense of historical information stored in the context units. Conversely, if the weights on the self-recurrent links are greater than those on the recurrent links more rigid memories are formed since more historical information is preserved in the context units at the expense of the more recent information being fed back from subsequent layers.

3. Data

In this section, the data used for the experiments undertaken in this paper are presented.

3.1. Oil price data

Orojo *et. al* [17] undertook experiments for the oil price prediction task with the MRN using just hidden and output feedback. In this paper, the in-sample is extended to December 2005 and the out-sample was from January 2006 to February 2015 and further experiments are carried out with all combinations of the different memory bank types (i.e. input, hidden and output recurrency).

3.2. Covid-19 data

Orojo *et. al* [18] presented Covid-19 forecasting for monthly and weekly data. In this paper, daily prediction of Covid-19 of confirmed cases from 7th February 2020 to 7th July 2020 and death cases from26th Feb to 7th July 2020 were used for this experimentation. The data were divided into training and testing sets, training data accounted for 80% of the data (80% of which was to train and 20% to validate) and the remaining 20% was out-sample testing.

3.3. Business cycle data

Understanding and identifying business cycle points before they occur is important for effective economic planning. Giusto and Piger [19] used four monthly coincident series collected from February 1967 to July 2013 to 'nowcast' business cycle phases *(periods of expansion or recession)*. These four time-series are commonly known as '*the big*

four economic indicators', represent growth and are *Non-farm payroll employment*, *Industrial production index*, *Real personal income excluding transfer receipts* and *Real manufacturing and trade sales*.

The series are obtained from the Federal Reserve Economic Data website: http://fred.stlouisfed.org/. The out-sample data is between October 1976 to July 2013, a period with five complete National Bureau of Economic Research (NBER) recession phases. The data consists of continuous-valued variables only and each variable was transformed using a basic standardisation (based on the mean and standard deviation of the variable values in the training set).

Three datasets were created for the real-time classification of NBER points. The first dataset comprises the four growth variables used in [19] and described above, the second comprises the change of direction[1] (COD) of each growth variable (thus, there are 4 input variables) and the third comprises the growth variables and their change of direction (totalling 8 input variables). *(Note: the change of direction is not standardised).*

3.4. M3 competition data

The M3 competition data is a widely used data benchmark for assessing the ability of time-series forecasting models. The M3-Competition data consists of 3003 time-series which include data from various sectors (micro, industry, macro, finance, demographic and other) and different time intervals (yearly, quarterly, monthly and other).

Ten monthly series are randomly selected from the following sectors: Micro, Macro & Industry, the series length are between 126 and 144, and the last 18 observations for each series are used as out-sample observations to assess the performance. The 10 series are *N1807*, *N1908*, *N1918*, *N2012*, *N2144*, *N2150*, *N2158*, *N2159*, *N2516* and *N2521*.

The data is standardised as explained in Section 3.3 using the mean and standard deviation. The change of direction or other pre-processing techniques were not utilised for comparability with published techniques.

4. Results

In this section, the MRN along with current notable models are applied to the different tasks discussed above and the results are presented. For each of the four data sets, we perform a set of experiments to find the optimum hyper-parameters for each models we implement.

4.1. Oil price prediction

The Jordan network, SRN, MRN and LSTM are applied for the oil price prediction task. Table 1 presents the results of the models for the oil price prediction task. *Note: a lower RMSE score indicates better performance.* Similar to [17]'s result, the MRN outperforms all the other models supporting [5] and [8] claim of superior performance. This superior performance is attributed to the MRNs ability to exploit and latch onto past information more effectively which informs its prediction. This superiority becomes clearer as predictions are made further in time. The Jordan network outperformed the SRN thus suggesting that output feedbacks are essential for the oil price prediction task. Similarly, all the best MRN models presented in Table 2 are indicative of this suggestion as they all utilise output feedbacks. Additionally, these results support Tepper's [8] viewpoint that SRNs have limited processing ability which lead to rapid knowledge decay.

The results show that the MRN is able to capture the temporal dependencies in crude oil price data within the evolving oil market. Particularly, the results are indicative of the MRN's ability to be used for long and short term prediction. The enhanced performance in the MRN is consistent with the improved learning, which can be attributed to the varying degrees of memory embedded within the MRN.

[1] This is the difference in magnitude between any given variable at time, t and the same variable at time, t - 1 which informs the direction (-1: negative change, 0: no change, 1: positive change).

Table 1: RMSE comparative results of the MRN and LSTM

Model	't + 1'	't + 3'	't + 6'	't + 12'
MRN	**0.321**	0.693	**0.864**	**0.892**
LSTM	0.448	0.688	0.932	1.161
Jordan	0.33	0.694	0.947	0.920
SRN	0.332	0.7	1.061	1.409

Table 2: Memory banks *[# input, # hidden, # output]* for the best MRN model

't + 1'	't + 3'	't + 6'	't + 12'
[0, 4, 2]	[0, 4, 4]	[4, 3, 4]	[0, 0, 3]

4.2. Covid-19 forecasting

The LSTM has been widely applied for Covid-19 forecasting and will therefore act as a high-quality benchmark for the MRN [20, 21, 22, 23]. In addition, there is no universally agreed metric used by researchers to evaluate the performance of their models with respect to others and unfortunately, this precludes direct comparisons. As both the MRN and LSTM will be optimised for the Covid-19 data in this paper and the same evaluation metrics used, it will offer insight into the relative performance of each model and ascertain which model is more efficacious with respect to complexity and generalisation. The MRN is applied for the task and then compared to the LSTM. The models are assessed using the Mean Absolute Percentage Error (MAPE) *(Note: a lower MAPE indicates better performance.)*

Table 3 presents the results for the best MRN and LSTM models for Covid-19 forecasting and as seen the MRN outperformed the LSTM. Further experiments are conducted with the LSTM and as seen increasing the number of epochs and the units leads to significant improvements, nonetheless, the MRN still outperformed the LSTM.

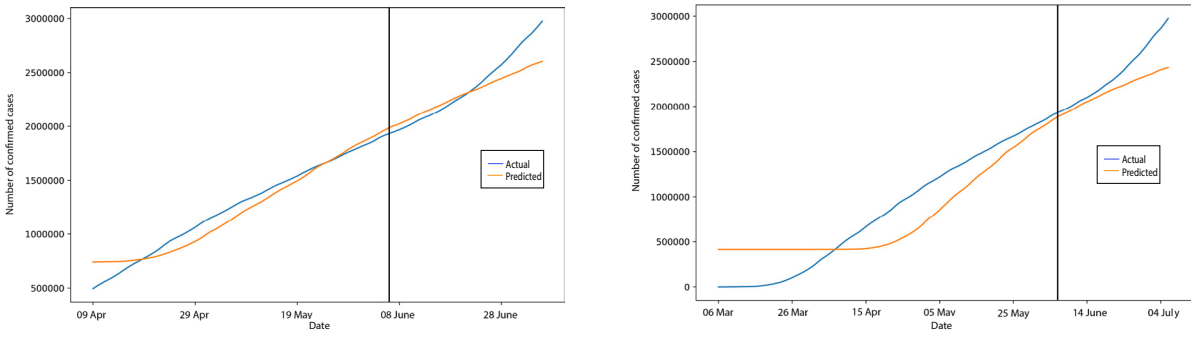
Table 3: MAPE comparison for Covid-19 forecasting of confirmed and death cases in the USA

Model *(*epochs)	Hidden units	Confirmed cases	Death cases
MRN *(*500)	20	**4.11**	**0.3**
LSTM *(*500)	20	7.03	22.67
LSTM *(*1000)	20	6.998	7.62
LSTM *(*1000)	100	5.14	1.33

The predicted values of the MRN and LSTM along with the observed values for the confirmed cases are presented in Figure 2a and Figure 2b. The MRN appears to follow the trend of the confirmed cases closely, it does however miss the sharp increase from mid-late June. The LSTM on the other hand, appears to map the trend of the signal, albeit not as well as the MRN and also misses the sharp increase in confirmed cases from mid-late June.

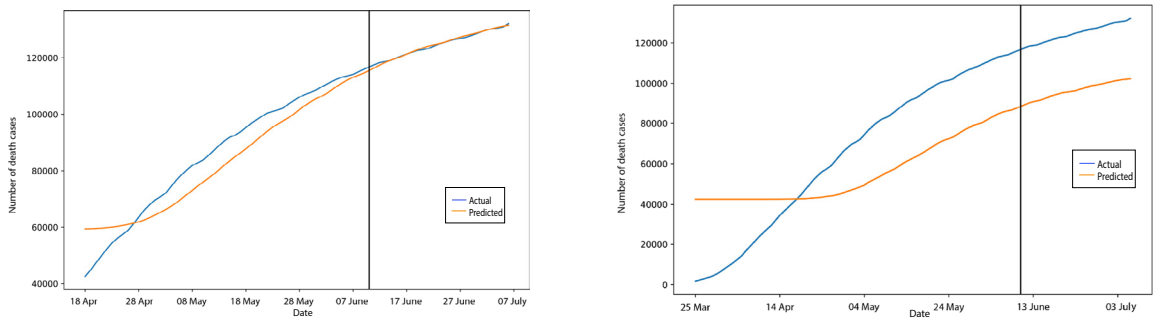
Figure 3a and Figure 3b present the predicted values of the MRN and LSTM along with the observed values for the death cases. The MRN appears to have learnt the underlying signal and predicts values for the death cases similar to those observed. While, the LSTM loosely follows the trend of the death cases although on a smaller scale.

Interestingly, the MRN, albeit a much simpler and less computationally intensive class of model, outperformed the LSTM. More specifically, the MRN requires fewer parameters than the LSTM for time-series processing of Covid-



(a) MRN (b) LSTM

Fig. 2: Confirmed cases



(a) MRN (b) LSTM

Fig. 3: Death cases

19 cases. The MRN's memory mechanism enables it to'remember' different spatio-temporal relationships within the signal and provides a descriptive overview of the information available. The results are indicative of the strong predictive abilities of the MRN for time-series forecasting. In particular, the MRN should be further explored and endowed to enhance performance.

4.3. Business cycle prediction

Giusto and Piger [19] applied the Learning Vector Quantisation model (LVQ) to identify turning points in real time, due to its computationally simple structure and its superiority compared to a mis-specified parametric Bayesian Classifier. The performance of the MRN is compared to the LVQ model, and also with the LSTM and Support Vector Machine (SVM). The Matthew Correlation Coefficient (MCC) see [19] for formula is used to assess the performance of the models. Giusto and Piger [19] nowcasted U.S. business cycle turning points (i.e. the forecast horizon was 't + 0', as the turning points occur) and this is particularly significant as the NBER's Business Cycle Dating Committee historically confirm turning points *after* they occur. In this section, the prediction task is taken a step further as predictions are made for 't + 1' steps ahead (that is a month ahead). Table 4 shows the results for all the experiments carried out and as seen the MRN performed better than the other models including the LVQ proposed by [19] for the three datasets. *Note: a higher MCC score indicates better performance.*

In particular, using the change of direction of the growth variable significantly improved the performance of the SVM with a sigmoid kernel whilst other models had relatively the same or slightly worse performance. This improvement for the SVM can be attributed to its effective handling and processing of binary variables for which it is historically acclaimed. Discretisation with SVMs encourages faster, easier and more effective identification of decision

Table 4: MCC comparison of models for the NBER turning points prediction task

Model	Optimser	MCC Growth	COD	Growth & COD
LVQ	Clustering	0.578	0.558	0.574
LSTM	Adam	0.715	0.68	0.645
	Stochastic Gradient Descent	0.605	0.63	0.655
	Sigmoid	0.13	0.635	0.627
SVM	Polynomial	0.57	0.416	0.49
	Radial Basis Function	0.396	0.395	0.276
MRN	BPTT	**0.787**	**0.79**	**0.799**

boundaries required to separate the classes. Similarly, with the third dataset (the growth variables and the change of direction), most models had a similar or slightly worse performance. Overall, the MRN provides the best results for this task. This experiment is particularly indicative that different models require different data transformations depending on the internal processing of the architecture to provide meaningful insight and harness the model's dynamics.

Table 5: Memory banks *[# input, # hidden, # output]* for the best MRN model

Growth	COD	Growth & COD
[4, 2, 0]	[2, 3, 2]	[4, 0, 3]

The MRN performed best with the third dataset, Growth & COD variables, which is indicative of the usefulness of discretisation for time-series processing to enhance performance. In particular, the ability of the MRN to employ different memory combinations, encourages appropriate selection and utilisation of historical data to effectively inform predictions. Overall, training the MRN with the three datasets provides meaningful results, highlighting its suitability.

4.4. M3 competition data

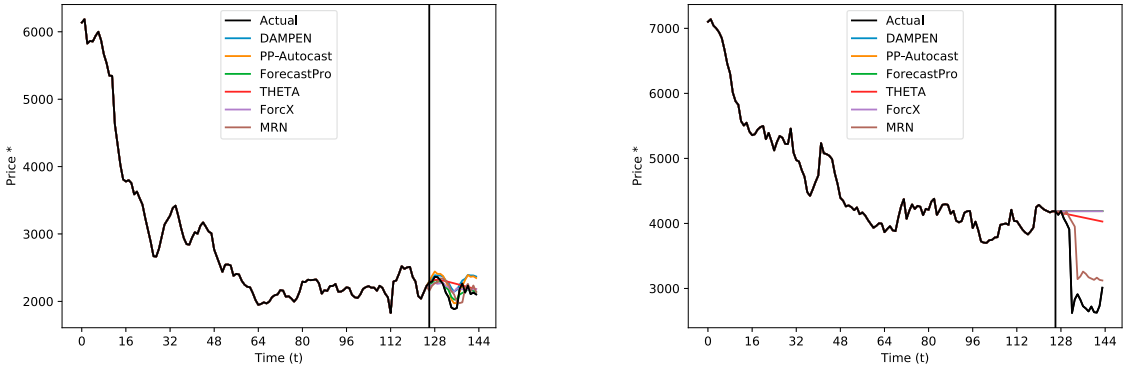
A number of models are applied to the M3 competition dataset. The forecast for the initial 24 models are available at: https://forecasters.org/resources/time-series-data/m3-competition/. The MRN is compared to the 5 best of these models. Experiments with varying memory banks, 10 hidden units and window sizes of [10, 40] are employed to obtain the best MRN. Table 6 shows the RMSEs for each model for the randomly selected series and similar to the work presented by [24], a simple average of all the RMSEs for each model is used to identify the best model and is presented in the table. The MRN performs best for 2 of the series and overall obtains the most consistent results, obtaining the overall best average RMSE.

All the models had the lowest RMSEs for series *N1918* and *N2150*, indicating the models could accurately model the signal, while the models had the highest RMSEs for series *N2144* and *N2521*, indicating the models had difficulties modelling and mapping the signal. The average RMSEs are indicative of the overall performance of the models. However, they are affected by relatively small or large values *(for example RMSE scores for series N2521)* and may hide disparities. The model predictions and observations for series *N2150* and *N2144* are visualised in Figure 4.

Series *N2150* is shown in Figure 4a, all the models appear to closely follow the series trend. Particularly, the PP-Autocast and Forecast Pro appear to make predictions that follow the signal, while the MRN appears to make predictions (including the trough) a few time-steps after it has occurred. The THETA appears to predict a straight line with a downward trend whilst the remaining models appear to follow the signal, although mapping the trough on a smaller scale. The PP-Autocast and DAMPEN appear to make significantly different predictions to the observed

Table 6: Comparative results of six models applied to the M3 competition data

	Model	MRN	Theta	Forecast Pro	ForcX	PP-Autocast	Dampen
Series	N2516	**187.1**	882.9	920.7	920.6	919.3	919.6
	N2521	2017.2	2028.5	2008.5	2018.7	2011	2011.7
	N1807	268.3	250.6	478.1	424.1	299.0	456.2
	N1908	453.9	362.1	317	313.1	333.5	373.4
	N2012	517.8	297.6	220.5	229.2	360	383
	N2159	521.9	478.5	466.5	494.4	509.3	437.8
	N2158	651.1	489.6	485.4	510	512.7	468.6
	N2150	141.6	171.7	82.2	128.4	141.2	183.5
	N2144	**538**	1091.5	1182.7	1181.7	1182.9	1182.8
	N1918	206.7	143.3	157.6	157.5	194	137.6
Average		**550.4**	619.6	631.9	637.8	646.3	655.4



(a) Series N2150

(b) Series N2144

Fig. 4: M3 Series (N1918, N2144)

towards the end of the series, overall the Forecast Pro's has the lowest score. Most models appear to have difficulty modelling series *N2144* and *N2521*. In particular, most of the models appear to predict a straight line for series *N2144* (as shown in Figure 4b and could not identify the drastic drop in price. The MRN is the only model that could map the signal and provide useful predictions.

Results in this section and work in [25] *(highlighting a key bias of Makridakis et al. [26] work and conclusion against a shift towards ML methods)*, demonstrate that ML models offered enhanced modelling abilities.

These series reflect real-world dynamics which can be volatile and chaotic. In particular, the MRN showed consistency with accurate predictions and learnt the mapping despite having the highest RMSE for some series. In particular, for two of the series, *N2144* and *N2521* where the other models had difficulties learning, the MRN was able to model and map the signal, specifically due to its state-based mechanism, which enables the modelling of different states available in the series. In conclusion, the M3 series is a notable benchmark for time-series forecasting, the results in this section along with the application of the MRN to other domains point to its suitability for time-series processing across a number of different fields and domains unlike traditional techniques or other ML techniques. *It is worth noting that where the MRN did not perform well, more data will be required to find a stable solution [25].*

5. Discussion

Over the last twenty years there has been a shift away from simple recurrent networks *(such as SRNs and Jordan networks)* to more complex recurrent networks *(such as ESNs and LSTMs)* and more recently, to much more computationally intensive feed-forward models, such as Transformers [27] where the temporal domain is mapped onto the spatial domain in a complex manner so that GPUs can be effectively applied and thus large volumes of language data processed. The shift towards ESNs and LSTMs is largely due to the vanishing gradient problem, which limits processing of time-dependence variables over a long period of time. However, due to the sequential operation of these models through time, training times on large data sets remains prohibitive as parallelism within GPU-based systems cannot be sufficiently exploited. Subsequently, over the last five years, significant interest and investment has been made in the Transformer-based solutions. The 'attention mechanism' embedded within these Transformer models radically changed the landscape in large scale Natural Language Processing (NLP) models with innovations such as *ChatGPT* showing human-like language generation capability. There is very few research work investigating the abilities of Transformers outside of NLP and Computer Vision particularly, there are very few works in the context of time-series processing [28, 29]. Murray *et al.* [30] conducted a comparison between a number of state-of-the-art models *(LSTM, Convolutional Neural Network and Transformer networks)*.

They identified that Transformers outperformed other models and LSTMs achieved comparable performance to Transformers. Similarly, Lara-Benítez *et al.* [31] evaluated Transformer for time series forecasting and compared its performance to LSTM and CNN networks (current state-of-the-art). They identified that transfomers outperformed CNNs and achieved similar results to the LSTM. Interestingly Zeng *et al.* [32] experiments demonstrated that Transformers were outperformed by simple one-layer linear models, stating they are not suitable for long-term time-series forecasting. Research work employing Transformers for numerical time-series forecasting is in the early stages and preliminary work indicates they do not appear to have the same impact as in the language domain.

We revisit this simple yet comprehensive dynamical recurrent network developed by Ulbricht [5]. Applying the MRN to four distinctive domains highlights it's ability to model temporal data. It specifically employs multiple memory banks to encourage strong information latching. It's flexible memory structure enables different suitable memory dynamics to be trained and employed for specific task. For example, as seen from Table 2, the MRN memory composition varied for the different time horizon, highlight the need for different times of information even within a specific task in a given domain. Similarly, modelling with different transformations of a given variable, the MRN is able to adapt it's memory composition to learn the underlying signal. The results in this paper particularly indicate that the wholesale shift towards more complex RNNs may be premature and simpler recurrent networks such as the MRN offer strong, if not superior, predictive performance. Such simple and effective models can support analytics as we shift towards a digitalised world. Future work will explore comparing the MRN's training time and complexity to other state-of-the-art models, extending the MRN and employing hybrid-extensions with Transformers.

6. Conclusion

In this paper, the MRN is applied to four real-world time-series forecasting tasks and its performance is comparatively assessed against widely known statistical and machine learning models. The experiments across all problem domains confirmed that the MRN is indeed a worthy competitor when compared to other state-of-the-art techniques and worthy of further exploration and optimisation.

In particular, the MRN's sluggish state-based memory mechanism appears to better capture the critical temporal dependencies than the LSTM. It appears that forming different memory types of varying information rigidity, through specifying importance weightings of layer-recurrent links and self-recurrent links, allows more informative internal representations to be formed during learning. However, a key issue is the need for the user to manually specify, and therefore to empirically establish, the optimum ratio between past and current information retained within each memory bank. This limitation together with that of a high dimensional input layer provoke the following questions: *"How can the weightings between past and current information by automatically determined by the learning process to form more useful memories?"* and *"How can the high dimensionality of the MRN's memory banks be minimised without losing predictive accuracy?"*. We have begun preliminary work to address these questions with a view to developing simple innovations that robustly answer these questions whilst maintaining effective modelling capabilities.

Acknowledgements

The authors thank Nottingham Trent University for supporting this research with the Vice Chancellor Scholarship.

References

[1] G. Natarajan, A. Ashok, Multivariate forecasting of crude oil spot prices using neural networks, ArXiv abs/1811.08963.
[2] U. Güçlü, M. A. J. van Gerven, Modeling the dynamics of human brain activity with recurrent neural networks, Front. Comput. Neurosci. 11, arXiv: 1606.03071.
[3] P. Werbos, Backpropagation through time: what it does and how to do it, Proceedings of the IEEE 78 (10) (1990) 1550–1560. doi:10.1109/5.58337.
[4] G. Dorffner, Neural networks for time series processing, Neural Network World 6 (1996) 447–468.
[5] C. Ulbricht, Multi-recurrent Networks for Traffic Forecasting, Proceedings of the National Conference on Artificial Intelligence 2 (1994) 883–888.
[6] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks, GMD Report 148, GMD - German National Research Institute for Computer Science (2001).
URL http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf
[7] C. Tallec, Y. Ollivier, Can recurrent neural networks warp time?, ArXiv abs/1804.11188.
[8] J. A. Tepper, M. S. Shertil, H. M. Powell, On the importance of sluggish state memory for learning long term dependency, Knowl. Based Syst. 96 (2016) 104–114.
[9] D. Jirak, S. Tietz, H. Ali, S. Wermter, Echo state networks and long short-term memory for continuous gesture recognition: a comparative study, Cognitive Computationdoi:10.1007/s12559-020-09754-0.
[10] T. N. Sainath, O. Vinyals, A. Senior, H. Sak, Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings 2015-August (2015) 4580–4584. doi:10.1109/ICASSP.2015.7178838.
[11] I. Danihelka, G. Wayne, B. Una, N. Kalchbrenner, A. Graves, Associative long short-term memory, 33rd International Conference on Machine Learning, ICML 2016 4 (2016) 2929–2938. arXiv:1602.03032.
[12] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures, Neural Computation 31 (7) (2019) 1235–1270. doi:10.1162/neco_a_01199.
[13] S. Jordan, Analysis and approximation of a jit production line, Decision Sciences 19 (1988) 672–681.
[14] J. L. Elman, Finding Structure in Time, Cognitive Science 14 (1990) 179—-211.
[15] J. Binner, P. Tino, J. Tepper, R. Anderson, B. Jones, G. Kendall, Does money matter in inflation forecasting?, Physica A 389 (21) (2010) 4793–4808.
[16] M. S. Shertil, On the Induction of Temporal Structure by Recurrent Neural Networks, Ph.D. thesis, Nottingham Trent University (2014).
[17] O. Orojo, J. Tepper, T. M. McGinnity, M. Mahmud, A Multi-recurrent Network for Crude Oil Price Prediction, in: Proc. IEEE SSCI, IEEE, 2019, pp. 2953–2958.
[18] Orojo, J. Tepper, T. M. McGinnity, M. Mahmud, Sluggish state-based neural networks provide state-of-the-art forecasts of covid-19 cases, in: Analogical and Inductive Inference, 2021.
[19] A. Giusto, J. Piger, Nowcasting U.S. Business Cycle Turning Points with Vector Quantization, Working papers, Dalhousie University, Department of Economics (Sep. 2013).
URL https://ideas.repec.org/p/dal/wpaper/daleconwp2013-02.html
[20] V. K. R. Chimmula, L. Zhang, Time series forecasting of covid-19 transmission in canada using lstm networks, Chaos, Solitons, and Fractals 135 (2020) 109864 – 109864.
[21] A. Tomar, N. Gupta, Prediction for the spread of covid-19 in india and effectiveness of preventive measures, The Science of the Total Environment 728 (2020) 138762 – 138762.
[22] S. M. Ayyoubzadeh, S. M. Ayyoubzadeh, H. Zahedi, M. Ahmadi, S. R Niakan Kalhori, Predicting covid-19 incidence through analysis of google trends data in iran: Data mining and deep learning pilot study, JMIR Public Health Surveill 6 (2) (2020) e18828. doi:10.2196/18828.
[23] A. Barman, Time series analysis and forecasting of covid-19 cases using lstm and arima models (2020). arXiv:2006.13852.
[24] Y. Li, R. Gault, T. M. McGinnity, Probabilistic, recurrent, fuzzy neural network for processing noisy time-series data, IEEE Transactions on Neural Networks and Learning Systems (2021) 1–10doi:10.1109/TNNLS.2021.3061432.
[25] V. Cerqueira, L. Torgo, C. Soares, Machine learning vs statistical methods for time series forecasting: Size matters (2019). arXiv:1909.13316.
[26] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and Machine Learning forecasting methods: Concerns and ways forward, PLoS ONE 13 (3) (2018) e0194889. doi:10.1371/journal.pone.0194889.
URL https://dx.plos.org/10.1371/journal.pone.0194889
[27] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, ArXiv abs/1706.03762.
[28] S. Chaudhari, G. Polatkan, R. Ramanath, V. Mithal, An attentive survey of attention models, ACM Transactions on Intelligent Systems and Technology (TIST) 12 (2019) 1 – 32.
[29] A. de Santana Correia, E. L. Colombini, Attention, please! a survey of neural attention models in deep learning (2021). arXiv:2103.16775.

[30] C. Murray, P. Chaurasia, L. Hollywood, D. Coyle, A comparative analysis of state-of-the-art-time series forecasting algorithms, in: International Conference on Computational Science and Computational Intelligence, IEEE, United States, 2022.
[31] P. Lara-Benítez, L. Gallego-Ledesma, M. Carranza-García, J. M. Luna-Romera, Evaluation of the transformer architecture for univariate time series forecasting, in: Conferencia de la Asociación Española para la Inteligencia Artificial, 2021.
[32] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting? (2022). arXiv:2205.13504.