Using Deep Learning Techniques to Predict Network Traffic on the South African Research and Education Network

Antony Fleischer
flsant005@myuct.ac.za
University of Cape Town
Cape Town, Western Cape, South Africa

ABSTRACT

This paper critically evaluates the use of artificial neural networks as network traffic predictors. Network traffic prediction is crucial for the maintenance and optimization of large, federated networks. Without the ability to forecast traffic trends, networks face congestion and resource mismanagement. These are some of the issues faced by the South African Research and Education Network. To solve these common problems, researchers have begun to use artificial neural networks to make predictions with time-series data. Neural networks are more accurate and efficient than statistical methods, and can recognize long-dependencies and unusual temporal patterns. Various models have been considered for the network traffic prediction problem, including the Autoregressive Integrated Moving Average, Recurrent Neural Networks and Long Short-Term Memory (LSTM) models. The literature shows that the LSTM architecture is the best performing neural network, and is the ideal candidate for further research using data provided by the South African Research and Education Network. Overall, it is concluded that additional research on the Long Short-Term Memory model is needed in two facets. Firstly, it is necessary to evaluate whether the LSTM architecture can be fundamentally changed to produce improved performance. Secondly, the LSTM must be assessed for its appropriateness on the SANReN, and so the computational requirements and efficiency of the LSTM will also be evaluated.

CCS CONCEPTS

• **Networks** → **Network performance modeling**; Network performance analysis; • **Computing methodologies** → *Neural networks.*

KEYWORDS

Network traffic prediction, time-series, Recurrent Neural Network, LSTM, supervised learning

ACM Reference Format:
Antony Fleischer. 2021. Using Deep Learning Techniques to Predict Network Traffic on the South African Research and Education Network. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As the internet evolves into Web 3.0, the amount of data produced by the global Internet of Things (IoT) is increasing. Existing computer networks and nodes are facing the challenges associated with transferring unprecedented amounts of data. One of these challenges is that large computer networks exhibit fluctuating network traffic volume [26]. Networks can attempt to implement traditional network monitoring or data analysis techniques to preemptively optimize the network for varying traffic volumes. However, such techniques have shown to be insufficient for precise, real-time processing of time-series data.

An accurate and efficient network traffic prediction model would be beneficial to network operators and users for two reasons. Firstly, the network would be able to predict when segments of the network will experience high traffic volumes and congestion; and secondly, the network would be able to optimize the allocation of its resources based on forecasted network traffic metrics [28]. Additionally, novel network analysis techniques may capture unknown trends in time-series datasets and their associated dependencies.

This research paper will evaluate the state-of-the-art of network traffic prediction by assessing the existing research on artificial neural network (ANN) models for network traffic prediction - particularly the Multi-layer Perceptron (MLP), Long Short-Term Memory (LSTM) and Stacked Auto-encoder (SAE) models– against traditional statistical models used for prediction – namely the Autoregressive Integrated Moving Average (ARIMA), Support Vector Machine (SVM) and Holt-Winters Forecasting models.

The supplementary objective of this paper is to critically evaluate an ANN that best supports network traffic prediction on the South African National Research Network (SANReN). This may be a previously mentioned ANNs or a new, evolved ANN that does not have as much research support, such as a Dual-Stage Attention-Based RNN. By providing network traffic prediction capabilities to the network, the network would be able to manage network congestion and allocate network resources effectually [5].

Furthermore, this research paper evaluates the effects of pre-processing data on a network traffic predictor's accuracy, as well as the trade-off between computational complexity and accuracy in various models, to realise an ANN model that is useful and relevant for the SANReN use-case.

2 BACKGROUND

2.1 Network Traffic Flows

Network traffic flows refer to the volume of bytes being sent – from any node to any other node - across a network [22]. Nodes in this context are any type of device that is connected to the network, not just routers [22]. As done by Ramakrishnan et al. [22], this paper will investigate the ability of neural network models to predict the number of bytes that will be sent in a future time period, as well as the number of packets sent during a time period. Time series data is a sequence of observed data points - $(y_1, y_2, \ldots, y_t)$ - indexed by time order [29]. Usually, the data points are measured at successive, equidistant points in time. It is important for this research paper that network traffic flows are defined as time-series data. Additionally, it is important to identify whether network traffic data is linear or not. Seasonality and trends are the main reasons that linear predictions models are inaccurate [26, 29], and so it is necessary to use ANNs to capture the non-linearity that network traffic exhibits [26]. Both neural networks and traditional statistical models can use time-series data in different ways. For example, these predictive models can implement different timescales and distinctive forecasting horizons [5]. Predictions for hourly traffic volumes rather than every 5 minutes is an instance of an altered timescale, whereas an adjusted forecasting horizon would change how many time periods ahead a predictor forecasts for [5]. Additionally, categorical variables such as timestamps and weekday may be encoded to numerical values and provide to a neural network to improve its accuracy. Various encoding methods – namely label and one-hot encoding - have been implemented to improve a neural network's predictive power [24]. The effect of these two encoding methods will be discussed later. Network traffic data may be provided to a neural network to predict when a burst in traffic volume will occur, rather than to forecast a pattern in the future of the time-series data [17]. Burst flow prediction can be done in parallel with regular neural network prediction [17], and so it is included in this research paper.

2.2 Project Context

There are a few motivations as to why network traffic prediction requires neural network predictors. Primarily, any prediction model depends on the statistical distribution of a data set [22]. It also depends on the notion that the data is from a time-series dataset [22]. We have already defined network traffic flows as time-series datasets, so the difficulty in network traffic prediction must stem from the statistical nature of the data. Network traffic flows are characterized by self-similarity and non-linearity [15] and are inefficiently modelled by Gaussian or Poisson distribution models [15]. This inefficiency introduces the need for ANNs and is reinforced later when evaluating the performance of traditional statistical models on network traffic data.

Whereas previous researchers have used publicly available datasets such as the GEANT [10, 14, 27] and Abilene [22, 25] networks, this paper uses network traffic data from the South African National Research and Education Network (SANReN). SANReN is a country-wide network of education and research institutions in South Africa. As a large, federated network, SANReN is unable to use traditional network monitoring and analysis techniques and would benefit from an ANN model that allows for preemptive network actions. The researchers would like to thank SANReN for their contribution to this paper.

2.3 Deep Learning for Network Traffic Prediction

2.3.1 ANNs are Deep Learning. Deep learning refers to the implementation of artificial neural networks (ANNs) with complex layers of internal nodes [4]. Additionally, deep learning is not distinct from machine learning, but rather a subset of machine learning [3]. Although the terms 'deep learning model' and 'ANN' are synonymous, this paper will use ANN throughout to refer to the parent class of models that this paper evaluates. Essentially, an ANN is an information processing model that is modelled according to the biological nervous system of the human brain [3]. Just as the brain is made up of neurons interconnected by synapses, an ANN is made up of elements called 'neurons' interconnected by weighted paths [2, 3]. The various ANNs will be elaborated on further in this paper.

2.3.2 ANNs for Prediction. As ANNs have evolved, they have provided new ways to deal with difficult problems regarding prediction and pattern recognition [3]. One such problem is network traffic volume prediction. As mentioned earlier, network traffic data is not comparable to other pattern recognition and prediction problems due to its non-linearity and stochasticity [17]. However, ANNs can accurately model almost any non-linear relationship [2] – including wind speed [16] and MPEG-4 video traffic [2], and so it is sensical to evaluate a variety of ANNs on network traffic data.

3 NETWORK TRAFFIC PREDICTION

3.1 Pre-processing

Before an ANN receives data as input, there is an opportunity for the researcher to pre-process the data. Without pre-processing, the ANN would receive raw data – which is data in its source form [1]. Pre-processing is made up of two steps: data engineering and feature engineering. By aggregating, summarizing, and filtering data, data engineering is the process of converting data from the source into prepared data [1]. Prepared data can then be appended with other ANN-specific features, such as encoding or scaling to increase the accuracy of an ANN [1, 19].

3.1.1 Discrete Wavelet Transformation. Furthermore, certain data transformations can be applied to non-linear data sets such as network traffic data. The Discrete Wavelet Transformation (DWT) is an example of one such transformation. DWT iteratively decomposes a given non-linear pattern into distinct non-linear approximations and linear details - which can then be to train non-linear and linear models respectively [18], [19]. DWTs also support reconstruction, which combines the predictions of two models to create a final forecast [19]. Existing research [18, 19] concludes that an ANN model with DWT has a better predictive ability than the same ANN model without DWT.

3.1.2 Model Section. Considering that traditional statistical models – such as ARIMA, Holt-Winters and SVM – are less suited in predicting non-linear data [10], ANNs are used to predict data that takes this form. However, Katris et al. [13] propose that a test for non-linearity should be the foundation of the model selection process. This approach would be useful if this paper's research applied to a broader audience or more use-cases, as models could then be selected and trained based on the scale and behaviour of independent networks. However, this is not the case, and this paper will not include model selection as a task to be performed by an ANN.

3.2 Traditional Statistical Methods

Traditional methods refer to statistical methods that have been used for prediction and pattern recognition before ANNs. Despite the differences in how they function, they are united by their inability to effectively forecast non-linear data. The most used models for linear prediction are the Holt-Winters forecasting model and ARIMA [5], whilst SVMs have also been used in a broader machine learning context [10].

3.2.1 ARIMA. ARIMA is a traditional statistical model used for predictions with time-series data. It fundamentally assumes that future data observations can be predicted using the values and errors of past data points [19].

The ARIMA is extremely efficient and accurate when modelling non-linear time-series [11, 19, 22], however, its shortcoming is that it cannot accurately model time-series data that does not show stationarity [19]. A time series has stationarity if, as time passes, the basic properties of the distribution – namely the mean, variance, and covariance – are constant [29]. Additionally, the absence of stationarity results in non-linearity in the mean [29]. Network traffic data is non-linear, and as a result, is not efficiently modelled by linear time series models such as ARIMA [19].

ANNs' elevated performance on network traffic data is supported extensively by research [11, 14, 22], with simple LSTMs having a better Root Mean Square Error than an ARIMA model by up to 69% in some instances [14]. However, ARIMAs provide similar accuracy to RNNs for some datasets – such as the GEANT dataset [22]. This introduces the trade-off between prediction rate and accuracy, and this trade-off will be assessed in this paper.

3.3 Artificial Neural Networks

Multiple forms of ANN have been used to predict network traffic volumes. Some studies [11, 21, 27] use feed-forward neural networks (FFNN), whereas others [10–12, 22] use neural networks called Recurrent Neural Networks (RNN). The difference between these two types of ANNs is their memory state. Feed-forward neural networks such as MLP don't have an internal memory state, and information is passed from internal layers in the direction of output only [5]. RNNs have cycles between internal layers, which creates a form of artificial memory and facilitates the learning of long-range temporal dependencies and other sequential behaviours that network traffic data may exhibit [20, 22]. However, there are some common features in all ANNs. For example, each neuron in a neural network is characterized by having multiple inputs, but only a single output [2]. Additionally, implementors of ANNs must consider the time it takes to train a machine learning model. This is the longest step of the ANN process, but once an ANN is trained, it can provide real-time results with less error than traditional time-series approaches [5].

3.3.1 Training Algorithms. ANNs learn from training algorithms [7]. The algorithm trains the ANN to identify and separate data in a set. Usually, the training set is a subset of the entire data set provided to the ANN [7]. Once the ANN has trained, it would be ready to perform pattern recognition and prediction. In the context of this paper, ANNs train using SANReN data and will predict the future patterns of the network traffic data. According to Cortez et al. [5], complex ANNs are comparable to human brains when solving problems of this nature. Training ANNs can take two forms: supervised, and unsupervised. Supervised learning is done by providing the ANN with a labelled dataset. This could be done by using a pre-labelled dataset, or by data engineering raw data – as described earlier [1]. Unsupervised learning uses ANNs to evaluate and group unlabelled data sets. Essentially, an ANN working with unlabelled data will try to define the intrinsic structure of the data. Due to these differences, some researchers have unsupervised ANNs on unlabelled data and provided the output to a supervised ANN for prediction and pattern recognition [23]. Another important consideration when training an ANN is the training-test split. The training data is used by an ANN to train itself to recognize patterns and predict, whereas the test data is used to measure the accuracy of its predictions [6]. In existing research, there are a few conventions for splitting the data. The size of the training subset has been 50% [2], 67% [22], 75% and 90% [10]. The training subset is not the minority of the network traffic data in existing studies, and this paper will maintain that.

3.3.2 Multi-Layer Perceptron. As mentioned earlier, a Multi-Layer Perceptron is a type of feed-forward neural network. As shown in Figure 1, an MLP has an input layer, multiple hidden layers, and a single output layer [20]. Every neuron in layer l_n is connected to every neuron in layer l_{n+1} [20]. However, a node in l_n is not connected to any other nodes in the same layer. The qualifier 'feed-forward arises from the fact that data is only passed in the direction of output – unlike other ANNs [20]. An MLP is trained using a supervised learning algorithm called backpropagation. The principle of backpropagation is to let the MLP adjust the weightings of its hidden layers' nodes so that its prediction matches that of an expected result [20]. By iteratively minimizing the difference between its prediction and the expected output, the MLP trains itself [20]. Additionally, the output of the MLP does not depend on the number of layers, as the accuracy of the MLP is affected by the total number of neurons in the hidden layers, rather than the number of the layers that are hidden [9]. MLP is usually compared to other ANNs to assess its performance. When using Mean Squared Error as an indicator of accuracy against RNN, LSTM, and GRU, MLP was the worst-performing ANN by 35% [27]. Conversely, when assessed against an SAE over various time periods, the MLP provided more accurate predictions in all periods but was more accurate by a larger amount as the time period approached 1 hour [20]. However, there is no research available containing a direct comparison between MLP and traditional statistical models such as ARIMA. Presumably, the better-performing, more complex ANNs are used more often to show the predictive value of the ANN class.

3.3.3 Recurrent Neural Networks. A Recurrent Neural Network is a type of ANN that can have one or more connection between neurons to form a cycle [20]. These cycles allow neurons to store information and pass feedback in the opposite direction to the flow of inputted data [20]. FFNN do not have this ability, and so this cyclical passage of information is how RNNs can create a form of internal memory

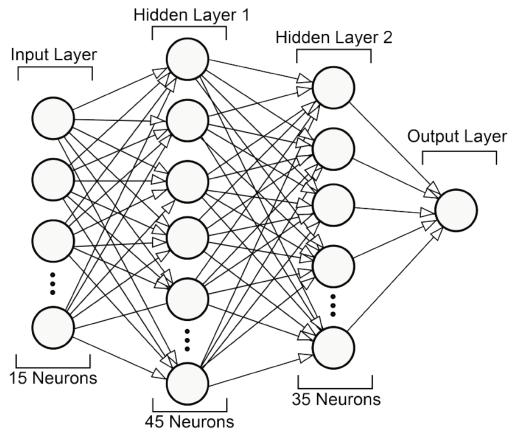

Figure 1: The feed-forward structure the MLP model [20].

The MLP model shows that the input, output and hidden layers are made up of multiple neurons, each only connected to neurons in the layer in front of itself.

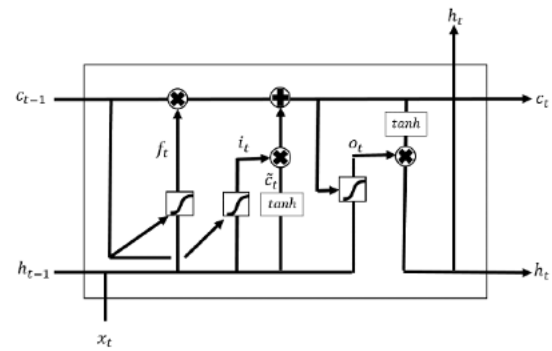

Figure 2: The memory unit of an LSTM [22].

An LSTM contains multiple memory units at each layer, with the forget, update and output gates determining the state of each memory unit. Each gate feeds information to a tensor product operation to compute the unit's current state and activation function.

[7]. These information cycles do not only occur within the hidden layer but could initiate at an output node and end at the input layer or be self-reflective [7]. This unique architecture trait allows an RNN to determine the dependencies in the data. Consequently, the increased accuracy of RNNs – and its derivatives: Long Short-Term Memory and Gated Recurrent Unit – stems from their ability to model more complex non-linear behaviour than FFNNs and traditional statistical models [22]. However, one shortcoming of RNNs is the gradient disappearance problem [16]. As the RNN is trained, and the nodes iteratively update one another, the gradient – the applied effect – of each update becomes smaller. Essentially, observations at the start of the network traffic data would not be considered or 'remembered' as the RNN processes more data [16]. Krishnaswamy et al. [14] argue that this makes RNNs unsuitable for network traffic prediction because it is unable to completely consider long-term dependencies. This has not been suggested by other research, as RNNs are still more accurate than ARIMA and Holt-Winter models on the same datasets [22]. However, by comparing an RNN with a more memory-dependent variation, such as LSTM, the effect of the gradient disappearance problem will be evident. When using MSE as a measure of error, traditional RNNs are shown to be more accurate than ARIMA models when forecasting 5 minutes ahead [19]. However, on small datasets with less than 100 data points, ARIMA's MSE is comparable to RNN [19]. Despite being more accurate than traditional statistical models at most forecast periods, RNN prediction error can increase for long term predictions, going from 3.5% when predicting one hour ahead to 12.23% for a 24-hour lookahead [5].

In general, the mentioned RNNs - traditional RNN, LSTM and GRU – demonstrate high accuracy when used on large datasets [22]. In terms of model construction, all three variations can be implemented using Python [22], which is useful for integration with Jupyter Notebook, sci-kit learn, and TensorFlow.

3.3.4 Long Short-Term Memory. The LSTM is one of the RNN models that has been designed to solve the gradient disappearance problem. The LSTM improves on the RNN by adding a memory unit [22]. The structure of each memory unit includes three gates, the forget gate f_t, the update gate i_t and the output gate o_t as shown in Figure 2 [11]. The update gate determines to what extent the unit updates itself, the forget gate determines to what extent existing memory is removed, and the output gate determines how much memory is exposed to the next step. The unit's current state, c_t, is determined as a mathematical function of the previous activation function h_{t-1}, the previous cell state c_{t-1}, the input x_t, and each of the gates [22].

These gates give the memory unit the ability to consider its input, and then either keep existing memory or overwrite it with new information. RNNs do not have this ability and will overwrite memory by default [22]. This singular difference allows LSTM to capture long-term dependencies that an RNN may not [22]. Using MSE as a measure of accuracy and error, the LSTM is more accurate than RNN and ARIMA on the Abilene and GEANT datasets [22]. Crucially, LSTM is never less accurate than memory-less RNN and traditional statistical methods and is more accurate in multiple studies [11, 12, 22, 27]. In one study, despite RNN being 35% more accurate than an FFNN, the implemented LSTM outperformed the RNN by 37% [27]. The only occasion when LSTMs are comparable to RNNs is when predicting packet data flow, rather than the flow of bytes [22].

Additionally, LSTM has various improvements that are not extensively researched. Models such as Dual-Staged Attention-Based LSTM [21] and Parallel LSTM [17] have led to accuracy improvements over traditional LSTM and could be an area of further research.

3.3.5 Gated Recurrent Unit. The Gated Recurrent Unit is another variant of RNN. It is a very similar design to LRU, with gated units that can determine the exposure of information within the unit [22]. As in LSTM, the GRU can learn long-term dependencies because of its update gate [25]. If the GRU has found a significant feature, then the update gate will maintain it rather than overwrite it [25]. This is in converse to RNN, where the neuron will always replace its content [25]. The GRU is not as extensively investigated as LSTM models, possibly due to its performance as a predictor. As with all the ANNs evaluated thus far, the GRU is more accurate than traditional statistical prediction techniques [22, 27]. However, when compared to LSTM, the GRU is significantly less accurate [27]. For example, when implemented on the public GEANT dataset, GRU exhibited an MSE of 0.051, compared to an LSTM MSE of 0.042 [27]. This is a difference in error of 27% when using the LSTM's score as the basis.

3.3.6 *Stacked Auto-Encoder.*

An autoencoder is a special type of unsupervised FFNN that is trained to copy its input to its output [6]. However, it is designed to be unable to copy perfectly, so that it must prioritize which aspects of the data to copy, and therefore learn the characteristics of the data [6]. A stacked autoencoder (SAE) is an ANN that uses multiple autoencoders, where the output of each autoencoder is linked to the input of the next [20], [6]. SAEs use a greedy layer-wise approach to learning, where each layer is trained independently [6]. SAEs will not be researched in this paper as they are ineffective on time-series data [20] and are consistently outperformed by other ANN models. As an ANN with an unsupervised training algorithm, the SAE takes extremely long to train relative to MLP and RNN. When using a 5-minute network traffic dataset, an RNN took 33981 milliseconds to train, whilst an SAE took over 6 million milliseconds – a 19000% increase [20]. This combination of relative inaccuracy and long training times makes SAE an unsuitable candidate for the SANReN use case.

4 DISCUSSION

All the ANNs mentioned above have shown, to varying degrees, that they can predict network traffic volumes accurately. However, this paper aims to find the best candidate ANN for network traffic prediction. Therefore, each of the ANNs mentioned must be pairwise assessed to choose a model that is accurate, efficient and has the theoretical capacity to be improved. Efficiency and computational complexity can also be considered attributes by which to assess ANNs as training times and prediction rates are a considerable part of the ANN research process. SAEs are a perfect example of this. They perform better than traditional statistical models but require an unsupervised pre-training period [20]. Consequently, they take relatively long to train and offer no increase in prediction accuracy when compared to others [20]. Therefore, the SAE will not be considered in this paper.

The traditional RNN seems to be the baseline for most research [10, 16, 22], [11], [20], [5], however, it also has its known shortcomings – such as the gradient disappearance problem [16]. The RNN model is also more prone to overfitting than its child models – the LSTM and GRU [22]. Therefore, the LSTM and GRU architectures should be considered suitable models for network volume prediction. However, the GRU does not outperform the LSTM in the few studies it appears in [25, 27], and will also not be considered in this paper.

It is also worth mentioning that the best performing RNN models only outperform traditional statistical models by 11% in smaller, more linear datasets such as GEANT [22]. However, when non-linearity and burst patterns are present, RNNs are up to 78% better than traditional models [22]. Therefore, this paper will aim to use a large sample of non-linear SANReN data with burst flow activity. Another consideration is how long the ANN is allowed to train for. When limited to 20 epochs – complete training cycles - MLPs suffer from severe performance issues. For instance, when compared to an ARIMA on 892 data points, limited MLPs performed worse [11]. This is because the MLP did not fully train the network in the restricted time. Conversely, an LSTM model limited in the same way maintained its performance [11].

Vinayakumar et al. [27] propose that the learning rate is directly correlated with the speed at which the model converges on the pre-determined error range. This suggests that the LSTM is a faster learner than the MLP, more efficient, and more accurate, and so the LSTM is still the prime candidate for further ANN research for network traffic prediction. Whilst considerations of the learning rate, data size, and training time are important, the primary measure of an ANN's performance is its accuracy. Across all the examined research on ANNs, LSTM is the best predictor [8, 11, 12, 22, 27]. Considering the LSTM's pertinence as an accurate network traffic predictor, this paper will explore evolutions of the LSTM. Two examples of LSTM improvements are the Dual-Staged Attention-Based LSTM model [21] and the Parallel LSTM [17]. The Dual-Stage Attention-Based LSTM can identify the relevant pattern in the data. Essentially, it can decide whether the current data point under consideration is influenced by the long-range temporal pattern or by the short-term burst flow [21]. This adapted LSTM was shown to outperform traditional LSTM models when trained and tested on a NASDAQ dataset [21]. Similarly, the parallel-LSTM can predict burst flows with greater accuracy, and so its general prediction accuracy is improved relative to traditional LSTMs [17]. Lastly, it is important to clarify that, according to an analysis of existing research, the LSTM model is this paper's best option for further evaluation. MLP, GRU and SAE may outperform the LSTM in other deep learning contexts, but that is beyond the scope of this paper.

5 CONCLUSIONS

Throughout existing research, it has become evident that all three of the main RNN architectures are low-error predictors for network traffic volumes. Across various datasets, RNN, LSTM and GRU better statical forecasting models [22]. Traditional statistical models have fallen behind in terms of prediction accuracy and efficiency [10], and so they will not be evaluated or used for comparisons in this paper. Similarly, there is sufficient research on ANNs to identify that traditional SAEs, RNNs and MLPs no longer compete with their respective variations. The feasibility of the LSTM model for non-linear time-series data has also been demonstrated, which makes an LSTM derivative the ideal focal point of this paper. Furthermore, considering the strength of the LSTM model derivatives when handling large samples and burst traffic, this paper will further research LSTM variations to use as predictors for network traffic data volumes. Lastly, it must be considered if the LSTM is appropriate for the SANReN use case, and so the computational requirements and efficiency of the LSTM derivative models will also be investigated.

REFERENCES

[1] 2021. https://cloud.google.com/architecture/data-preprocessing-for-ml-with-tf-transform-pt1

[2] Adel Abdennour. 2006. Evaluation of neural network architectures for MPEG-4 video traffic prediction. *IEEE transactions on broadcasting* 52, 2 (2006), 184–192.

[3] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, 11 (2018), e00938.

[4] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*. Ieee, 1–6.

[5] Paulo Cortez, Miguel Rio, Miguel Rocha, and Pedro Sousa. 2006. Internet traffic forecasting using neural networks. In *The 2006 IEEE international joint conference on neural network proceedings*. IEEE, 2635–2642.

[6] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[7] Simon Haykin and N Network. 2004. A comprehensive foundation. *Neural networks* 2, 2004 (2004), 41.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[10] Yuxiu Hua, Zhifeng Zhao, Rongpeng Li, Xianfu Chen, Zhiming Liu, and Honggang Zhang. 2019. Deep learning with long short-term memory for time series prediction. *IEEE Communications Magazine* 57, 6 (2019), 114–119.

[11] Shan Jaffry. 2020. Cellular traffic prediction with recurrent neural network. *arXiv preprint arXiv:2003.02807* (2020).

[12] Nan Jiang, Yansha Deng, Osvaldo Simeone, and Arumugam Nallanathan. 2019. Online supervised learning for traffic load prediction in framed-ALOHA networks. *IEEE Communications Letters* 23, 10 (2019), 1778–1782.

[13] Christos Katris and Sophia Daskalaki. 2014. Prediction of internet traffic using time series and neural networks. In *Proceedings of international work-conference on time series analysis (ITISE 2014)*, Vol. 1. 594–605.

[14] Nandini Krishnaswamy, Mariam Kiran, Kunal Singh, and Bashir Mohammed. 2020. Data-driven Learning to Predict WAN Network Traffic. In *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*. 11–18.

[15] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on networking* 2, 1 (1994), 1–15.

[16] Sisheng Liang, Long Nguyen, and Fang Jin. 2018. A multi-variable stacked long-short term memory network for wind speed forecasting. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 4561–4564.

[17] Huang Lin, Wang Diangang, Liu Xiao, Zhuo Yongning, and Zeng Yong. 2020. A Predictor Based on Parallel LSTM for Burst Network Traffic Flow. In *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*. 476–480.

[18] Haipeng Lu and Fan Yang. 2018. A network traffic prediction model based on wavelet transformation and LSTM network. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 1–4.

[19] Rishabh Madan and Partha Sarathi Mangipudi. 2018. Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*. IEEE, 1–5.

[20] Tiago Prado Oliveira, Jamil Salem Barbar, and Alexsandro Santos Soares. 2016. Computer network traffic prediction: a comparison between traditional and deep learning neural networks. *International Journal of Big Data Intelligence* 3, 1 (2016), 28–37.

[21] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971* (2017).

[22] Nipun Ramakrishnan and Tarun Soni. 2018. Network traffic prediction using recurrent neural networks. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 187–193.

[23] Alaa Sagheer and Mostafa Kotb. 2019. Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems. *Scientific reports* 9, 1 (2019), 1–16.

[24] Yusuke Tokuyama, Ryo Miki, Yukinobu Fukushima, Yuya Tarutani, and Tokumi Yokohira. 2020. Performance Evaluation of Feature Encoding Methods in Network Traffic Prediction Using Recurrent Neural Networks. In *Proceedings of the 2020 8th International Conference on Information and Education Technology*. 279–283.

[25] Sebastian Troia, Rodolfo Alvizu, Youduo Zhou, Guido Maier, and Achille Pattavina. 2018. Deep learning-based traffic prediction for network optimization. In *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 1–4.

[26] Sebastian Troia, Gao Sheng, Rodolfo Alvizu, Guido Alberto Maier, and Achille Pattavina. 2017. Identification of tidal-traffic patterns in metro-area mobile networks via matrix factorization based model. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 297–301.

[27] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. 2017. Applying deep learning approaches for network traffic prediction. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2353–2358.

[28] Weitao Wang, Yuebin Bai, Chao Yu, Yuhao Gu, Peng Feng, Xiaojing Wang, and Rui Wang. 2018. A network traffic flow prediction with deep learning approach for large-scale metropolitan area network. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–9.

[29] Steven Wheelwright, Spyros Makridakis, and Rob J Hyndman. 1998. *Forecasting: methods and applications*. John Wiley & Sons.