Om *et al. Hum. Cent. Comput. Inf. Sci.* (2020) 10:39
https://doi.org/10.1186/s13673-020-00242-w

Human-centric Computing and Information Sciences

RESEARCH Open Access

Modelling email traffic workloads with RNN and LSTM models

Khandu Om[1], Spyros Boukoros[2], Anupiya Nugaliyadde[1], Tanya McGill[1], Michael Dixon[1], Polychronis Koutsakis[1*] and Kok Wai Wong[1]

*Correspondence: p.koutsakis@murdoch.edu.au
[1] Discipline of Information Technology, Mathematics & Statistics, Murdoch University, Science and Computing Building 245, SC1.012, 90 South Street, Murdoch, WA 6150, Australia
Full list of author information is available at the end of the article

Abstract

Analysis of time series data has been a challenging research subject for decades. Email traffic has recently been modelled as a time series function using a Recurrent Neural Network (RNN) and RNNs were shown to provide higher prediction accuracy than previous probabilistic models from the literature. Given the exponential rise of email workloads which need to be handled by email servers, in this paper we first present and discuss the literature on modelling email traffic. We then explain the advantages and limitations of different approaches as well as their points of agreement and disagreement. Finally, we present a comprehensive comparison between the performance of RNN and Long Short Term Memory (LSTM) models. Our experimental results demonstrate that both approaches can achieve high accuracy over four large datasets acquired from different universities' servers, outperforming existing work, and show that the use of LSTM and RNN is very promising for modelling email traffic.

Keywords: Email traffic, Recurrent neural networks, Long short term memory model

Introduction

In 2019, the total number of business and consumer emails sent and received per day was expected to exceed 293 billion and this is forecasted to grow to over 347 billion by the end of 2023 [1]. The two main causes of email traffic overload are the volume of email traffic and the lack of effective traffic management [2]. Problems associated with the growing volume of email traffic and especially spam emails include server performance, user quality of service, financial losses and productivity decrease.

Network and server performance are highly dependent on how the systems react to load fluctuations. Therefore, accurate modelling of email workload, which is generated by interactions between email clients and email servers, can help in predicting and improving network performance. Accurate email traffic prediction can help network/cloud administrators take actions to optimize the way they allocate the storage space and the bandwidth that they have at their disposal.

There have been significant developments in the characterization of network traffic workload using probability distribution models. However, the majority of the workload models pertaining to email traffic focus on the marginal distribution and first

order statistics properties, while correlation and time-varying properties have received less attention [3, 4]. A possible reason could be the limitations of probability distribution models in capturing time-sequential data. The unique characteristics of time series data makes it different from other data types. The sampled time series data often contain noise and have high dimensionality properties, which make them challenging to analyse and model with conventional probabilistic techniques. In order to overcome the limitations of probabilistic models, email traffic was modelled using RNNs in [5], treating email traffic workload as a time series problem. The prediction accuracy was found to be substantially higher than that of the probabilistic modelling approach in [6].

Despite the significant developments in the understanding of the statistical properties of network traffic, email workload modelling in particular has only been addressed to a limited extent. A possible reason for this lack of a significant number of relevant works is the difficulty in obtaining large datasets. This problem does not exist in our work, as we have collected a large amount of email data from four different universities.

The main objectives of this work are to: (a) present and discuss the relevant literature on email traffic modelling, (b) develop a comprehensive structured approach to maximize the generalization ability of a RNN model to avoid the vanishing gradient problem when using backpropagation. The LSTM model [7] and a different implementation of the RNN model compared to the one used in [5] are investigated in our work to examine if they can provide efficient modelling for all types of emails in our datasets.

The remainder of the paper is organized as follows. "Related work" section presents and discusses existing work on modelling email traffic. "Datasets and models" section gives a brief overview of the RNN and LSTM models and discusses their parameters and implementation. "Experiments" section shows the experimental results that validate our work and "Conclusion and future work" section contains a summary of our main findings and our proposal for future research directions.

Related work

Gomes et al. [3] examined various key workload aspects of email traffic such as email arrival process, size, popularity distribution and temporal locality of email using probabilistic models to distinguish between spam and non-spam traffic. Their results show that the message size is more accurately fitted with the lognormal distribution, and this is consistent with the earlier work reported in [2]. The arrival process was shown to follow a Poisson distribution and a Zipf-like distribution provided the best fit for the distribution of the number of emails, which is consistent with the findings from [8].

Shah and Noble [4] found that message sizes can be represented by log-normal distributions at the body (similar to the findings of [2]) and by Pareto distributions at the tail, which is in contrast to the finding of Gomes et al. [3] which found that the tail is also log-normally distributed. The work in [2] did not analyse heavy-tailed characteristics. Spam messages were found in [4] to be larger than non-spam, however Gomes [3] observed that spam messages are typically smaller than non-spam emails.

Bertolotti and Calzarossa [2] provided a characterization of workload traffic generated from the SMTP and POP3 protocols collected from four mail servers covering different workload aspects such as time stamp, message size, number of recipients of the messages

and IP address details. Consistent with the findings of [4], the lognormal distribution was found to be the best fit for the message size.

The aforementioned studies focused on mining the logs of email traffic workload characteristics and did not consider certain network traffic properties which are critical to validating the models, such as Long-Range Dependence (LRD) and self-similarity characteristics.

The work in [9] by Dada et al. [10] concluded that there is a need to apply deep learning to spam filtering in order to exploit its numerous processing layers and many levels of abstraction to learn representations of data. The work in discusses the importance of traffic prediction in order to eliminate traffic redundancy in the cloud.

Leland et al. [11] provided the first empirical evidence of self-similarity characteristics in LAN traffic. Self-similarity describes the phenomenon in which the behaviour of a process is preserved irrespective of the scale in time. Paxson and Floyd [12] showed that packet inter-arrival times for wide area Internet traffic was characterised by heavy-tailed distributions and burstiness, which indicated that the Poisson process underestimated both burstiness and variability. Additionally, the probabilistic modelling approach does not take into account the time varying property of email traffic which exhibits dependencies across time as shown through its correlated behaviour across widely separated times (long term memory) [11].

Short term forecast plays an important role to manage bandwidth and demand for users' reliability, while medium term forecast helps to formulate the scheduling of maintenance and long term forecast helps to reduce investment risk. However, developing and selecting an accurate time series model is a challenging task, as this requires training several different models for selecting the best amongst them. This is normally done along with substantial feature engineering to derive informative features and finding optimal time lags. The derived informative features and the optimal time lags are commonly used as the input features for time series models. Given that there are not many investigations of the use of RNNs and LSTMs for modelling email traffic in the literature, we provide below information on research in forecasting electric consumption which is qualitative similar to the work we conducted in this paper.

Bouktif et al. [13] has used two variants of deep neural networks such as LSTM and gated recurrent neural network (GRU) to forecast electric consumption. The experiment result shows that LSTM and GRU deep learning models with multi sequence time lags achieve higher performance as compared to single sequence and that they achieved the most accurate and stable results using 1 day and 1 week input sequence. The research is similar to our study as it considers the periodic characteristic of the traffic workload using single and multiple sequence input time lags.

In another paper by the same authors, Bouktif et al. [14] investigated features selection using genetic algorithm (GA) to find the optimal time lags and number of layers for LSTM model performance optimization using electricity consumption data for short and medium term forecasting horizons. The rationale behind focusing on these two forecasting models is that deterministic models can be used successfully for both of them. However, in the case of long term forecasting, stochastic models are needed to deal with uncertainties of forecasting parameters that always have a probability of occurrence [15]. Two challenges are associated with the

targeted forecasting horizons. In the short term case, the accuracy is crucial for optimal day-to-day operational efficiency electrical power delivery and in the medium term case, the prediction stability is needed for the precise scheduling of fuel supplies and timely maintenance operations.

He [16] has used a Convolutional Neural Network (CNN) for feature extraction of maximum and minimum temperatures, whether is during holiday, the hour of the day and day of week features from historical load sequence and recurrent neural network (RNN) for forecasting 1 day ahead hourly electric loads of a North China city. The proposed method outperformed all baseline models, reducing the prediction error.

Marino et al. [17] showed the superiority of the standard LSTM model as compared to the sequence to sequence (seq2seq) based architecture for forecasting 1 h and 1-min time step electricity consumption data from one residential customer.

Janardhanan and Barrett [18] compared the performance of the LSTM model with the traditional autoregressive integrated moving average (ARIMA) model for forecasting CPU usage of resources in a datacentre. The LSTM model clearly outperformed the ARIMA model.

Cao et al. [19] investigated ensemble techniques for univariate time series forecasting of CPU workload of machines in a datacenter and compared the performance with ARIMA model. The ensemble model performed better in terms of prediction accuracy and adaptability to the dynamic pattern change in time dataset as compared to the ARIMA model.

Zheng et al. [20] also used Neural Networks in their work for load forecasting in the smart grid. They proposed a novel Long-Short-Term Memory (LSTM) algorithm combined with Recurrent Neural Network (RNN). This algorithm accurately forecasts non-stationary and non-seasonal electrical loads. According to their simulation results, SARIMA had the best performance followed by LSTM in their first test scenario. However, LSTM had the best performance among all the other algorithms when feeding it the regional electrical consumption data.

Returning to the topic of email traffic prediction, Boukoros et al. [6] divided email traffic into five categories: system incoming/outgoing, users incoming/outgoing and spam traffic. The datasets were collected from the Technical University of Crete (TUC) in Greece for nine non-consecutive weeks between February and October 2014. In contrast to the previous results [2–4], the best fits were found to be provided by the log-logistic and Generalized Extreme Value distributions. The models were evaluated via several statistical tests such as Q–Q plots, Kolmogorov–Smirnov (KS), Anderson–Darling (AD), Kullback–Leibler (KL) Divergence and Relative Percentage Error (RPE). The average accuracy achieved was 83% excluding some outliers.

In order to address the limitations of probabilistic models, email traffic was evaluated as a time series problem using Recurrent Neural Networks in [5] and the prediction accuracy was found to be substantially higher than the probabilistic modelling approach in [6]. In this work we provide further investigation with the use of tuned hyper-parameters. RNN and LSTM models are evaluated to propose a model which could best fit all email traffic categories.

Table 1 Related literature on email traffic workload modelling

Authors	Objective and dataset	Key parameters	Techniques & findings
Shah and Noble [4]	Large scale study of email patterns. The dataset was collected over 7 months (2.85 million messages) from a departmental server	Message size, content type, temporal locality	Lognormal distribution is the best fit for the size of the message body, Pareto distribution is the best fit for the tail. Spam email sizes are larger than that of legitimate email
Gomes, et al. [3]	Focus on identifying the characteristics that significantly distinguish spam from non-spam traffic. The dataset consists of 8 days of SMTP incoming email logs collected from a university in Brazil	Email arrival process, size and popularity distribution and temporal locality	The inter-arrival time for spam traces is exponentially distributed Email sizes follow lognormal distribution for both spam and non-spam traces. However, the average size of non-spam emails is six to eight times larger than the average size of spam The distribution of the number of recipients per email is modelled with a Zipf-like distribution and is heavier tailed in the spam workload Temporal locality is much weaker among spam recipients than for non-spam
Bertolotti and Calzarossa [2]	Focus on the accurate characterization of the email traffic workload. The datasets were collected from the mail servers of an ISP, two enterprises and a university in Italy	Arrival process, size and the number of recipients of messages	Weibull distribution model is found to provide the best fit for modelling inter-arrival times whose values are smaller than a threshold, where Pareto distribution is the best fit for inter-arrival time larger than a threshold. The empirical inter-arrival time distribution threshold value is approximately equal to 7 s
Lee and Kim [32]	Focus on the coexistence of the Poisson process and self-similarity. The dataset consists of 9 months of SMTP traces collected from a web portal in South Korea	Inter-arrival time of SMTP traces	The Q–Q plot and Chi square test demonstrate that the inter-arrival time of SMTP traces follows a Poisson process. On the other hand, the inter-arrival time also exhibits self-similarity and long range dependence
Boukoros, et al. [6]	Focus on modelling workload of email servers for all categories of traffic using probability distribution models and statistical test. The datasets were collected over 9 months from a university in Greece	Users' incoming and outgoing email sizes, system incoming and outgoing email sizes and spam email sizes	In contrast to several of the above works, the lognormal distribution was found unable to provide the best fit for any of the categories. The best fit was provided by the log-logistic and Generalized Extreme Value distributions
Boukoros, et al. [5]	Focus on modelling email traffic as a time series problem. The datasets were collected from four universities over several months	As in [6]	The Recurrent Neural Network model has achieved significantly higher accuracy compared to the probability distribution models

Table 1 provides a snapshot of the most relevant work on email traffic modelling.

Datasets and models

Our goal is to predict, through our models, the sizes of upcoming emails.

The proposed models are evaluated with the datasets collected from four universities, namely the Technical University of Crete (TUC), Greece, the University of Peloponnese (UoP), Greece, Murdoch University, Australia and Liverpool John Moores University (LJMU), UK. The email traffic logs were collected for ten non-consecutive weeks from TUC, four consecutive weeks from LJMU, 5 consecutive weeks from UoP, and 52 consecutive weeks from Murdoch university. The datasets consist of spam, system incoming and outgoing and users' incoming and outgoing email traffic respectively. Workloads are characterized based on temporal dependence and email size.

In this section, we analyse characteristics that have significant impact on the email traffic workload to obtain some intuition on how each one of them influences the load forecasting. The dataset used for the analysis below is the one captured at Murdoch University with a timeframe frequency in seconds. Similar results were acquired for all datasets.

Figure 1 illustrates how the weekdays' (left hand side) and weekends' (right hand side) traffic load varies periodically. The weekdays' traffic load profile is much heavier as compared to the weekends, as expected.

The dataset consists of only one feature (the email sizes), therefore the option for feature improvement is limited. However, we imputed the missing values using linear imputation method to capture email trend patterns uniformly to provide a way to make use of patterns that are missing. Further, the dataset is normalized using MinMaxScaler function and the data range is scaled between $[-1, 1]$.

Statistical hypothesis test for stationarity

Time series data are considered stationary if they do not have trends or seasonal effects. If a time series is stationary, it is easier to make predictions about its values, as the way the time series changes is predictable. Therefore, the first step is to check whether there is any evidence of a trend or a seasonal effect in our dataset. The raw dataset is evaluated to check whether it is stationary with respect to mean and standard deviation using the Augmented Dickey-Fuller test (ADF) [21]. The ADF test is a type of statistical test called a unit root test. The null hypothesis (H0) of the test is that time series can be represented by a unit root, meaning that it is not stationary. The alternative hypothesis (H1) of the

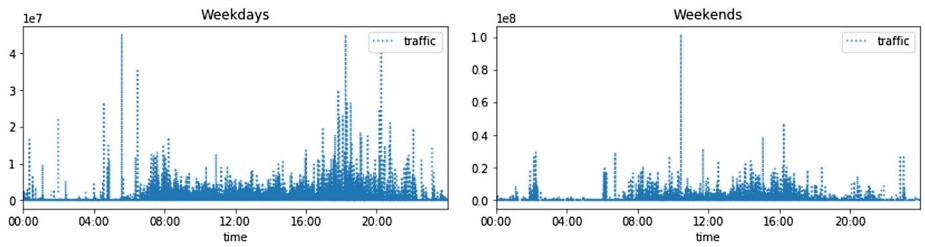

Fig. 1 Email traffic workload on weekdays versus weekends

test suggests that it does not have a unit root, meaning that the time series is stationary. In ADF statistics, if the p value > 0.05, the data has a unit root and is non-stationary, and if p-value ≤ 0.05, the data does not have a unit root and is stationary.

In all the tests we ran for the Murdoch University dataset, the ADF statistics ranged between (-7.37) and (-10.73) for users' and system emails. Similarly, for the TUC dataset the ADF statistics ranged between (-12.92) and (-55.27). The p value was less than 0.05 for all email types, which means that the time series is stationary, thus we reject the null hypothesis.

Figure 2 shows the rolling average (moving average) and the rolling standard deviation of the time series.

Recurrent neural networks

Neural networks have a very wide range of applications, from video activity detection [22] and face recognition [23] to classification of 3D objects [24] and emotion classification [25].

RNN models are based on similar ideas to those of feedforward neural networks (FFNNs). The main difference is that the output of FFNNs at any time t, is a function of current input and weight, while the output of RNNs at time t, depends not only on the current input and weight but also on previous inputs. Modelling temporal data is critical in most real-world applications, since natural signals like network traffic, speech and video have time varying properties and are characterized by having dependencies across time. Feedforward neural networks (FFNNs) are limited since they are unable to capture temporal dependencies. Simple RNNs, also known as Elman networks and Jordan networks, were introduced to address this limitation. RNNs are artificial neural networks that can capture temporal dependencies. Instead of training the network with single input and single output at each time-step, RNNs use sequences as inputs in the training phase.

Despite the elegance of these networks, it was recognized in the early'90 s that all these networks suffer from the vanishing and exploding gradient problem [7]. While training the network, weight matrices are adjusted with the use of gradient in the backpropagation process by continuous multiplications of derivatives. The value of these derivatives may be so small, that these continuous multiplications may cause the gradient to practically vanish. Hence, capturing relationships that span more than eight or ten steps back

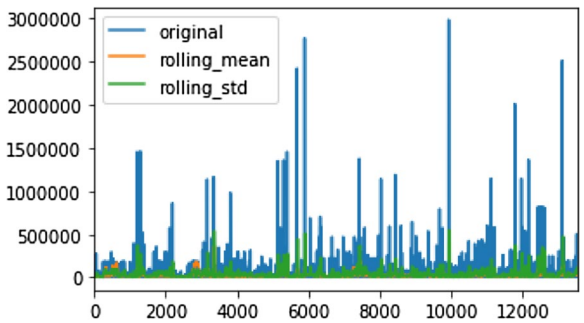

Fig. 2 Moving average

is practically impossible and this makes it difficult to train the network using long range dependence [26].

In the mid-90 s, LSTM was invented to address this problem [7]. The key novelty in LSTM was that the ideal state variables can be kept fixed by using gates and be reused at an appropriate time in the future.

Part of the difficulty in training deep neural networks lies in determining the proper weight initialization strategy to deal with the problem of shrinking variance in deeper layers. The first systematic analysis of this problem was conducted by [27], in which samples were drawn from a truncated normal distribution centered at 0, which is widely used in the initialization of neural networks and commonly referred to as the Glorot (Xavier) initialization. In contrast, the work carried out in [28] argued that the Glorot initialization does not work well with the *relu* activation function, and proposed a different initialization, commonly known as He initialization. A third initializer was introduced by [29], commonly referred to as the LeCun initializer.

To evaluate the accuracy of using time series for email traffic prediction, we present in the following section a study on the aforementioned initialization strategies and on several well-known activation functions for efficient training of RNN models to address the vanishing and exploding gradient problem. Figure 3 presents the proposed model architecture. The raw input data is pre-processed and the missing values are filled using interpolate linear function. It is further processed to check for the outliers, scale to a given range and then divided into training and testing subsets. To further improve the performance of the selected model, the hyper-parameters such as weight initializers, activation and loss function are tuned to choose the best set of parameters. The optimal number of LSTM layers and window size are selected using a genetic algorithm. The best parameters are modelled to compare the performance of the RNN and LSTM models.

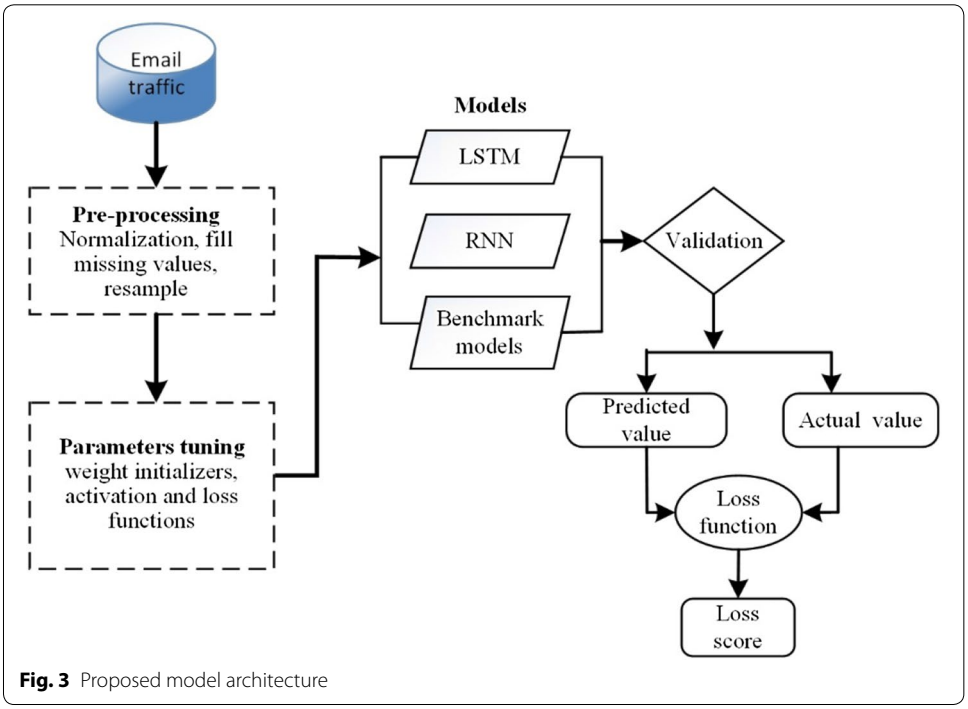

Fig. 3 Proposed model architecture

Experiments

In order to evaluate our modelling/prediction results, we use the Relative Percentage Error (RPE) metric. The RPE evaluates the differences between the actual values and the corresponding predicted values by the models, and expresses them as a percentage [30]:

$$RPE = \frac{|Y - X|}{X} * 100\% \tag{1}$$

where Y is the predicted value and X is the real observation

A baseline RNN model with the same network architecture is used for all the methods described in this experiment to identify the optimal hyper-parameters addressing the vanishing and exploding gradient problem. We are using the baseline RNN model because it is sufficient to represent the category of deep neural network models, as well as to make the experiment feasible. Three experiments were undertaken:

- Experiment 1 compares popular weight initializer methods for handling the vanishing and exploding gradient problem.
- Experiment 2 compares commonly used activation functions.
- Experiment 3 compares the RNN and LSTM models in regard to their accuracy for our email traffic datasets.

For comparison purposes, we have also used Linear Regression, K-Nearest Neighbor and Random Forest, ARIMA and SARIMA. We have used random grid search [31] to fine-tune the models. All the implemented models used in this study used the mean squared error as the loss function and RPE to measure their performance.

Experiment 1: weight initializers

Task

The purpose of this experiment is to evaluate the convergence of a deep neural network model given different weight initialization strategies to minimize potential vanishing and exploding gradient problems.

Architectures

A general rule for setting weights is to set them to be close to zero without being too small [27]. The proposed RNN model is built with input hidden layers of 128, 64 and 32 units for the LSTM and RNN models respectively and one output layer. The *relu* activation function is applied on each of the hidden layers with some added dropout in-between. We have also defined a mean squared error loss function along with an Adam optimizer function, and trained the model choosing 1000 epochs. Finally, the RPE is evaluated by calculating the differences between the actual and predicted values over the actual values, for the respective weight initializers. All parameters are selected based on the grid search result of the best performing estimates.

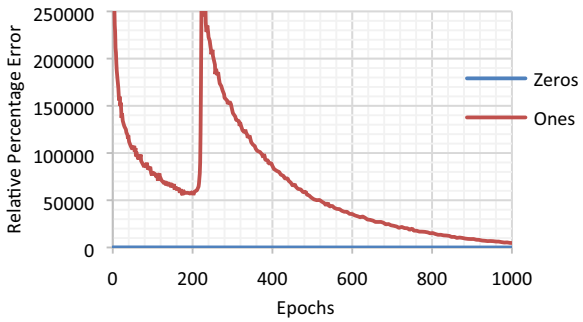

Fig. 4 RPE for the RNN model with constant weights of ones and zeros

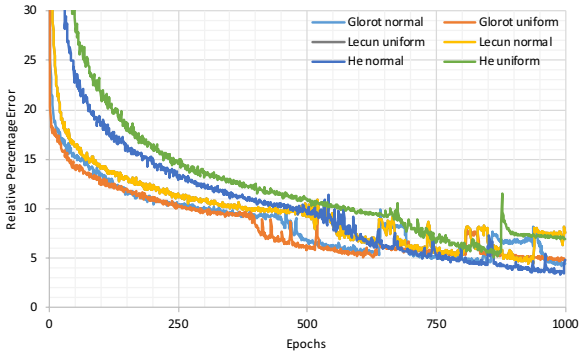

Fig. 5 RPE for the Glorot, He and Lecun weight initializers

Case 1: Ones and zeros constant weights

Figure 4 shows the comparison, for the TUC spam email dataset, of error rates with respect to all *ones* and *zeros* constant weight initializers, which are iterated over 1000 epochs. When the model is initialized with all *ones* constant weights, the error rate starts out with an extremely high initial rate and fluctuates randomly over the training sample data. On the other hand, when the model is initialized with *zero* constant weights, the error rates remains flat over time. This clearly shows that the weights of a model should never be initialized to constant *zeros* or *ones* because if every neuron in the network computes the same output, then they will also compute the same gradients and weight updates during the backpropagation. This makes it hard to decide which weight to adjust and the algorithm will not be able to learn to minimize the loss.

Case 2: Glorot, He and Lecun weight initializers

Figure 5 presents the RPE results for RNN baseline modelling using *Glorot, He and Lecun* weight initializers. These results indicate the difficulty of training using sample TUC spam email data. As shown in the figure, the *Glorot, He and Lecun* weight initializers start with high RPEs for all activation functions, but the errors decrease substantially as the number of epochs increases. The results show that *Glorot* initialization strategies perform relatively better with minimum 4.47% and 4.83% RPE values for normal and

uniform initializers, respectively, followed by the *He* normal initializer. This result is consistent for all datasets.

Experiment 2: Activation functions

Task

This experiment evaluates the *sigmoid, tangent, softsign and relu* activation functions to evaluate their performance in training the neural network by measuring the error rates.

Architecture

The RNN model was trained with the r*elu* activation function applied on the input-to-hidden and hidden-to-hidden layers and the same architecture was trained from scratch with *tanh, sigmoid and softsign.* The model is built with an input layer, three hidden layers with 128, 64, and 32 nodes respectively and a final output layer. We used mean squared error loss and Adam optimizer functions and iterated with 1000 epochs. All these parameters are selected based on the grid search result of the best performing estimates.

Figure 6 shows the RPE for the softsign, relu, tanh and sigmoid activation functions evaluated using the sample baseline spam email dataset from TUC. The performance of the relu activation function is relatively better with a minimum RPE of 5.95%. This result is again consistent for all datasets.

Experiment 3: Performance comparison between RNN vs LSTM

Task

LSTM models have been introduced to overcome the vanishing gradient problem of the RNN model. However, a comparison of the performance of RNN and LSTM for email traffic datasets has not been reported, to the best of our knowledge. This experiment examines the performance of the models on all of our datasets.

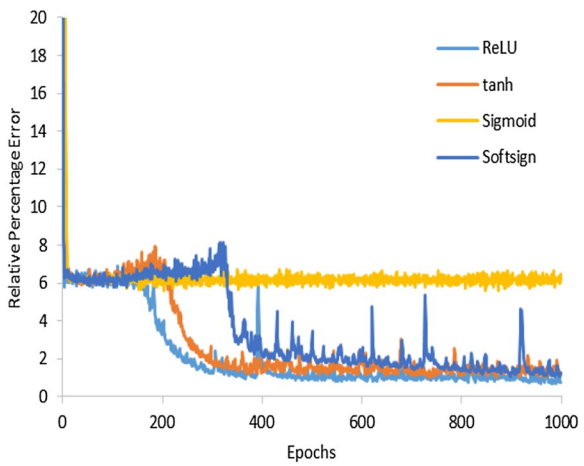

Fig. 6 RPE for different activation functions

Architecture

We have used the same baseline experimental setting for RNN and LSTM models in order to draw a fair comparison of their performances.

The first key difference between this study and previous work is with regard to the approaches we used and the experimental environment. The work in [6] was based on statistical modelling and excluded outliers. The authors did not investigate the time series properties of email traffic data. The work in [5] used a RNN model to model email traffic workloads as a time series, including the outliers [5]. However, the authors did not evaluate the best fit hyper-parameters for the categories of email traffic to overcome the vanishing gradient problem.

This gap led us to evaluate various hyper-parameters that best fitted the different categories of emails in our datasets, to improve the performance of the model. Email traffic is passed as a single sequence with input time lags. As mentioned earlier, it is normalized using MinMaxScaler with a feature range from $(-1,1)$ and the values of Nan and zero are dropped, similarly to [5]. The proposed model is tuned using different activation functions, weight initializers, optimizers, neural network units and batch sizes for the different categories of the datasets and the best fit parameters are selected. The model is initialized using the first training window with three input sequences to predict the next sequence. In the next iteration, the same window data points are included as part of the next training dataset and subsequent data points are forecast in the next iteration and so forth. The model is validated against predicted and actual values and the RPE is recorded to evaluate the performance of the model.

We used a z-score outlier detection threshold value of 10% on the UoP incoming emails dataset because the frequency of email traffic was steady over the first 4 weeks and there was a sudden rise and fluctuation in the email sizes in the last (fifth) week. Hence, the model could not be trained adequately to predict the outliers. For this reason, we have used outlier detection to remove the extreme data points.

Table 2 shows the relative percentage error values for the other models used (Linear Regression, K-Nearest Neighbor and Random Forest, ARIMA and SARIMA) using dataset from Murdoch university and Technical University of Crete. The performance of KNN and Linear regression are better as compared to the other models, but as it will be shown below, their performance is very poor compared to our approach.

Table 3 presents the RPE values over all our datasets for the models used in our work and those from [5, 6] for comparison purposes. The performance of the proposed RNN model is clearly better than that of the models in [5, 6] for all email categories in our four datasets. The significant improvement in the performance compared to the RNN used in [5] is due to the tuning of the hyper-parameters, which are adjusted with respect to the different email categories to get the best performance possible. Still, the LSTM model outperforms RNN in 12 out of the 16 categories. The reason is that RNN uses feedback connections to store representations of recent input events in the form of activations (short term memory) as opposed to long term memory gate features which are used in LSTM to control the constant error flow [7]. Long term memory is shown in our results to be necessary for improved model performance when modelling email traffic.

Table 2 Other models' performance

Email traffic	Technical University of Crete					Murdoch University				
	KNN	Random Forest	Linear regression	ARIMA	SARIMA	KNN	Random Forest	Linear regression	ARIMA	SARIMA
Incoming										
Users	240.28	673.92	240.28	1697.90	99.99	71.46	74.06	82.33	469.21	198.00
System	19.39	183.80	29.25	122.37	99.99	92.97	3181.38	92.98	27.13	133.66
Outgoing										
Users	424.68	13436.85	424.68	14119.89	99.72	66.88	1254.35	66.85	409.85	840.70
System	18.33	318.15	23.08	128.39	99.99	52.80	112.80	78.91	159.99	57.28
Spam	28.84	554.92	28.80	567.08	99.89					

Table 3 RPEs for all models and all datasets

Prediction error (RPE %)								
Category	**Technical University of Crete**				**University of Peloponnese**			
	Previous works		**Proposed models**		**Previous works**		**Proposed models**	
Incoming traffic	**Probability distribution**	**RNN**	**RNN**	**LSTM**	**Probability distribution**	**RNN**	**RNN**	**LSTM**
Users incoming	21.5	13.9	6.22	5.64	16.7	9.20	3.29	3.33
System incoming	20.8	2.1	1.97	2.04	23.00	7.00	9.96	6.08
Outgoing traffic								
Users incoming	14.7	9.4	2.28	1.34	29.60	13.70	4.96	5.22
System incoming	10.00	5.30	2.25	2.22	20.40	4.40	2.83	1.38
Spam traffic								
Spam traffic	17.7	17.7	4.71	3.95	25.00	57.10	4.73	4.62
Category	**Murdoch University**				**Liverpool John Moores University**			
	Previous works		**Proposed models**		**Previous works**		**Proposed models**	
Incoming traffic	**Probability distribution**	**RNN**	**RNN**	**LSTM**	**Probability distribution**	**RNN**	**RNN**	**LSTM**
Users incoming	32.70	14.20	0.37	0.19	8.4	4.2	1.35	1.09
System incoming	9.30	4.20	0.29	0.25				
Outgoing traffic					Spam traffic			
Users incoming	40.80	25.30	0.97	0.92	36.9	18.7	2.45	2.26
System incoming	22.60	23.30	0.28	0.40				

Conclusion and future work

In this work we have focused on the problem of modelling email traffic workloads by treating traffic as a time series function. We have discussed the existing literature and we have used RNN and LSTM models for modelling email traffic gathered from four different universities. We have shown that with the use of appropriate initialization of the training weights, proper activation functions and hyper-parameters the performance of the RNN model can be substantially improved for modelling email traffic. However, the highest accuracy achieved by RNN is smaller for most email traffic categories in our datasets than the performance achieved by LSTM. Our models clearly outperform a large number of other modelling approaches from the literature.

Our results reveal that model selection is crucial and that the prediction of future email traffic loads with very high accuracy is possible. Our future work will focus on the load variations across different time periods and on outlier detection, for possible further improvement of the models' accuracy through feature extraction. We also intend to focus on approaches to automatically tune the hyper-parameters and the deep learning architecture.

Acknowledgements

We would like to thank our colleagues Profs. Costas Vassilakis from the University of Peloponnese and Angelos Marnerides from Lancaster University for their help with collecting the datasets and collaborating in their previous analysis. We would also like to thank Mr. Panagiotis Kontogiannis, Head of the Educational Computational Infrastructure at the Technical University of Crete, Mr. Martin Connell, Senior Systems Engineer at Liverpool John Moores University and Mr. Mario Pinelli, Manager of Computer Services and IT at Murdoch University. Without their help with collecting the datasets this research would not have been possible.

Authors' contribution
KO worked on the literature review and the implementation of the RNN and LSTM models, as well as on the analysis of the experimental results. SB worked on the analysis of the experimental results. AN worked on the implementation of the RNN and LSTM models and on the analysis of the experimental results. TMG worked on the literature review and on the structure of the paper. Michael Dixon (MD) worked on the literature review and on the structure of the paper. PK worked on the literature review and the implementation of the RNN and LSTM models, as well as on the analysis of the experimental results. KWW worked on the literature review and the implementation of the RNN and LSTM models, as well as on the analysis of the experimental results. All authors read and approved the final manuscript.

Funding
This was not a funded research project.

Availability of data and materials
The datasets used and analysed in this study are available from the corresponding author on reasonable request.

Competing interests
The authors declared that they have no competing interests.

Author details
[1] Discipline of Information Technology, Mathematics & Statistics, Murdoch University, Science and Computing Building 245, SC1.012, 90 South Street, Murdoch, WA 6150, Australia. [2] Technische Universität Darmstadt, Darmstadt, Germany.

Received: 2 February 2020 Accepted: 11 August 2020
Published online: 02 September 2020

References

1. The Radicati Group (2019) Email statistics report, 2019–2023, [Online]:https://www.radicati.com/wp/wp-content/uploads/2019/01/Email_Statistics_Report,_2019-2023_Executive_Summary.pdf
2. Bertolotti L, Calzarossa MC (2001) Models of mail server workloads. Perform Eval 46:65–76
3. Gomes LH, Cazita C, Almeida JM, Almeida V, Meira W Jr (2007) Workload models of spam and legitimate e-mails. Perform Eval 64:690–714
4. Shah S, Noble BD (2007) A study of email patterns. Softw Pract Exp 37:1515–1538
5. Boukoros S, Nugaliyadde A, Marnerides A, Vassilakis C, Koutsakis P, Wong KW (2017) Modeling server workloads for campus email traffic using recurrent neural networks. In: Paper presented at the International Conference on Neural Information Processing (ICONIP), Guangzhou, China, November 2017
6. Boukoros S, Kalampogia A, Koutsakis P (2016) A new highly accurate workload model for campus email traffic. In: Paper presented at the IEEE International Conference on Computing, Networking and Communications (ICNC), Kauai, Hawaii, February 2016
7. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780
8. Newman ME, Forrest S, Balthrop J (2002) Email networks and the spread of computer viruses. Phys Rev E 66:035101
9. Dada EG, Bassi JS, Chiroma H, Abdulhamid SM, Adetunmbi AO, Ajibuwa OE (2019) Machine learning for email spam filtering: review, approaches and open research problems. Heliyon 5:e01802
10. Zohar E, Cidon I, Mokryn O (2013) PACK: prediction-based cloud bandwidth and cost reduction system. IEEE/ACM Trans Netw 22:39–51
11. Leland WE, Taqqu MS, Willinger W, Wilson DV (1994) On the self-similar nature of ethernet traffic (extended version). IEEE/ACM Trans Netw 2:1–15
12. Paxson V, Floyd S (1995) Wide area traffic: the failure of poisson modeling. IEEE/ACM Trans Netw 3:226–244
13. Bouktif S, Fiaz A, Ouni A, Serhani M (2019) Single and multi-sequence deep learning models for short and medium term electric load forecasting. Energies 12:149
14. Bouktif S, Fiaz A, Ouni A, Serhani M (2018) Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: comparison with machine learning approaches. Energies 11:1636
15. Chui F, Elkamel A, Surit R, Croiset E, Douglas P (2009) Long-term electricity demand forecasting for power system planning using economic, demographic and climatic variables. Eur J Ind Eng 3:277–304
16. He W (2017) Load forecasting via deep neural networks. Procedia Comput Sci 122:308–314
17. Marino DL, Amarasinghe K, Manic M (2016) Building energy load forecasting using deep neural networks. In: Paper presented at the 42nd annual conference of the IEEE industrial electronics society, Florence, Italy, October 2016
18. Janardhanan D, Barrett E (2017) CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models. In: Paper presented at the 12th international conference for internet technology and secured transactions, Cambridge, UK, December 2017
19. Cao J, Fu J, Li M, Chen J (2014) CPU load prediction for cloud environment based on a dynamic ensemble model. Softw Pract Exp 44:793–804
20. Zheng J, Xu C, Zhang Z, Li X (2017) Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In: Paper presented at the 51st annual conference on information sciences and systems Baltimore, USA, March 2017
21. Dickey DG (2011) Dickey-Fuller Tests. In: Lovric M (ed) International encyclopedia of statistical science. Springer, Berlin
22. Song Y, Kim I (2018) DeepAct: a deep neural network model for activity detection in untrimmed videos. J Info Process Syst 14:150–161

23. Zhang J, Jin X, Liu Y, Sangaiah AK, Wang J (2018) Small sample face recognition algorithm based on novel siamese network. J Info Process Syst 14:1464–1479
24. Song W, Zou S, Tian Y, Fong S, Cho K (2018) Classifying 3D Objects in LiDAR point clouds with a back-propagation neural network. Hum-Cent Comput Info Sci 8:29
25. Li T-M, Chao H-C, Zhang J (2019) Emotion classification based on brain wave: a survey. Hum-Cent Comput Info Sci 9:42
26. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J (2001) Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: A field guide to dynamical recurrent neural networks, IEEE Press
27. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Paper presented at the thirteenth international conference on artificial intelligence and statistics (AISTATS), Sardinia, Italy, May 2010
28. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Paper presented at the IEEE international conference on computer vision (ICCV), Santiago, Chile, December 2015
29. LeCun YA, Bottou L, Orr GB, Müller K-R (2012) Efficient backprop, neural networks: tricks of the trade. Lect Notes Comput Sci 7700:9–48
30. Lanfranchi LI, Bing BK (2008) MPEG-4 bandwidth prediction for broadband cable networks. IEEE Trans Broadcast 54:741–751
31. (2019) Scikit-learn.org, Parameter estimation using grid search with cross validation. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
32. Lee Y, Kim J-S (2008) Characterization of large-scale SMTP traffic: the coexistence of the poisson process and self-similarity. In: Paper presented at the IEEE international symposium on modeling, analysis and simulation of computers and telecommunication systems (MASCOTS), Baltimore, USA, September 2008

Publisher's Note