Predicting Host CPU Utilization in Cloud Computing using Recurrent Neural Networks

Conference Paper · November 2017

5 authors, including:

Martin Duggan
National University of Ireland, Galway


Karl Mason
National University of Ireland, Galway


Jim Duggan
National University of Ireland, Galway

Enda Barrett

Predicting Host CPU Utilization in Cloud Computing using Recurrent Neural Networks

Martin Duggan*, Karl Mason, Jim Duggan, Enda Howley, Enda Barrett
National University of Ireland, Galway
Email*: m.duggan1@nuigalway.ie

***Abstract*—One of the major challenges facing cloud computing is to accurately predict future resource usage for future demands. Cloud resource consumption is constantly changing, which makes it difficult for forecasting algorithms to produce accurate predictions. This motivates the research presented in this paper which aims to predict host machines CPU consumption for a single time-step and multiple time-steps into the future. This research implements a Recurrent Neural Network to predict CPU utilisation, due to their ability to retain information and accurately make predictions for time series problems, making it a promising candidate to predict CPU utilization with greater accuracy when compared to traditional approaches.**

Keywords—Cloud Computing, CPU Prediction, Neural Networks

I. Introduction

Armbrust et. al. have listed resource prediction as one of the ten biggest obstacles facing the continued growth of cloud computing [1]. One of the major difficulties for prediction algorithms in cloud computing is that cloud resources are in a constant state of flux. Traditional forecasting techniques such as for as ARIMA rely on patterns in historical data to make future predictions [19]. These approaches are not suitable when the data is not stationary or when there is a significant amount of random variation in the data. This paper uses a recurrent neural network to improve upon traditional forecasting techniques to make accurate time series prediction of host machines CPU utilization as they are much more adaptable and robust than these traditional approaches. CPU is the resource with the highest level of demands in virtualized environments and therefore is a major cause of resource shortages on host machines. CPU is one of the most important metrics for measuring the performance of host machines and is a popular metric for researchers to test when predicting host performance [28], [9], [6]. These studies have examined one-step ahead forecasting using methods such as LOESS and feed-forward Neural Network to predict CPU utilization. However, one step ahead prediction time models (usually 5 minutes ahead) give insufficient time for the cloud resources to be adjusted, when sudden heavy demands occur. Research has shown [5] predicting a workload on a short time scale such as 5 minute intervals is more difficult to produce accurate results than for long-term forecasting (i.e., time steps of days or weeks). This is due to the fact that cloud resources in these short time scales can be extremely unpredictable. The further into the future an algorithm can accurately predict the demand on data centre resources is critical to how well a data centre is able to perform. This is one of the key ideas that has motivated this research.

In recent years machine learning algorithms have received a lot of attention and are becoming popular to use in cloud computing. One of the most effective and diverse machine learning methods is the Neural Network [7], which is inspired by the brain. Neural networks act as function approximators which makes them widely applicable to a broad range of problems from regression to robotics. The Recurrent Neural Networks are of interest in this research due to their ability to retain information making it a promising candidate to predict CPU utilization with greater accuracy when compared to traditional approaches.

In this paper, we predict host CPU utilization using Recurrent Neural Networks. The aims of this research are to:

1) Investigate the accuracy of a Recurrent Neural Network for predicting CPU utilization when compared to traditional methods.
2) To determine how far into the future the Recurrent Network can accurately predict host CPU utilization.

The outline of the paper is as follows. Section II gives an overview of forecasting in cloud computing, and neural networks. The experimental procedure will be explained in Section III. Section IV will present the experimental results. These results will then be discussed in Section V. Finally, Section VI will conclude the paper.

II. Related Work

Cloud resources such as CPU are in a constant state of flux and are difficult to predict on a short time scale (e.g. 20-30 minutes). Approaches such as one-step-ahead prediction give very little time for the data centre to re-adjust resource required when bursts of high traffic occur. An algorithm that can produce accurate prediction 20 to 30 minutes into the future could inform the data centre management systems to perform suitable actions such as turn host machines on/off to deal to deal with future demands. The objective of this paper is to use a recurrent neural network to predict host machines CPU utilization with a high degree of accuracy.

A. Forecasting in Cloud Computing

Host machine CPU is one of the most studied metrics when it comes to performance, as it is a major cause of resource shortage. Dinda and O Hallaron used different linear forecasting models to predict tasks running times, based on CPU load predictions [9]. Zhang et al. employed a multi-step ahead CPU load prediction approach for grid tasks to predict the future performance of the resources [28].

Recently there has been a move towards integrating Artificial Intelligence (AI) and Machine Learning (ML) techniques to improve the overall efficiency of a cloud data centre. Several works show how AI and ML algorithms can provide cloud systems with the abilities to better adapt to the changes in cloud resource consumption to improve resource scaling, VM live migration [13], [11], [10], [12], [25] and resource allocation [3], [2] in cloud computing. Neural networks are one of the most effective and versatile machine learning algorithms and have been successfully applied to areas of cloud computing such as scheduling [14], intrusion detection [26], DDoS attack defence [18] and load forecasting [24]. Neural networks have previously been used to forecast resource demands in cloud computing. Duy et al. employ a neural network predictor for optimising server power consumption in a data centre [14]. They use a feed-forward neural network to predict future load demands based on historical demands to turn on/off servers to minimise the energy usage. Prevost et al. implement neural networks and a linear predictor algorithms to forecast future workloads [24]. Bey et al. use several different models for time series prediction. They use an adaptive network to estimate the future value of CPU load for distributed computing [6]. However, their hybrid predictors were designed to perform for one-step-ahead prediction and the work presented in this paper builds on this work by predicting both one-step and multi-steps ahead. All of the research highlighted above outlines how neural networks are effective at addressing many of the problems in cloud computing, in particular, CPU forecasting. The research presented in this paper makes the novel contribution of applying Recurrent Neural Networks for CPU forecasting.

B. Neural Networks

Neural Networks are function approximators that are inspired by the biological neural networks that constitute the human brain [7]. Some of the applications of neural networks include: power generation [21], control [22] and watershed management [23]. Fig. 1 illustrates the architecture for a neural network which is arranged in a number of layers. The input layer is responsible for taking in the inputs to the model, the hidden layer is where the vast majority of the computation is done and the output layer produces the output of the model.

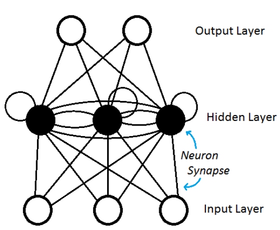

Fig. 1. Recurrent Neural Network [21]. This figure illustrates a Recurrent Neural Network. Neurons are connected weighted synapses that pass signals between neurons. The recurrent synapses can be seen in the hidden layer of neurons. This gives the recurrent network the abilities to retain information.

The standard feed-forward network consists of an input layer of neurons, one or multiple hidden layers of neurons and an output layer. The neural networks receive information in a form of a signal (normalised between 0 and 1) through the input layer neurons and then outputs a signal using the sigmoid function. The signal or input that the network receives in this paper is in the form of two CPU utilisation values from a host machine (normalised between 0 and 1). This two CPU input is propagated forward through the hidden layers of neurons via synapses (weighted connections). Then the network calculates an output at the output layer neuron or neurons. In this paper, only one output is needed and the output signal corresponds to a future CPU values of a host machine. An error signal is calculated by finding the difference between the actual CPU value and the predicted value. This error is then propagated back through the network, and the weights (synapses) are adjusted to correct the error of the prediction.

Aside from the input layer, a neuron in any other layer will have as input. The sum of the weighted signals that are outputted from other connected neurons. A neurons input signal is described by Equation 1.

$$v_j = \sum_{i=1}^{N} w_{i,j} a_i \tag{1}$$

Where v_j is the input to a neuron in the j^{th} layer, layer i is the preceding layer to j that contains N neurons, each neuron in layer i has output a_i and each of these output signals are weighted by the value $w_{i,j}$ as they are passed to each neuron in layer j.

Each neuron a_i outputs a value between 0 and 1. This output value is determined by the activation function of the neuron. The most commonly used activation function is the sigmoid function. This is described by Equation 2

$$a_j = \frac{1}{1 + \exp -v_j} \tag{2}$$

This research will implement a Recurrent Neural Network, illustrated in Figure 1. Recurrent networks are different from the standard feed-forward networks as the hidden layers neurons have recurrent connections. These connections allow the hidden layer neurons to connect to itself. Thus, giving the neural network memory of previous predictions which makes it well suited to the problem of predicting CPU demand. The recurrent network in this paper is trained using the popular Back-Propagation-Through-Time (BPTT) algorithm [27]. The idea of BPTT is the unfolding of the recurrent neural network at a discrete-time into a multilayer feedforward neural network each time a sequence is processed. The BPTT is different to the feed-forward neural networks as it enables the recurrent neural network to store past information, thus suitable for sequential models. The standard feed-forward algorithm is updated as follows:

$$v_j = \sum_{i=1}^{N} w_{i,j} a_i + \sum_{h=1}^{m} s_h(t-1) u_{ih} \tag{3}$$

where U is the recurrent weight matrix and $s_h(t-1)$ is the previous hidden layer

A normal back propagation algorithm weights are updated by calculation the cost function (the error from the actual answer and predicted answer)

$$C = 1/2 \sum_{p=1}^{k} \sum_{e=1}^{o} (d_{pk} - y_{pk})^2 \tag{4}$$

where d is the desired output, k is the total number of training samples and o is the number of output units. Then the change in weights for the output nodes can be calculated as:

$$\delta_{pk} = (d_{pk} - y_{pk}) g'(net_{pk}) \tag{5}$$

where g is the activate function where net represent inputs. The changes in weights for the hidden layer weight can be represented as:

$$\delta_{pj} = \sum_{k=1}^{o} \delta_{pk} w_{kj} g'(net_{pj}) \tag{6}$$

Therefore, the recurrent weights can be then back-propagated back through the network:

$$\Delta u_{ih} = \sum_{p=1}^{N} \delta_{pj} s_{ph}(t-1) \tag{7}$$

C. Network Parameter Selection

The reason for using the recurrent neural network to predict CPU utilization over a feed-forward neural network is due to their ability to retain information and accurately make predictions for time series problems. This makes it a promising candidate for predicting CPU utilization with greater accuracy when compared to traditional approaches.

The recurrent neural network used in this research has three hidden neuron in the hidden layer and has two inputs from the input layer. The inputs into the network are the current and previous CPU utilization values. When a parameter sweeps was conducted, it showed that that a network with three hidden neurons produced the greatest performance. The parameter sweeps also highlighted that having greater than two inputs of CPU utilisation did not increase the recurrent networks performance. The network had one output that corresponded to the network's prediction of future CPU utilization.

D. Network Training

The recurrent neural network algorithm will be trained over 10,000 evaluations and will be evaluated on unseen test data. The experiments are repeated over 10 runs to ensure statistically significant results.

III. Experiment Details

A. Data Models

The data-set used to train and test the recurrent neural network comprised of CPU utilization that was generated by the CoMon project, a monitoring infrastructure for PlanetLab. The project contains CPU utilization data, which was obtained from more than a thousand VMs from 500 data centres around the world. There are ten folders worth of workloads, containing CPU utilization values measured every five minutes in VMs. Each file contains 288 values. We ran the CloudSim simulator using the Lr-mmt algorithm to generate hosts CPU values [8]. In the cloudsim simulation, over 800 host are used. We selected host number 3's CPU values for each of the ten days worth of workloads for our experiments. The first nine workload traces (containing 2296 CPU values data-set) were used to train the recurrent network and the tenth workload (containing 288 CPU values data-set) to test the network. The reason for using these planetlabs files is that they have proven to be useful CPU work-flow data-sets when conducting experiment on simulated cloud host machines [12].

B. Comparative Forecasting Methods

The recurrent neural network will be compared to the following methods:

1) Back-propagation (BP).
2) Random walk forecasting (RWF).
3) Moving Average (MA).

The Back-propagation (BP) algorithm works by calculating the error between the target output and the observed output. This error is then propagated back through the network and is used to update the weights. BP is different to BPTT as it does not store any memory. In this research the BP network has 2 inputs, 3 hidden neurons and 1 output, keeping consistent with the recurrent network's implementation. Random walk forecasting is a basic forecasting method that is implemented as a benchmark algorithm. This approach predicts the next future value as equal to the currently observed value. The moving average method is another commonly using forecasting approach. This method consists of predicting a future value by averaging n previous values. In this paper, the two previous times steps were averaged to give a future prediction.

C. Experiments Conducted

There will be three experiments conducted in this paper. The first experiment involves comparing all of the algorithms and methods performances on the training data. The second experiment will evaluate the performance of each algorithm on the testing data. The purpose of this experiment is to examine if the trained recurrent networks (BPTT) are capable of giving a good general performance on data it has not seen. The third experiment will evaluate how far into the future the BPTT network can predict. The results of these experiment will be interesting as it would be beneficial to data centres management systems to in advance how much CPU is used on each host before events such as live migration can occur. This experiment will evaluate the accuracy of the network for predicting CPU utilization further than one step into the future.

IV. Results

This section presents the results of each of the experiments outlined above followed by a discussion in order to highlight their significance for real world data centre challenges.

A. Training Data

Figure 2 shows the convergence of both the BPTT and BP trained networks on the training data. This graph highlights the average Mean Absolute Error (MAE) at each time step. The graph shows that BPTT converges to a better solution than BP and highlights that BPTT found a better solution faster also. One reason being that the recurrent network can store memory

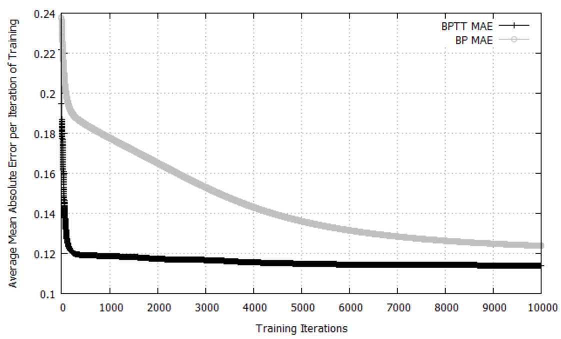

Fig. 2. Convergence of BPTT and BP. This figure illustrates the MAE at each iteration of training for Back-propagation-through-time and Back-propagation neural networks.

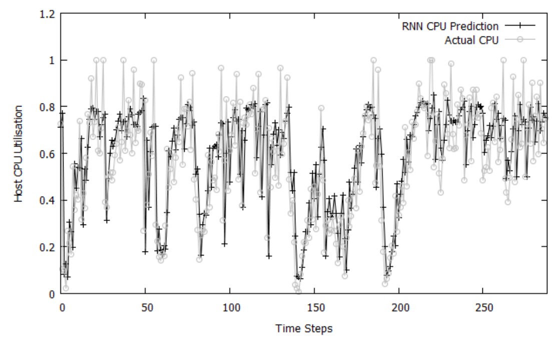

Fig. 3. Host Utilization Predictions for Test Data. This figure illustrates the predicted host utilization of the recurrent neural network on unseen test data.

from the previous time step, allowing the algorithm to achieve a lower error and produce better predictions than the standard feed-forward neural network.

Table 1 presents the results for all forecasting methods in this paper. The Mean Absolute Error (MAE) and Mean Squared Error (MSE) were used to determine how accurate each forecasting method performed on the training data. From the results in Table I, the random walk, moving average and back-propagation neural network performs significantly worse than BPTT algorithm. This validates the choice of using a recurrent neural network trained for host CPU utilisation prediction. When comparing the performances of each method results reveal that MA was the worst performing for the MAE and MSE on the training data, with BPTT producing the best results.

TABLE I. TRAINING DATA ACCURACY

Algorithm	MAE (Std Dev)	MSE (Std Dev)
BPTT	0.1162 0.001)	0.0219 (0.0003)
Random Walk	0.1427 (0.00)	0.0354 (0.0000)
Moving Avg	0.1492 (0.00)	0.0367 (0.0000)
Backpropagation	0.1301 (0.01)	0.031 (0.004)

B. Test Data

The second experiment conducted involved evaluating how well the recurrent neural network can predict unseen test data from the same host machine data that it was previously trained on. Figure 3 plots the prediction of the BPTT and the actual CPU utilisation. The graph shows that BPTT predicts accurately for values between 0.01 and 0.8, however, it struggles to predict values higher than this threshold. One reason for this being that the data has a sudden variation of CPU utilisation. Another reason being that the implementation of the recurrent neural network in this paper only hold the previous step CPU value, if the algorithm stored a longer sequence of data the prediction potentially could improve future predictions.

Table II presents how accurate each of the forecasting methods is when evaluated on the test data. As with the training data, the recurrent BPTT trained network performed the best. Random walk performed significantly worse on the testing data than BP and moving average. The back-propagation algorithm performed worst on both training and testing data when compared to BPTT. This shows that the network trained using BP does not generalise well to time series data.

The results in Table II highlight that the standard deviations for the recurrent network are higher than for the training data. However, this is to be expected as it was being evaluated on previously unseen data so more deviation in the prediction accuracy is to be expected.

TABLE II. TEST DATA ACCURACY

Algorithm	MAE (Std Dev)	MSE (Std Dev)
BPTT	0.1498 (0.001)	0.0468 (0.0008)
Random Walk	0.1716 (0.0000)	0.0513 (0.0000)
Moving Avg	0.1587 (0.0000)	0.0428 (0.0000)
Backpropagation	0.160 (0.010)	0.041 (0.004)

C. Multi-step Ahead Prediction

The aim of the multiple time steps ahead prediction experiment was to evaluate how far into the future the neural network could predict CPU utilization and to establish how much the accuracy of the prediction decreases the further into the future the network attempts to predict. Since BPTT trained network had the best accuracy in both the training and testing data sets, this recurrent neural network was implemented to predict CPU utilization for multiple steps prediction. This experiment involved predicting the CPU utilization of a host machine at 1, 2 and 3-time steps into the future. Each of these time steps corresponds to 5 minutes (i.e. the aim was to predict 15 minutes into the near future of CPU utilisation with high accuracy).

Table III presents the accuracy of the prediction at each of the first, second and third time steps. As presented in the table the further into that the future the recurrent network predicts, the accuracy decreases linearly. This is true for both the training and testing data as present in Table III.

TABLE III. MULTI-STEP PREDICTION ACCURACY

Number of Steps	Training MAE (Std Dev)	Training MSE (Std Dev)	Test MAE (Std Dev)	Test MSE (Std Dev)
1 Step Ahead	0.116 (0.001)	0.021 (0.0003)	0.153 (0.008)	0.038 (0.003)
2 Step Ahead	0.133 (0.003)	0.029 (0.001)	0.167 (0.015	0.046 (0.008)
3 Step Ahead	0.146 (0.0008)	0.035 (0.0001)	0.212 (0.018)	0.066 (0.009)

Figure 4 displays the MAE for both the 1 and 3 steps into the future on the test data. The reason for time step 2 been omitted is for clarity and readability of the graph. This graph highlights the time steps the recurrent network's accuracy is performing the greatest and where the error in predictions are the highest. Figure 4 shows when there are sudden changes in the host machine CPU utilisation the larger the error is in the accuracy of prediction. For instance consider time step 50 in Figure 3. The actual CPU utilisation values show a sudden decrease from 0.9 to 0.2. During this period, Figure 4 shows that there was a sharp increase in the error from the recurrent network prediction at each of the time steps. One reason for the sudden decrease in prediction accuracy is that the network find it difficult to perform well when extreme changes occur in CPU utilisation.

Another observation highlighted from Figure 4 is the difference between the prediction error of the 1 and 3 step ahead predictions on the testing data. Examining time steps 200 to 288 in Figure 3 shows the actual CPU is constant with little sudden changes in CPU utilisation. Figure 4 highlights that the one step ahead predictions produces better results. The third time step ahead prediction errors shows how difficult it is to produce accurate results with a noisy data set used in this experiment. Considering how well the recurrent neural network performed in the multi-time step ahead predicts, from the MAE and MSE results from predicting two-time steps ahead shows that it out performed the random walks results even when that algorithm was only predicting one-time step ahead. The overall average mean squared error for each time step was 0.038, 0.046 and 0.066. These results show a steady increase in the error the further out the network tries to predict.

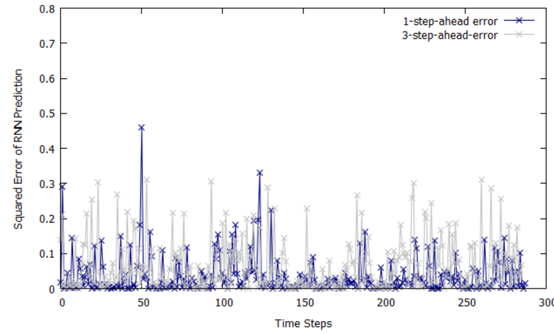

Fig. 4. The MAE of each of the multi-step ahead prediction. This figure illustrates the error of the prediction of CPU utilization for 1-time-step-ahead and 3-time-steps-ahead for the recurrent network.

V. Discussion

The results of the experiments show that the recurrent neural network has the capabilities to improve upon traditional prediction methods such as random walk, moving average and Back-propagation to predict CPU utilization with a high degree of accuracy. This is shown both for the results for one-step and multi-step prediction. The first experiment conducted determine how a recurrent neural network could outperform tradition forecasting methods. The results indicate that even with a large amount of noise in the CPU utilization data the recurrent neural network could produce accurate results on the training data-set compared to the traditional prediction methods. The aim of the second experiment was to test the how well the recurrent neural network could perform on previously unseen data. Again shown in the results the recurrent neural network provided the best prediction accuracy. The third experiment conducted examined how far into the future the recurrent network could predict with a high degree of accuracy. The results indicate that recurrent neural network can produce a reasonable degree of accuracy when predicting multiple time steps into the future. Forecasting multi-time steps ahead for cloud resource has proven to be a difficult area in time series research. The recurrent neural network presented in this research could potentially be integrated with many areas of cloud computing such as host migration and VM scheduling to improve overall performance. For instance, research has shown that instantiating a new virtual machine takes between 5-15 minutes [17]. The results presented in this paper demonstrate that recurrent neural networks are capable of predicting CPU utilisation 15 minutes into the future and still retain a relatively high degree of accuracy. The recurrent neural network could inform the cloud management system when a host is going to become over-utilized so appropriate actions such as live migration or boot up a new VM instances potentially could be initiated prior to a host becoming over-utilized. Power consumption is another key area in cloud computing where accurate forecasting algorithms could enhance the performance of cloud data centre. Gartner et al. highlighted that the ICT industry contributed to about 2% of global CO_2 emitted each year, aligning itself on the same level with the aviation industry [16] Koomey has stated that in 2010 1.3% of all power consumed worldwide was due to data centre usage [20] and this was increasing each year. Cloud data centres integrating more advanced AI and ML algorithm could reduce energy consumption. Energy is constantly being wasted on a substantial portion of the host machines that operate at 10-50% of their full capacity [4], which in-turn results in a significant increase in energy costs. The results presented by Duy et al. have shown how neural networks can be utilized as a predictor to reduce energy consumption in a data centre to turn off host when the traffic load is light [14]. The results in this paper demonstrate that recurrent neural networks can provide more accurate forecasts, therefore having additional benefits of reducing the overall energy consumption of the data centre. For instance, data centres that have a portion of it host machines operating between 0-10% utilization could be predicted by a recurrent neural network for the next 20-30 minutes and shut down these machines to reduce energy consumption and by extension decrease CO_2 emissions from powering the cloud data centres. Recently, companies such as Google have implemented their own DeepMind neural network tool to reduce their data centre energy cost by 40% [15].

VI. Conclusion

The aim of this research was to investigate if recurrent neural networks are capable of accurately predicting CPU utilization for short time periods. The results from this paper indicate that it is possible to predict CPU utilization with a high degree of accuracy for data sets that have sudden extreme changes. The recurrent neural network train with BPTT was able to accurately predict CPU utilization within 10,000 evaluations of the training data. The recurrent neural network performed best on both training and testing data when compared to tradition prediction methods such as random walk, moving average and backpropagation. Results show however that the prediction of the CPU utilization is a difficult task due to the occasional sudden extreme change in CPU utilization. The results also highlight that the recurrent neural networks prediction accuracy decreases as it predicts further into the future. However, on average the network is capable of predicting with a reasonable level of accuracy 3 steps (15 minutes) into the future. In summary, the contributions of this research are:

1) Recurrent neural networks have the capabilities to accurately predicting noisy host CPU utilization.
2) The Recurrent neural networks produce relatively high accuracy when predicting 15 minutes into the future. The accuracy of the network predictions decreases in a linearly the further into the future the network attempts to predict.

A. Future Work

There are several potential routes for future research that have arisen from this research, which will include using different algorithms such as Long-Short-Term-Memory to train the recurrent neural network to compare and improve the accuracy of the predictions of back-propagation-through-time. Other interesting future work would include predicting other metric such as RAM and disk utilisation of a host machine.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[2] E. Barrett, E. Howley, and J. Duggan. A learning architecture for scheduling workflow applications in the cloud. In *Web Services (ECOWS), 2011 Ninth IEEE European Conference on*, pages 83–90. IEEE, 2011.

[3] E. Barrett, E. Howley, and J. Duggan. Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. *Concurrency and Computation: Practice and Experience*, 25(12):1656–1674, 2013.

[4] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12), 2007.

[5] T. Benson, A. Anand, A. Akella, and M. Zhang. Microte: Fine grained traffic engineering for data centers. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, page 8. ACM, 2011.

[6] K. B. Bey, F. Benhammadi, A. Mokhtari, and Z. Guessoum. Cpu load prediction model for distributed computing. In *Parallel and Distributed Computing, 2009. ISPDC'09. Eighth International Symposium on*, pages 39–45. IEEE, 2009.

[7] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[8] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, 2011.

[9] P. A. Dinda and D. R. O'hallaron. Host load prediction using linear models. *Cluster Computing*, 3(4):265–280, 2000.

[10] M. Duggan, J. Duggan, E. Howley, and E. Barrett. A reinforcement learning approach for the scheduling of live migration from under utilised hosts. *Memetic Computing.*, Volume 8.(DOI 10.1007/s12293-016-0218-x.):pp 111., 09 December 2016.

[11] M. Duggan, J. Duggan, E. Howley, and E. Barrett. An autonomous network aware vm migration strategy in cloud data centres. In *Cloud and Autonomic Computing (ICCAC), 2016 International Conference on*, pages 24–32. IEEE, 2016.

[12] M. Duggan, J. Duggan, E. Howley, and E. Barrett. A reinforcement learning approach for dynamic selection of virtual machines in cloud data centres. In *Conference: Innovative Computing Technology (INTECH 2016)*, 2016.

[13] M. Duggan, J. Duggan, E. Howley, and E. Barrett. A network aware approach for the scheduling of virtual machine migration during peak loads. *Cluster Computing*, pages 1–12, 2017.

[14] T. V. T. Duy, Y. Sato, and Y. Inoguchi. Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1–8. IEEE, 2010.

[15] J. Gao and R. Jamidar. Machine learning applications for data center optimization. *Google White Paper*, 2014.

[16] G. Inc. Gartner estimates ict industry accounts for 2 percent of global co2 emissions. In *http://www.gartner.com/newsroom/id/503867*, 2007.

[17] S. Islam, J. Keung, K. Lee, and A. Liu. Empirical prediction models for adaptive resource provisioning in the cloud. *Future Generation Computer Systems*, 28(1):155–162, 2012.

[18] B. Joshi, A. S. Vijayan, and B. K. Joshi. Securing cloud computing environment against ddos attacks. In *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, pages 1–5. IEEE, 2012.

[19] M. Khashei, M. Bijari, and S. R. Hejazi. Combining seasonal arima models with computational intelligence techniques for time series forecasting. *Soft Computing*, 16(6):1091–1105, 2012.

[20] J. G. Koomey. *Estimating total power consumption by servers in the US and the world*. February, 2007.

[21] K. Mason, J. Duggan, and E. Howley. Evolving multi-objective neural networks using differential evolution for dynamic economic emission dispatch. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1287–1294. ACM, 2017.

[22] K. Mason, J. Duggan, and E. Howley. Neural network topology and weight optimization through neuro differential evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 213–214. ACM, 2017.

[23] K. Mason, J. Duggan, and E. Howley. A meta optimisation analysis of particle swarm optimisation velocity update equations for watershed management learning. *Applied Soft Computing*, 2017 (In Press).

[24] J. J. Prevost, K. Nagothu, B. Kelley, and M. Jamshidi. Prediction of cloud data center networks loads using stochastic and neural models. In *System of Systems Engineering (SoSE), 2011 6th International Conference on*, pages 276–281. IEEE, 2011.

[25] R. Shaw, E. Howley, and E. Barrett. Predicting the available bandwidth on intra cloud network links for deadline constrained workflow scheduling in public clouds, 01 2017.

[26] K. Vieira, A. Schulter, C. Westphall, and C. Westphall. Intrusion detection techniques in grid and cloud computing environment. *IT Professional, IEEE Computer Society*, 12(4):38–43, 2010.

[27] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[28] Y. Zhang, W. Sun, and Y. Inoguchi. Predict task running time in grid environments based on cpu load predictions. *Future Generation Computer Systems*, 24(6):489–497, 2008.