# Cloud Storage Prediction with Neural Networks

Stefan Frey, Simon Disch, Christoph Reich, Martin Knahl
Furtwangen University
Cloud Research Lab
Furtwangen, Germany
{Stefan.Frey, Ch.Reich, Martin.Knahl}@hs-furtwangen.de

Nathan Clarke
Plymouth University
Centre for Security, Communication and Network Research
Plymouth, United Kingdom
cscan@plymouth.ac.uk

***Abstract*—In this work, we present an Artificial Neural Network approach to predict the usage, size and type of a cloud storage to enable better compliance with Service Level Agreements (SLAs). One of the biggest advantage of cloud infrastructures is scalability on demand. Cloud services are monitored and based on utilization and performance need, they get scaled up or down, by provision or deprovision of resources. The goal of the presented approach is to predict and thereof select the right amount of storage with a minimum of preallocated resources, as well as the corresponding storage type based on the predicted performance needs in order to reduce SLA violations. Evaluation of the results obtained by simulation confirm that, by using this approach, SLA violations decreased compared to a threshold value control system.**

***Keywords–Cloud Computing, Storage, Prediction, Neural Networks, SLA, QoS***

## I. INTRODUCTION

After an initial hype, cloud computing has established itself as an adequate means of providing resources on demand on a self-service basis and gives customers access to a large pool of computational power and storage. With cloud computing, customers do not have to manage and maintain their own Information Technology (IT) assets and are not bound to their locally limited resources. In order for both customers and providers to be confident that their cloud services are usable at an adequate level, Quality of Service (QoS) guarantees are needed [1]. For this, service requirements stated in Service Level Agreements (SLAs) need to be monitored and the corresponding resources need to be managed. Currently, cloud providers typically support very simple metrics such as availability, or global best effort guarantees.

In cloud systems, resources are being provided dynamically, which means the quality of a service can be directly dependent on the provisioning mechanism [2]. In order to improve the QoS for cloud computing services QoS monitoring, provisioning strategies, as well as detection and prediction of possible SLA violations must be investigated. In this paper, an approach is proposed to regulate cloud storage through the use of Artificial Neural Networks (ANN). Artificial Neural Networks are computational structures modeled after the biological processes of the brain. According to the Defense Advanced Research Projects Agency (DARPA) Neural Networks are systems composed of many simple processing elements operating in parallel whose function is determined by the network structure, connection strengths, and the processing performed at the elements or nodes [3]. Neural networks have been successfully used for decision support systems and show high potential for the use in forecasting and prediction systems [4]. If one could predict the usage of a service, looking ahead further than the provisioning delay time, one could guarantee the QoS for that specific service. The approach presented in this paper aims to improve SLA compliance through the prediction of cloud storage usage. This addresses particularly the storage allocation and dynamic storage capacity guarantees specified in SLAs.

The remainder of the paper is organized as follows. In Section II, the related research efforts are discussed. Section III presents the cloud QoS model and external factors. In Section IV, the specific approach neural networks for controlling the storage of cloud services is introduced. The proof of concept is reported in Section V. Finally, a conclusion is drawn and future work is suggested in Section VI.

## II. RELATED WORK

Neural Networks are widely used in forecasting problems. One of the earliest successful application of ANNs in forecasting is reported by Lapedes and Farber [5]. They used a feedfoward neural network with deterministic chaotic time series generated by the Glass-Mackey equation, to predict such dynamic nonlinear systems.

Atrificial Neural Networks are proven universal approximators [6][7] and are able to forecast both linear [8] and nonlinear time series [9]. Adya and Collopy investigated in the effectiveness of Neural Networks (NN) for forecasting and prediction [4]. They came to the conclusion that NN are well suited for the use of prediction, but need to be validated against a simple and well-accepted alternative method to show the direct value of this approach. Since forecasting problems are common to many different disciplines and diverse fields of research, it is very hard to be aware of all the work done in this area. Some examples are forecasting applications such as: temperature and weather [10][11][12], tourism [13], electricity load [14][15], financial and economics [16][17][18][19] and medical [20][21] to name a few. Zhang, Patuwo, and Hu [9] show multiple other fields where prediction by ANN was successfully implemented.

## III. CLOUD STORAGE QOS

SLAs specify the expected performance characteristics between service providers and customers. The most important component of an SLA is the exact description of the service quality (service levels). These descriptions are called Service Level Objectives (SLOs), which contain Key Performance Indicators (KPIs) consisting of metrics and the specific value

 ISBN: 978-1-61208-388-9 

to be guaranteed. These metrics are constantly monitored and the SLOs are guaranteed over a relatively long time interval. If the guaranteed service levels are not met, the SLA is violated and penalty costs may have to be paid to the customer by the provider. For storage, typical KPIs stated in an SLA can be the read- and write-speed, storage capacity, random input/outputs per second (IOPS) and bandwidth. Since cloud computing resources can be allocated dynamically at runtime additional, dynamic service level objectives arise. For example, this could comprise a constant growth of the storage capacity or the compliance with a certain maximum deployment time or guarantee a constant minimum of available free memory.

Cloud storage resources are usually multi tenant, which means for the provider that it can economically be very important to distribute the storage as efficiently as possible. This means allocating as close to the minimum guaranteed amount of storage as possible. In practice, this can lead to problems because the memory usage of clients can vary greatly and therefore SLA violations can happen easily. For this, a method shall be found that allows to determine the needed amount of memory close to the optimum and allocate it ahead of time. With such an efficient provisioning method it would be possible for providers to maximize the usability of their infrastructure while at the same time guarantee customers a high quality service.

## IV. Artificial Neural Network

The aim of this work was to create a prototype application which enables efficient provisioning of cloud storage resources with the use of Artificial Neural Networks to achieve better compliance with SLAs. The most common type of ANNs used for forecasting is the feedforward multilayer perceptron (ffMLP), as seen in Figure 1.

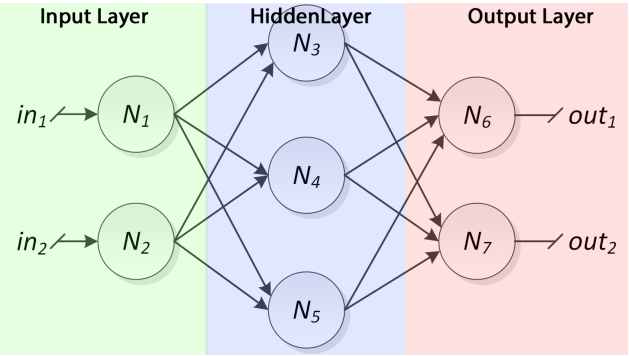


Figure 1. Simple 3-tier Feedforward Multilayer Perceptron.

These are Neural Networks, which consist of one input layer, $n$-hidden processing layers and one output layer. Feedforward networks are classified by each neuron in one layer having only direct connections to the neurons of the next layer, which means they have no feedback. In feedforward multilayer perceptrons, a neuron is often connected to all neurons of the next layer, which is called completely linked. So, there is no direct or indirect connection path from neuron $N_x$ which leads back to a neuron $N_{x-z}$. To compute a one-step-ahead forecast, these NNs are using lagged observations inputs of time series or other explanatory variables.

For the creation of the Neural Network model we used the graphical editor and simulator MemBrain [22]. The presented Neural Network consists of 119 neurons, which are aligned into 5 layers, and corresponds to a ffMLP where not all neurons are completely linked. An architectural overview of the presented model is shown in Figure 2 below.

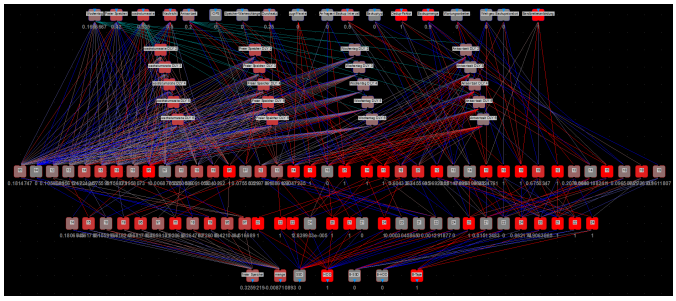

Figure 2. Feedforward Multilayer Perceptron Architecture.

Training of ANNs can be seen as a complex nonlinear optimization problem, and sometimes the network can get trapped into a local minimum. ANNs can theoretically learn by developing new or deleting existing connections, changing the connection weights or threshold values, altering one or more of the three neuron functions (activation, propagation and output) and developing new or deleting existing neurons. In order to improve outputs, the input neurons should get normalized variables. This can simply be done by the equation below.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

In order to avoid local minima and bad results, the training should be initialized several times with different starting weights and alignments. For the training of the proposed model, data sets were created in the form of Comma Separated Value (CSV) files. Each file contains storage usage patters with input and output vectors. Here, 60% of the samples were used for training and the remaining 40% were used for the validation of the network. The output behavior was modeled by depending on a input vector, where the desired output values where manually entered into the input vector. Thus, situations in which the responsible output neuron shall increase the amount of allocated memory were mapped.

To teach the network the prediction capability of future memory usage, the input vector was extended. The entire course of the used memory amount was added for the period of $t_0$ to $t_n$. The desired output for this input vector at the given time $t_i$ shall be the predicted amount of memory used at time $t_{i+x}$. To achieve this, the value of the output vector at any point $t_i$ in the time period $t_0$ to $t_n$ was set to the input vector of the point $t_{i+x}$, by which $x$ determines the length of the forecast period. Through this shift in values the network can be trained for a prognosis. During each training session the network error was checked with validation data. MemBrain calculates this using the following formula:

$$NetError = \frac{\sum_{i=1}^{n} (Target_i - Output_i)^2}{n} \tag{2}$$

 ISBN: 978-1-61208-388-9

The desired activation of the output neurons is here referred to as $Target$ and the actual calculated activation is the $Output$. The squared deviations are summed and divided by the number of training data sets. To determine whether the Neural Network shows good results of the output behavior, it has been trained and validated with 10 different training data sets. The result for the network error after each learning processes is shown in Table I below.

TABLE I. INFRASTRUCTURE SENSOR PARAMETER.

| TrainingNr. | NetError(Training) | NetError(Validation) |
|---|---|---|
| 1 | 0,0000573 | 0,046 |
| 2 | 0,0000586 | 0,040 |
| 3 | 0,0000713 | 0,040 |
| 4 | 0,0000702 | 0,112 |
| 5 | 0,0000611 | 0,040 |
| 6 | 0,0000783 | 0,083 |
| 7 | 0,0000703 | 0,046 |
| 8 | 0,0000627 | 0,038 |
| 9 | 0,0000645 | 0,061 |
| 10 | 0,0000630 | 0,046 |

Here, it can be seen that the NetError reaches overall good values close to zero and not only for a particular dataset. The average total error for all training runs from Table I is 0.0000657 for trained and 0.0573 for untrained (unknown) input data.

## V. EVALUATION

The aim of this work was to investigate, whether or not the use of a Artificial Neural Network for the provisioning of a cloud storage resources has a positive effect on SLAs compliance, and whether this can lead to a better resource utilization compared to a classic threshold value system. For this purpose we created a simulation environment where storage requests (read, write, and delete) form a generator were sent trough a QoS monitor. Inside the QoS control module, the Artificial Neural Network and the threshold value system were used to regulate the amount of allocated storage capacity. Figure 3 shows the architectural overview of the simulation environment.

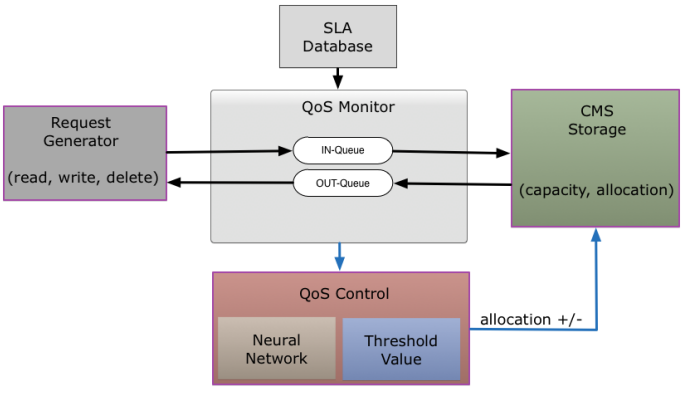


Figure 3. Simulation Architecture.

In the simulation, the impact of regulatory mechanisms on the following key performance indicators was considered:

- Free memory amount: providing an optimal amount of memory by the control logic.
- Response time: compliance with the KPI response time by adjusting the storage medium.
- Backup Media: proposal of a suitable backup medium.

For this, the used Neural Network consisted of 11 different input neurons. Table II lists the used input neurons and describes the used input factors. As output neurons, there is one neuron that gives the expected used memory amount for the next simulation step, a neuron that determines the amount of memory to be added or removed, as well as other neurons that recommend the optimal backup medium.

TABLE II. SIMULATION INPUT NEURONS.

| Neuron | Description |
|---|---|
| Time | Point $t_i$ in $t_0 \ldots t_n$ |
| Weekday | Day of week for point $t_i$ |
| Free Storage Capacity | Free storage capacity at point $t_i$ |
| Growth Rate | Change of capacity from $t_{i-1}$ to $t_i$ |
| Response time | Mean response of last 5 inputs $\frac{\sum_{i=i-5}^{i} t_i}{n}$ |
| Queue Length | Still open request at point $t_i$ |
| Troughput | Troughput at point $t_i$ |
| Access Rate | Amount of requests per time slot |
| Request Type | Distinction between large and small requests |
| Backup Amount | Size of backup data |
| Bandwidth | Usable bandwidth at point $t_i$ |

In order to compare the results of the Neural Network with a common, in practice widely used method, a threshold value based scaling was implemented. This regulation system is controlled by predefined thresholds for the monitored KPI values. The implementation for the threshold rules for adding and removing allocated storage can be seen below in Figure 4, as simple pseudocode if then rules.

```
// addStorage
IF AllocatedCapacity      UsedCapacity < SLAFreeCapacity +2
THEN
        IF (UsedCapacity < 20)     AllocatedCapacity += 10;
        ELSE IF (UsedCapacity > 80)     AllocatedCapacity += 20;
        ELSE    AllocatedCapacity += 15;

// removeStorage
IF AllocatedCapacity      UsedCapacity > SLAFreeCapacity +15
THEN
        IF (UsedCapacity < 20)     AllocatedCapacity -= 20;
        ELSE IF (UsedCapacity > 80)     AllocatedCapacity -= 10;
        ELSE    AllocatedCapacity -= 15;
```

Figure 4. IF THEN rules for threshold system.

Here, it can be seen that, by falling below a 2% buffer of the storage value defined in the SLA, the allocation will be increased and by exceeding 15% over the amount of storage defined in the SLA, the allocation will be lowered. The amount of which the allocated storage will be changed is dependent on how much the overall storage usage is. In case of an usage of over 80 %) increase will be 20%, with an usage of below 20% the increase will be 10% and in between the increase is 15 % of the overall volume. These settings are reversed for the deallocation of the storage.

For the scenario in this simulation, a dynamic storage SLA, in which a customer gets granted 10GB of free space and up to

100GB of overall usage, was assumed. With such a dynamic limit described in the SLA, it is particularly important for the provider to find a solution that is as close as possible to the guaranteed amount of storage, since this will ensure a high economic efficiency. In practice, however, this usually is not possible. For this reason and because a violation of the SLAs can have monetary consequences, bigger buffer zones are installed. Figure 5 shows the resulting graph of the simulation with the conventional threshold value rules.

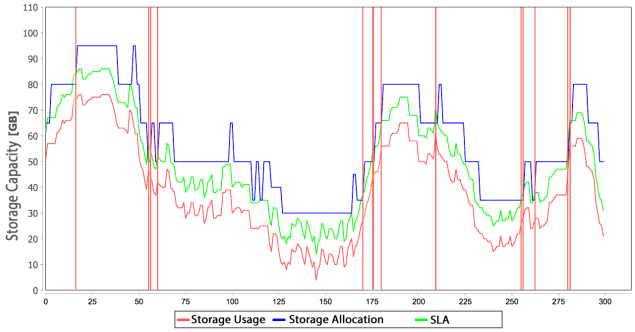


Figure 5. Storage allocation results for threshold rules.

The red graph in Figure 5 and Figure 6 shows the course of the memory usage in GB, by the user during the simulation. The usage has been pre-generated for the simulation purpose and shall resemble a system, where a user regularly creates and deletes files with up to 15GB size, as well as generate larger files with up to 50GB. This type of usage may occur while working with different media files, like in the post-processing of movie projects. The green line marks the guaranteed amount of storage available to the user, granted by the SLA. It proceeds synchronous to the red graph, since the user gets guaranteed 10GB more than they currently use. The blue graph shows the pre-allocated amount of storage, which is directly usable by the user.

If we compare these results with those obtained by the Neural Network controlled storage allocation, shown in Figure 6, it becomes clear that the efficiency is marginally improved. With an average of 18.68% of memory over provisioned the threshold value system is almost as effective as the neural network, with a 18.22% overhead. The slight difference arises from the fact that the allocation offered by constantly adopting, fits to the SLA limits with a relatively constant overhead. In contrast, the threshold value system initially provides too much memory, and then only adopts the amount of allocated storage shortly before a violation of the SLA it to occur.

While comparing the two graphs, we see that the threshold system due to the fixed thresholds less often adjusts the amount of memory (blue curve). Since the added / removed amount of memory operates with a fixed predefined value, often too much memory is provided and then immediately gets removed again. This happens likewise when reducing the amount of memory allocated, which often leads to falling below the specified minimum amount in the SLA. However, the Neural Network determines constantly, based on the learned training data, a variable amount of memory that is to be added or removed, which leads to adequate reactions and a slightly better economic result.

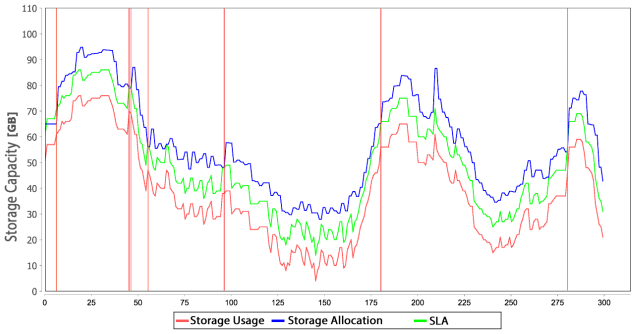


Figure 6. Storage allocation results for NN.

However, when comparing the number of SLA violations, it becomes clear that the Neural Network approach delivers a significantly better solution. This is also evident in the resulted graph seen in Figures 6 and 5, where the SLA violations are indicated by vertical red lines. These exemplary results of the simulation show that the Neural Network produces 7 and the threshold value system 13 SLA violations. These results were also confirmed within the other test runs, where the Neural Network generates an average of 7.45 violations per run and the threshold values system of 15.03 SLA violations per run.

Overall, the Neural Network generated solution for the provisioning of storage is better suited, since the number of SLA violations is significantly lower. Together with the slightly lower overhead makes this a reasonably good solution.

## VI. Conclusion and Future Work

The aim of this paper was to investigate whether QoS parameters of a cloud computing storage could be more effectively managed using the predictive capabilities of neural networks. In particular, this study sought to improve the overhead amount of pre-allocated storage and reduce SLA violations. For this, a feedforward multilayer perceptron Artificial Neural Network was presented and its structure and functionality has been delineated. As proof of concept, several tests were performed in order to prove the effectiveness of the approach compared to a threshold value system.

It was found that the over-provisioning of the allocated storage amount could be improved by 0.46% with the ANN prediction. Here improvements with respect to an optimization of the provisioning amount should be carried out. In terms of the SLA compliance, the presented approach significantly won over the threshold value system with nearly halve as many violations. These results shed a positive light on the presented approach, which could lead to an increase in efficiency and economics of cloud storage.

Future work will seek to evaluate this approach within a real cloud environment and with real life differentiating user work loads. Research will also investigate other QoS parameters to understand and improve upon the prediction of cloud storage. Furthermore, the one-step-ahead prediction capability of the used Neural Network should be stretched ahead further into the future in order to improve the forecast and adaptability.

 ISBN: 978-1-61208-388-9

Finally, a deeper evaluation against other prediction methods (e.g Bayesian or Markov Models) is needed, to determine whether or not ANN present the best approach.

## ACKNOWLEDGMENT

This research is supported by the German Federal Ministry of Education and Research (BMBF) through the research grant number 03FH046PX2.

## REFERENCES

[1] P. Patel, A. Ranabahu, and A. Sheth, "Service level agreement in cloud computing," UKPEW, 2009.

[2] R. Calheiros, R. Ranjan, and R. Buyya, "Virtual machine provisioning based on analytical performance and qos in cloud computing environments," in Parallel Processing (ICPP), 2011 International Conference on, Sept 2011, pp. 295–304.

[3] DARPA Neural Network Study (U.S.), DARPA Neural Network Stdy, Widrow, Morrow, and Gschwendtner, Eds. AFCEA Intl, 1988.

[4] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? a review and evaluation," Journal of Forecasting - Special Issue: Neural Networks and Financial Economics, vol. 17, no. 5-6, September 1998, pp. 481–495.

[5] A. Lapedes and R. Farber, "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modelling," Los Alamos National Laboratory, Los Alamos, NM, Tech. Rep. LA-UR-87-2662, 1987.

[6] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, no. 5, 1989, pp. 359–366.

[7] K. Hornik, "Approximation capabilities of multilayer feedforward networks," Neural Networks, vol. 4, no. 2, 1991, pp. 251–257.

[8] G. P. Zhang, "An investigation of neural networks for linear time-series forecasting," Computers and Operations Research, vol. 28, no. 12, 2001, pp. 1183–1202.

[9] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," International Journal of Forecasting, vol. 14, no. 1, 1998, pp. 35–62.

[10] G. Langella, A. Basile, A. Bonfante, and F. Terribile, "High-resolution spacetime rainfall analysis using integrated {ANN} inference systems," Journal of Hydrology, vol. 387, no. 34, 2010, pp. 328–342.

[11] J. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," Power Systems, IEEE Transactions on, vol. 17, no. 3, Aug 2002, pp. 626–632.

[12] P. J. Roebber, M. R. Butt, S. J. Reinke, and T. J. Grafenauer, "Real-time forecasting of snowfall using a neural network," Weather and Forecasting, vol. 22, no. 3, 2014/07/15 2007, pp. 676–684.

[13] D. C. Pattie and J. Snyder, "Using a neural network to forecast visitor behavior," Annals of Tourism Research, vol. 23, no. 1, 1996, pp. 151–164.

[14] D. Park, M. El-Sharkawi, I. Marks, R.J., L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," Power Systems, IEEE Transactions on, vol. 6, no. 2, May 1991, pp. 442–449.

[15] H. Hippert, C. Pedreira, and R. Souza, "Neural networks for short-term load forecasting: a review and evaluation," Power Systems, IEEE Transactions on, vol. 16, no. 1, Feb 2001, pp. 44–55.

[16] Y. Bodyanskiy and S. Popov, "Neural network approach to forecasting of quasiperiodic financial time series," European Journal of Operational Research, vol. 175, no. 3, 2006, pp. 1357–1366.

[17] P. McAdam and P. McNelis, "Forecasting inflation with thick models and neural networks," Economic Modelling, vol. 22, no. 5, 2005, pp. 848–867.

[18] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," Neurocomputing, vol. 10, no. 3, 1996, pp. 215–236, financial Applications, Part II.

[19] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," Expert Systems with Applications, vol. 38, no. 8, 2011, pp. 10 389–10 397.

[20] A. M. Vukicevic, G. R. Jovicic, M. M. Stojadinovic, R. I. Prelevic, and N. D. Filipovic, "Evolutionary assembled neural networks for making medical decisions with minimal regret: Application for predicting advanced bladder cancer outcome," Expert Systems with Applications, 2014.

[21] C. Arizmendi, D. A. Sierra, A. Vellido, and E. Romero, "Automated classification of brain tumours from short echo time in vivo mrs data using gaussian decomposition and bayesian neural networks," Expert Systems with Applications, vol. 41, no. 11, 2014, pp. 5296–5307.

[22] T. Jetter, "Membrain neural network editor and simulator," [Online] Available: http://www.membrain-nn.de Retrived: 9 August 2014.

 ISBN: 978-1-61208-388-9 