University of Wollongong

Evaluation of Neural Network Models for Australian Energy Market Operators Five Minute Electricity Demand Forecasting

– *Prepared for* –

Australian Energy Market Commission (AEMC)
PO Box A2449
Sydney South, NSW 1235

– *Prepared by* –

A. Prof. Markus Hagenbuchner and Em. Prof. Ah Chung Tsoi
University of Wollongong
1 Northfields Avenue
Wollongong, NSW 2500

13 December 2016

Version: Final Report

Contents

List of Figures

Glossary

AEMO The Australian Energy Market Operator is a body corporate that is responsible for the administration and operation of the wholesale national electricity market in accordance with Australia's Electricity Code.

AI Artificial Intelligence encompasses all computational methods that can either reason when equipped with an internal logic obtained from observations on how the system behaves or learn from its past historical record of inputs and outputs. ANNs (defined below) are an example of algorithms that can learn whereas Expert Systems (not relevant in this report) are an example of algorithms that can reason.

ANN Artificial Neural Networks aim at emulating the behaviour of biological neurons or biological neural assemblies (consisting of a few biological neurons connected together to perform some elementary behaviours) in the mammalian brain. A more detailed background on ANNs is provided in Section 3.1 on page 4.

DNN Deep Neural Networks are MLPs (defined below) that consist of numerous hidden layers.

FRPN Fully Recursive Perceptron Networks are a recent development which improve on the prediction capabilities of MLPs while at the same time are more scalable than MLP or Deep Learning systems.

ML Machine Learning encompasses the general class of algorithms that enable computers to learn from past historical data.

MLP A Multilayer Perceptron is a neural network model based on artificial neurons that are arranged in layers. A more detailed description is provided in Section 3.1.2 on page 5.

SOM Self Organizing Map, an artificial neural network model where artificial neurons are arranged on an n-dimensional grid, with $n = 2$ most commonly. This is often used for the projection of high dimensional data to one with lower dimensions, with grid points being represented by artificial neurons.

SOES The Self-Organizing Ensemble System is a neural network ensemble method consisting of a SOM and a set of base predictors for electricity demand prediction. of a prediction system.

RNN Recursive Neural networks are able to model higher order dependencies such as the dependency of a demand at a given time on the demand one week prior, one month prior, one year prior, etc. RNNs are described in Section 4.

1 Executive Summary

The key findings in this report are:

1. The current AEMO neural network model is limited in its capability, and should not be considered suitable to accurately perform dispatch demand forecast without considering more modern approaches.
2. Significant advances in Machine Learning research introduced modern learning systems which are more suitable for dispatch demand forecasts.
3. The benefit arising from the proposed rule change would be very limited if AEMO continues to use the current neural network model.
4. AEMO's current neural network model cannot deal with volatility, spikes, shocks, price responses, and any other situations which might require the modelling of context (i.e. seasonal context) for accurate prediction.
5. Neural networks are dominant in electricity demand prediction internationally.
6. A purpose build system which would be able to deal with spikes, shocks, and price responses that is suitable for Australian conditions can be developed.

These key findings are a result of answers provided to issues currently faced by the Australian Energy Market Operator (AEMO):

Issue 1: Is the current AEMO neural network model suitable to perform accurate dispatch demand forecast?
The report investigates the suitability of the current AEMO neural network model in relation to its capability to perform accurate dispatch demand forecast. The report provides strong evidence that **the current AEMO neural network model is not suited to accurately perform dispatch demand forecast**. The reason to why AEMO adopted its current model is due to the lack of suitable alternatives at the time of the model's introduction.

Issue 2: Are there prediction systems that are better suited to perform accurate dispatch demand forecast than the neural network prediction model currently used by AEMO?
The report finds that the type of neural network used by AEMO is a first generation neural network that is over 20 years old. **Significant advances in the area of research in Machine Learning introduced modern learning systems which are much more suitable for dispatch demand forecasts.**

Issue 3: Are there similar applications for which neural network models are used?
The report shows that neural networks are popularly used in data mining and big data. **Neural networks are commonly deployed in prediction applications** (i.e. prediction in financial markets, weather, yield, etc.).

Issue 4: What are the forecasting capabilities of the AEMO's current neural network model?
The report shows that AEMO's current neural network model engages a number of "tricks" to mask the deficiencies of the model. These tricks allow the model to make reasonable

predictions under normal (i.e. frequently observed) conditions. It is demonstrated and explained that **the current model cannot deal with abnormal conditions that arise out of volatility, spikes, shocks, price responses, and any other situation which require the modelling of context for accurate predictions** .

Issue 5: What are possible solutions to the identified problems in AEMO's current model? Much more **appropriate methods have been developed in the years since the adoption of AEMO's current neural network model**. This report presents and explains methods which overcome relevant limitations of the current model. These modern methods are capable of modelling the temporal context for more accurate prediction. Moreover, **a purpose build system is proposed which would be able to deal with spikes, shocks, and price responses**. Such a system would hence negate the need for manual adjustments to compensate for the prediction error of the currently used model. The purpose built system could automate the processes involved for accurate dispatch demand forecast.

This report recommends that the current neural network model be substituted by one based on modern techniques. Several options exist. Two main approaches are:

Approach 1: Adopt a more modern "general purpose" neural network method such as the Elman network or the recursive neural network. Unlike AEMO's current neural network model these neural networks have been designed to work on data sequences and can hence be expected to produce better predictions in time series electricity demand forecast. The advantage of this approach is that it can be deployed relatively quickly and at a low cost. The disadvantage is that the prediction accuracy will not be as good as a purpose build system (see approach 2).

Approach 2: Develop a purpose build prediction system such as one which combines the best of a number of indepdent techniques as they complement one another in their individual shortcomings. The system can be built such that it can produce accurate prediction even in difficult-to-predict cases that arise out of volatility, spikes, shocks, and price responses. The main disadvantage of this approach is the time implication. It would take time to develop, implement, and test the system. Nevertheless, some of the leaders in machine learning research are right here in Australia. This provides the unique opportunity to engage the expertise of local researchers to develop a prediction system that is fit for Australian purposes.

2 Objectives

This report has been prepared in accordance with our engagement letter dated 20/July/2016. The objective is to provide the Australian Energy Market Commission (AEMC) with:

1. an assessment of the current Australian Energy Market Operator's (AEMO)'s Neural Network Model in relation to its capabilities to perform accurate generator supply forecast and accurate dispatch demand forecast.
2. an overview of the development and latest advances in Neural Network Models and relevant forecasting models.
3. an overview of the applications of Neural Network Model to forecast dynamic, fluctuation entities which might exhibit seasonality, pseudo-periodicity, and trend behaviours, in both Australia and internationally.
4. the international experience of energy market forecasting and lessons in which Australia can draw on to improve its current forecasting accuracy.
5. a high level description of possible solutions to the identified problems in AEMO's current model.

3 Background

AEMO is the National Energy Market Operator and planner. AEMO operates the Australia's National Electricity Market (NEM), and one of its key functions is to conduct a centrally-coordinated dispatch process that pools (aggregates) geographically distributed heterogeneous generators from producers and delivers required quantities of electricity from the aggregated pool to wholesale consumers which are geographically distributed. The dispatch process requires a bid stack with information provided by market participants (representing the supply side information) and a demand forecast prepared by AEMO (representing the demand side information) to dispatch generators and loads every five-minutes to balance the supply and demand of the electricity market in real-time.

Not all generators and loads are required to submit bids into the central dispatch process. The obligation to submit bids depends on whether they are scheduled, semi-scheduled or non-scheduled. Scheduled and semi-scheduled market participants need to provide AEMO with bids which are used to create a bid stack representing the known supply and demand intentions of these participants. AEMO separately forecasts the demand and supply of all participants who are not scheduled. Supply from non-scheduled generators is treated currently as negative demand.

AEMO uses an Artificial Neural Network model to forecast five-minute ahead electricity demand and performs adjustments to minimize the impact of prediction errors and to address known deficiencies of the current model. AEMO's model is based on a Neural Network model developed by Professor David Edelman for TRANSGRID in 1997. For simplicity, we will refer to this neural network model as the Edelman model [1].

[1] Relevant information on this model can be obtained from AEMO's web site via the following link: https://www.aemo.com.au/-/media/Files/PDF/SO_FD_01__Five_Minute_Electricity_Demand_Forecasting-_Neural_Network_Documentation.ashx

Artificial Neural Networks (ANN) has been an active area of research for many years albeit research efforts have accelerated in recent years due to significant advances in the design of scalable and intelligent solutions for solving numerous data mining and big data applications. ANN concerns the design of algorithms that enable computers to learn from data. The topic can be considered as part of machine learning research which, in turn, is a subdivision of the wider area of Artificial Intelligence (AI) research.

Artificial Neural Networks had been a mainstay technique for forecasting and control of electricity generation and supply systems since their early popularization in the late 1980s. It was shown by a number of researchers, that a neural network based forecasting model could, if properly constructed and trained, outperform the then popular forecasting model of autoregressive moving average model, or autoregressive model. Professor David Edelman's work with TRANSGRID in the 1990s can be considered as an application of the neural network model to forecasting electricity. That was prior to the creation and development of an energy spot market system in Australia. Professor Edelman's model subsequently was developed and adapted to be deployed as the forecasting method in AEMO's system. Todate, the competitive electricity market is a reality, while the Edelman model was developed prior to the emergence of the national electricity market and prior to the uptake of large scale and distributed renewable energy. It is therefore important to evaluate whether:

1. the Edelman model is still "fit for purpose", and
2. to evaluate the suitability of more modern approaches for providing an improved forecast.

The following provides a background on ANNs to aid the understanding of subsequent sections in this report. A historical background and an insight into the workings of the Edelman Network is also provided.

3.1 Background on ANNs

Artificial Neural Networks (ANNs) are computer algorithms which are designed to simulate learning capabilities of the mammalian biological neural systems (human brains). Such algorithms differ significantly from conventional computer algorithms. While conventional algorithms are designed to implement a specific functionality, ANN algorithms are designed to learn the functionality from data. A good introduction to ANN can be found in [12].

In general, ANNs are suitable for problems that do not have algorithmic solutions, do not provide a sufficient solution at all times, or do not scale with the availability or more data.

ANN is a very active area of research within the broader field of Machine Learning (ML). ML encompasses ANNs as well as other methods, so called non-parametric learning methods, that do not aim at simulating neural systems [2].

The class of ANNs and the class of non-parametric learning systems often produce results of comparable accuracy in many application areas. A main advantage of ANN is that they simulate a massively parallel system which can be considered as a physical model of the mammalian brain: there are numerous small processing units, called neurons. These neurons exchange information via interconnecting links (via dendrites and synapses) but generally each neuron works at its own pace and in parallel. Since ANNs simulate these processing units and hence the corresponding algorithms can be implemented very efficiently on modern multi-core processing units such as

[2]Some well-known examples of non-parametric learning methods are Support Vector Machines, Bayesian networks, Decision Trees, Association Rule Learning, and others.

CPUs (central processing units) and GPUs (graphics processing units). This renders ANNs scalable and hence, ANNs are often the approach of choice in many data mining and big data applications.

ANNs are being developed for systems that require a degree of intelligence. Research in ANNs have made great strides in recent years. It is generally recognized that we are witness to an era at which computer implemented learning methods surpass human abilities and consequently there is significant investment by industry and government in AI.

3.1.1 A Brief History of ANNs

The first attempt to simulating learning abilities of neural systems was made by Frank Rosenblatt in 1957. Rosenblatt introduced the concept of perceptron [3] networks. Rosenblatt's work was merely a proof-of-concept as its learning abilities were very limited. While work on developing ANNs started early it was not until 1985 when a work by David Rumelhart and Jay McClelland started a research frenzy. Rumelhart and McClelland extended the concept of a perceptron network and introduced the concept of a multilayer perceptron (MLP) neural network model. The significance of their work arises out of a formal proof which states that an MLP with a single hidden layer with non-linear activation functions is a universal approximator. In layman terms this means that MLPs are proven to be suitable for a very large class of learning problems.

MLPs found wide spread applications and are a de-facto standard in off-the-shelf data mining and machine learning software packages. The ANN model deployed by Professor David Edelman for the purpose of predicting demand in the Australian energy market is in fact an MLP. The Edelman network is a special instance of an MLP and which has been designed and trained for the purpose of 5-minute ahead demand prediction.

3.1.2 How MLPs work

MLPs consist of (simulated, artificial) neurons that are organized in layers and which are connected via weighted links. An MLP consists of at least two layers:

1. The input layer.
2. The output layer.

In addition, MLPs can have an arbitrary number of hidden layers between the input and output layers. These intermediate layers are called hidden because they do not have direct connections to either the inputs or the outputs. They are internal variables to the MLP, not directly accessible either from the input end or from the output end. It is assumed that these internal variables are arranged in layers, with the inputs to the current data being taken from the outputs of the previous layer, and the outputs of the current layer form the inputs of the succeeding layer. No cross layer connections are allowed, and no feedback from previous layers are allowed. In other words, it is assumed that the signals travel in one direction only, from the input end to the output end. An example of an MLP is shown in Figure 1. The MLP in Figure 1 can accept a four-dimensional input and produces a one-dimensional output. The input dimension and output dimension can be arbitrary and these dimensionality are generally aligned with the learning problem at hand.

[3] Rosenblatt called neurons *perceptrons* to avoid confusion with biological neurons although the terms are used interchangeably in ANN literature.

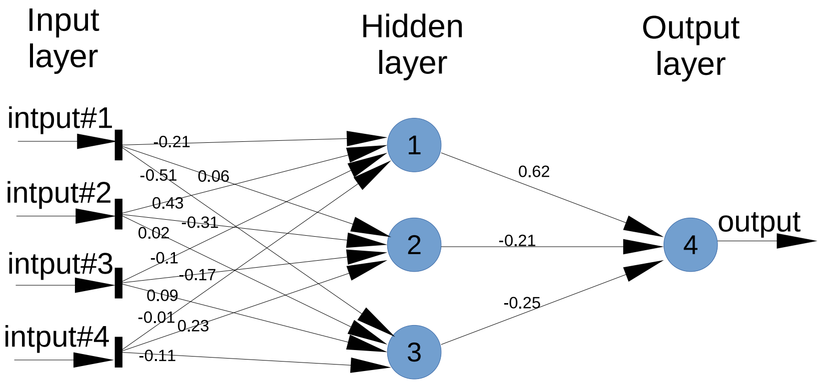

Figure 1: Example: An MLP consisting of a four dimensional input layer, four internal (hidden) neurons and one output neuron. The neurons are connected via weighted links.

Each neuron receives a weighted sum of variables from the previous layer, as input. For example, the neuron numbered 1 in Figure 1 would receive as input: the value of input#1 multiplied by the size of the weight of the link connecting input#1 with neuron 1 plus the value of input#2 multiplied by the size of the weight of the link that connects input #2 value with neuron 1 plus the value of input #3 multiplied by the size of the weight of the link that connects input #3 with neuron 1 plus the value of input #4 multiplied by the size of the weight of the link that connects input #4 with neuron 1. The same procedure is repeated for neuron 2 and neuron 3. Since there is a different set of links connecting the input layer with these neurons and hence the weighted sum will be different for each of these neurons. The response of a neuron is computed once the weighted sum has been obtained. The output of a neuron is guided by an activation function [4]. Thus, if the weighted sum of a neuron is x then its output is the response of the activation for x. The output of each of the hidden layer neurons is computed accordingly. If a neurons is in the output layer then its output is the output of the neural network. Note that this system processes information in a strict feed-forward fashion (from the input layer to the output layer). Hence, an MLP is often called a feed-forward neural network.

An MLP is created by defining the number of hidden layers and the number of neurons in each of the hidden layers. Both of these are parameters that can be chosen arbitrarily. The input dimension and the output dimension of the MLP is defined by the learning problem. Once the layers and neurons are defined, weighted links are added such that the input layer is fully connected with the neurons in the hidden layer, and the hidden layer neurons are fully connected with the neurons in the output layer. The completion of this stage defines the *architecture* of the MLP. Once the architecture is defined it remains fixed. This means that the number of neurons and the number of links will not change. When an MLP network is created then initially all the

[4] Common activation functions are either hyperbolic tangent functions or sigmoidal functions. For large positive input values, both the sigmoidal function and the hyperbolic tangent function will give an output approaching 1, while for large negative input values, the sigmoidal function will give an output approaching 0 and the hyperbolic tangent function will have an output approaching -1. It is common to find the hidden layer neurons to have sigmoidal functions, while the output layer neurons could either be hyperbolic tangent function, sigmoidal function, or linear activation function; while the input layer neurons almost always have linear activation functions.

weights in the network contain random values. The purpose of an MLP training algorithm is to adjust the weights in the network such that the network produces a correct response (output) for any given input. The design of such a training algorithm is a main focus in ANN research.

3.1.3 How the Weights in the MLP are Obtained.

A training algorithm changes the weights such that the desired output value is produced for any given input. The most common training algorithm for MLP networks is the Error Back-Propagation Algorithm. The algorithm iteratively adjusts the weights such that the output of the network becomes increasingly similar to the desired output for any given given input. The algorithm requires a set of input-output pairs of training data. Such a set, when used to train an MLP, is called the *training set*. Thus, a training set consists of a number of input samples and for each of the input samples we know the corresponding desired response. Details of the training algorithm is described in literature such as in [12].

Properly trained, an MLP can be expected to produce the correct response to inputs which are drawn from the same domain as the samples in the training set. The MLP cannot be expected to produce correct responses to inputs that are drawn from a different domain.

Note that the MLP algorithm requires a training set of samples for which the desired output is known. Learning algorithms that have such a requirement are called *supervised* learning algorithms. There are ANNs which do not have such requirements. The corresponding learning algorithms are then said to be *unsupervised*. One notable representative of an unsupervised ANN is the Self-Organizing Map.

3.2 Limitations of the MLP

MLPs work well under a number of assumptions:

Assumption 1: The input dimension is fixed. MLPs can only accept fixed sized input vectors. This renders MLPs not useful in applications where the input is of variable length. In time series prediction such as in demand prediction in the electricity market the sequence of historical data is not fixed in size since the sequence of historical data continues to grow in size. In the case of the electricity market, the sequence of historical data grows at the rate of the dispatch interval (one data point is added to the sequence of historical data every 5 minutes).

Assumption 2: Input vectors are independent. MLPs process each input vector independently to other input vectors. The MLP does not have the facility of using information from other input vectors when processing a given input. Thus, an MLP is not able to learn from errors made for other similar input samples. As a consequence, if the MLP produces an error for an input sample then the MLP is prone to repeat the error for similar (or related) input samples.

Assumption 3: All data is from the same domain. The training data and the test data are assumed to be from the same domain or derived from the observations made on the same underlying system or process. MLPs have difficulties in dealing with data from hybrid (different) domains. For example, the MLP currently used by AEMO for predicting energy demand uses as input data from the domain of historical electricity usage data in the 1990s. Its ability to incorporate external data such as weather data or significant event data is very limited since the MLP has no facility of distinguishing the relationship of external data with the domain data.

Assumption 4: The domain is static. After training, the weights in the MLP are fixed. This means that the MLP can only expected to work in domains that were covered by the training data. If the domain changes (i.e. previously unseen patterns emerge over time) then the MLP will have an undefined behaviour which can result in large network errors.

Assumption 5: The function which describes all target values is continuous. An MLP cannot model well discontinuity in the output space. For example, a sudden change in electricity demand (e.g., due to a price spike) cannot be modelled well by an MLP. While in theory this can be addressed by using an infinitely large number of neurons in the hidden layer this is not practical. As a result, the MLP is expected to perform poorly in cases where sudden changes in electricity demand or wholesale price are observed.

These assumptions render the MLP challenging to be used for the task of electricity demand prediction (and electricity pricing prediction) in the Australian energy market because:

1. The input is a temporal sequence of observations. This time sequence is of variable length and continues to grow in length with time.
2. The demand is seasonal, in that the changes in demand at a certain time of the year differ from changes in demand at other time of the year.
3. The prediction at a given time instance depends on previous time instances.
4. Supply and demand data depend on additional factors: Weather, availability of resources and cost of generation, etc. Such relevant external information cannot be included without the loss of some information or modification to the MLP model.
5. The domain is not static as network usage, list of generators initiated to supply the likely load, type of generation, etc. change over time.

Having provided such limitations of the MLP, one needs to hasten to add that this does not mean MLP as a modelling methodology is outdated. Indeed, the very opposite is true. One finds more and more adoption of the MLP model or its modifications in practical systems. It is universally understood that the MLP and its variants can be considered as one of the most natural generalizations of the linear model, e.g., autoregressive model, autoregressive moving average model, in traditional time series modelling techniques. MLPs remain the most widely used in time series modelling or forecasting, like the very short term (5 minute ahead) energy market demand prediction.

3.3 The Edelman Model

The Edelman model which is currently used in AEMO's model of forecasting the five minute ahead aggregated demand values, is a first generation MLP (as described above) with one hidden layer. The Edelman model accepts 9-dimensional inputs (consisting of aggregated demand values in previous times, the details of which will be explained in a later section), a single hidden layer consisting of 4 hidden layer neurons, and one output neuron. Prof. Edelman represents the set of weights that connect the input with the hidden neurons by an input-to-hidden weight matrix, and the weights that connect the hidden neurons with the output neurons by a hidden-to-output weight matrix. Note that in AEMO's documentation the weights are called *coefficients*. The meaning of the two terms ("weight" and "coefficient") is equivalent and can be used interchangeably. This report uses the term *weights* since this is the prevalent established terminology in the ANN research community.

3.3.1 Forecasting of aggregate electricity demand

Prof. Edelman addressed some of the limitation of MLP networks by using a number of tricks. For example, Prof Edelman addresses assumption 1 and partially addresses assumption 2 (see section 3.2) as follows: Instead of using data as an input sequence he uses two fixed size time windows which are one week apart. Moreover, he transforms the input in order to smooth the effects of sudden changes in demand. This can be explained on an example as follows: At 20 minutes past midnight on any given day, the demand values within Edelman's time window are provided in the following table:

Current day	
0:00	z_1
0:05	z_2
0:10	z_3
0:15	z_4
0:20	z_5
0:25	z_6

and

One week prior	
0:00	z_7
0:05	z_8
0:10	z_9
0:15	z_{10}
0:20	z_{11}

Prof Edelman wished to obtain an input which is more responsive to small changes. So he performed the following simple transformation:

u_1	$\log \frac{z_2}{z_1}$
u_2	$\log \frac{z_3}{z_2}$
u_3	$\log \frac{z_4}{z_3}$
u_4	$\log \frac{z_5}{z_4}$
u_5	$\log \frac{z_6}{z_5}$

u_6	$\log \frac{z_8}{z_7}$
u_7	$\log \frac{z_9}{z_8}$
u_8	$\log \frac{z_{10}}{z_9}$
u_9	$\log \frac{z_{11}}{z_{10}}$

In other words, out of the 11 measured demand values, he produced 9 inputs u_i to the MLP. The transformed inputs have a nice property in that if $z_i = z_{i-1}$, then $u_i = 0$. If $z_i > z_{i-1}$, then $u_i > 0$. If $z_i < z_{i-1}$ then $u_i < 0$. Usually, as the difference in time between z_i and z_{i-1} is 5 minutes, z_i and z_{i-1} would be very similar. The logarithm value of their ratio becomes a small number. So this trick transforms a cohort of measurements to input values of much smaller values, and moreover, it is fluctuating in the region of 0. This makes it convenient for the MLP. Had we input the raw demand values z_i into the MLP, then it is found that the output of the MLP will contain a bias, which would be very difficult to eliminate (as the bias would depend on the magnitude of values of the input to hidden layer weights and the hidden to output layer weights).

The desired output of the MLP is given as $y = \log \frac{z_{12}}{z_{11}}$. In other words, at the output of the MLP, we may obtain the output as y and this represents $\log \frac{z_{12}}{z_{11}}$. So, once y is obtained from the MLP, then the value of z_{12} can be obtained as $z_{12} = z_{11} \exp(y)$.

The easiest way in which a MLP will reproduce such an output would be for the output activation function to be a hyperbolic tangent function. A hyperbolic function is one which will stretch from $-\infty$ to ∞ in the x axis, and in the y axis, the corresponding values are between -1, and +1. So if the MLP has an output activation function of hyperbolic tangent function, then this will be the y value indicated above, and then the forecast for z_{12} will be $z_{11} \exp(y)$.

However, the Edelman uses a sigmoid activation function for the output neurons. The problem here is that the sigmoid function is confined to within $[0;1]$ meaning that it cannot produce a negative output value. As a result, the Edelman model is required to perform a transformation of the network output by multiplying the network output by 2 and subtracting 1. This step increases the computational burden unnecessarily.

While the use of the logarithm of the ratio in change of electricity demand is more responsive to small changes, the approach smoothes large changes (making it difficult to predict large changes) and will fail completely when the changes are very large. Note that the network output is confined to $[0;1]$ and that even after scaling the Edelman model cannot possibly produce outputs smaller than -1 or larger than +1. However, the 5-minute ahead value u_{10} can be smaller than -1 or larger than +1 as the logarithm of $\frac{z+11}{z_{12}}$ can exceed the value range of $[-1;1]$ in cases of a sudden strong decline in demand (the ration is smaller than 0.368) or in cases of a sudden strong increase in demand (the ratio is larger than 2.718). Such large changes are exceptionally rare (i.e. a failure of the grid). However, the inability to predict such large changes worsens a situation which is already exceptional. Moreover, the smoothing increases the vulnerability of the Edelman model in its ability to accurately predict electricity demand during periods of shock, spikes, or sudden changes.

Some limitations of the input transformation technique are as follows:

1. Note that the network input at time i is very similar to the network input at time $i + 1$. In fact, seven of the nine inputs carry the same informative value. For example, the network input at time 16:00 contains information about the electricity demand at time 16:00 15:55, 15:50, 15:45, 15:40, and the demands at time 16:00, 15:55, 15:50, 15:45 exactly one week prior. Then, 5 minutes later for the next prediction the network input contains information about the demand at time 16:05, 16:00 15:55, 15:50, 15:45 as well as at time 16:05, 16:00 15:55, 15:50 one week prior. Since the time window shifts by just one element every five minutes and hence the information conveyed by the network input is largely unchanged. A consequence is that if the network made an error at time i then the MLP is prone to make a similar error at time $i + 1$. Moreover, since the network input at time i is processed independently to the inputs at time $i + 1$ and hence, the Edelman model is unable to learn from, or adjust to, mistakes made earlier in the time sequence. This problem is observed in practice.

2. A cohort of 45 minutes length does not capture climatic or seasonal variables and does not capture the sequential time series nature of the domain.

3. The time windows can only capture time dependencies between the two consecutive weeks. In practice there are time dependency patterns such as annual time dependency patterns, seasonal time dependency patterns, hourly time dependency patterns etc which cannot be captured by the Edelman model.

4. Occasional large shocks are considered a type of noise [5]. However, in the literature the term “noise” is generally used to describe measurement or data entry errors. Large shocks are actual observations and not usually caused by measurement errors. It is surmised that the actual measurement errors at AEMO should be quite small and quite possibly negligible, except when the sensors malfunctioned. However, large jumps in the input do occur: i.e. in response to a pricing signal, a user suddenly switched off or on a substantial load, such reduction or increase in the demand curve will be fed into the inputs of the MLP. Prof Edelman’s model appears not to take such effects into consideration.

A second aspect of the Edelman model is that it assumes that the inputs are all independent of one another. Thus ignores the temporal dependencies of the inputs. There are neural network models, which can express the temporal dependencies among the inputs. Therefore it would be possible to experiment on the effects of temporal dependencies of inputs on the 5 minute ahead forecasting accuracies.

A third aspect is that the Edelman model was trained using the data collected in the 1990s. The model has been subsequently adapted and applied as the forecast of the aggregate demand of the AEMO operations. The model would be over 15 years old. In the meantime, the underlying domain has changed much, in that the list of new energy sources has changed, behavioral patterns have changed, etc. Therefore it is anticipated that the Edelman model being based on training data which is over 15 years old, can incur large error values when it is being deployed to forecast the aggregate demand of today, as the underlying domain has evolved much during the past 15 years. AEMO attempted to re-train the MLP on newer data with mixed results. In most cases the new model performed worse than the old model. However, as will be shown in Section 4.4 of this report, the failure in updating the model is due to limitations of the software used as well as due to errors in the use of the software.

4 Overview of the development and latest advancement of Neural Network Model and relevant forecasting models

An impressive number of more modern machine learning methods have been developed over the years. A historical overview of ANNs is given in Figure 2. The ANNs shown in Figure 2 are limited to those that have some relevancy to time series prediction. We also omitted some of the most recent developments since those methods are not yet fully understood or are yet to demonstrate reliability in real world applications. The following will explain these methods based on whether they are supervised or unsupervised learning methods.

4.1 Supervised learning Methods

A Hopfield network is a neural memory designed to act as a fault tolerant storage (or memory). Due to its fault tolerant properties it can be used to reduce noise and to restore missing information in data streams.

Much more relevant to prediction in time series data are the Elman networks introduced by J.L. Elman in 1989 [7]. The Elman network is a simple recurrent neural network designed to process data sequences and which has found widespread applications in speech processing and

[5]Source: A document prepared by De Mare Consultants in 2010 for AEMO.

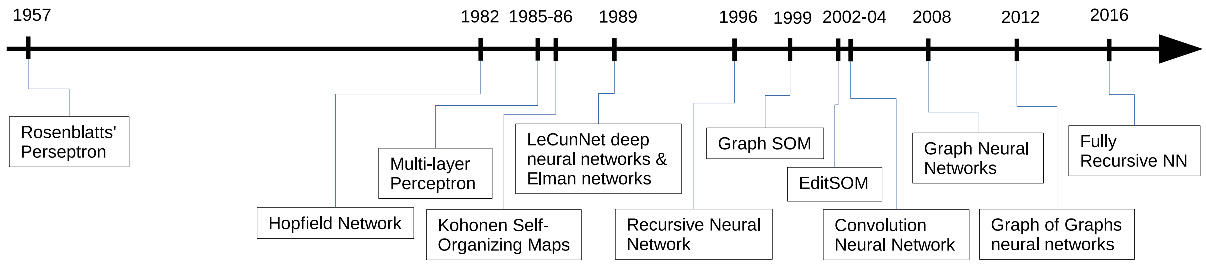

Figure 2: The evolution of Artificial Neural Networks.

financial forecasting. Elman networks can be trained on time series information of any length and they do not require the input stream to be chopped into small time windows or into small sub-sequences [6]. More importantly, Elman networks are able to model the time dependency of the data in a data stream. Hence, Elman networks can learn from a stream of historical electricity demand data the time dependencies of the data. In layman terms this means that an Elman network is able to identify changes in electricity demand based on the day of the week or time of the day. The Elman network is in fact an extension of the MLP network architecture and can hence be referred to as a *second generation* MLP.

Y. LeCun et al. (1989) introduced the concept of deep learning which improved the prediction accuracy of single hidden layer MLPs. This was an interesting development because until 1989 it was widely believed that MLPs do not require more than one hidden layer [7] in order to produce good results. The belief stemmed from a formal proof which showed that MLPs with a single hidden layer are universal approximators. However, in practise it was found that MLPs with many hidden layers often outperform MLPs with few hidden layers. There are a number of reasons to why the theoretical results deviate from practical observations. A main reason being that the theory assumes that the hidden layer can be arbitrarily large (i.e. it can have thousands or millions of neurons in its hidden layer). This however is not practical due to implications on the computational demand. Another reason is that a computer's floating point precision (the precision by which real values can be represented by a computer) is limited whereas the theory assumes an arbitrarily accurate precision. It turned out that MLPs with many hidden layers can overcome these problems and thus produce better results practically. Deep neural networks existed prior to 1989 but it was not until 1989 when Y. LeCun described an effective way of training such deep neural networks which, in turn, made deep learning useful for solving real world learning problems. In the context of electricity demand prediction this means that the capabilities of the model currently used by AEMO can be improved by adding more hidden layers and to train the system by using algorithms such as those proposed by Y. LeCun. Deep learning is very much in the news these days. Many of today's most powerful neural systems (i.e. as found in AlphaGo, autonomous vehicle control, etc) use deep learning as a fundamental building block.

[6] A recurrent neural network can be understood as follows: the output at current time is dependent on a weighted sum of previous output values through their manifestation as outputs of hidden layer neurons – manifestation here is taken to mean that the past outputs were used as inputs to the hidden layer neurons and the outputs of these hidden layer neurons provide manifestation of the inputs. Hence, the designation of "recurrent" in the name of the model.

[7] There is no theoretical benefit to having more than one hidden layer as long as we have sufficient number of hidden layer neurons.

Recursive neural networks [8] have been introduced by a team of Australian researchers in 1996 [17]. RNNs can encode any number of causal relationships between input data. Conceptually it is quite similar to the recurrent neural network. It considers the input data, and does the connections between the inputs and the hidden layer neurons in exactly the same manner as the MLP. What is different for the recursive neural network from that of the MLP is that there are instantaneous connections among the hidden layer neurons. The hidden layer neurons are fully connected. The outputs are derived from the hidden layer neurons in the same manner as the MLP. In other words, the recursive neural network has exactly the same architecture as the MLP, except in the recursive neural network case, the hidden layer neurons are fully connected with instantaneous connections through unknown weights. One would ask: what are the significance of these instantaneous connections in the hidden layer. Imagine that each of the neuron encodes some spatial relationships (the meaning of instantaneous connections in the hidden layer), then the recursive neural network would be an excellent neural architecture to encode spatial relationships. Thus in the electricity grid, the recursive neural network will be able to consider the spatial relationships among the generators, or loads, as well as encode the temporal relationships in the demand.

The Fully Recursive Perceptron Network (FRPN) is a very recent development. What is exciting about the FRPN is that it was shown that FRPNs can at the very least match, or surpass, the capabilities on MLPs while being only a fraction in size. In other words, an FRPN can match the computational abilities of much larger (i.e. very deep) MLP networks and consequently can be deployed to any learning problem to which MLPs can be applied to but at greatly reduced computational cost. FRPNs are hence suitable in real time systems which may require a frequent (or continuous) retraining of the model (i.e. as in very dynamic domains). FRPNs can surpass MLPs in applications that feature infinite periodic patterns (i.e. such as the sine curve). MLPs and deep learning methods cannot model infinite periodic patterns unless they feature an impossible large number of neurons in the hidden layer. The ability of modeling infinite periodic patterns can be of relevance in time series prediction since the FRPN is able to model fundamental signals that occur periodically on a daily or weekly basis.

4.2 Unsupervised Learning Methods

Unsupervised Learning Methods [9] can be quite good in providing what people usually termed “intraday” demand forecast. Main reason is that unsupervised learning methods do not require (and do not work with) target values. Instead, unsupervised learning methods model data based on similarity. Thus, unsupervised learning models can model historical demand curves (via a training set) and then render a prediction by detecting/finding a historical demand curve (profile) that is most similar to the current situation and use the next data point from

[8] The terminologies are somewhat confusing as they were developed by different groups of people. We cannot change them unilaterally without causing further confusion. It suffices to say that for recursive neural networks, the major difference from that of the recurrent neural network is that a recursive neural network contains instantaneous feedback, while a recurrent neural network contains only delayed feedbacks. This subtle difference causes quite different behaviours between a recursive neural network and a recurrent neural network.

[9] There are two kinds of learning methods: a supervised learning method, which has a target associated with the learning problem, and an unsupervised learning method which does not have a target associated with the learning problem. The provision of a target will “teach” the learning system while unsupervised learning problem trains a learning system without the help of a target. Unsupervised learning methods can be used i.e. to group similar sequences into clusters.

the historically similar profile as the prediction of the output of the learning model. The rationale is that if a current demand curve follows a previously observed demand curve then it is reasonable to expect that a future demand will be similar to the demand that was observed after the (matching) previous demand curve.

Unsupervised learning systems are often more scalable, robust to noise, and less sensitive to sparse or high dimensional input than supervised learning methods.

The Self-Organizing Map (SOM) introduced by T. Kohonen in 1986 [12]. The SOM is able to project high dimensional data (data which exist in high dimensional space) onto a low dimensional *display* space. The display space is usually 2 dimensional for easy visualization. The underlying algorithm is very efficient such that SOMs are popularly applied in many data mining and big data applications and in order to reduce the dimensionality of a learning problem. One of the strongest properties of a SOM is that it performs a topology preserving mapping. This means that high dimensional data which are close to each other in the input space will be mapped close to each other in the low dimensional display space. SOMs are hence also used commonly for clustering tasks (grouping of similar data items together into a group). Moreover, SOMs can be applied to data sequences. These properties of a SOM render it useful for addressing one of the main problems with predictions in temporal sequences: to account for discontinuities in the target set caused by e.g. price spikes and price responses.

The SOM could overcome a very common problem in the grouping of demand curves. Very often, the shapes of the demand curves are not exactly the same. There might be some minute variations, like one of the demand at 0:25 on a Wednesday, instead of being 6,000 MW it is, say, 6010 MW. Should we call this a new class, or should we say that it is still acceptable to be in the same class as the one with 6.000 MW. The SOM could be deployed to overcome such an issue, as it works on the whole shape instead of just on the individual points. The advantages of using this method when compared with the traditional method is apparent. Traditionally, a 24 hour demand curve is known to exhibit the following effects:

- Similarity of demand today and yesterday at the same time. This is called a daily periodicity effect.
- The demand curve exhibits a diurnal pattern, i.e., two daily peaks, and a trough at night.
- The demand today, say, Wednesday, is similar to the demand last Wednesday. This is the weekly periodicity effect.
- Seasonal effects. It is known that the demand curve is higher during some of the consecutive days of the year, and relatively lower during some other consecutive days of the year.

Traditional technique would fit a deterministic model to each of these components. The demand curve is used to subtract these deterministic components, and arrive at a residual signal which hopefully will be zero mean. Traditional time series forecasting technique is to build say an autoregressive model for this residual component. The one step ahead forecast would be obtained by running the autoregressive model forwards one step, and then added the corresponding values from the deterministic components. This technique has been applied to electricity demand and load forecasting in time series literature [1, 8, 9, 10, 22].

But to the practitioners, it is always known that these so called deterministic components are approximate. Indeed it is not uncommon to find that they do not work. For example, while it is known that there are two peaks during the day, sometimes these peaks occur at some time like 10 am and 18:00 or 18:30. However, with the changing of living style, it has been observed that the evening peak sometimes would occur at 19:00. Moreover, it is sometimes observed that the evening peak stretches out to be a series of peaks and troughs over some defined level.

With the effect of El Nino, and global warming, it is sometimes found that the seasonal effects are not as pronounced as previously. So this makes it quite an art to extract the deterministic components underlying the demand or load curve for the method to work.

One might say that we are only interested in 5 minutes ahead forecast, and therefore, the daily periodicity, the seasonal components do not affect the very short term forecast that much. This statement might be true in the days when the main electricity generators were thermal, hydro, or geothermal. Nowadays we have increasing proportion of the energy generation from renewable resources, e.g., wind, sun. The inclusion of renewable resources in the electricity grid supply causes problems. The main issue is that their supply may be fluctuating. This causes fluctuation in the generation curve which was not there previously. As the availability of renewable resources is seasonal but not necessarily periodic this means the traditional time series forecasting techniques would not work well. Therefore a neural network model, like a SOM might be able to provide better prediction of the 24 hour ahead forecast than say one which is based on traditional time series model forecasting techniques.

The SOM has been extended and modified over the years. A GraphSOM has been introduced in 1999. A main property of the GraphSOM is that is can map data graphs in linear time to a fixed dimensional display space. Data graphs, consisting of a number of nodes connected to one another (not necessarily fully connected) by links, are structures that are more general than vectors and sequences in that an arbitrary number of dependencies can be modelled by a graph. Data sequences are limited to representing linear dependencies in which each entry in a sequence can have at most one ancestor and at most one descendent. Graphs on the other hand can represent any number of dependencies such as, for example, the dependency of historical electricity supply and demand on individual loads and generators [10]. GraphSOMs can be engaged to detect whether a given situation will cause a price response [11]. This is modelled from past observations (the training set) and can detect if a current situation is very similar to a previous situation which triggered a price response. Once a response triggered by a price is predicted, a second model can be engaged to predict the subsequent outcome of the equilibrium adjustment process. This could be used to model a bid process in which the demand curve as a result of a price trigger; and multiple demand and price triggered responses can be represented. However for this to work well, we might need a large amount of data in which the system responded to various pricing situations are recorded. Systems that combine more than one model are very common and are called *ensemble systems*.

The EditSOM introduced by Bunke et al. (2002) has the same capabilities although the underlying learning algorithm is computationally very expensive and is generally not useful for deployment to real world and large scale applications. Thus, the GraphSOM is useful for

[10]Conceptually one can imagine each node to represent a generator or a load; the generators are connected to supply the loads through links (a transmission line connecting the generator and the load. Therefore a power system can be represented by a graph. Interconnected power systems can be represented by a number of graphs connected together in a hierarchy of graphs.

[11]One can conceptually think of the following: each node in a graph can be represented by a SOM which takes the inputs into that node. The SOM in each node will group inputs with similar characteristics into that node together.

electricity demand (and price response) prediction if it is desired to model individual loads and generators. Otherwise, if aggregate data is to be modelled than a standard SOM can suffice.

4.3 Ensemble and Hybrid Systems

Ensemble systems consist of a combination of two or more learning methods. Hybrid systems are ensemble systems which consist of different types of learning methods. Thus, every hybrid system is an ensemble system but not every ensemble system is a hybrid system. The main objective is to combine the capabilities of individual learning systems into a single holistic system.

We distinguish between horizontal and vertical ensemble systems. This is illustrated on an example in Figure 3. It can be seen that horizontal ensemble systems organize the learning systems in a horizontal layer then combine their outputs to produce a single response to a given input. Vertical ensemble systems organize the learning systems in a vertical layer such that the output of one learning system is used as an additional input to the next learning system. Note that the vertical ensemble system shown in Figure 3 is a hybrid system. Horizontal ensembles combine the predictive power of the individual methods. Hence, horizontal ensembles are insensitive to errors produced by a single method and are hence more suitable for critical applications. Vertical ensembles are used to reduce the residual error of a learning system. For example, a SOM may have been trained to predict energy demand and, after training, the output of the SOM may still deviate by 10% from the correct demand on average. In a vertical ensemble, the output (the prediction) of the SOM can be fed to a second learning system such as an RNN. The RNN is then trained on the input data as well as on the things that have already been learned by the SOM. In other words, the RNN is trained to reduce the residual error of the SOM. After training it may be found that the RNN still produces an error of 5% on average. A third learning method can be engaged to reduce this residual error further, and so on, until the residual error does no longer decrease.

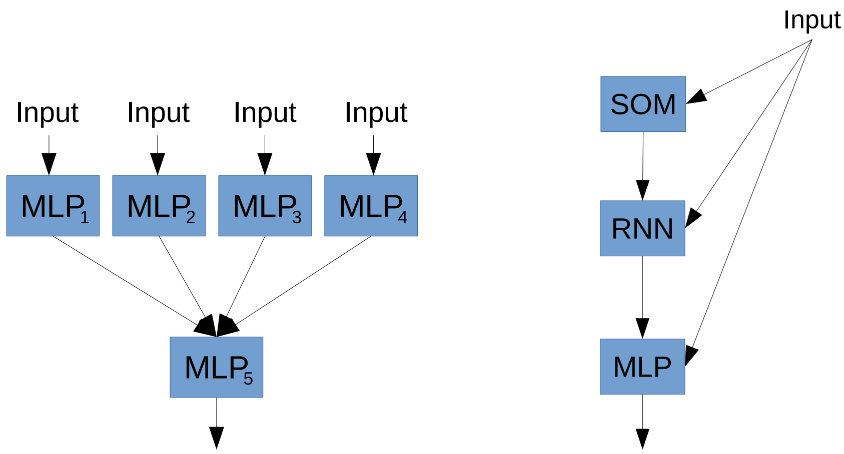

Figure 3: A horizontal ensemble system (left) and a vertical ensemble system (right).

Vertical ensembles can produce better results than vertical ensembles. However, horizontal ensembles are sensitive to a failure of any one of the learning components. It is possible to combine both concepts although this can make such systems very complex and hard to maintain.

4.4 Predicting Energy Demand

AEMO uses an MLP to predict 5-minute ahead energy demand and found that the MLP produces much better predictions than naïve (no-change) forecast models and linear regression forecast models. AEMO found that the prediction errors of the MLP is about half that of those other prediction models [4, 6].

The currently used MLP model has been generated and trained in 2005 (Tasmania), 2007 (Queensland), and 1999 (Victoria, NSW, and South Australia). An attempt to update the 1999 network model was made in 2007 by re-training the MLP on newer data. It was found that the old MLP performed better [12]. The reasons to why the MLP did not improve its performance are caused by limitations of the software used by AEMO and in the way the neural network was trained [2]. The reasons can be summarized as follows:

- The problem faced by AEMO is a data mining problem. As with any data mining problem, and before considering any of the possible approaches, it is essential to have a good domain understanding. It does not suffice to merely collect data and then to feed the machine in the hope for some good results. The data needs to be properly analysed and, if necessary pre-processed. A good understanding of the data and a good understating of the problem enables a data miner to choose the correct approach and training parameters.
- Training MLPs is guided by a set of training parameters which need to be set manually. These parameters include the learning rate, number of training iterations, number of hidden layers, number of hidden neurons. The software used by AEMO is called MetrixND. The software supports two training modes which are called "Train" and "Learn" [13]. The AEMO user manual recommends the use of the mode "Train" [2]. However, that mode does not allow a user to set the learning rate, the number of hidden layers, nor the number of hidden layer neurons. We therefore need to presume that the software uses fixed default values. This can be a major problem because an MLP trained on more recent data would almost certainly need more neurons in its hidden layer(s). Main reason is that the volume of data increases with time and the shape of the demand curve increases in complexity due to increased dynamics in the electricity demand and supply. An increase in the number of hidden neurons would increase the MLP ability to model complex patterns. Moreover, the MetrixND uses 100 training iterations as a default value. The software allows its users to change this value. However, AEMO's manual makes no mentioning that this value should be changed. MLPs are known to be *slow learners* often requiring several thousand training iterations [20]. It is very likely that when leaving the number of training iterations at 100 then the resulting MLP has not been trained fully. This is particularly detrimental in the case when an MLP is to be trained on complex patterns as in such a case the MLP would need many more training iterations to fully model the training set. Consequently, it appears that AEMO attempted to train an MLP on new set of data but by using the same network architecture and parameters as those used in the

[12] Source: Memo from AEMC

[13] In the MetrixND software, the difference between these two modes is that a "training" session does not allow the user to set training parameters (it uses default values) whereas in the "learning" mode the software allows the user to alter some of the training parameters such as the initial network condition (via a seed value for the random number generator), the number of neurons in the hidden layer. Other training parameters such as the learning rate and number of hidden layer cannot be adjusted in MetrixND; hard-coded default values appear to be used instead.

training of the old model. The correct approach would have been to adjust the training parameters in order to account for the properties of the new training set.

- The MLP is inherently limited in its abilities to model time series information. These cannot be overcome by simply training (or re-training) the MLP. These limitations are explained in much more detail in Section 6.

Given that MLPs have been superseded by more modern ANNs and which are designed for time series prediction learning problems (as opposed to MLP which were never meant for time series applications) the MLP should not be considered suitable for predicting energy demand.

4.5 Pricing Policies

The MLP used for prediction of demand assumes that pricing policies, once fixed, do not change. The problem is that the current model does not take into account responses by the consumers as a result of current pricing policies. Say, for example, the demand is predicted to grow by a certain amount. The increase in demand may require AEMO to schedule high cost generators which in turn increases the cost of the electricity. Loads may react to the higher costs by delaying the high load event to a later time. Pumping stations and aluminium smelters, for example, can reduce operational costs by delaying operations (such as pumping or the heating of the aluminium ore) at times of high energy cost to times when the energy cost is lower. Thus, the prediction of a higher load may lead to a high cost of energy which in turn causes the demand to drop. The main problem with this is that such a response to price may push the demand to a level at which energy could have been supplied at a cheaper cost. However, since the pricing policy is fixed and hence, the cost of electricity may be higher than is necessary for meeting the actual demand.

If, on the other hand, the price response had been predicted then, because the demand would drop, the cost of electricity generation would also drop. It can be that this new price would not trigger a price response and hence, the demand would remain high as originally predicted which, in turn pushes up the price. The situation could cause a feedback loop which may not converge. Moreover, there is often a consequence to every price response. For example, an aluminium smelter may delay an activity during a high price event. High price events do not normally last more than two hours, and there is a (unspecified) limit to the time by which a smelter can delay an operation. Thus, the smelter will engage the high load event at some time after the price response event. This means that the demand can be higher than normal for some time after a price response.

Due to its limitations, the Edelman model could not possibly predict the correct demand when a price response occurs, nor can it predict the consequences of a price response.

The techniques considered in this section may be considered as dealing with two situations: the incorporation of contextual information and the external inputs in the short term load forecasting problem under normal operating conditions, and the consideration of modles which can handle abnormal operating conditions, like the occurrence of spikes, shocks in the demand curve. The models which are proposed to handle these two situations are designed based on different techniques. So, for a load forecasting system to work in an automatic fashion, it would be desirable if they can be combined so that the two approaches can work well together be it in the normal or abnormal operating condition. This would require a purpose built system which could incorporate situations which might be unique to the Australian environment.

5 The international experience of energy market forecasting and lessons in which Australia can draw on to improve its current forecasting accuracy

It is well-known that each electricity supply and demand network is unique or different. Therefore some care would need to be exercised in working out the general underlying network characteristics first before making use of the conclusions reached by others. The unfortunate fact is that often researchers do not clearly state the characteristics of their underlying network, and thus, it makes good detective work trying to uncover what can be learned from international experience, as a general rule, or what might be considered as peculiar features of that particular network, and hence it would not be true for other networks.

Common are situations where there might be large consumers, like aluminium smelters, or data centres. These industrial concerns use large amount of electricity and electricity charges form a substantial part of their operating costs. Often such industries would have built close to power generating stations. For example, it is known that Microsoft Google built data centres close to energy production sources which are often cheap energy production sites. Often their load is inelastic, i.e., it cannot be switched off. However there are consumers the load of which are elastic, i.e., it can be switched off and on without affecting the operation of the plant, e.g., incinerators, electric furnaces. Often such consumers would have automatic mechanisms which will control the on or off of their load in response to pricing signals. These consumers essentially produce a jump in the demand curve. If they come online then the demand will jump up, while if they get offline then the demand will be dropped by a certain amount. The forecast of such jumps in demand is notoriously difficult, as consumers do not need to inform the suppliers beforehand when they wish to switch on or off their load. However, through a consideration for the underlying economics of the response of the supply and demand in continuous time, like one modelled by the Orstein-Uhlbeck process, where the supply and demand is implicitly modelled in the mean reversal process, it is possible to predict the probability that the demand curve will jump. There are extensions of the Orstein-Uhlbeck process which is used to consider the jump in electricity demand. Such models might be possible to provide some prediction on the jump probability of the demand curve.

5.1 Case study: Electricity demand forecasting in Germany

Electricity demand prediction has been studied intensely for the past 24 years. The Institut für Elektrische Anlagen und Energiewirtschaft (IEAW)[14] is a research institution specializing in electricity and energy markets, network planning and operation, and system stability and security of supply. They work closely with industry and government institutions such as the BDEW (German Association of Energy and Water), European network operators, German network electricity operators, German utilities, the federal ministry for economics and technology, German power companies, the federal electricity network agency, and others. Some of their earliest considerations with relevance to this report date back as far as 1992. A 1992 report[15] by

[14]Institute for Electrical Systems and Energy Industry. Web: http://www.iaew.rwth-aachen.de/

[15]The report was published in the form of a thesis. Weblink: http://www.iaew.rwth-aachen.de/index.php?id=63&tx_iaewtheses_pi1%5Buid%5D=202&cHash=4fd41b823586b7c608594f0a145f69f3 accessed on 15/Aug/2016.

IEAW tested existing methods for electricity demand prediction, and proposed the use of MLPs as a way to improve the prognosis of node loads. They found that MLPs generally outperform standard prediction methods.

Neural networks for load prediction have been studied more explicitly shortly after in 1993 [11]. The work provides a comparison of neural networks with standard regression methods and explains the advantages of neural networks over standard regression methods by utilizing the proven computational capabilities of MLP networks [11].

Studies continued past the introduction date of AEMOs current model: A 1998 study investigated methods for reliable load forecast in a liberalised electricity market [14]. The study found that artificial neural networks have distinct advantages especially to conventional multiple regression. Nevertheless, they emphasize that a main problem of current (as of 1998) neural networks "is based, to a great extent, on trial and error" [14]. They confirm the now well understood requirement that neural networks must be adapted to the specific requirements of the respective company or application. In other words, neural networks need to be tailored for deployment to a specific problem.

A 2004 working paper on "Trends in German and European Electricity Forecasting of Electrical Load" (published in German) provides a comprehensive review of methods for forecasting electrical load [15]. The paper studies the suitability of a range of prediction methods for predicting long term, mid term, and short term electricity demand. For long term prediction they investigate the ability of predictors to predict load six months ahead, one week ahead for mid term predictions, and one day ahead for short term predictions. The methods compared include linear regression methods, auto regression methods, fuzzy systems, and artificial neural networks. They also study the influence of calendric attributed influence on the electrical load curve, the influence of daily load curves, the influence of rhythms (cyclic and a-cyclic rhythms) and patterns in weekly load cycles, and seasonal influences. They include an investigation of the effects of meteorological parameters such as temperature and cloud cover on electrical load. The most significant findings of their work include:

- The improvement in prediction capability arising from the inclusion of meteorological information is insignificant although an improvement on prediction accuracy is observed when predicting regional demand.

- Price response is recognized as a problem.

While the working paper provides a comprehensive study on prediction methods of 2004 it does not offer an analysis of the reasons that led to these findings nor does it offer a forward looking approach. Hence, this should be considered an incomplete work.

One may ask: what had happened since 2004 when the report [15] was written. The short answer is not much. Most application oriented researchers continued to use MLP based techniques for short term load forecasting, with minor adjustments (see e.g., [22]) [16]. To the best of our knowledge, nobody has yet applied deep learning, or recursive neural networks, FRPN,

[16]One may argue that the neuro-fuzzy approach based on adaptive neural fuzzy inference systems (ANFIS) [18], fuzzy techniques, as summarized and presented in [13] would be an advance in conceptual thinking in using artificial intelligence based techniques to short term load forecasting. However, it may be observed that ANFIS is an MLP with radial basis functions as hidden layer neuron activations [16, 18], and that fuzzy-type algorithms are offline methods which might be challenging to be applied to online learning like in the short term forecasting problem.

as indicated in the previous sections, to short term forecasting of electricity load. These could, if handled properly, represent the next level of advance in using the state-of-the-art machine learning techniques to the problem of short term electricity load forecasting.

Smart meters are in use in Germany since the 1990's. Smart meters were initially used to meter just the major loads and were introduced to improve prediction accuracy. Since 2010 smart meters are being increasingly adopted in private households as well. The effects of smart meter data on prediction accuracy has been studied widely. For example a 2012 study on the inclusion of smart meter data reports a 38% improvement in prediction accuracy [3]. These findings motivated the German government to introduce a law in July 2016 which stipulates that all new households must install smart meters from 2017 onwards and all existing households must have smart meters installed by 2032.

More recent studies on load prediction investigate the possibility of predicting individual loads. A 2014 project report on the "Creation of Load Forecasts for Electricity Demand of Family Homes" considers prediction algorithms such as regression, ANN, genetic algorithms, and fuzzy logic [21]. Some of the main findings are:

- Genetic algorithms [17] are too slow in learning from data to be useful in real time prediction systems.
- Fuzzy logic (a non-parametric system) [18] is suitable for the creation of reference models.
- ANN are the most successful and most widely studied approach to demand and load prediction.

It appears that in general 15 minute ahead prediction is most common for scheduling generators in German. The German electricity market is tightly interwoven with the European energy market. Electricity is being traded with other countries in the European union.

5.1.1 Lessons learned (Germany)

Some of the lessons learned in Germany are:

- ANNs are most widely studied and most popularly used for demand prediction. They generally produce the best results and are suitable for real time prediction systems.
- The inclusion of climate data has no significant effect on improving the prediction accuracy of aggregate electricity demand.
- Prediction accuracy of regional loads is improved with the inclusion of climate data.
- The inclusion of smart meter data significantly enhances the prediction accuracy.

Moreover, they are facing similar problems as encountered in Australia. Most notable are current efforts in addressing problems that arise out of price response.

[17] These are based on the observation that biological genes can split, mutate, and these can be used as a way to randomly select parameters of the learning problem in such a way that the problem will eventually settle down to an equilibrium.

[18] These are using the observations that human reasoning takes place not as black or white, but somewhere in between in the grey area. These require extensive hand-tuning for it to work well.

The trend also appears to go towards an increase in granularity by predicting the demand of individual loads. This is most likely motivated by the penetration of renewable generators. The number of households with roof mounted photovoltaic systems has increased to about 1.5 million in 2015[19]. Combined they produce 38.5 TWhr or 7.5% of Germany's energy demand. Solar systems are also increasingly installed on industrial sites. The energy generated offsets the demand at a site, increases the volatility of the aggregate demand and hence, motivates the study in the prediction of individual loads rather than prediction of aggregates.

5.2 Lessons in which Australia can draw on

In summary, the following experience could be learned from international practices in operating the power grid:

- The German experience has shown that weather variable would be important to be incorporated in the forecasting model of regional loads but would be of limited use for forecasting an aggregate load. The experience may be transferable to the Australian case because most of its population (and therefore the major loads) are located along the coastal, more temperate, fringe. Nevertheless, an important finding of the German models is that the inclusion climate data does not harm the prediction accuracy but that it has the potential to improve prediction accuracy under suitable circumstances.
- The German models have shown that seasonal effects play a much more significant role on prediction accuracy than climatic data and hence seasonal data should be included in the prediction model. The neural network model currently used by AEMO does not include seasonal information as part of its input. Hence, an improvement can be made by incorporating seasonal effects.
- Neural networks, because of their ability to model nonlinear functions, are some of the best models for forecasting. The experience in Europe has shown that neural networks are at least as good, and often better, as traditional methods. The solutions proposed in this report to addressing temporal differences, effects of price response, spikes, and shocks, and effects of volatility are in line with German efforts to addressing similar issues.
- Electricity demand prediction has been the topic of intensive research over the past 20 years in Europe. Ongoing efforts are made on improving prediction accuracy in a changing landscape. The result is that some of the European countries use prediction systems which have been adopted to significant changes in the energy market. Moreover, the systems are tailored to work best for specific applications (i.e. prediction of demand in Germany). Australia can learn from this by considering the issue of electricity prediction an ongoing effort for optimizing electricity prediction accuracy. These efforts go hand in hand with developments in data mining and machine learning research. Moreover, it should be best to aim for a purpose built system that designed to work best for Australian conditions.
- German studies include the analysis of prediction methods, their dependence on external factors, and sensitivity to rhythmic patterns. The analysis leads to a better understanding of existing limitations and to a better understanding of suitable data preparation methods.

[19]Source: https://www.ise.fraunhofer.de/de/veroeffentlichungen/veroeffentlichungen-pdf-dateien/studien-und-konzeptpapiere/aktuelle-fakten-zur-photovoltaik-in-deutschland.pdf

A similar analysis would allow Australia whether, for example, climatic information does indeed not affect the prediction accuracy of aggregate loads, or would help establish a proof of concept for new methods (i.e. such as those proposed in this report).

- It is also observed that the pricing strategy is an important mechanism in inducing the consumers to behave in particular fashion. There is a tradeoff between the desire of inducing a desired consumer behaviour and the effects of price response on prediction accuracy. An investment in the development of systems that are robust to price response would provide a greater degree of freedom to inducing a desired consumer behavior. Moreover, this would render Australia's system less sensitive to the effects of the very expensive peak demand (i.e. the need to use diesel generators).
- The German experience has shown that the use of smart meters improves prediction accuracy very significantly. The observation is made in conjunction the adoption of solar generation in industry and in private households. Given that Australia experiences a steady increase in the number of photovoltaic systems and hence, a wider adoption of smart meters and the inclusion of smart meter data in electricity demand prediction systems should be considered.
- De-synchronization might occur in power grids, especially one which incorporates renewable energy sources. This is because of the possible extreme behavior of some of these renewable energy sources, e.g., wind turbines, which in particular high wind situations could de-synchronize the network, and lead to the need to isolate such sources from the network, otherwise the network might become unstable.
- German energy market operators employ data mining experts for addressing the problem of prediction or outsource required services to i.e. Enercast Inc or Siemens. By doing so they do not need to rely on static models and instead have access to state-of-the-art models and can react promptly to regulatory and market condition changes.

6 An evaluation of AEMO's Neural Network Model in relation to its forecasting capability

This section provides a high level evaluation of the Edelman model in relation to its suitability towards forecasting electricity demand. The model is designed for making 5-minute ahead demand predictions. More specifically, the model is designed to predict the amount by which the demand changes within 5 minutes. If the predicted value deviates from the actually observed value then this is referred to as the prediction error. AEMO can react promptly to prediction errors but the cost implications of significant forecast errors cannot be neglected. Hence, it is important to address the problem of large errors. AEMO observes large prediction errors quite frequently (several times per day on average). The rest of this section will explain why AEMO's current prediction model produces errors and why large errors are common.

6.1 The Effect of Time Dependencies

The Edelman model cannot model temporal dependencies nor model (effects of the) time itself. As a consequence, the model cannot deal with time dependent demand patterns nor can it deal with time specific demand patterns. The model will produce large forecast errors in such cases. This is best explained as follows: Recall that the Edelman model receives as input a 9-dimensional numerical input vector at any given time instance i. The time i itself is not included in the network input. The model thus makes a prediction in absence of knowledge of time. Lets for convenience call the input vector a *demand sequence*, and lets assume that a particular demand sequence S is observed at time i_1 and again at time i_2. Thus, S is observed more than once in the historical data. Recall that historical data is used to create a training set (and test set) for the MLP. Given that S is observed at different times say, during a weekday and during a weekend, hence it is likely that the demand value d_{i_1} at time i_1 differs from demand value d_{i_2} at time i_2. Thus, the dataset would contain the input-output pairs (S, d_{i_1}) and (S, d_{i_2}). If both samples are in the training set then the training algorithm would face the impossible task of mapping the same input to different targets. It is then said that the training set contains *conflicts*. The size of the network weights (after training) can provide an indication to whether a training set contained conflicts. The weight matrices presented by D. Edelman contain weights as large as 125.6. This is an extremely large value and a strong indication of a problem with the data or the model. Large weight values lead to a problem called *saturation* and indicate a problem with the model. In general, saturation reduces the models ability to serve its purpose. On the other hand, if only one of these samples (i.e. (S, d_{i_1})) is in the training set then the network would produce as output d_{i_1} whenever S is presented as input and hence produce a large forecast error at time i_2.

For each year we have approximately 260 weekdays, 104 days fall on a weekend, but there are only 13 “special” days (public holidays). It is well known that electricity demand patterns on weekday differs from electricity demand patterns on weekdays or on public holidays. There are demand sequences on weekdays that are not normally observed on weekends. Similarly, there are demand sequences that are observed on public holidays but which are not normally observed during a weekday, and so on. Moreover, demand sequences can differ significantly for the different public holidays because. This means that demand sequences that are observed on i.e. a Boxing Day occur 260 times less commonly than demand sequences for weekdays. In machine learning this is referred to as a severely *unbalanced* learning problem. The problem with this is that the MLP training algorithm is being updated 260 more frequently on demand sequences observed on weekdays than on demand sequences observed during special days. The frequent cases cause the MLP to “forget” demand sequences that are rarely presented and hence, the MLP is likely to produce large forecast errors during special days.

6.2 Adaptability to Volatility

The Australian energy market is dynamic. Demand patterns and generator profiles change over time. There are now many more energy generators from renewable resources than a few years ago, new generators emerge and, on the consumer side, loads are becoming more and more energy efficient, and energy storage becomes more and more commonplace. As a result, energy demand sequences change with time such that i.e. sequences that were observed some years ago are no longer observed in the current market situation, and demand sequences observed in the current situation may not have occurred some years ago.

The MLP is being trained on a *static* set of historical data and, once the training session has completed, the weights in the MLP remain fixed. This means that the MLP cannot accurately predict demand values in a changed electricity market. The level of accuracy diminishes the more the energy market changes. Moreover, the MLP produces predictions that are based on information that was used to train the network. Some of the information is out of date and hence, the MLP performs predictions which do not reflect current energy market conditions.

Similarly, the function which describes the demand sequences is not continuous. Sudden changes can cause abrupt changes to demand (i.e. a sudden increase or a sudden decrease). The training algorithm of an MLP is only defined for learning problems that can be described by a continuous function. In all other cases the MLP is prone to generate large errors when sudden changes occur.

Thus, the Edelman model cannot react to dynamics in the energy market and will make prediction errors that increase in frequency and magnitude as the time gap between current time and the time at which the training set was sampled increases. While the Edelman model limits these errors through the application of the log function, this only masks rather than addresses the problem.

6.3 Effects of external Factors

Future events often depend on a multitude of factors. Many historical prediction systems simplify a problem by assuming independence, or by neglecting the dependency, to additional factors. Many of these factors can indeed be neglected. For example, the maintenance window for equipment at a major load may affect the demand but redundancies and protocols at the site are designed to minimize any effects and hence, is invalid to assume that the effects of maintenance are negligible under normal conditions. On the other hand, there are dependencies on factors that affect demand patterns. Weather and special events can have a pronounced effect on demand patterns. For example, wind generators may be disabled abruptly during high wind conditions. This may require the scheduling of high cost generators which will drive up the cost and may trigger a price response causing the demand to drop suddenly. High wind conditions are often predictable. However, since the Edelman model does not take climate data into account and hence it is not able to adjust to external factors when making a prediction. The Edelman model always assumes that the prediction occurs under "normal" conditions (i.e. conditions that are not affected significantly by external factors). Similarly, climate control systems require much more energy on hot days, snow generation is conducted only under suitable weather conditions, solar generators are affected by seasons, cloud cover and temperature, and so on and on.

Edelman's model assumes independence of demand sequences to any other factors. The Edelman model cannot predict demand that is responding to external factors such as weather. Hence the model can be expected to produce large errors in such situations.

7 Possible solutions to the identified problems in AEMO's current model.

The problem faced by AEMO is a data mining problem. There simply are no good general purpose off-the-shelf solutions to data mining problems because every data mining problem differs. Solutions are generally problem specific and need to be tailored to suitably address a given problem. While there are some general purpose products (such as those based on MLP) those systems are generally not able to produce a quality of results that can be expected from a special purpose build system. Recent research introduced models that serve the needs of a much wider spectrum of data mining problems. Such systems have the potential to develop into well performing general purpose systems although none of these have matured sufficiently yet nor have they been implemented for commercial applications. It will be some time until off-the-shelf solutions would become available to electricity demand forecasting.

The following presents some approaches and solutions which can be considered for improving AEMO's prediction model. We will start with the simplest approach which has the potential of improving prediction accuracy with very little effort although the expected improvements would be small. We will then list increasingly powerful systems up to self-monitoring systems which can handle shocks and automate some of the procedures done manually at current.

7.1 Improvements of the Current Model.

The current model should experience improvement in prediction ability by re-training it on newer data while increasing the number of hidden layer neurons and increasing the number of training iterations. The question to how many hidden layer neurons are required, and the number of required training iteration can be answered with the help of a validation dataset. The validation dataset consists of historical samples that are not in the training set. Unfortunately, the software currently used by AEMO does not support validation sets. There are many other software implementations of the MLP such as in the data mining software package R or in MatLab (and many other) which could be used instead. The use of a validation dataset has the following effect: The MLP is trained for a number of iterations on a given training set. At each iteration the network is being tested on the validation set. Since the validation set is disjoint from the training set and hence the validation results are a reflection of the prediction ability of the MLP. The training continues until the the validation results do not improve any further. This is the point at which training stops and hence, this determines the number of required training iterations. Network training is repeated by increasing the number of hidden layer neurons (by one at a time) and noting down the best validation result. This is repeated until the validation results do not improve further.

7.2 Modelling Time Series Information.

An Elman network (see Section 3.1) should improve the prediction results due to its abilities to model the time dependencies in the sequence of 5-minute demand data. However, the Elman network is not able to model very long time sequences. The sequence of historical demand data can be longer than 500,000, too long for an Elman network. Pre-processing of the data addresses the problem: Define a time window over the input sequence. A time windows of size N would contain N consecutive 5-minute demand data. The input to the Elman network would then become an N dimensional input, one for each time window. The time window can be shifted by i.e. $N/2$ along the sequence. This reduces the length of the input sequence to $500,000/N * 2$. Thus a trade-off between input dimension and sequence length is created. This comes at a cost to the precision of the prediction. The larger N the more information about time dependencies is lost. Hence, N should not be chosen too large as otherwise we loose the benefit of knowing the time dependencies and N should not be too small as otherwise we experience the long-term dependency problem. The best trade-off can be obtained experimentally and with the aid of a validation dataset.

7.3 Modelling Higher Level Temporal Dependencies.

RNNs offer a more effective mechanism for dealing with long-term dependencies as well as time dependencies. With RNNs it is possible to explicitly model arbitrary time dependencies by i.e. incorporating links to energy demand values exactly one hour prior, one day prior, one week prior, one month prior, and one year prior. These links can be added to every time instance in order to capture cyclic energy demand patterns which cannot be captured by the Edelman model. Additional links can be added to capture demand patterns on special days (i.e. public holidays). The RNN would hence be able to model temporal dependencies as well as daily/weekly/yearly cycles in electricity demand.

As far as we know, no mainstream data mining software package has incorporated the RNN. However, there is a publicly available command line driven software available [20]. It would be possible to add a more user friendly graphical user interface without much effort.

7.4 Dealing with Volatility

A limitation of all of the afore mentioned methods is that they remain static once training has completed. This means that the model by itself cannot react to changes in the energy market. Agile learning is the answer to this problem. A number of approaches can be taken:

Periodic retraining: Re-train the model on a regular basis and use a validation set to assure prediction capability. Substitute by the new model if validation results exceed that of the old model. Re-training could be done on a frequent basis to promptly react to changes in market conditions. When implemented on suitable computing hardware, most of the previously mentioned machine learning algorithms should be able to re-train within the 5-minute time window.

Event based retraining: Retrain the ANN only when found inadequate to dealing with a given (i.e. new) situation. The event of a large prediction error triggers the retraining of the ANN and on data that includes those that caused the prediction error. The approach promptly react to errors made and hence, limit the duration of consecutive errors.

Continuous training: It is possible to consider a software implementation which takes as input the growing number of historical data and trains the model indefinitely. However, continuous training brings the well known risk of *overfitting*. Moreover, continuous training means that we are bound to a given network architecture (i.e. the number of hidden neurons must remain fixed). Thus, in practise it should be better to consider periodic retraining or event based retraining.

7.5 Dealing with spikes, shocks.

Shocks create a discontinuity in the output space. The function or curve that describes the electricity demand has a sudden jump (a discontinuity) during a shock or spike. Shocks can differ in magnitude and duration. The aftermath of shocks can also deviate from normal observation patterns. MLPs do not work well during periods of shocks.

The SOM (see Section 4.2) is a neural network that can work with discontinuities in the output space. Although unsuitable for predictions, the method can be combined with a suitable

[20] Web link: http://www.artificial-neural.net

predictor to create a Self-Organizing Ensemble System (SOES) as follows:
Step 1: Train an SOM so as to *cluster* the electricity demand curves. Since a SOM algorithm is topology preserving and hence, similar or related demand curves would fall into the same cluster.
Step 2: Train one predictor for each cluster. Data that is mapped to a particular cluster will serve as a training set for the predictor that is associated with that cluster.

This proposed system can be expected to significantly improve prediction capabilities since:

- We are no longer relying on a single predictor.
- The dataset is segmented into sub-sets that share a common property (i.e. a subset of sub-sequences that feature a particular type of spike or shock), and thus
- Spikes and shocks of any type and form are accounted for.
- Each predictor is an expert in handling a given energy market condition.

The model can be modified and extended very flexibly by using RNN, MLP, or Deep Learning as the base predictor. If RNNs are used then such a system could also incorporate and model the influence of external factors such as weather and special events.

An automated self-adjusting prediction system can be created by combining the SOES method with the event based agile training approach. The system can be extended further into a self-monitoring system by using several SOESs and combining them as a horizontal ensemble. Such a system would be self-monitoring because an error of one (or few) SOESs is masked by the other SOESs in the ensemble. The robustness to errors and failures can be controlled by the number of SOMSE's in the ensemble. The system would be self-adjusting and self-monitoring, be able to deal with shocks and cyclic patterns, and hence reduce or avoid the need for manual intervention in demand prediction.

7.6 Dealing with Equilibrium Adjustment Processes (i.e. price response)

The problem of price response is described in Section 4.5. From that description it is found that a price response is an equilibrium adjustment process that may or may not converge. There can be three objectives when dealing with price response:

1. Detect whether a given situation will cause a price response.
2. Model the equilibrium adjustment process of the price response. This would model the bid and response cycle.
3. Predict the outcome of the equilibrium adjustment process. This would model result of the bid response cycle.

Relevant in the context of 5-minute electricity demand prediction are the objectives 1 and 3. Thus, the prediction model should be able to predict whether or not there will be a price response in 5 minutes, and to predict the equilibrium (the point at which the price response system converges) or, if the system does not convergence, to predict the upper limit of the expected demand. There are a number of approaches that can be taken to predict the outcome of a price response. For example:

Q-learning: Q-learning can model the future effects of pricing policies and possible responses from consumers to guide better pricing policy designs. This is a non-parametric learning system (it is not a ANN) that is often used to train so called *intelligent agents* or robots. The model is being trained on a number of runs (sequences) so as to predict the best possible sequence of action. Q-learning has been applied successfully to prediction in time series data (i.e. stock trading) where Q-learning is used to compute an optimal policy (when to buy and when to sell) [5, 19]. In the context of dealing with price response Q-learning can be adopted to predict whether there will be a price response and consequently, the generated model would describe the course of events that led to the price response. In other words, the generated model can describe the effects of pricing policies on possible (price) responses.

SOES: A SOES can predict the outcome of equilibrium adjustment processes because there would be a predictor that has been trained on historical demand sequences which exhibited a price response. Thus, there is a predictor which is an *expert* at predicting the most likely outcome of a converging price response situation and there would be another predictor which can predict the outcome of a non-converging price response situation. Hence, given a sufficient number of historical examples the SOES can be expected to accurately predict both: (1) whether or not there a price response situation converges and (2) if the system converges it can predict the convergence point.

7.7 Comparison of Prediction Models

Table 1 compares the capabilities of prediction systems. The methods are listed in order of introduction (most recent is listed last). The rating is relative to each other.

	Suitability for modelling time series	Suitable for predictions in time series	Can model effects of price response	Robust to shocks and spikes	Modelling higher level dependencies	Model dependencies with external factors	Availability of readily available software	Cost-benefit
MLP	--	-	⊗	⊗	-	-	+++	◯
Elman Network	++	+	--	--	-	-	++	+
Q-learning	+	+	+	+	-	-	+	+
RNN	+++	++	-	-	++	-	◯	+
DNN or FRPN	++	++	-	-[21]	++	-	◯	+
SOES	+++	+++	++	++	++	++	◯	+

Legend: +++(best), ++(very good), +(good), ◯(satisfactory), -(poor), --(very poor), ⊗(unsuitable)

Table 1: An overview of the capabilities of prediction methods.

[21] Assuming that spikes and shocks are rare events.

8 Summary

Neural Networks are established as the tool of choice internationally for forecasting dispatch demand and for the prediction of electricity generation from renewable sources. AEMO also uses a neural network model for dispatch demand forecast although the type of neural network currently used is a first generation neural network method which has been superseded by more modern approaches. Numerous types of neural networks have been developed in the past 20 years. This report has shown that AEMO's current neural network has significant deficiencies in prediction from time series information. In particular, the currently used model is unable to react to volatility of the energy market and is unable to deal with spikes, shocks, price responses, and any other situation which requires the modelling of context for accurate prediction. The currently used model was adopted because of the lack of alternatives at the time of its introduction and because the model has been tweaked so as to produce reasonably accurate predictions under normal conditions [22]. It is suggested that we can build a forecasting system which combines a number of independent techniques, to take advantages of their strengths in handling unforeseen situations, and overcome their individual shortcomings.

This report concludes with the recommendation that AEMO considers the substitution of the currently used prediction model with a more modern approach like those suggested in this report.

[22] Normal conditions are conditions which are not affected by spikes, shocks, price responses, or any situation which require the modelling of context for accurate prediction.

References

[1] M. A. Abu-El-Magd and N. K. Sinha. Short-term load demand modeling and forecasting: A review. *IEEE Transactions on Systems, Man and Cybernetics*, 12:370–382, 1982.

[2] Australian Energy Market Operator. *5 Minute Demand Forecast TOD-based MetrixND Training Manual*, 6 February 2014.

[3] Dustin Baranek, Alexander Probst, and Stefan Tenbohlen. Optimierung der lastprognose mittels smart meter daten. *Hg. v. Institut für Energieübertragung und Hochspannungstechnik. Universität Stuttgart. Available online via http://www.unistuttgart.de/ieh/forschung/veroeffentlichungen/2013_Probst_Optimierung_der_Lastprognose_mittel_Smart_Meter_Daten.pdf*, 10:2013, 2013.

[4] De Mare Consultants. Five minute electricity demand forecasting - neural network model documentation. Technical report, Australian Energy Market Operator, 01 July 2010.

[5] James Cumming. An investigation into the use of reinforcement learning techniques within the algorithmic trading domain. Master's thesis, Department of Computing, Imperial College London, June 2015.

[6] Maianh Diesendorf. 5-minute electricity demand forecast methodology (internal AEMO report). Technical report, Australian Energy Market Operator, 22 February 2013.

[7] Jeffreyl L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[8] M. Espinoza, J. A. K. Suykens, R. Belmans, and B. De Moor. Electric load forecasting,. *IEEE Control Systems Magazine*, 27:43–57, 2007.

[9] G. Gross and F. D. Galiana. Short-term load forecasting. *Proceedings of the IEEE*, 75:1558–1573, 1987.

[10] M. T. Hagan and S. M. Behr. The time series approach to short term load forecasting. *IEEE Transactions on Power Systems*, 2:785–791, 1987.

[11] Richard F. Hartl, Thomas Staffelmayr, and Gerold Petritsch. *Lastprognose in der Energiewirtschaft mittels Neuronaler Netze*, pages 366–373. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[12] S Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edition, 1999.

[13] T. Hong. Short term electric load forecasting. Technical Report 9/10/2010, PhD thesis, North Carolina State University, 2010.

[14] Kai Hufendiek and Martin Kaltschmitt. Einsatz künstlicher neuronaler netze bei der kurzfristigen lastprognose. In *VGB-Konferenz Forschung für die Kraftwerkstechnik 1998*, number http://dx.doi.org/10.18419/opus-1478, pages 1–5, 1998.

[15] Christoph Höft. *Bewertung von Verfahren zur Prognose der elektrischen Last – eine empirische Analyse*. PhD thesis, Dresden University of Technology, 10 August 2004.

[16] R. Jang. Neuro-fuzzy modeling. *Proceedings of the IEEE*, 83(3), 1995.

[17] S. Lawrence, A.C. Tsoi, and A.D. Back. The gamma mlp for speech phoneme recognition. In *Advances in Neural Information Processing Systems 8*, pages 785–791. MIT Press, 1996.

[18] A. Lotfi and A. C. Tsoi. Learning fuzzy inference systems using an adaptive membership function scheme. *IEEE Transactions on Systems, Man and Cybernetics, Part B - Cybernetics*, 26(2):326–331, 1996.

[19] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 673–680, New York, NY, USA, 2006. ACM.

[20] Genevieve B. Orr and Klaus-Robert Mueller, editors. *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*. Springer, 1998.

[21] Felix Schnorr and Heinrich Hinze. Erstellung von lastprognosen für den elektrischen strombedarf von einfamilienhäusern. Technical report, Hochschule für Technik und Wirtschaft Berlin, 2014.

[22] J. W. Taylor and P. E. McSharry. Short-term load forecasting methods: An evaluation based on european data. *IEEE Transactions on Power Systems*, 22:2213–2219, 2007.